DATA COMMUNICATION SERIES

IPv6 On-Path Telemetry

Driving New Transformation in Network O&M

Zhenbin Li, Pingan Yang, Tao Han,
and Tianran Zhou

CRC Press
Taylor & Francis Group

人民邮电出版社
POSTS & TELECOM PRESS

DATA COMMUNICATION SERIES

IPv6 On-Path Telemetry

Driving New Transformation in Network O&M

Zhenbin Li, Pingan Yang, Tao Han,
and Tianran Zhou

CRC Press
Taylor & Francis Group

人民邮电出版社
POSTS & TELECOM PRESS

IPv6 On-Path Telemetry

Built on Huawei's pioneering research, this book provides a comprehensive introduction to the architecture, techniques, and deployment of IPv6 on-path telemetry, highlighting the most recent developments in IP network technologies.

IPv6 on-path telemetry is a network Operations and Maintenance technology that has emerged alongside the development of IP network services in the 5G and cloud computing era. It has gradually achieved large-scale deployment and application. Starting with the challenges that 5G and cloud computing pose to IP networks and the shortcomings of traditional O&M, this book analyzes the differences between IPv6 on-path telemetry and traditional Operations, Administration, and Maintenance (OAM) technologies. It emphasizes the technical value of IPv6 on-path telemetry from a service perspective in the 5G and cloud era. Then, this book illustrates the technical implementation and deployment of IPv6 on-path telemetry, providing instructions supported by Huawei use cases. Finally, it explores the prospects of IPv6 on-path telemetry industry and related techniques, drawing on Huawei's expertise in IP networks and its research progress.

This book is intended for network planning engineers, technical support engineers, and network administrators. It will also appeal to researchers, students, and professionals interested in IP network technologies and communication networks.

Zhenbin Li is the chief protocol expert responsible for IP protocol research and standards promotion at Huawei. To date, he has acted as the editor-in-chief for the development of multiple technical books, including *SRv6 Network Programming: Ushering in a New Era of IP Networks*, *IPv6 Network Slicing: Offering New Experience for Industries*, and *Guide to SRv6 Network Deployment*.

Pingan Yang is a senior expert responsible for data communication protocols at Huawei, taking the lead in the architecture design and delivery of Huawei's box-shaped routers and CloudBRAS products, as well as the design and deployment of MPLS, Virtual Private Network (VPN), SRv6, and on-path telemetry.

Tao Han is a chief architect at Huawei Data Communication Product Line, responsible for the architecture and system design of routers, switches, and other products; accumulating a wealth of experience in the data communication field.

Tianran Zhou is a datacom protocol expert at Huawei, who has accumulated extensive experience in fields including SDN, AI-powered intelligent O&M, IPv6 Enhanced, and cloud-network synergy. He has submitted more than 50 patents, published more than 10 papers, and participated in the development of the technical book *SRv6 Network Programming*.

Data Communication Series

Cloud Data Center Network Architectures and Technologies
Lei Zhang and Le Chen

Campus Network Architectures and Technologies
Ningguo Shen, Bin Yu, Mingxiang Huang, and Hailin Xu

Enterprise Wireless Local Area Network Architectures and Technologies
Rihai Wu, Xun Yang, Xia Zhou, and Yibo Wang

Software-Defined Wide Area Network Architectures and Technologies
Cheng Sheng, Jie Bai, and Qi Sun

SRv6 Network Programming: Ushering in a New Era of IP Networks
Zhenbin Li, Zhibo Hu, and Cheng Li

IPv6 Network Slicing: Offering New Experience for Industries
Zhenbin Li, Jie Dong, Yawei Zhang, and Xinzong Zeng

Guide to SRv6 Network Deployment
Edited by Minwei Jin, Zhenbin Li, Lanjun Luo, and Weidong Li

IPv6 On-Path Telemetry: Driving New Transformation in Network O&M
Zhenbin Li, Pingan Yang, Tao Han, and Tianran Zhou

For more information on this series, please visit: https://www.routledge.com/Data-Communication-Series/book-series/DCSHW

IPv6 On-Path Telemetry

Driving New Transformation in Network O&M

Zhenbin Li, Pingan Yang, Tao Han,
and Tianran Zhou

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an informa business

人民邮电出版社
POSTS & TELECOM PRESS

The cover image is from Huawei's internal gallery and is copyrighted for all purposes.

First edition published 2026
by CRC Press
2385 NW Executive Center Drive, Suite 320, Boca Raton FL 33431

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Translated by Zhaodi Zhang, Wei Li, and Chen Jiang

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

For Product Safety Concerns and Information please contact our EU representative GPSR@taylorandfrancis.com. Taylor & Francis Verlag GmbH, Kaufingerstraße 24, 80331 München, Germany.

English Version by permission of Posts and Telecom Press Co., Ltd.

ISBN: 978-1-041-14514-1 (hbk)
ISBN: 978-1-041-15128-9 (pbk)
ISBN: 978-1-003-67790-1 (ebk)

DOI: 10.1201/9781003677901

Typeset in Minion
by codeMantra

Contents

Foreword I

Packet loss and delay are the concerns of network operators when new network topologies are being deployed or current network topologies suddenly change due to operational or configuration changes.

In the past, packet loss and delay were measured and observed through probing. As the name implies, probes are of limited scale and do not exactly reflect customers' packet loss and delay experience since they implement emulation. They are also unable to answer why and where packets are dropped or delayed on the network.

Customers sign Service Level Agreements (SLAs) with their network operators, where network operators have Service Level Objectives (SLOs) defining when to act. With IPFIX, network operators enabled provider and customer data plane visibility. Flow Aggregation measurements can be aggregated across data, control, and management planes. This enables network operators to gain a statistical view of their networks.

One of the current limitations of IPFIX implementation is that packets are usually sampled. Although sampling and aggregation are needed to reduce the amount of data being exported from a network element, sampling has the negative impact that operators do not have the ability to follow the forwarding path of a single packet throughout a virtualized Segment Routing-enabled network. It is necessary to measure the on-path delay. Thanks to the introduction of options headers in IPv6, tracing information, such as the direct export bit and observation timestamp, can be added to a packet when it enters the Segment Routing domain. This enables Hybrid Type 2 Passive use cases. Combined with the ability to export the measured and aggregated on-path delay in IPFIX, operators are now able to measure not only the loss with ForwardingStatus IE89 as a Service Level

Indicator (SLI), but also the on-path delay without increasing the amount of records exported.

Since Segment Routing adds segment lists into the IPv6 data plane, and thanks to it being visible in IPFIX, we have now almost all the statistical information to understand where the loss and delay for customer packets for a specific traffic-engineered Segment Routing path are coming from. We are almost where we want to be.

With flow queuing having been established in the industry to fight Buffer Bloat and by extending IPFIX in the future to also give the flow queue context, we will finally arrive at the point where we have a cohesive, well-aligned system between control, forwarding, and management planes and network telemetry data export. This will be the critical foundation for a closed-loop, Autonomous Driving Network (ADN).

This book gives valuable details in described technologies. They are essential for operating networks to meet today's customer expectations.

Thomas Graf
Distinguished Network Engineer and Network Analytics Architect at Swisscom

Foreword II

WITH THE RAPID DEVELOPMENT of information technologies, the commercial deployment of Internet Protocol version 6 (IPv6) has become a global consensus. IPv6 offers significant improvements over IPv4, including a larger address space, higher security, and better performance. Building on IPv6, IPv6 Enhanced has emerged, further promoting Internet innovation and development. IPv6 Enhanced is an upgrade and innovation of IPv6, covering multiple key technologies such as Segment Routing, on-path telemetry, and network slicing. These technologies not only enhance the capabilities of IPv6 but also promote its application in various scenarios, including basic networks, industry networks, and data centers. Among these technologies, on-path telemetry plays an essential role in IPv6 Enhanced, leading network O&M into the intelligent era.

As the Internet grows in popularity, and applications become more diverse, the complexity of network O&M is increasing. Conventional methods used in network O&M mainly rely on active measurement techniques, such as ping and TWAMP. However, these techniques cannot accurately reflect the channel performance at the same time point due to discrepancies between the measured and actual service traffic. This issue makes it impossible to meet requirements for high-precision and real-time performance. In addition, the rapid development of video, voice, and other real-time services poses higher requirements on the capabilities of networks to process traffic bursts. This means that conventional active measurement methods cannot meet the requirements for fast fault locating and intelligent O&M.

Against this backdrop, on-path telemetry is introduced to measure real service traffic. It does this by embedding telemetry information — such as the forwarding path, forwarding time, and device information — into the data packets of user services. As these packets traverse the network, each network device measures channel performance based on the embedded telemetry information and reports the measurement results to the data analysis platform for result calculation and display.

On-path telemetry can monitor network performance in real time based on key indicators, including the packet loss rate, delay, and jitter. These indicators are uploaded to an intelligent brain for analysis, allowing O&M personnel to quickly detect and eliminate potential network problems. This approach eliminates the need for tedious manual troubleshooting involved in conventional O&M, thereby improving O&M efficiency while also reducing O&M costs.

Using key indicators, including bandwidth usage and delay, on-path telemetry can monitor the performance of service traffic in real time. These indicators enable carriers to promptly detect and resolve problems in service traffic, ensuring stable service running and optimal user experience. In particular, in some scenarios with high requirements on real-time performance, such as 5G bearer services and the finance industry’s smart outlets, deploying on-path telemetry can significantly improve user experience.

Furthermore, on-path telemetry can monitor network resource usage in real time based on key information, including bandwidth usage and device load. Such information allows carriers to more accurately understand the allocation and usage of network resources, facilitating resource configuration and optimization. This not only improves the utilization of network resources but also reduces network operation costs.

What’s more, on-path telemetry can be used for network security protection. By enabling carriers to monitor and analyze network traffic data in real time, on-path telemetry allows them to promptly detect and defend against network threats. For example, carriers can monitor abnormal traffic and leverage pattern recognition technologies to detect and defend against network threats such as DDoS attacks. This not only improves network security but also protects user privacy and data security.

As an important innovation in the field of network performance measurement, on-path telemetry is driving the intelligent development of networks. By monitoring and analyzing network performance data in real time, carriers can gain a better understanding of how the network is operating and requirements are changing, thereby implementing more precise network planning and optimization. In addition to improving network stability and reliability, this promotes the continuous innovation and development of network technologies.

As network technologies evolve and application requirements change, on-path telemetry techniques are also advancing, especially in terms of standardization, intelligentization, multi-scenario application, and security assurance.

On-path telemetry techniques will become increasingly standardized and normalized as IPv6 Enhanced technologies become more widespread and standardization advances. This promotes technical compatibility and interoperability between different vendors and also boosts the application and development of on-path telemetry techniques.

With advancements in computing and storage technologies, the performance and intelligentization of on-path telemetry will continue to improve. In the future, on-path telemetry will be able to implement more efficient and accurate monitoring and analysis functions, providing more intelligent support and services for network O&M.

As IPv6 Enhanced is applied to more scenarios, the application scope of on-path telemetry will expand. For example, in emerging fields such as autonomous driving and industrial Internet, on-path telemetry will play a more important role and provide stronger support for network O&M and management.

And, as network security and privacy protection issues become more prominent, on-path telemetry will focus more on security assurance and privacy protection. By implementing data flow path tracing, it facilitates the management of cross-border data flows. In the future, it will use more advanced security protection technologies and privacy protection mechanisms to ensure data security and privacy.

At present, China is actively driving IPv6 network upgrade and reconstruction. In the context of the accelerated digital transformation of enterprises, IPv6 on-path telemetry will have a broader development prospect. By enabling high-precision, real-time network performance measurement and fast fault locating, it provides powerful support for network O&M. As technologies advance and application scenarios expand, IPv6 on-path telemetry will play a more important role in the future, laying a solid network foundation for the development of the digital era. Through the collaboration of related enterprises and organizations, IPv6 on-path telemetry techniques are expected to see continuous innovation and improvement, facilitating the construction of more intelligent, secure, and reliable networks.

Huawei's IPv6 Enhanced technology innovation team actively collaborates with industry partners and has achieved significant milestones in the innovation, standardization, product development, and deployment of IPv6 on-path telemetry techniques. Building on this, the team completed the book *IPv6 On-Path Telemetry: Driving New Transformation in Network O&M*. This book systematically describes the architecture, key techniques, and deployment of IPv6 on-path telemetry, providing a valuable reference for practitioners. We believe that this book will have a positive impact on the development of the IPv6 industry.

Let us look forward to the development of IPv6 on-path telemetry in the future and encourage more people to contribute to the innovation of IPv6 technologies and their applications.

Hequan Wu
Chairman China's Expert Committee of Promoting Large-Scale IPv6 Deployment

Preface

NETWORK MEASUREMENT IS A crucial foundation for IP network Operations and Maintenance (O&M), but it has long posed significant technical challenges for IP networks. These challenges arise for multiple reasons. Primarily, they stem from the absence of an on-path measurement mechanism, coupled with insufficient capabilities of reporting and analyzing massive amounts of data. With the emergence of SDN, the issues associated with handling massive amounts of data have been mitigated to some degree thanks to the introduction of high-performance data reporting techniques like Internet Protocol Flow Information Export (IPFIX) and Google Remote Procedure Call (gRPC), along with a centralized controller's ability to analyze massive amounts of data. Since 2017, the rapid growth of emerging services like 5G and cloud computing has driven the development of IPv6 Enhanced technologies, such as SRv6 network programming and IPv6 network slicing. These technologies leverage IPv6 extension headers to carry instructions for extending network functions, thereby propelling advancements in IPv6 on-path measurement techniques. The establishment of a systematic technical framework, encompassing techniques of east-west on-path measurement, south-north high-performance data reporting, and analysis of massive amounts of data by a controller, offers a viable solution for solving O&M challenges on IP networks. Moreover, these techniques have already been deployed at scale on live networks. This technical framework is referred to as IPv6 on-path telemetry, the title of this book.

Huawei's Data Communication Product Line started research on the SDN transition many years ago. From 2013 to 2017, Huawei worked with industry partners to develop southbound interface protocols for controllers, including BGP, Path Computation Element Communication Protocol (PCEP), and Network Configuration Protocol (NETCONF)/Yet Another Next Generation (YANG), and defined related standards. Since 2014, Huawei's Data Communication Product Line has conducted research on IP Flow Performance Measurement (IP FPM) and defined related standards within the IETF framework. And since 2017, Huawei has been devoted to innovating and standardizing IPv6 on-path telemetry techniques. This work has included collaborating with industry partners to drive innovation in IPv6 on-path measurement techniques such as alternate marking and In-situ Operations, Administration, and Maintenance (IOAM); pioneering distributed high-performance data reporting techniques, including IPFIX protocol extensions and User Datagram Protocol (UDP)–based telemetry; and developing standards like the Network Telemetry Framework (NTF) and In-situ Flow Information Telemetry (IFIT) — a framework for implementing IP on-path telemetry – based on the systematic research into on-path telemetry. This technical research and the work on these standards have laid a solid foundation for the establishment of the IPv6 on-path telemetry technical system. Drawing on our extensive expertise in R&D, standardization, and deployment, we have compiled this book, which systematically explains the IPv6 on-path telemetry technical system. Our goal is to facilitate a deeper understanding of IPv6 on-path telemetry techniques in the industry and contribute to the industry's development.

This book offers a thorough exploration of IPv6 on-path telemetry. To facilitate easier reading and comprehension, there are two key points to keep in mind:

- The relationship between IPv6 on-path telemetry and IPv6 on-path measurement: IPv6 on-path telemetry is a framework that encompasses a suite of techniques, including IPv6 on-path measurement, telemetry data reporting, and associated data analysis across controllers and the data, control, and management planes. At the heart of IPv6 on-path telemetry lies the IPv6 on-path measurement mechanism, which is fundamentally tied to the data plane. This book aims to provide a comprehensive introduction to IPv6 on-path telemetry techniques, rather than focusing exclusively on IPv6 On-Path measurement mechanisms, which is why we titled it *IPv6 On-Path Telemetry*. Both IPv6 on-path telemetry and IPv6 on-path measurement are used in this book, so it is important to distinguish between them.

- The pivotal role that IPv6 plays in IP on-path measurement: IP on-path measurement techniques (including alternate marking and IOAM) require the encapsulation of instruction data of on-path measurement in the data plane, posing new requirements for data plane extensions. These extensions are not specific to IPv6 and may also be implemented in other data planes. However, the IPv6 extension header mechanism was originally designed as part of IPv6. Utilizing IPv6 extension headers to carry instructions of on-path measurement can effectively address the on-path measurement needs of IP networks. Furthermore, compared with other data plane extension mechanisms, the IPv6 extension header mechanism boasts greater maturity in terms of standardization, commercial delivery, and deployment. Consequently, this book focuses on IPv6 on-path telemetry. Related techniques can also serve as a reference for implementing on-path telemetry in other data planes. We believe that IPv6 on-path telemetry will emerge as a critical application of IPv6, alongside SRv6 network programming and IPv6 network slicing, thereby accelerating the widespread adoption and deployment of IPv6.

MAIN CONTENTS IN THIS BOOK

This book consists of nine chapters that have been divided into three parts. The first part – Chapter 1 – focuses on the background and benefits of IPv6 on-path telemetry techniques. Part II – Chapters 2–7 – describes the various techniques involved in IPv6 on-path telemetry. Part III– Chapters 8–9 – wraps up this book by summarizing the development of the IPv6 on-path telemetry industry and exploring its future.

CHAPTER 1 OVERVIEW OF IPV6 ON-PATH TELEMETRY

This chapter analyzes the pain points of traditional O&M methods based on service and architecture evolution in the 5G and cloud era, introduces the background of IPv6 on-path telemetry, and describes its technical benefits.

CHAPTER 2 IPV6 ON-PATH TELEMETRY ARCHITECTURE

This chapter outlines the overall architecture, functional modules, and data acquisition mechanisms of network telemetry and delves into IFIT – a reference architecture for implementing IPv6 on-path telemetry and related key techniques.

CHAPTER 3 DATA PLANE OF IPV6 ON-PATH TELEMETRY

This chapter describes data plane solutions of IPv6 on-path telemetry, including the alternate marking and IOAM methods.

CHAPTER 4 CONTROL PLANE OF IPV6 ON-PATH TELEMETRY

This chapter introduces control plane solutions of IPv6 on-path telemetry and offers a detailed explanation of control plane protocol extensions for IPv6 on-path telemetry. During IPv6 on-path telemetry deployment, control plane protocol extensions can be used to advertise and negotiate the measurement capabilities of nodes and links. In addition, telemetry instances can be automatically deployed through control plane protocols, making deployment far easier.

CHAPTER 5 IPV6 ON-PATH TELEMETRY INFORMATION REPORTING

This chapter explains how IPv6 on-path telemetry information is reported. Telemetry is a technique that can remotely and quickly collect data from devices. It also supports multiple data push methods. This chapter provides a detailed description of gRPC, UDP, and IPFIX.

CHAPTER 6 IPV6 ON-PATH TELEMETRY CONTROLLER

This chapter describes the architecture and functions of the IPv6 on-path telemetry controller, Network Digital Map technology implemented based on on-path telemetry, and southbound and northbound interface capabilities. It also comprehensively explains the key technical principles and working processes of the IPv6 on-path telemetry controller.

CHAPTER 7 DEPLOYMENT OF IPV6 ON-PATH TELEMETRY

This chapter describes how to deploy IPv6 on-path telemetry. An implementation of IPv6 on-path telemetry that has been deployed on many carrier and enterprise networks is the IFIT solution based on the alternate marking method, delivering enhanced user experience while making network quality observable in real time. This chapter first demonstrates the successful application of the alternate marking-based IFIT solution on the live network, and then explains the deployment procedures from both device and controller perspectives.

CHAPTER 8 INDUSTRY DEVELOPMENT AND TECHNOLOGY PROSPECTS OF IPV6 ON-PATH TELEMETRY

This chapter summarizes the development of the IPv6 on-path telemetry industry and offers insights into its future development.

IPv6 on-path measurement has strong ties to active measurement and clock synchronization techniques. Consequently, the appendix of this book introduces the technical fundamentals of IPv6 active measurement techniques, including TWAMP and Simple Two-Way Active Measurement Protocol (STAMP), and clock synchronization techniques, including Network Time Protocol (NTP) and Precision Time Protocol (PTP).

CHAPTER 9 JOURNEY ALONG IPV6 ON-PATH TELEMETRY

In "Journey Along IPv6 On-Path Telemetry" in Chapter 9, Zhenbin Li summarizes the development history of IPv6 on-path telemetry techniques and Huawei's participation in innovation and standards promotion. Each chapter also ends with some of the stories behind IPv6 on-path telemetry design, summarizing the design experience of involved techniques and helping readers further understand the design and deepen their understanding of such techniques. Some of the content constitutes the author's opinion and should be used for reference only.

EDITORS

Zhenbin Li and Pingan Yang are the editors-in-chief of this book, and Tao Han and Tianran Zhou are the deputy editors-in-chief. This book has been meticulously compiled by Zhenbin Li. The editorial board comprises a diverse team from Huawei's Data Communication Product Line. Hang Ruan, Jinming Huang, and other colleagues helped write the book and provided extensive technical materials. Documentation department staff, including Jingyi Chen, Yajia Hu, Fan Zhang, Lili Peng, Yanran Li, and Jiahui Wang, carefully edited the text and worked on figures to ensure overall quality. And Yanmiao Wang, Yusheng Zhang, and Keyi Zhu reviewed the technical aspects of this book and offered valuable technical suggestions.

Acknowledgments

While promoting the innovation and standardization of IPv6 on-path telemetry, we have received extensive support and help from both inside and outside Huawei. On this occasion of the book's publication, we would like to express our sincere appreciation to the following Huawei executives and colleagues for their support: Kewen Hu, Shaowei Liu, Lei Wang, Zhipeng Zhao, Juye Wu, Chenxi Wang, Yuefeng Qiu, Su Feng, Jinzhu Chen, Meng Zuo, Zhigang Wang, Hui Wang, Zhiqiang Du, Xiao Qian, Jianbing Wang, Liang Zhang, Zhaokun Ding, Jian Jin, Minwei Jin, Shixin Shao, Xiaojun Yu, Dawei Fan, Jianping Sun, Rui Gu, Huizhi Wen, Xiaopan Li, Dandan Tang, Jiandong Zhang, Yue Liu, Shucheng Liu, Zhe Jiang, Guangtao Ren, Weidong Li, Juhua Xu, Lei Bao, Shuying Liu, Yi Zeng, Xiaoqi Gao, Jialing Li, Peng Zheng, Peng Wu, Songfeng Liu, Li Fan, Fuyou Miao, Guoyi Chen, Yali Wang, Min Liu, Bo Lu, Yunan Gu, Longfei Dai, Ling Xu, Qin Wu, Bo Wu, Xin Yan, Hong Shen, Wenxia Dong, Guanjun Zhou, Zhijun Jing, Yuanyi Sun, Leyan Wang, Shuhui Wang, Xiaohui Tong, Mingyan Xi, Xiaoling Wang, Yonghua Mao, Lu Huang, Kaichun Wang, Huaguo Mo, Hongkun Li, Taixu Tian, Yang Xia, Gang Yan, Fenghua Zhao, Cheng Sheng, Zhibo Hu, Haibo Wang, Dapeng Chen, Guofeng Qian, Yongpeng Zhao, Xinzong Zeng, Zhong Chen, Hui Zhang, Zhidong Yin, Chun Liu, Jinhui Wang, Jingfei Lv, Fang Xin, Jianfeng Yang, Jiaxin Pan, Pingping Yu, Wei Yu, Zhaodi Zhang, Wei Li, Chen Jiang, Yadong Deng, Xue Zhou, Ruijuan Li, George Fahy, and Samuel Luke Winfield-D'Arcy. Additionally, we would like to extend our heartfelt thanks to the following technical experts in China's IP field who have been longtime supporters of our technological innovation and standards promotion efforts: Hui Tian, Wei Gao, Feng Zhao, Yunqing Chen, Huiling

Zhao, Chongfeng Xie, Aijun Wang, Yongqing Zhu, Huanan Chen, Xiaodong Duan, Weiqiang Cheng, Zhenqiang Li, Wenying Jiang, Fengwei Qin, Xiongyan Tang, Chang Cao, Ran Pang, and Zhuangzhuang Qin. Finally, we would like to express our gratitude to Thomas Graf and academician Hequan Wu for writing the Forewords to this book.

We have written this book with the intention of creating a comprehensive reference that presents the IPv6 on-path telemetry framework, related techniques, and deployment. While we have made every effort to ensure the accuracy and completeness of this book, IPv6 on-path telemetry is evolving, and errors or omissions may occur. We would therefore be grateful for any feedback about such issues.

Recommendations

THE BOOK ON IPV6 on-path telemetry is a complete manual on the high-precision measurement techniques for IPv6 networks. It covers all the aspects that are necessary to know for a complete deployment, including the data plane, control plane, and management plane. Indeed, the book describes how the on-path telemetry methods are encoded into each IPv6 packet, which protocols can be used to advertise and negotiate the measurement capabilities, how to configure proper monitoring, and which alternatives are available for reporting the IPv6 on-path telemetry data. It also covers the framework architecture and the core functions of the controller for fully automated and intelligent OAM. Several commercial deployment scenarios are presented to show their feasibility. The standardization progress and the related industry initiatives are reported too. This book is comprehensive and exhaustive, so I would recommend it to both technical experts and network administrators. Technical experts will be interested in the state-of-the-art of the on-path telemetry methods, which overcome the pain points of the traditional network OAM methods and address the measurement challenges of the new 5G and cloud services, while network administrators will be interested in the effective implementation of the on-path telemetry, which is crucial nowadays for performing real-time monitoring that allows for measuring the quality of network services with high accuracy.

– Giuseppe Fioccola
IETF BMWG Chair

As a researcher and engineer who has been actively involved in the evolution of network operations and maintenance, I find IPv6 On-Path

Telemetry: Driving New Transformation in Network O&M to be an invaluable contribution to this field. It provides a comprehensive and insightful exploration of IPv6 on-path telemetry, a core technology driving intelligent O&M in the 5G and cloud era. It doesn't merely describe the technical specifications; it shares the compelling "stories behind" the development of key concepts and standards, offering a unique perspective on the journey from initial ideas to large-scale commercial deployment. The authors, who were at the forefront of these innovations, detailed the evolution of IP O&M techniques, the pain points of traditional methods, and how techniques like alternate marking and IOAM have been enhanced and combined with IPv6 and telemetry mechanisms to enable on-path measurement. Drawing on personal experience, the book highlights the challenges encountered and the limitations of previous approaches. It transparently discusses the standardization process within the IETF, including the debates and collaborations that led to important RFCs and drafts related to alternate marking, IOAM, NTF, and IFIT capability advertisement. The book also covers the practical aspects, including the data plane mechanisms, control plane extensions for capability advertisement using IGP and BGP extensions, and various information reporting methods like gRPC, UDP Telemetry, and IPFIX. Real-world application scenarios and deployment examples of IFIT-AM on carrier and enterprise networks are presented, demonstrating its value in areas like IP RAN, premium private lines, and financial WANs. Ultimately, this book is a testament to the relentless exploration and hard practice required for true technological innovation. It not only provides a deep technical understanding of IPv6 on-path telemetry but also offers valuable lessons from the journey of making IP networks healthier and simplifying O&M. For anyone invested in the future of network operations and the capabilities of IPv6 Enhanced networks, this book is strongly recommended.

– Haoyu Song
Futurewei Network Expert

In the era of 5G and cloud computing, network performance monitoring and maintenance are becoming increasingly complex and critical. The book *IPv6 On-Path Telemetry* focuses on the latest developments in this field and reveals how on-path telemetry combines with IPv6 – the next-generation Internet Protocol — to achieve real-time, visualized monitoring of network service forwarding quality. As well as diving into the on-path telemetry system, this book provides detailed deployment instructions, offering excellent technical references for network researchers, O&M personnel, and even industry managers.

– Wei Gao
Director of Internet Center Institute of Technology and Standards China Academy of Information and Communications Technology (CAICT)

IPv6 is the foundation of the next-generation digital information infrastructure, with IPv6 Enhanced as its long-term evolution direction. *IPv6 On-Path Telemetry* is the third classic in the IPv6 Enhanced series of books, written by authors with extensive experience in IP protocol formulation and systematic R&D. IPv6 on-path telemetry is key to evolving networks into autonomous systems and efficiently implementing computing networks. Incorporating deployment and engineering practices, this book provides a systematic, in-depth, and multi-perspective exploration into the protocol architecture and technical implementation of IPv6 on-path telemetry. This book is a valuable reference for technical R&D and engineering personnel and will further promote the expansion and practice of IPv6 Enhanced-related systematic engineering.

– Yunqing Chen
Consultant of China Telecom Research Institute and Director of the Information and Communication Network Technology Committee of China Institute of Communications (CIC)

IP network on-path telemetry is one of the core technologies for IPv6 evolution. By providing real-time, refined, and E2E network performance measurement, it enhances network troubleshooting and O&M efficiency and serves as a key technology for future network evolution toward intelligentization and servicization. This book delves into the principles and applications of this technology and provides a valuable reference covering network research and engineering practice.

– Xiaodong Duan
Vice President of China Mobile Research Institute

As one of the key IPv6 Enhanced technologies, IPv6 on-path telemetry plays a vital part in building computing networks. It implements high-precision service flow measurement in real time and intuitively displays the network service quality indicators of flows for users, facilitating fault locating and path optimization while also improving network service quality. China Unicom attaches great importance to the deployment and application of IPv6 Enhanced technologies. It has deployed SRv6, IFIT, and network slicing on multiple networks across China, including in Beijing, Hebei, and Hubei, significantly enhancing network service capabilities. *IPv6 On-Path Telemetry* is another masterpiece in the IPv6 Enhanced series of books. It describes the background, technical system, deployment, and future prospects of IPv6 on-path telemetry, providing a reference for the research, deployment, and application of intelligent network O&M. This book is expected to better promote network intelligentization and accelerate the prosperity of the IPv6 Enhanced industry.

– Xiongyan Tang
Vice President of China Unicom Research Institute

Before the emergence of IPv6 on-path telemetry, WAN transmission quality was typically measured using BFD, NMS probes, traffic backtracking, or other methods. These methods indirectly or partially reflected the SLA status, meaning that the network was like a mysterious, invisible black box. In IPv6 on-path telemetry, IFIT monitoring information is embedded into service packets to monitor the forwarding path and service quality of the packets in real time. This turns the network into an open and visible white box, eliminating the industry's pain points in achieving fast fault demarcation/locating and network innocence self-proving. The Agricultural Bank of China is a pioneer and beneficiary of IPv6 on-path telemetry. By integrating a self-developed IFIT management module into the SRv6 backbone network and SRv6 branch access network, it has built E2E service quality visualization capabilities. Information such as the service path, delay, packet loss rate, and traffic rate from the head office data center to outlets is clearly displayed, achieving fast fault locating and service recovery while also effectively ensuring financial service continuity. This book describes the architecture and protocol principles of IPv6 on-path telemetry technology in simple terms, helping readers quickly grasp the technology's full scope and application prospects. It is a great resource for promoting IPv6 on-path telemetry and improving the industry’s network O&M level.

– Qingbang Xu
Architect of Agricultural Bank of China

IPv6 on-path telemetry is an important technological innovation of IPv6. It not only promotes the development of network technologies but also supports the construction of future high-speed data networks in China, facilitating flexible, secure, and efficient data circulation. Zhenbin Li and his team have been deeply involved in the IPv6 field for many years. This book illustrates the system architecture, key techniques, and practices of IPv6 on-path telemetry, providing strong support for the development and continuous innovation of IPv6 technologies.

– Dong Liu
Director of China Future Internet Engineering Center (CFIEC)

Technical Committee

Li, Jiahui Wang

Technical Reviewers: Yanmiao Wang, Yusheng Zhang, Keyi Zhu

Translators: Zhaodi Zhang, Wei Li, Chen Jiang, Yadong Deng, Xue Zhou, Ruijuan Li

Content Editors: George Fahy, Samuel Luke Winfield-D’Arcy

Authors

Zhenbin Li is the chief protocol expert responsible for IP protocol research and standards promotion at Huawei. He joined the company in 2000 and was in charge of the architecture, design, and development of Huawei's IP operating system Versatile Routing Platform (VRP), Multiprotocol Label Switching (MPLS) sub-system, and Software Defined Network (SDN) controller. Since 2009, he has been active in standards innovation in the Internet Engineering Task Force (IETF), continuously contributing to the innovation and standardization of numerous protocols, such as SDN southbound protocols, Segment Routing over IPv6 (SRv6), 5G transport, telemetry, and Application-Aware Networking (APN). To date, he has led and participated in more than 100 IETF Requests for Comments (RFCs)/drafts, submitted more than 110 patents, and acted as the editor-in-chief for the development of multiple technical books including *SRv6 Network Programming: Ushering in a New Era of IP Networks*, *IPv6 Network Slicing: Offering New Experience for Industries*, and *Guide to SRv6 Network Deployment*. During the period from 2019 to 2023, he undertook Internet architecture management as a member of the IETF Internet Architecture Board (IAB).

Pingan Yang is a senior expert responsible for data communication protocols at Huawei. Since joining the company in 2001, he has been engaged in the architecture and system design of IP products and protocols. Taking the lead in the architecture design and delivery of Huawei's box-shaped routers and CloudBRAS products, as well as the design and deployment of MPLS, Virtual Private Network (VPN), SRv6, and on-path telemetry, he is dedicated to introducing new technologies such as SDN, cloudification, and intelligence into the networks of carriers and various industries.

Tao Han is the chief architect at Huawei Data Communication Product Line. Having joined the company in 2001, he took responsibility for the architecture and system design of routers, switches, and other products, accumulating a wealth of experience in the data communication field. While leading 5G transport network architecture planning and product system design, he integrated technologies – such as SRv6, network slicing, and on-path telemetry – into system hardware/chipset design, helping to establish a future-oriented evolution roadmap for the IP transport network architecture.

Tianran Zhou is a datacom protocol expert at Huawei. Having spent over 15 years in innovative research and standardization, he has accumulated extensive experience in fields including SDN, Artificial Intelligence (AI)-powered intelligent Operations and Maintenance (O&M), IPv6 Enhanced, and cloud-network synergy. To date, he has submitted more than 50 patents, published more than 10 papers, and participated in the development of the technical book *SRv6 Network Programming: Ushering in a New Era of IP Networks*. He worked as the co-chair of the Operations and Management Area Working Group (OPSAWG) and as a member of the directorate for this area, and published seven RFCs in the IETF. He also took charge of architecture design in many open-source organizations, involving projects such as OpenStack, OpenDaylight, Open Platform for NFV (OPNFV), and Open Network Operating System (ONOS), as well as working as a project team leader.

Technical Reviewers

Yanmiao Wang is the director of Huawei Service Router PDU. Since joining the company in 2006, he has held a number of positions, including the director of the Router Maintenance Dept., director of the Router Solution Dept., and director of the Technology Development Dept. He has also been responsible for the innovation of key technologies and solutions, product R&D, and full lifecycle management in the router field.

Yusheng Zhang is the general manager of the VRP Technology Development Team (TDT) at Huawei Data Communication Product Line. He joined the company in 2007, taking charge of the design and development of Border Gateway Protocol (BGP), VPN, and SDN. From 2019 to 2022, he worked as the chief expert in Middle East and Africa carrier product management. There, he led the innovative planning of the IPv6 Enhanced-oriented network solution for regional benchmark carriers, helping them implement large-scale deployment of SDN and SRv6, and accumulating extensive experience in SRv6 network planning and design. Since 2023, he has been responsible for the competitiveness planning and delivery of data communication protocols.

Keyi Zhu is the director of the Data Communication Standard & Patent Dept. at Huawei Data Communication Product Line. He has served as the director of Huawei Data Communication Enterprise Router Domain, chief network engineer of Huawei Mobile Account Dept., and director of the Carrier IP Solution Sales Dept. outside China. In addition, he is the deputy secretary of the Network Innovation and Development Alliance (NIDA). He takes charge of the standardization work in multiple standards organizations, such as IETF, Institute of Electrical and Electronics Engineers (IEEE), International Telecommunication Union (ITU), China Communications Standards Association (CCSA), and NIDA on behalf of Huawei Data Communication Product Line, and was deeply involved in the formulation of IPv6 Enhanced policies, technical standards, and certification standards. He also participated in the development of multiple books, such as *Guide to SRv6 Network Deployment*.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the Vul. Response Process. For details about this process, visit the following web page:

https://www.huawei.com/en/psirt/vul-response-process

For vulnerability information, enterprise customers can visit the following web page:

https://securitybulletin.huawei.com/enterprise/en/security-advisory

I

CHAPTER 1

Overview of IPv6 On-Path Telemetry

DOI: 10.1201/9781003677901-2

INTERNET PROTOCOL VERSION 6 (IPv6) on-path telemetry is a network performance monitoring technical system built upon on-path measurement techniques. Developed alongside network services in the 5th Generation of Mobile Communication Technology (5G) and cloud era, it provides network administrators with real-time, visualized indicators on the forwarding quality of network services, laying the foundation for intelligent Operations and Maintenance (O&M). This chapter analyzes the pain points of traditional O&M methods based on service and architecture evolution in the 5G and cloud era, introduces the background of IPv6 on-path telemetry, and describes its technical benefits.

1.1 OVERVIEW OF IP OAM TECHNIQUES

IPv6 on-path telemetry is closely related to the Operations, Administration and Maintenance (OAM) techniques of IP networks. OAM is a general term that refers to a toolset for detecting, isolating, and reporting faults and measuring performance. It is widely used in network O&M activities. In terms of functionality, OAM provides both Fault Management (FM) and Performance Measurement (PM), as shown in Figure 1.1[1,2].

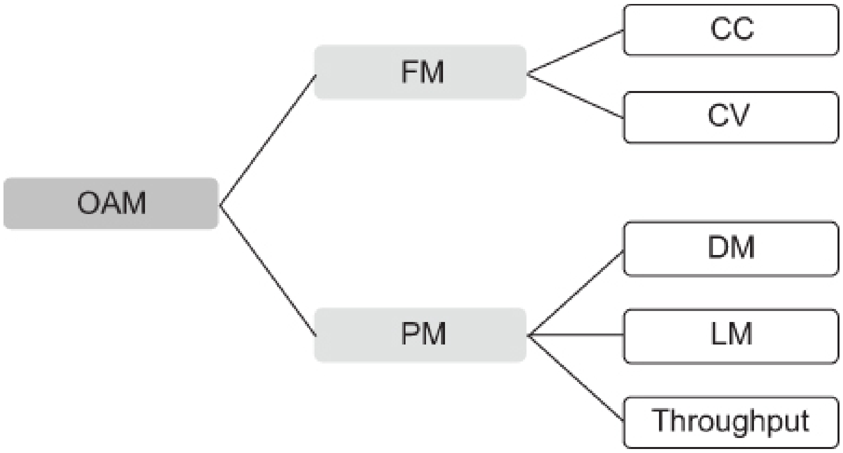


FIGURE 1.1 Classification of OAM functions. ⏎

FM mainly covers the following two aspects:

- Continuity Check (CC) detects address reachability by using mechanisms such as IP ping[3] and Bidirectional Forwarding Detection (BFD)[4].
- Connectivity Verification (CV) verifies paths and locates faults by using mechanisms such as IP traceroute[3] and BFD.

Different OAM FM mechanisms may be used on different networks. For example, to implement FM functions, IP ping and traceroute are used on an IP network, Label Switched Path (LSP)[5] ping and traceroute can be used on a Multi-Protocol Label Switching (MPLS) network, and IPv6 ping and traceroute or End.OP in Segment Routing over IPv6 (SRv6) OAM protocol extension[6] can be used on an SRv6 network. And to implement fast CC and CV, BFD can be used on IP, MPLS, and SRv6 networks.

PM mainly involves the following aspects:

- Delay Measurement (DM) measures metrics such as delay and jitter.
- Loss Measurement (LM) measures metrics such as the number of lost packets and packet loss rate.
- Throughput measurement measures metrics such as interface bandwidth, link bandwidth, and the packet processing capability per unit of time.

OAM implements PM in one of the following three ways[7] according to whether OAM packets need to be actively sent.

- Active PM actively generates OAM packets, measures their performance, and uses the measurement result to infer network performance. A typical example of active PM is Two-Way Active Measurement Protocol (TWAMP) [8].

- Passive PM monitors undisturbed and unmodified service data packets to obtain performance indicators. Unlike active PM, passive PM observes the data packets directly, without modifying them or generating additional OAM packets. It can therefore accurately reflect network performance. A typical example of passive PM is IP Flow Information Export (IPFIX)[9,10].
- Hybrid PM combines both active and passive PM. It measures network performance by modifying only certain fields in service packets (e.g., marking certain fields in packet headers). It does not send additional OAM packets to the network for measurement. A typical example of hybrid PM is IP Flow Performance Measurement (IP FPM)[11], which directly monitors real data flows by marking packets. Hybrid PM achieves measurement accuracy comparable to passive PM, thanks to it not generating additional OAM packets on the network.

PM can also be classified into out-of-band and in-band measurements. The former indirectly simulates service data packets and periodically sends them to collect statistics and measure the performance of End-to-End (E2E) paths. Conversely, the latter marks real service packets to collect statistics and measure the performance of real service flows. According to this classification, active measurement is an out-of-band measurement mode, while passive measurement and hybrid measurement are in-band measurement modes.

Take the following analogy as an example. Service flows on a network are like vehicles driving along a highway. Out-of-band measurement is similar to placing monitoring cameras at fixed locations on both sides of the highway. Because the collected data is limited and blind spots may exist between these cameras, the entire journey of vehicles cannot be depicted. In contrast, in-band measurement is similar to installing a dashcam in each vehicle. This enables collection of a vehicle's driving information, which is used to accurately restore the vehicle's driving path. Figure 1.2 compares the two measurement modes.

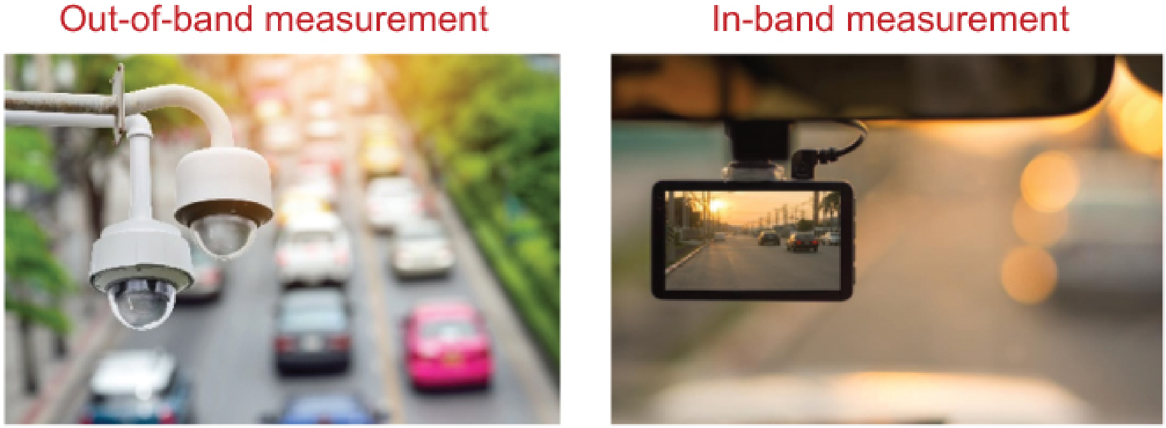


FIGURE 1.2 Comparison between out-of-band measurement and in-band measurement. ⏎

TWAMP, a typical out-of-band measurement technique, was one of the first such techniques proposed. It has been widely adopted thanks to it being easy to deploy. However, it lacks precision and cannot locate specific failure points or reflect the quality of the network that real services traverse.

1.2 BACKGROUND OF IPV6 ON-PATH TELEMETRY

As network services evolve in the 5G and cloud era, IPv6 on-path telemetry techniques have emerged to meet the increasingly stricter Service Level Agreement (SLA) requirements and address challenges in network O&M.

1.2.1 Network O&M Challenges

The services and architecture of IP networks have changed dramatically in the 5G and cloud era. These changes pose significant challenges to network O&M. For example, the development of 5G has given rise to new services, such as High Definition (HD) video, Virtual Reality (VR), and Internet of Vehicles (IoV). And the cloudification of network devices and services has become an inevitable trend to facilitate unified management and reduce O&M costs. As shown in Figure 1.3, new services and architecture pose the following challenges to the current transport network:

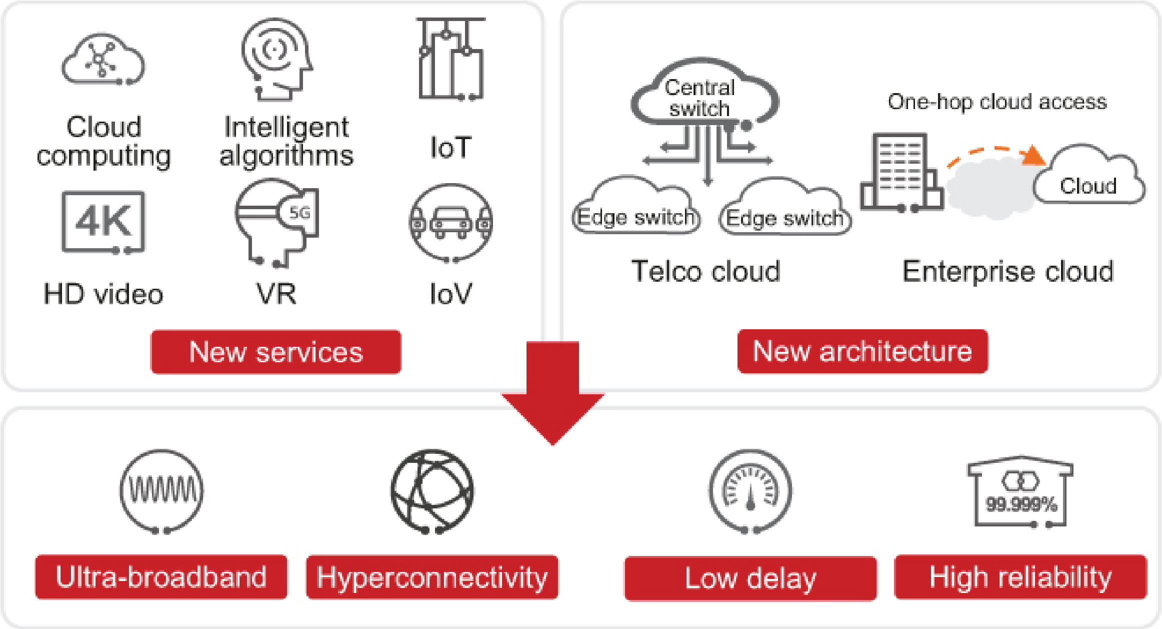


FIGURE 1.3 Challenges posed by new services and architecture to transport networks. ⏎

- Ultra-bandwidth: To carry massive amounts of service data, there is a need for bandwidth to be continuously increased, its utilization maximized, and its growth predictable.
- Hyperconnectivity: To support the vast number of intelligent terminals accessing the network, on-demand dynamic connections and automated service deployment are required. Furthermore, differentiated SLA assurance needs to be implemented for different service connections.
- Low delay: To optimize user experience by delivering smooth and responsive network access, the network delay needs to be slashed from 20 ms to as low as 2 ms. For example, it must be no more than 10 ms for telemedicine, 5 ms for IoV, and 2 ms for industrial control networks. In addition, the delay must be deterministic.
- High reliability: To improve network reliability, proactive fault detection and fast fault demarcation and locating are required, and the network self-healing capability needs to be further developed.

1.2.2 Pain Points of Traditional Network O&M Methods

As shown in Figure 1.4, traditional network O&M methods cannot meet the SLA requirements of new applications in the 5G and cloud era. The two main causes of this are as follows:

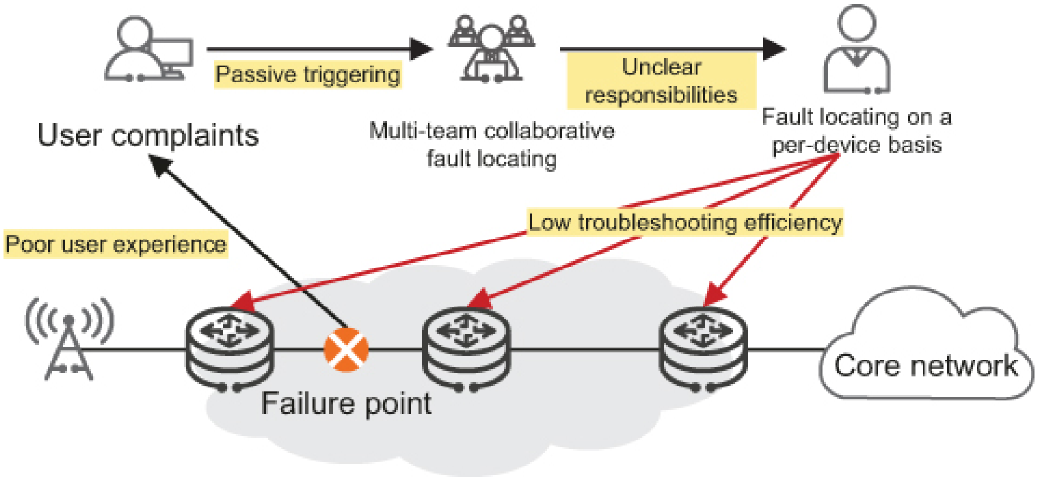

Passive triggering
Unclear responsibilities
User complaints
Multi-team collaborative fault locating
Fault locating on a per-device basis
Low troubleshooting efficiency
Poor user experience
Failure point
Core network

FIGURE 1.4 Pain points in traditional network O&M methods. ⏎

- Passive perception of service loss: O&M personnel often have to rely exclusively on complaints received from users or work orders dispatched by related service departments in order to determine the scope of faults. This approach puts greater pressure on troubleshooting and potentially compromises user experience because O&M personnel cannot perceive faults quickly and can only handle them passively. To resolve this problem, the network needs a service-level SLA measurement method that can actively detect service faults.
- Inefficient fault demarcation and locating: Multiple teams often need to collaborate in order to demarcate and locate faults. However, there is no clear demarcation mechanism to identify their respective responsibilities. Worse yet, troubleshooting is inefficient because devices must be manually checked one by one to identify which one is faulty. The faulty device then needs to be restarted or have its traffic switched to another device. In addition, traditional OAM techniques cannot precisely reproduce performance deterioration or fault scenarios of real services because these techniques use test packets to simulate service flows. As such, high-precision fast measurement based on real service flows is required on the live network.

1.2.3 Emergence of IPv6 On-Path Telemetry

High-precision measurement techniques for IP networks are crucial to address the network challenges brought by the development of 5G and cloud services and the pain points that affect traditional network O&M methods. IPv6 on-path measurement is such a technique. It directly measures network performance indicators, including the delay, packet loss, and jitter, by marking real service flows on a network. This technique works with telemetry to report measurement data in real time and displays quality indicators of the network experienced by users' service flows on the controller Graphical User Interface (GUI).

Request for Comments (RFC) 7799[7] defines on-path measurement as a hybrid PM method. In this method, each processing node needs to collect and process data according to the OAM instructions encapsulated into received packets. This method brings many benefits compared with active measurement — for example, it can measure real user traffic, implement per-packet monitoring, and obtain more data-plane information. And by using on-path measurement, more detailed OAM information can be obtained. Such information includes:

- Paths through which packets are forwarded on the network, including devices and inbound and outbound interfaces.
- Rules matched by packets on each network device that forwards the packets.
- Time taken to cache packets on each network device, accurate to within nanoseconds.
- Flows with which a packet competes for a queue during the queuing process.

Despite the many benefits that IP on-path measurement brings, it also faces several challenges that stem from IP OAM techniques. These challenges include:

- Designing an effective on-path measurement mechanism.
- Providing on-path measurement instructions and information encapsulation modes that can be easily extended in the data plane.
- Developing high-performance data reporting techniques that can handle vast amounts of node-level on-path measurement data.
- Centrally analyzing and processing massive amounts of network-level on-path measurement data.

IPv6 on-path telemetry is a technical system developed to address these technical challenges. It enables large-scale deployment and commercial use of IP on-path measurement techniques.

A representative on-path measurement technique developed during the early stages is IP FPM[11]. It adopts an alternate marking mechanism for on-path packet loss and delay measurement and introduces the novel concept of on-path measurement. IP FPM directly marks specific flag bits in the IP header of service packets, making PM far more accurate. However, it is difficult to deploy and unsuitable for large-scale applications on live networks due to the limited extensibility of the Internet Protocol version 4 (IPv4) header.

The subsequent In-Situ Operation, Administration, and Maintenance (IOAM) [12] further enriches the on-path measurement mechanisms. It implements fine-grained on-path measurement functions such as automatic path discovery, transmission path verification, and per-flow or per-packet measurement of packet loss and delay by defining a range of measurement options, including IOAM Trace, Proof-of-Transit, Edge-to-Edge, and Direct Export[13].

To address the problem of deploying on-path measurement at scale, related mechanisms were further optimized in the Internet Engineering Task Force (IETF), including:

- Optimizing the alternate marking mechanism: Since technologies such as SRv6 emerged around 2017, IPv6 extension headers have become a powerful tool for extending network functions. IPv6 extensions are used to carry on-path measurement instructions and information, resolving issues related to encapsulation scalability and standardization involved in the deployment of the alternate marking mechanism.
- Introducing the intelligent flow selection and reporting suppression mechanisms: Intelligent flow selection allows on-path measurement to be performed for high-priority flows, while reporting suppression ensures that on-path measurement results are reported only in exception scenarios such as path changes. Introducing these mechanisms can reduce the number of service flows for which on-path measurement is performed and the volume of reported data.

Around the same time, the use of telemetry gained wide attention. Telemetry is a high-performance technique that can remotely and quickly collect data from physical or virtual devices. With telemetry, devices can send information — including traffic statistics, Central Processing Unit (CPU) usage, and memory usage — to collectors in push mode, which enables data collection to be more real-time and faster than the conventional pull mode (request/response interaction).

To systematize network telemetry techniques, the Network Telemetry Framework (NTF)[14] was defined in the IETF. The NTF is used to classify and organize different telemetry data sources and types, define different components of a network telemetry system and their interactions, and facilitate cross-layer coordination and integration of multiple telemetry methods. An important source of telemetry data in the data plane is on-path measurement. And telemetry is well-suited to meet the requirements of reporting on-path measurement data. It includes data reporting mechanisms such as the distributed telemetry mechanism[15], User Datagram Protocol (UDP)–Notif[16], Google Remote

Procedure Call (gRPC)[17], and IPFIX[9,10]. Telemetry flexibly collects data by subscribing to different sampling paths, enabling the controller to manage more devices and obtain higher precision measurement data. This provides the big data necessary for quickly locating network faults and optimizing network quality.

Through ongoing evolution and optimization, on-path measurement is ready for large-scale commercial use, especially after being combined with IPv6 extension and telemetry mechanisms. At the same time, dynamic protocol extensions are required to better support its deployment and application on live networks. This is necessary to advertise the on-path measurement capabilities of network devices for implementing automated deployment by properly planning the service flows (and their forwarding paths) for which on-path measurement is performed. The comprehensive technical system described thus far is referred to as IPv6 on-path telemetry in this book.

In summary, a complete IPv6 on-path telemetry solution covers not only multiple types of on-path measurement mechanisms but also high-performance data reporting mechanisms, optimization mechanisms that can minimize on-path measurement data reporting, and automation mechanisms that can improve deployment efficiency. As such, the In-situ Flow Information Telemetry (IFIT) framework[18,19] — serving as an implementation framework of IP on-path telemetry — was first proposed in the IETF and officially released by the European Telecommunications Standards Institute (ETSI).

Building upon the existing on-path telemetry techniques, IFIT extends and improves on-path telemetry mechanisms. It forms a complete technical framework that encompasses on-path measurement, intelligent flow selection, data reporting suppression, dynamic probe, and automated deployment. This allows on-path telemetry mechanisms to be deployed and applied.

Currently, standards organizations including the IETF, ETSI, and China Communications Standards Association (CCSA) are ramping up efforts to formulate on-path telemetry standards in order to facilitate the large-scale commercial use of on-path telemetry.

1.3 TECHNICAL BENEFITS OF IPV6 ON-PATH TELEMETRY

IPv6 on-path telemetry accurately measures various quality indicators of the network through which real services pass, supports flexible deployment on networks that carry multiple types of services, and allows centralized analysis and visualization of measurement results. Furthermore, it can be used to build a closed-loop intelligent O&M system by combining big data analytics and intelligent algorithm capabilities.

1.3.1 High-Precision and Multi-Dimensional Quality Measurement of Real Services

Traditional OAM techniques use test packets that may be forwarded along paths different from those used for real service flows. Figure 1.5 compares out-of-band measurement and IPv6 on-path telemetry. IPv6 on-path telemetry provides the following advantages:

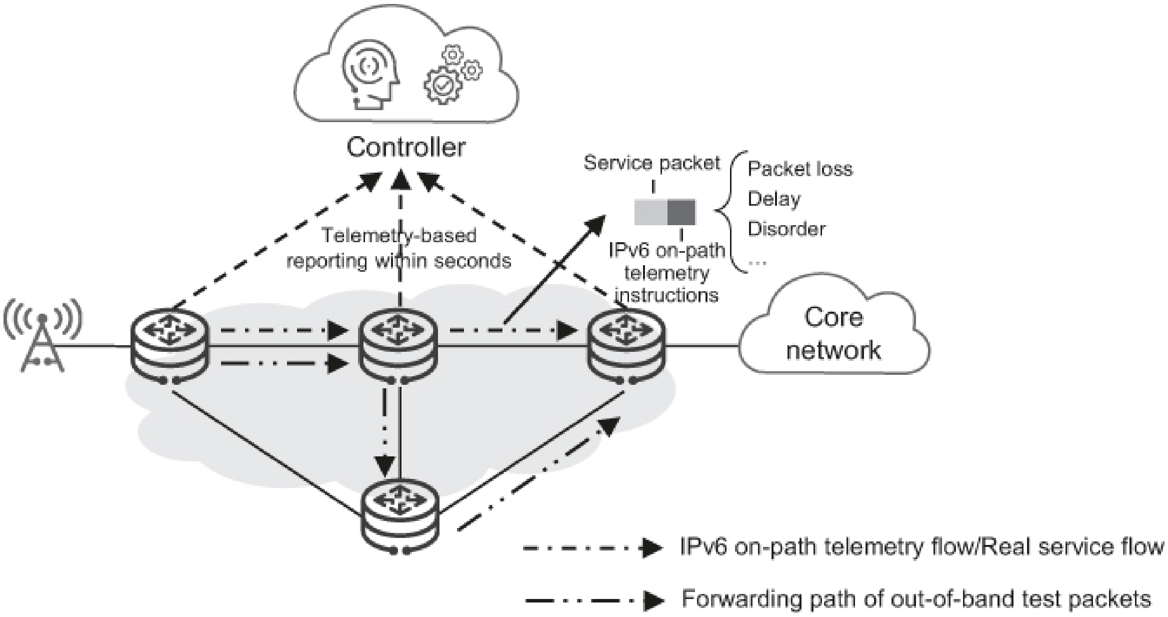

Controller
Telemetry-based reporting within seconds
Service packet
IPv6 on-path telemetry instructions
Packet loss
Delay
Disorder
...
Core network
IPv6 on-path telemetry flow/Real service flow
Forwarding path of out-of-band test packets

FIGURE 1.5 Comparison between out-of-band measurement and IPv6 on-path telemetry. ⏎

- Traces the actual forwarding path of packets and accurately measures the performance of each service in multiple dimensions, such as delay, packet loss, and packet disorder. The precision of packet loss measurement can reach 10^{-6}, while that of delay measurement can reach microseconds.
- Monitors network SLAs in real time and quickly demarcates and locates faults by working together with telemetry, which enables data collection within seconds.
- Identifies minor exceptions on the network and detects the loss of even one packet. Such precise packet loss measurement meets the requirements of “zero-packet-loss” services like accounting, telemedicine, industrial control, and power differential protection, ensuring high reliability for such services.

1.3.2 Flexible Adaptation to Multi-Type Service Scenarios

Network development is usually a lengthy process. Consequently, as requirements evolve, one network may contain multiple device types and carry various service types. Addressing this issue, IPv6 on-path telemetry is easy to deploy and can flexibly adapt to multi-type service scenarios, as shown in Figure 1.6. It has the following highlights:

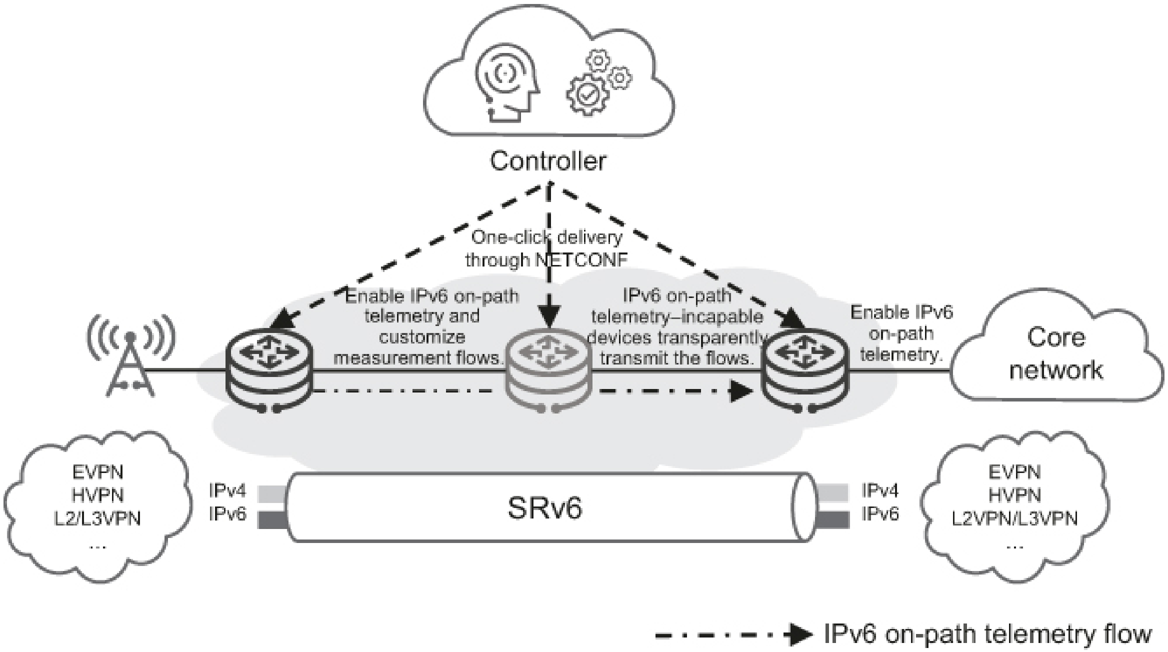

Controller
One-click delivery through NETCONF
Enable IPv6 on-path telemetry and customize measurement flows.
IPv6 on-path telemetry–incapable devices transparently transmit the flows.
Enable IPv6 on-path telemetry.
Core network
EVPN
HVPN
L2/L3VPN
...
IPv4
IPv6
SRv6
IPv4
IPv6
EVPN
HVPN
L2VPN/L3VPN
...
IPv6 on-path telemetry flow

FIGURE 1.6 Flexible adaptation of IPv6 on-path telemetry to multi-type service scenarios. ⏎

- Supports one-click delivery of network-wide configuration. E2E measurement and hop-by-hop measurement need to be customized only on the ingress of a network path, while IPv6 on-path telemetry can simply be enabled once on transit and egress nodes. In this way, IPv6 on-path telemetry can be implemented on large networks with many devices.
- Supports measurement of both specific service flows and E2E private line traffic. IPv6 on-path telemetry instances can be generated through user configuration (static measurement flows) or through either automatic learning or traffic with IPv6 on-path telemetry instructions (dynamic measurement flows). These instances can be instantiated for specific flows created based on unique information (such as IP 5-tuple), tunnel-level aggregation flows, or Virtual Private Network (VPN)-level aggregation flows.

- Features good compatibility with existing networks, even those with a diverse range of device types. Devices that lack support for IPv6 on-path telemetry can transparently transmit IPv6 on-path telemetry flows, ensuring seamless operation in scenarios featuring interconnection with third-party devices.
- Learns actual forwarding paths automatically. This eliminates the need for pre-defining a forwarding path to deploy measurement hop by hop for all Network Elements (NEs) along the path, simplifying planning and deployment.
- Supports a wide range of Layer 2 and Layer 3 networks and tunnels, meeting diverse requirements on the live network.

1.3.3 Visualized O&M Capabilities

Without visualization, O&M is inefficient because network O&M personnel need to manually configure devices one by one, and multiple departments then need to cooperate to check each item. Visualized O&M provides centralized management and control capabilities, supports online service planning and one-click deployment, and enables quick fault demarcation and locating through SLA visualization. IPv6 on-path telemetry can provide visualized O&M capabilities, as shown in Figure 1.7. With such capabilities, users can deliver different IPv6 on-path telemetry policies through the controller GUI to implement routine proactive O&M and quick fault handling. The following further explains this by using a 5G transport network as an example.

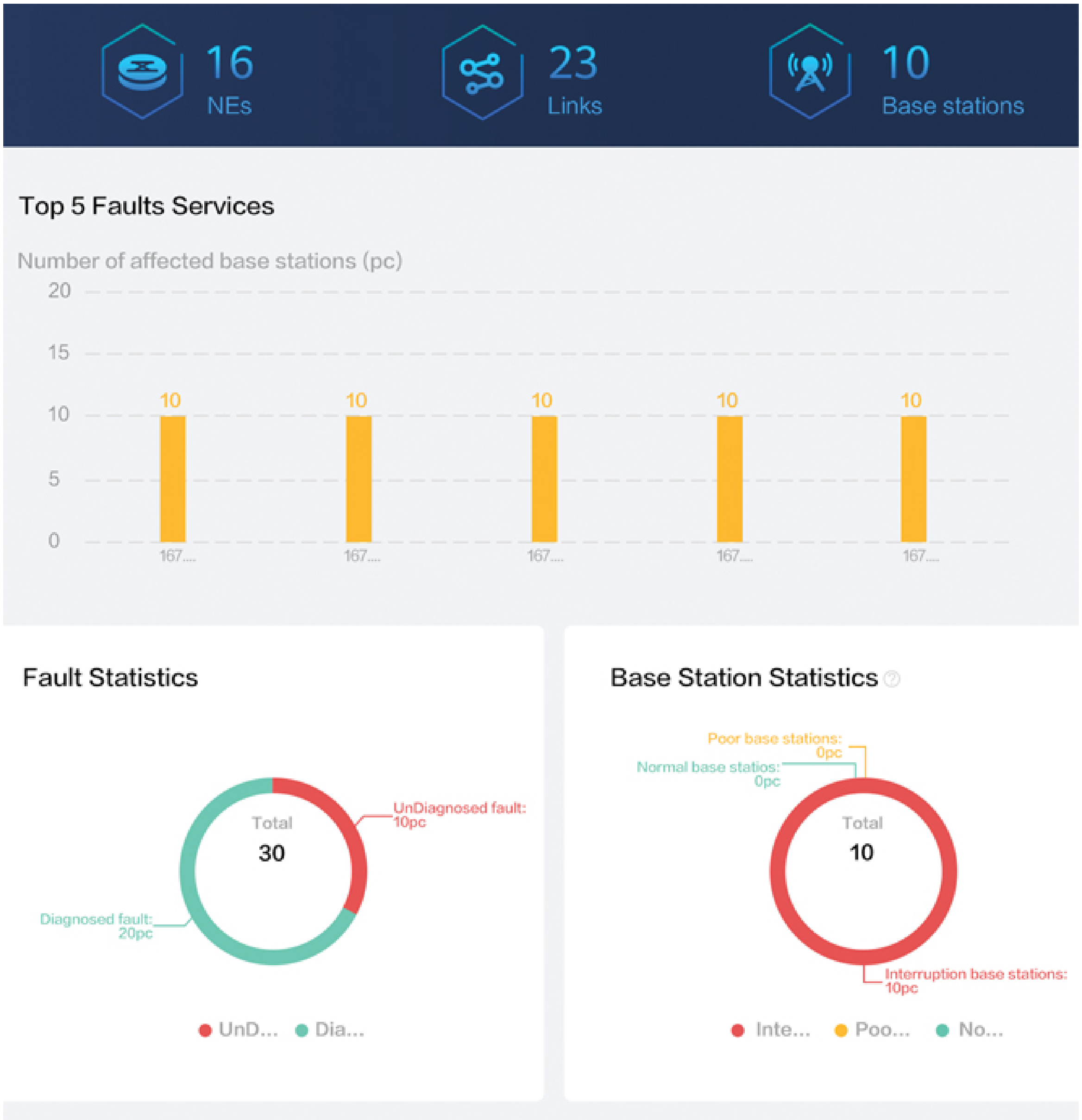


FIGURE 1.7 Visualized O&M capabilities provided by IPv6 on-path telemetry.

- Routine O&M: O&M personnel can routinely monitor the status of base stations, network fault trends, abnormal trends in base stations, and top five faults affecting base stations network-wide and per area. This enables O&M personnel to promptly learn about base station service status changes and the top faults across the network and in key areas through performance reports. And in VPN scenarios, detailed information about E2E service flows is provided to help O&M personnel identify and locate potential faults and ensure the overall SLAs of private line services.

- Quick troubleshooting: Upon receiving a fault report, O&M personnel can view the service topology and hop-by-hop flow indicators of IPv6 on-path telemetry simply by searching for the base station name or IP address. This allows them to quickly rectify the fault based on the fault location, possible causes, and rectification suggestions. They can also view information about topology paths and historical fault locating information collected over the past seven days.

Figure 1.7 provides an example showing the IPv6 on-path telemetry information displayed on the controller GUI. This visualized information achieves a superior O&M experience by helping users learn about the network status and quickly detect and rectify faults.

1.3.4 Closed-Loop Intelligent O&M System

The evolution of network architecture and services poses new challenges to transport networks. In particular, traditional O&M methods must be improved in order to deliver a high-quality network experience from end to end. To achieve this, it is necessary to transition from passive to proactive O&M and build an intelligent O&M system. Such a system will proactively detect exceptions in real services, automatically demarcate faults, implement fast fault locating and self-healing, and more. This helps create automated processes that can adapt to complex and changing network environments.

As shown in Figure 1.8, IPv6 on-path telemetry can be combined with technologies such as big data analytics and intelligent algorithms to build an intelligent O&M system. The working process of the system is as follows:

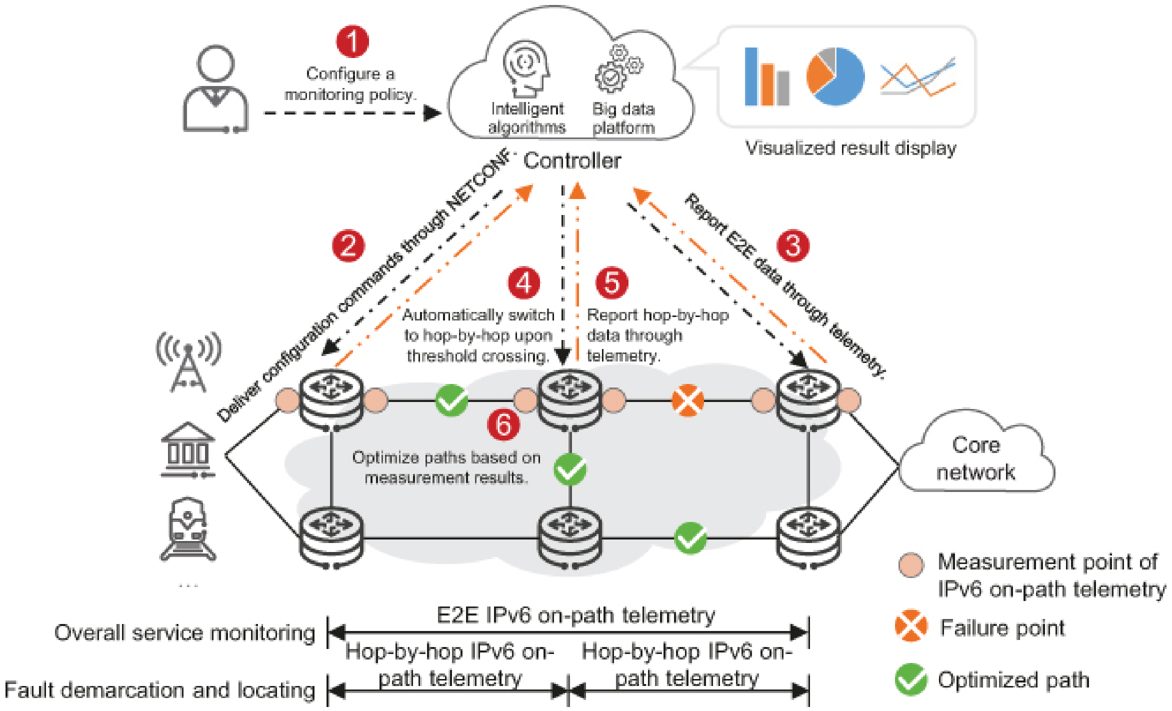

1
Configure a monitoring policy.
Intelligent algorithms
Big data platform
Controller
Visualized result display
2
Deliver configuration commands through NETCONF
3
Report E2E data through telemetry.
4
Automatically switch to hop-by-hop upon threshold crossing.
5
Report hop-by-hop data through telemetry.
6
Optimize paths based on measurement results.
Core network
...
Measurement point of IPv6 on-path telemetry
Failure point
Optimized path
E2E IPv6 on-path telemetry
Overall service monitoring
Hop-by-hop IPv6 on-path telemetry
Hop-by-hop IPv6 on-path telemetry
Fault demarcation and locating

FIGURE 1.8 Closed-loop intelligent O&M system combining IPv6 on-path telemetry with multiple technologies. ⏎

1. A user enables the IPv6 on-path telemetry capability across the network through the controller. The user can then configure a telemetry-based subscription, select the required ingress, egress, and links for a service, and configure an IPv6 on-path telemetry monitoring policy.
2. The controller converts the monitoring policy into configuration commands and delivers them to devices through the Network Configuration Protocol (NETCONF).
3. Devices generate E2E IPv6 on-path telemetry monitoring instances. The ingress and egress use telemetry to report service SLA data within seconds to the controller, which then processes the data using the big data platform and displays the measurement results on its GUI. E2E monitoring mode enables holistic monitoring of network services. In E2E monitoring mode, the overall monitoring of network services can be implemented.
4. If the packet loss or delay exceeds the defined monitoring thresholds, the controller automatically adjusts the monitoring policy from E2E to hop-by-hop and delivers it to devices through NETCONF.
5. After receiving the updated policy, the devices adjust the service monitoring mode to hop-by-hop and report service SLA data to the controller hop by hop through telemetry within seconds. The controller then processes the data using the big data platform and displays the measurement results on its GUI.

In hop-by-hop monitoring mode, network faults can be demarcated and located.

6. The controller performs intelligent analysis based on service SLA data and identifies potential root causes based on exception information obtained through device Key Performance Indexes (KPIs), logs, and more. It also provides handling suggestions and assigns work orders. In addition, the controller optimizes service paths to ensure service quality and implement fault self-healing.

In the preceding process, the results obtained through E2E and hop-by-hop monitoring using IPv6 on-path telemetry serve as data sources for the big data platform and intelligent algorithm analysis. These results underpin the ability of the intelligent O&M system to implement precise fault demarcation and locating and fast fault self-healing. The big data platform enables data queries within seconds and can efficiently process massive amounts of measurement data from IPv6 on-path telemetry. Furthermore, no data loss occurs if a single node fails, ensuring efficient and reliable data analysis and conversion. The intelligent algorithm can cluster poor-Quality of Experience (QoE) events into mass network faults by calculating the path similarity of poor-QoE service flows within the same period, thereby locating the common failure point. Specifically, poor-QoE service flows whose indicator reaches the algorithm threshold are considered to be caused by the same fault. This approach is more than 90% accurate, making O&M more efficient while also slashing the service interruption time. By combining the preceding techniques, it is possible to guarantee the closed-loop process of the intelligent O&M system and promote the optimization of intelligent O&M solutions, enabling them to adapt to future network evolution.

1.4 STORIES BEHIND IPV6 ON-PATH TELEMETRY

1.4.1 Development History of IP OAM

IP OAM has a long history, with the well-known IP ping and traceroute mechanisms in use for decades. The emergence of MPLS has driven the development of MPLS OAM, which includes the LSP ping and traceroute mechanisms and BFD.

MPLS OAM is primarily used for FM. Around 2006, PM became an important topic in the IP/MPLS OAM technical field, giving rise to many new mechanisms and standards. Through the cooperation and competition between two major standards organizations — International Telecommunication Union-

Telecommunication Standardization Sector (ITU-T) and IETF — the following standards systems emerged:

- ITU-T: Transport MPLS (T-MPLS).
- IETF: MPLS Transport Profile (MPLS-TP).

These two standards systems aimed to not only provide PM mechanisms but also establish an OAM system. The ITU-T hoped to establish an MPLS OAM system based on the Ethernet OAM system defined in Y.1731, whereas the IETF wanted to integrate existing mechanisms such as LSP ping/traceroute and BFD, with a focus on defining protocol extensions and standards for the missing PM mechanism. The competition between these two technical standards systems ultimately impaired the development, commercial delivery, and deployment of the MPLS OAM technical system.

While the PM system based on MPLS OAM was being developed, little consideration was given to the on-path measurement mechanism. The problems associated with out-of-band measurement were generally addressed by binding with non-load-balanced MPLS Traffic Engineering (TE) tunnels and by disabling Penultimate Hop Popping (PHP).

Around 2015, Huawei, Telecom Italia, and industry partners worked together to propose the IP FPM mechanism[20], which utilized the alternate marking method to implement on-path measurement of packet loss rate and delay. In terms of engineering, the alternative marking method was a feasible option because it provided controllability over the volume of data involved in on-path measurement. The method had been implemented on Huawei routers and switches based on IPv4. Because the IPv4 header had virtually no available fields, it was difficult to implement even two-bit-based marking. In the end, the solution employed two unused bits — one from the Type of Service (ToS) field and one from the Flag field in the IPv4 header — under the premise of clear network planning. However, this solution still brought some compatibility issues. And because only two bits were available, there was no space to carry flow identifiers if on-path measurement was required for specific flows. Consequently, the flows must be identified only through 5-tuple information, requiring every node along the path through which packets pass to be configured with such information. Furthermore, 5-tuple-based on-path measurement needs to be configured on almost every device, network-wide, given the potential switching of traffic paths when network faults occur. The configuration complexity has had a negative impact on the deployment and application of IP FPM.

Around 2016, the IETF proposed the IOAM mechanism[21], and the In-band Network Telemetry (INT) mechanism emerged[22], both of which have helped promote the further development of the on-path measurement mechanism. Initially, IOAM supported only on-path measurement in passport mode, but the postcard method[23] was later introduced to make the on-path measurement mechanism more complete.

Unlike IP FPM, the IOAM header was designed from the outset with an extensive range of fields, which cannot fit within IPv4. To support IOAM functions, the corresponding encapsulation mechanism needs to be defined and standardized — the IPv6 extension header mechanism[24] is a better choice, thanks to it being more mature than other encapsulation mechanisms.

Although IOAM has well-established mechanisms and standards in place, its per-flow and per-packet information reporting pose significant challenges to the system. In contrast, the combined use of the alternate marking method and IPv6 enables more information (such as flow identifiers) to be conveniently carried through IPv6 extension headers than is possible in IPv4. This eliminates the need for additional configuration of flow identification when performing on-path measurement because flow identifiers can be carried in packets, simplifying the configuration of the original IPv4-based alternate marking method. In addition, because the amount of reported data is controllable, the on-path measurement mechanism can be effectively deployed at scale and put into commercial use. Ultimately, the IP on-path measurement mechanism has opened up a new chapter.

1.4.2 Experience and Lessons of IP OAM Development

A review of the development history of IP OAM mechanisms and standards at the IETF, coupled with my many years of experience at the IETF, suggests that some important OAM-related work at the IETF has met with limited success. This is true for not only IP OAM but also for other standardization efforts in Simple Network Management Protocol (SNMP)/Management Information Base (MIB) and NETCONF/Yet Another Next Generation (YANG), to name a few. As a result, it may seem that the IETF is not highly capable at such work. Based on my experiences, the following lessons can be learned.

1.4.2.1 Organizational Level

The IETF is a loose standards organization that operates according to a bottom-up approach, making it challenging to conduct systematic standardization work. This was particularly evident in the IETF's standardization efforts regarding IP OAM, SNMP/MIB, and NETCONF/YANG. IP OAM involves complex mechanisms that require a reasonable top-level design and a robust organization to ensure task division and implementation. Despite this, a system-level task was distributed across dozens of Working Groups (WGs) in the IETF, operating in a loose manner without cohesive top-level design and collaboration. Consequently, this led to the OAM mechanisms and models being incomplete, or greatly delayed the standardization of dependent mechanisms and models because the key mechanisms and models faced repeated delays in standardization or underwent changes. Due to these issues, the IETF standards could not completely address real-world application needs, forcing users to rely on vendor-specific implementations. And while the IETF invested significant effort into developing standards, the opportunity to implement them had already passed, leaving users unmotivated to adopt these new mechanisms and standards.

On top of these issues, the IETF lacks enough expertise in the telecommunications sector compared with the ITU-T. The IETF has grown together with the Internet, which develops rapidly yet faces severe O&M deficiencies. And as the Internet has grown, it has also faced intense competition from traditional telecom networks supported by the ITU-T. These brought about issues such as the following:

- IETF experts lacked sufficient understanding of telecom networks.
- The IETF had a disapproving attitude toward the ITU-T.

Although personnel involved in traditional telecom networks had more extensive OAM experience, the IETF lacked expert experience and was reluctant to adopt ITU-T standards. This ultimately hampered the standardization of OAM technologies in the IETF.

1.4.2.2 Technology Level

Another important factor hindering the development of the IP OAM technical system is the technical challenges it faces. IP OAM is significantly more complex than the Point-to-Point (P2P) connection-based OAM of traditional telecom networks due to the flexibility of IP. Both IP OAM and MPLS OAM face the following typical challenges:

- Load balancing issue: This encompasses Equal-Cost Multiple Path (ECMP) and trunk. In load balancing scenarios where out-of-band OAM is used, OAM packets and service packets may take different paths. As a result, the OAM detection result may not accurately reflect the service experience.
- Inconsistent forward and return paths: MPLS tunnels are unidirectional. Consequently, when an MPLS tunnel is used for forward BFD packets but an IP path is used for the return BFD packets, traffic on the forward path

may be incorrectly switched over based on the BFD detection result due to the failure on the return path.

- PHP issue: Although MPLS supports PHP, the egress cannot identify flows or collect packet statistics based on the label of an MPLS-encapsulated service packet when this label is popped at the penultimate hop.
- Inability to distinguish sources in Multi-Point to Point (MP2P) mode: In MPLS Label Distribution Protocol (LDP) or MPLS VPN scenarios, labels distributed by a downstream node are sent to multiple upstream nodes. As a result, flows from different sources cannot be identified based on labels, and packet statistics cannot be collected.
- OAM complexity arising from the hierarchical network service architecture: Service transmission paths include native IP paths, MPLS tunnels, nested/stitched MPLS tunnels, MPLS VPN tunnels, and nested/stitched MPLS VPN tunnels. Such complex path transmission increases the complexity in OAM detection.

In summary, IP is essentially a Multipoint-to-Multipoint (MP2MP) service model. Using out-of-band OAM methods inherently introduces certain defects or issues. As such, on-path measurement has emerged as the only viable approach for resolving IP OAM issues. However, in an era where IP extensibility is limited, the only possible way is to make improvements based on out-of-band OAM methods until the IPv6-powered enhanced innovation era arrives. The extensibility offered by IPv6 creates opportunities to fundamentally solve the technical problems of IP OAM.

REFERENCES

1. ITU-T. Operation, Administration and maintenance (OAM) functions and mechanisms for ethernet-based networks (EB/OL). (2019-08-29) [2024-09-30]. G.8013/Y.1731. ⏎
2. Mizrahi T, Sprecher N, Bellagamba E et al. An overview of operations, administration, and maintenance (OAM) tools [EB/OL]. (2014-06) [2024-09-30]. RFC 7276. ⏎
3. Conta A, Deering S, Gupta M. Internet control message protocol (ICMPv6) for the internet protocol version 6 (IPv6) specification [EB/OL]. (2017-07-14) [2024-09-30] RFC 4443. ⏎
4. Katz D, Ward D. Bidirectional forwarding detection (BFD)[EB/OL]. (2020-01-21) [2024-09-30]. RFC 5880. ⏎

5. Kompella K, Swallow G. Detecting multi-protocol label switched (MPLS) data plane failures [EB/OL]. (2006-02) [2024-09-30] RFC 4379. ⏎
6. Ali Z, Filsfils C, Matsushima S et al. Operations, administration, and maintenance (OAM) in segment routing over IPv6 (SRv6) [EB/OL]. (2022-06-23) [2024-09-30]. RFC 9259. ⏎
7. Morton A. Active and passive metrics and methods (with hybrid types in-between) [EB/OL]. (2016-05) [2024-09-30]. RFC 7799. ⏎
8. Hedayat K, Krzanowski R, Morton A et al. A two-way active measurement protocol (TWAMP) [EB/OL]. (2020-01-21) [2024-09-30]. RFC 5357. ⏎
9. Claise B, Trammell B, Aitken P. Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information [EB/OL]. (2013-09) [2024-09-30]. RFC 7011. ⏎
10. Sadasivan G, Brownlee N, Claise B. Architecture for IP flow information export [EB/OL]. (2009-03)[2024-09-30]. RFC 5470. ⏎
11. Fioccola G, Capello A, Cociglio M et al. Alternate-marking method for passive and hybrid performance monitoring [EB/OL]. (2018-01-29) [2024-09-30]. RFC 8321. ⏎
12. Brockners F, Bhandari S, Mizrahi T. Data fields for in situ operations, administration, and maintenance (IOAM)[EB/OL]. (2022-05) [2024-09-30]. RFC 9197. ⏎
13. Song H, Gafni B, Brockners F et al. In situ operations, administration, and maintenance (IOAM) direct exporting [EB/OL]. (2022-11-15) [2024-09-30]. RFC 9326. ⏎
14. Song H, Qin F, Martinez-Julia P et al. Network telemetry framework [EB/OL]. (2022-05-27) [2024-09-30]. RFC 9232. ⏎
15. Zhou T, Zheng G, Voit E et al. Subscription to distributed notifications [EB/OL]. (2024-04-28) [2024-09-30]. Draft-ietf-netconf-distributed-notif-09. ⏎
16. Zheng G, Zhou T, Graf T et al. UDP-based transport for configured subscriptions [EB/OL]. (2024-07-04) [2024-09-30]. Draft-ietf-netconf-udp-notif-14. ⏎
17. Google. What is gRPC? [EB/OL]. (2024-09) [2024-09-30]. ⏎
18. Song H, Qin F, Chen H. Framework for in-situ flow information telemetry [EB/OL]. (2024-04-25)[2024-09-30]. Draft-song-opsawg-ifit-framework-21. ⏎
19. ETSI GR ENI 012. Reactive in-situ flow information telemetry [EB/OL]. (2022-03) [2024-09-30]. ⏎
20. Chen M, Zheng L, Mirsky G et al. IP flow performance measurement framework [EB/OL]. (2016-09-18) [2024-09-30]. Draft-chen-ippm-coloring-

based-ipfpm-framework-06. ⏎

21. Brockners F, Bhandari S, Pignataro C et al. Data fields for in-situ OAM [EB/OL]. (2018-01-03) [2024-09-30]. Draft-brockners-inband-oam-data-07. ⏎
22. In-band Network Telemetry (INT) dataplane specification v2.1 [EB/OL]. (2020-11-11) [2024-09-30]. ⏎
23. Song H, Gafni B, Zhou T et al. In-situ OAM direct exporting [EB/OL]. (2020-04-14) [2024-09-30]. Draft-ioamteam-ippm-ioam-direct-export-00. ⏎
24. Bhandari S, Brockners F. IPv6 options for in situ operations, administration, and maintenance (IOAM) [EB/OL]. (2023-09) [2024-09-30]. RFC 948 ⏎

II

CHAPTER 2

IPv6 On-Path Telemetry Architecture

DOI: 10.1201/9781003677901-4

FOLLOWING A SYSTEMATIC REVIEW of telemetry technology, the Network Telemetry Framework (NTF)[1] was proposed in the IETF. Designed to facilitate the development of network operation applications, this framework defines four modules — management plane telemetry, control plane telemetry, data plane telemetry, and external data and event telemetry — as well as their functional components and interactions. The IFIT framework was later proposed in the IETF for implementing IP network on-path telemetry, making it possible to deploy and apply the on-path telemetry mechanism. This chapter describes the two frameworks in detail.

2.1 NETWORK TELEMETRY FRAMEWORK

Network telemetry is a method for acquiring in-depth network information and involves a variety of techniques for purposes such as remote data generation, collection, correlation, and consumption. Such in-depth information not only helps improve network operation efficiency but also facilitates automated network management. However, given that network telemetry involves multiple techniques and deploying it in real-world O&M scenarios presents many challenges, a unified network telemetry framework is required.

2.1.1 Definition of Network Telemetry

Network carriers have long relied on SNMP[2], Command Line Interface (CLI), or Syslog[3] to acquire network information. Some OAM techniques described in RFC 7276 are also used for network troubleshooting[4]. However, these conventional techniques are unable to fully support existing network applications due to the following reasons:

- Most network applications need to continuously monitor the network, making polling-based low-frequency data collection unsuitable. A better option is to enable a data source (e.g., a forwarding chip) to proactively push subscription-based streaming data. Such an approach can provide a sufficient amount of data with the necessary precision.
- Network applications need to probe a wide range of data, from packet processing engines to traffic managers, line cards to main control boards, user flows to control protocol packets, device configurations to operations, and physical layers to application layers. However, conventional OAM covers only a limited range of data. For example, SNMP processes data from only the MIB. In addition, conventional network devices cannot provide all the necessary probes, highlighting the need for more open and programmable network devices.
- In many application scenarios, it is necessary to correlate network-wide data from multiple sources, such as distributed network devices, different components of a network device, or different network planes. However, conventional solutions are often piecemeal and cannot achieve this.
- Some conventional OAM techniques (e.g., CLI and Syslog) lack formal data models. Unstructured data impedes tool automation and application extensibility. Standardized data models are critical to supporting programmable networks.
- Although some conventional OAM techniques like SNMP Trap[5,6], Syslog, and Sampled Flow (sFlow)[7] support data push, the pushed data is limited to only predefined management plane warnings (e.g., SNMP Trap) or sampled user packets (e.g., sFlow). Network carriers need data from any source, at any granularity, and with any precision, but existing techniques cannot deliver this.
- Conventional techniques used for active measurement often interfere with user traffic and provide only indirect measurement results, while those used for passive measurement often consume excessive network resources and generate a lot of redundant data or provide inaccurate measurement results. It is far more desirable to directly collect data from user traffic on demand. These issues have been partially resolved by existing standards and techniques, such as IPFIX, NetFlow, Packet Sampling Protocol (PSAMP), IOAM, and YANG Push. However, these standards and techniques need to be recognized and adapted to a new framework.

Network telemetry is becoming more and more important given the emergence of automated network O&M. As its name suggests, telemetry is a technique for remotely acquiring measurement parameters. Examples of its use include those in the aerospace and geological fields, where it is used to acquire satellite or sensor data. When applied to a network, telemetry is used to remotely collect network node parameters. This automated network measurement and data collection technique measures and collects information about remote nodes and provides abundant, reliable data in real time for the information analysis system, playing an important role in the closed-loop network service control system.

The core of network telemetry is network data, which can be generated and acquired through multiple techniques to provide information about network devices, network planes (i.e., data, control, and management planes), and network states. Different data analysis techniques can be used to process network data for purposes such as network service assurance and network security protection.

Any data that can be extracted from a network (including the data, control, and management planes) is considered telemetry data. This data can be used to enhance network visibility and guide which actions to implement. It includes statistics, event records and logs, state snapshots, and configuration data, and covers the outputs of active or passive measurements. In some scenarios, data is processed on the network before being sent to data consumers — this data can also be considered telemetry data. If the cost is acceptable, small amounts of high-quality data may be more valuable than large amounts of low-quality data.

Network telemetry encompasses both the telemetry data itself and the techniques adopted in data generation, output, collection, and consumption. It offers higher flexibility, extensibility, and accuracy in comparison to conventional network OAM.

Currently, multiple network telemetry techniques and protocols (e.g., IPFIX and gRPC) have been widely deployed. Network telemetry allows independent entities to acquire data from network devices for visualization and analysis, facilitating network monitoring and O&M. It has a wider scope than conventional network OAM and can, for example, provide abundant data to support network automation and overcome the limitations of conventional OAM techniques. Unlike some conventional OAM tools, which are designed to help human operators monitor and diagnose networks and instruct them to manually operate networks, network telemetry typically assumes that machines (instead of human operators) are data consumers, meaning that it can directly trigger automated network operations.

Although new network telemetry techniques have emerged and evolved, the following characteristics of network telemetry are widely accepted:

- Push and streaming: Telemetry collectors do not poll data from network devices. Instead, they subscribe to streaming data pushed from data sources in network devices.
- Volume and velocity: Telemetry data is intended to be processed by machines rather than humans. This means that there may be large volumes of data, resulting in the need to optimize real-time automated processing.
- Normalization and unification: In order to meet requirements for network automation, telemetry needs data representations to be normalized and protocols to be unified. This is necessary to simplify data analysis and provide integrated analysis of heterogeneous devices and data sources across networks.
- Model-based: Telemetry data is pre-modeled, allowing applications to easily configure and use the data.
- Data fusion: The data for a single application can come from multiple data sources, such as cross-domain, cross-device, and cross-layer data sources. These data sources share a common name or Identifier (ID) and take effect only after being correlated.
- Dynamic and interactive: Because network telemetry is used in a closed control loop for network automation, it needs to run continuously and adapt to the dynamic and interactive queries from network controllers.

An ideal network telemetry solution may also have the following characteristics or properties in addition to the preceding ones:

- In-network customization: During network operation, generated data can be customized to meet the specific requirements of applications. This requires the support of a programmable data plane so that probes with custom functions can be deployed in flexible locations.
- In-network data aggregation and correlation: Network devices and aggregation points can perform calculations to determine which events and what data need to be stored, reported, or discarded. This helps reduce the load on central collection and processing points.
- In-network processing: Gathering all information to a central point for processing may not always be necessary or possible. In such cases, data processing can be completed in the network to ensure a quick local response.
- Direct data plane export: Data generated by data plane forwarding chips can be directly exported to data consumers for improved efficiency, especially when the data bandwidth is high and real-time processing is required.
- In-band data collection: In addition to the passive and active data collection methods, there is a new hybrid method that allows the data of all target flows to be directly collected on the entire forwarding path.

Note that a network telemetry system should not interfere with normal network operations. This is necessary to avoid the observer effect, which refers to an object being disturbed simply by being observed. Specifically, the network telemetry system should not change the network behavior nor affect the forwarding performance. Another point to note is that the high volume of traffic involved in network telemetry may cause congestion. This issue can be addressed by using appropriate isolation or traffic engineering techniques, reducing the amount of reported data through local processing, or employing a congestion control mechanism to ensure that telemetry traffic backs off if it exceeds the network capacity.

2.1.2 Overall Architecture of Network Telemetry

Network telemetry enables the automation of network O&M and requires rich and coherent network data. In many cases, static data collection that relies on a single data source is not sufficient to meet the telemetry data needs of applications. It is therefore necessary to integrate multiple data sources involving various techniques and standards. This requires a framework that can classify and organize different telemetry data sources and types, define different components of a network telemetry system and their interactions, and help to coordinate and integrate multiple telemetry methods across layers. Under this framework, different applications can flexibly combine telemetry data, and interfaces are normalized and simplified. In particular, such a framework facilitates the development of network operation applications for the following reasons:

- Future network O&M depends on comprehensive network visualization. An integrated, converged mechanism and common telemetry data should be used, if possible, to provide uniform and coherent support for applications. Protocols and mechanisms should therefore be integrated into a minimal but comprehensive set. Implementing a telemetry framework will help normalize the development of techniques.
- Network visibility presents diverse viewpoints. For example, the device viewpoint provides insights into the network topology and device status using the network infrastructure as its monitoring object, whereas the traffic viewpoint provides visualization of the traffic quality and path using flows or packets as its monitoring object. During operation, an application may need to switch between these viewpoints. It may also need to correlate services and their impact on user experience in order to acquire comprehensive information.
- To efficiently use network resources and reduce the impact of network telemetry-related processing on network performance, applications need

elastic network telemetry. For example, routine network monitoring should cover the entire network with a low data sampling rate. Telemetry data sources should be modified and telemetry data rates increased only when issues occur or critical trends emerge.

- Efficient data aggregation is critical for applications to reduce the total amount of data and improve analysis accuracy. A telemetry framework incorporates various telemetry techniques, enabling a comprehensive network telemetry system to be assembled and eliminating repetitive or unnecessary work.

RFC 9232 defines a two-level network telemetry architecture[1], which is described as follows:

- Network telemetry can be applied either to the data, control, and management planes on the network or to other sources outside the network, and is therefore partitioned into four top-level modules — management plane telemetry, control plane telemetry, data plane telemetry, and external data and event telemetry — as shown in Figure 2.1. Each module has its own network application programming interface. Network applications can acquire data from these modules, analyze data, and perform actions.

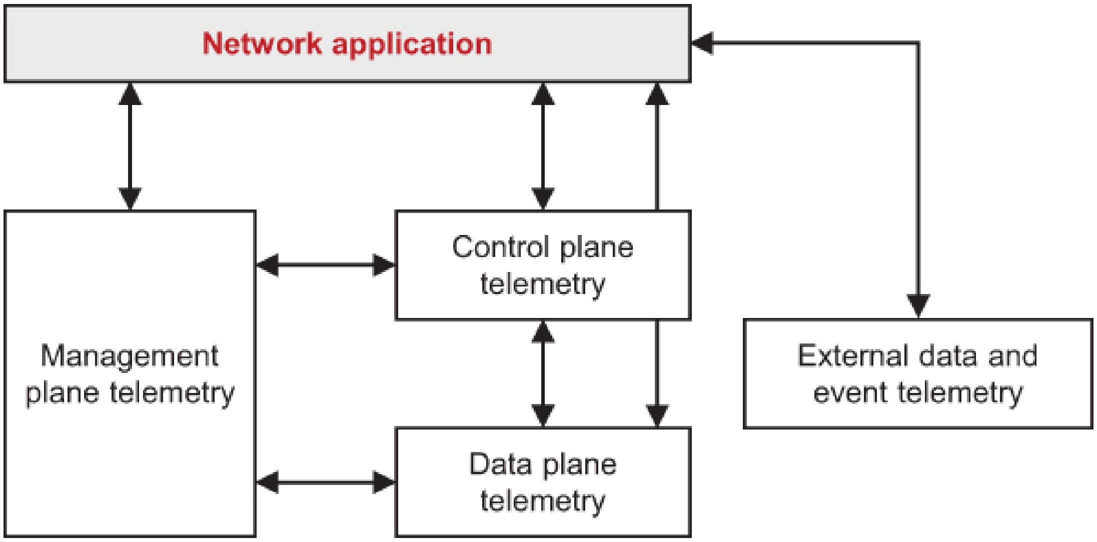


FIGURE 2.1 Top-level modules of the NTF. ⏎

- At the next level, each module is decomposed into different components. These are described in Section 2.1.4.

The two-level architecture with uniform data abstraction helps accurately locate a protocol or technique in a network telemetry system.

The top-level modules of the NTF are classified based on the differences in data sources and export locations. These differences stem from different telemetry data objects and affect the in-network data programmability and processing capability, data encoding and the transport protocol, and the required data bandwidth and delay. For example, data can be sent directly or proxied through the control and management planes. In some cases, the network controller may be the source of telemetry data, allowing the telemetry data collected from network devices to be further exported. Some principles and classifications for control plane and management plane telemetry can also be applied to the controller. Table 2.1 lists the differences between the four modules, which can be compared from the following six dimensions:

TABLE 2.1 Differences between the Four Network Telemetry Modules ⏎

Dimension	Management Plane Telemetry	Control Plane Telemetry	Data Plane Telemetry	External Data and Event Telemetry
Data object	Configuration and operating status	Control protocol, signaling, and Routing Information Base (RIB)	Flow and packet Quality of Service (QoS), traffic statistics, buffer and queue status, Forward Information Base (FIB), and Access Control List (ACL)	Terminal, social, and environmental
Data export location	Main control CPU	Main control or line card CPU	Forwarding chip or line card CPU	Various

Dimension	Management Plane Telemetry	Control Plane Telemetry	Data Plane Telemetry	External Data and Event Telemetry
Data model	YANG, MIB, and Syslog	YANG and customized model	YANG and customized model	YANG and customized model
Data encoding	Google Protocol Buffers (GPB), JavaScript Object Notation (JSON), and Extensible Markup Language (XML)	GPB, JSON, XML, and plain text	GPB and plain text	GPB, JSON, XML, and plain text
Data application protocol	gRPC, NETCONF, and Representational State Transfer Configuration Protocol (RESTCONF)	gRPC, NETCONF, and BGP Monitoring Protocol (BMP)	IPFIX, traffic mirroring, gRPC, NetFlow, and UDP telemetry	gRPC
Data transfer protocol	Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), and Transmission Control Protocol (TCP)	HTTP, HTTPS, and TCP	UDP	HTTP, HTTPS, TCP, and UDP

- Data object.
- Data export location.
- Data model.

- Data encoding.
- Data application protocol.
- Data transfer protocol.

A data object is the target and source of each module. The most convenient location for exporting data varies according to the data source. For example, forwarding hardware is the main source of data plane telemetry data, whereas protocol daemons running on CPUs are the main source of control plane telemetry data. Exporting data from a device location close to its source is more convenient and efficient. But because the locations where data can be exported have different capabilities, it is necessary to select appropriate data models, encodings, and transport protocols in order to balance performance and costs. For example, a forwarding chip has a high throughput but limited capabilities in processing complex data and maintaining states, whereas the opposite is true for a main control CPU. It is therefore not possible to use a universal protocol to meet all network telemetry requirements, as different telemetry protocols may be more suitable for different modules. Table 2.1 lists some of the representative techniques of the four modules in each dimension.

Note that a data source and the application that consumes data may interact indirectly, and some data may be transmitted within the device itself. Take management plane telemetry as an example. In some operating states, data (e.g., interface status and statistics) can only be derived from data plane data sources, meaning that the management plane first needs to acquire data from the data plane. Another example is the need to access the data plane FIB in order to acquire control plane telemetry data. An application may involve multiple planes and interact with different planes at the same time. For example, an SLA compliance application may require both data plane telemetry and control plane telemetry.

2.1.3 Network Telemetry Modules

Network telemetry covers four modules: management plane telemetry, control plane telemetry, data plane telemetry, and external data and event telemetry.

2.1.3.1 Management Plane Telemetry

In management plane telemetry, the management plane of a network device interacts with a Network Management System (NMS) to provide information such as performance data, network log data, network warning and defect data, network statistics, and state data. This process involves many protocols, such as classical SNMP and Syslog. Regardless of the protocol used, management plane telemetry must meet the following requirements:

- Convenient data subscription: An application can freely select the data to be exported and both the data export mode and frequency (i.e., real-time subscription or periodic subscription).
- Structured data: In network O&M automation, machines replace humans in interpreting network data. Data modeling languages such as YANG can effectively describe structured data and normalize data encoding and conversion.
- High-speed data transmission: A data source needs to send large amounts of data at a high frequency. This requires a compact encoding format or a data compression solution to be used to reduce the data volume and improve data transmission efficiency. Additionally, by replacing the query mode with the subscription mode, it is possible to reduce interactions between the client and server and improve the data source's efficiency.
- Network congestion avoidance: An application must avoid the impact of telemetry on network services through a congestion control mechanism or at least SLA isolation.

2.1.3.2 Control Plane Telemetry

Control plane telemetry monitors the health conditions of different network control protocols at all layers of the protocol stack. Tracing the operating states of these protocols facilitates real-time, fine-grained detection, location, and even prediction of various network issues, thereby implementing network optimization. Some of the challenges and issues faced by control plane telemetry are as follows:

- Correlating E2E KPIs with the KPIs of a specific layer is required. For example, Internet Protocol Television (IPTV) users can describe their experience based on video smoothness and definition. If the user experience KPI is abnormal or a service interruption occurs, it is necessary to first demarcate and locate the issue in the involved layer and the specific protocol, for example, Intermediate System to Intermediate System (IS-IS) or Border Gateway Protocol (BGP) at the network layer. Then, the problematic device and the root cause of the issue need to be determined.
- Conventional OAM-based control plane KPI measurement methods, such as ping, traceroute, and Y.1731[8], only measure KPIs and do not reflect the actual operating states of related protocols. As such, these methods are less effective and efficient for control plane troubleshooting and network optimization.
- More control plane monitoring tools are required. One example of control plane telemetry is BMP, which is mainly used for BGP route monitoring. It also helps implement a variety of applications such as BGP peer analysis,

Autonomous System (AS) analysis, prefix analysis, and security analysis. However, the monitoring of other layers and protocols, and the cross-layer, cross-protocol correlation of KPIs, are still at the initial stage. For example, Interior Gateway Protocol (IGP) monitoring is not used as extensively as BMP monitoring. Therefore, further research is still required.

2.1.3.3 Data Plane Telemetry

Effective data plane telemetry relies on network devices being able to expose data. However, the quality, quantity, and timeliness of such data must meet some strict requirements. This poses the following challenges to the data planes of network devices in providing first-hand data sources:

- The main function of the data plane is to process and forward user traffic. Although network visibility is important, telemetry is only an auxiliary function and should not affect normal traffic processing or forwarding. This means that the forwarding behavior should remain unchanged, and that a balance between forwarding performance and telemetry needs to be achieved.
- Network applications require data from various data sources to achieve E2E visibility, potentially involving large amounts of data. However, this data must not exhaust the network bandwidth, regardless of whether in-band or out-of-band measurement is carried out.
- The data plane of a device must provide real-time data with the minimum delay. Long delays in processing, transmission, storage, and analysis negatively affect how effective the control loop is and may even render the data useless.
- Applications require data that is structured, labeled, and easy to parse and use. They also require data types that may vary significantly. In order to provide accurate data for applications, the data plane of a device needs to deliver sufficient flexibility and programmability.
- The requirements and solutions for network congestion avoidance also apply to data plane telemetry.

Although these challenges are not specific to the data plane, its limited resources and flexibility make them more difficult for it to overcome. An important enabler in supporting network telemetry is the programmability of the data plane. Newer data plane forwarding chips feature advanced telemetry functions and support flexible telemetry function customization.

Data plane telemetry relies on performance measurement mechanisms. In terms of their usage, there are various possible dimensions for classifying performance measurement. The typical dimensions are as follows:

- Active, passive, and hybrid: For details, see Section 1.1.
- In-band and out-of-band: For details, see Section 1.1.
- E2E and in-network: E2E measurement starts from and ends at network end hosts, whereas in-network measurement is performed on the network and is transparent to end hosts. The in-network technique can be easily extended to end hosts if needed.
- Data type: The measurement can be flow-, path-, or node-based. Data objects can be such items as data packets, flow records, measurement values, and device status.

2.1.3.4 External Data and Event Telemetry

Events outside the network system boundary are also an important source of network telemetry data. To optimize network O&M, internal telemetry data and external events can be correlated with the network system's requirements. Like other sources of telemetry data, data and events must meet strict requirements, particularly in terms of timeliness, due to the need for incorporating external event data correctly into network management applications. The challenges faced by external data and event telemetry are as follows:

- Because both hardware (e.g., physical sensors) and software (e.g., data sources that can analyze information streams) can function as external event detectors, the transmitted data needs to comply with a universal and extensible mode in addition to supporting different detectors.
- External event detectors have a high requirement on timeliness due to being mainly used for notification purposes. Messages can be inserted into control plane queues by priority — more important sources and events have higher allocated priorities.
- The data models used by external event detectors must be sufficiently compatible and extensible to support existing and future devices and applications.
- Congestion is likely to occur due to the communication with entities outside a service provider network's boundary traversing the Internet, meaning that suitable measures must be taken to mitigate congestion risks.

2.1.4 Functional Components of Network Telemetry Modules

Each telemetry module can be further divided into five different functional components, as shown in Figure 2.2. These functional components are described as follows:

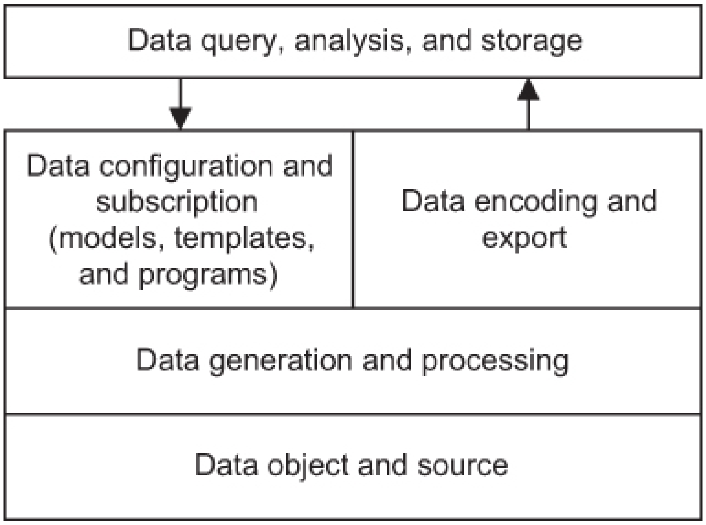

Data query, analysis, and storage
Data configuration and subscription (models, templates, and programs)
Data encoding and export
Data generation and processing
Data object and source

FIGURE 2.2 Functional components of network telemetry. ⏎

- Data query, analysis, and storage: This component is typically integrated into the network management system at the end that receives data. It is responsible for issuing data requirements, which can be either one-time data queries or subscriptions to events or streaming data. It also receives, stores, and processes data returned from network devices, and can perform data analysis interactively (which initiates further data queries). The component itself can be centralized or distributed, and involves one or more instances.
- Data configuration and subscription: This component manages data queries on devices and determines which protocol and channel to use for an application to acquire the data it needs. It also configures data that potentially cannot be directly acquired from data sources. Subscription data can be described using items such as models, templates, or programs.
- Data encoding and export: This component determines how telemetry data is transferred to the data query, analysis, and storage component. Data encoding and the transport protocol may vary depending on where the data is exported.
- Data generation and processing: This component captures, filters, processes, and formats data from raw data sources. Such operations may involve in-network computation and processing on the fast or slow path of network devices.
- Data object and source: This component determines the monitoring objects and raw data sources configured on devices. Typically, the data provided by data sources is in a raw format and needs to be further processed. Each data source can be considered a probe. Some data sources can be dynamically installed, whereas the others are static.

2.1.5 Data Acquisition Mechanism of Network Telemetry

In a broad sense, network data can be acquired through subscriptions or queries. A subscription is a contract between a publisher and a subscriber, whereby the subscribed data is automatically sent to the subscriber until the subscription expires. Subscriptions can be predefined or they can be configured and customized according to subscriber requirements. Queries are used if a client wants on-demand feedback from a network device. The queried data can be acquired directly from a specific data source or acquired by synthesizing and processing raw data. Queries are best suited for interactive network telemetry applications. In general, data can be queried when needed, but in many cases, subscription is more efficient and can reduce the delay for a client to detect a change. From the perspective of data consumers, a telemetry data consumer can subscribe to or query the following four types of data from network devices:

- Simple data: stably acquired from datastores or static probes in network devices.
- Derived data: obtained by synthesizing or processing the raw data obtained from one or more network devices. The data processing function can be statically or dynamically loaded into network devices.
- Event-triggered data: conditionally acquired when certain events occur, such as when an interface starts or shuts down. This type of data can be actively pushed through subscriptions or passively polled through queries. Numerous event modeling methods are available, including the Finite State Machine (FSM) and Event-Condition-Action (ECA)[9].
- Streaming data: continuously generated by time sequence, such as an interface packet counter exported every second. This type of data is actively pushed to subscribers and can reflect the real-time network status and indicators, but requires high bandwidth and powerful processing capabilities.

These telemetry data types are usually interrelated instead of conflicting with each other. Specifically, derived data is comprised of simple data, event-triggered data can be either simple or derived, and streaming data can be generated based on recurring events. Figure 2.3 shows the relationships between network telemetry data types.

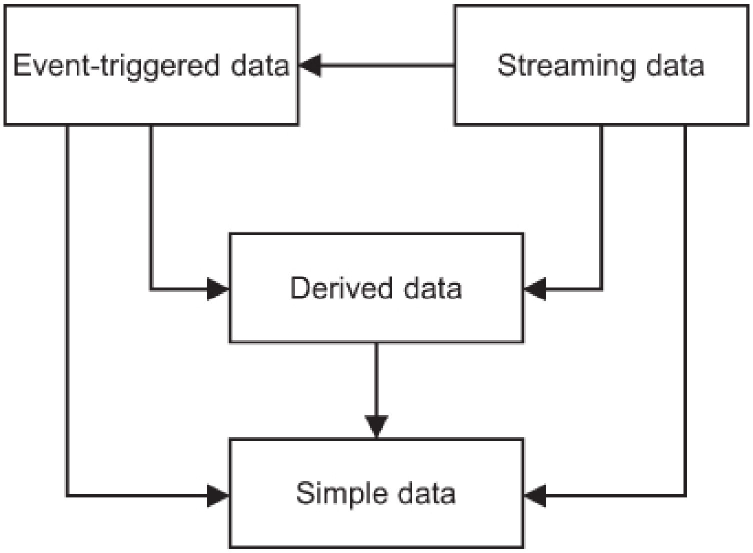


FIGURE 2.3 Relationships between network telemetry data types. ⏎

Subscriptions typically deal with event-triggered and streaming data, and queries typically deal with simple and derived data. However, other processing methods are also possible. Network telemetry techniques are mainly designed for subscribing to event-triggered or streaming data and querying derived data.

2.2 IFIT FRAMEWORK

The IFIT framework was proposed by the IETF to enable normalized on-path telemetry to be implemented efficiently on IP networks. This framework facilitates the application of on-path telemetry by defining a complete set of techniques, including on-path measurement, intelligent flow selection, efficient data reporting, Dynamic Network Probe (DNP), and automated deployment.

2.2.1 On-Path Measurement Modes Supported by IFIT

IFIT supports multiple on-path measurement techniques, such as IOAM and alternate marking. These techniques can be implemented in passport or postcard mode, depending on how the collected data is processed. Figure 2.4 illustrates data processing in the two modes.

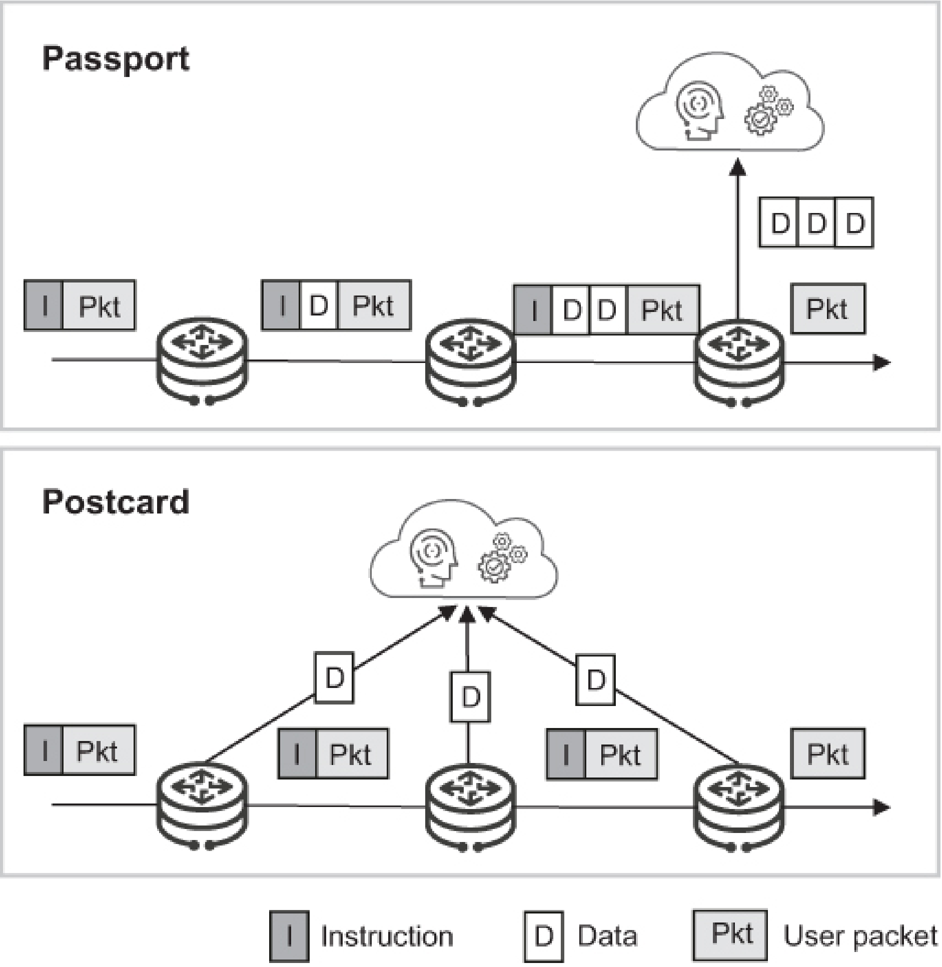


FIGURE 2.4 Data processing in passport and postcard modes. ⏎

In passport mode, the ingress node of a measurement domain adds a Telemetry Information Header (TIH) to the packets to be measured. This header includes the data collection instruction. Transit nodes collect data hop by hop according to this instruction and record the data in the packets. The egress node of the measurement domain reports all the data collected along the path for processing, removes the TIH and data, and restores the user packets. Network telemetry in passport mode is like a tourist who receives an entry/exit stamp in their passport for each country they visit.

The postcard mode allows each node in the measurement domain to generate additional packets that carry collected data and send them to the collector. This differs from the passport mode, which records the collected data in the received user packets that contain TIHs. Network telemetry in postcard mode is like a tourist sending a postcard home each time they arrive at a tourist spot.

The two modes each apply to different scenarios and have unique advantages and disadvantages. Table 2.2 compares the two modes.

TABLE 2.2 Comparison between the Passport and Postcard Modes ⏎

Item	Passport Mode	Postcard Mode
Advantages	• Provides hop-by-hop data correlation, reducing the collector's workload. • Requires only the egress node to send data, reducing the overhead.	• Supports the detection of the specific packet loss location. • Features short packet headers of a fixed length. • Supports easy hardware-based implementation.
Disadvantages	• Does not support packet loss demarcation. • Increases the packet header size as the number of hops grows (tracing mode).	Requires the collector to correlate packets with data generated by nodes on the path.
Typical techniques	IOAM trace	Alternate marking and IOAM Direct Export (DEX)

2.2.2 IFIT Framework and Core Functions

Although on-path measurement has many advantages, it faces several challenges in real-world network deployments.

- On-path measurement involves specifying the flow object to be monitored on a network device and allocating monitoring resources for performing corresponding operations. Such operations include inserting a data collection instruction, collecting data, and stripping an instruction and data. However, the limited processing capabilities of a network device restrict the number of flow objects it can monitor. This poses a challenge to the large-scale deployment of on-path measurement.
- The additional processing that on-path measurement introduces in the data plane of a device may compromise the device's forwarding performance. And due to the observer effect, network measurement results may not accurately reflect the status of the measured object.
- Per-packet monitoring generates a large amount of measurement data, which will consume a lot of network bandwidth if all data is reported. Because a data analyzer may process data from hundreds of forwarding devices on the network, server performance will be significantly compromised due to

receiving, storing, and analyzing the massive amount of data sent from the devices.

- Predefined datasets can provide only limited data and cannot meet the changing data requirements in the future. As a result, a new method is needed to define data in a flexible and extensible manner and deliver the required data to applications for analysis.

IFIT is a reference framework that, as shown in Figure 2.5, encompasses the application and management system, controller, and IFIT-capable forwarders. It can flexibly integrate multiple techniques for both data plane measurement and data export to provide comprehensive performance information for network OAM, meeting the measurement requirements of different applications. For example, different types of information can be collected using the IOAM or alternate marking technique. If the latter is used, the mode can be switched from E2E to hop-by-hop for fault locating. After telemetry data is processed and analyzed, the analysis result can be used to instruct the controller to modify the configurations of devices in the IFIT domain and adjust the data collected by IFIT. As such, this process is dynamic and interactive.

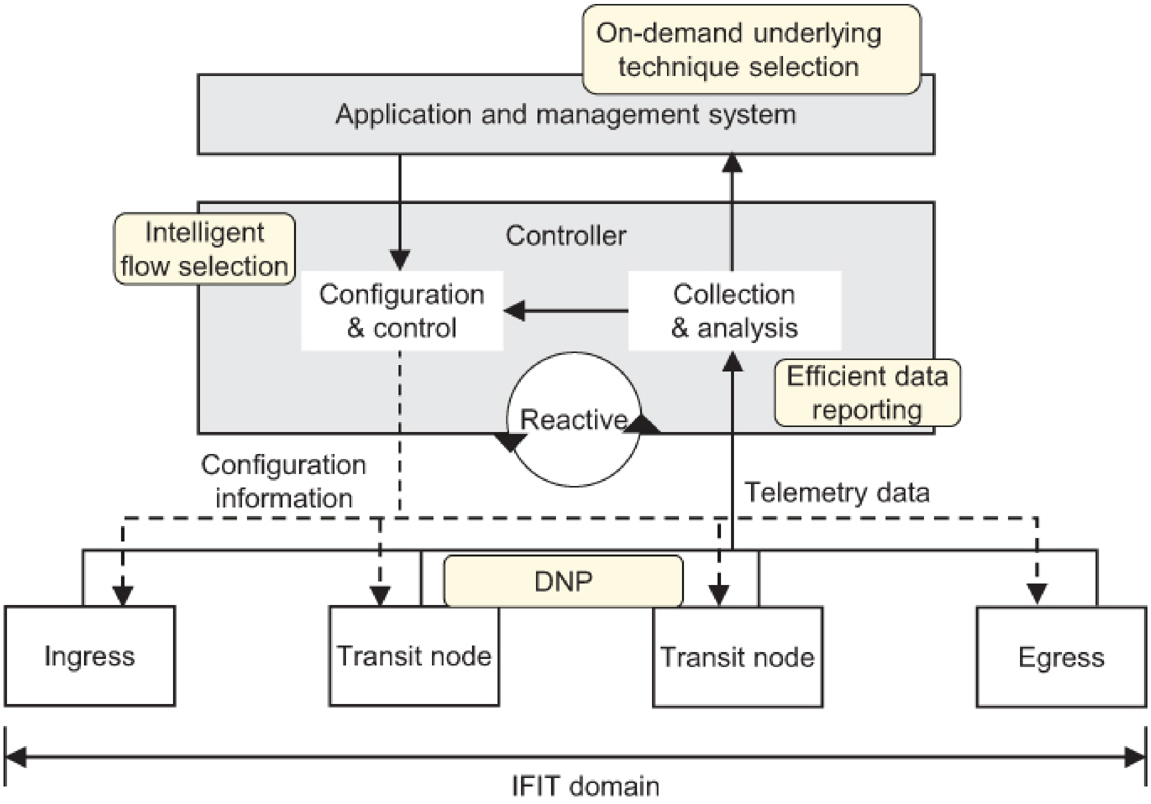


FIGURE 2.5 IFIT framework. ⏎

The IFIT framework offers four core functions: intelligent flow selection, efficient data reporting, DNP, and on-demand underlying technique selection.

2.2.2.1 *Intelligent Flow Selection*

Limited hardware resources often make it impossible to monitor all traffic on the network and collect data on a per-packet basis. Doing so would not only affect the normal forwarding of devices but also consume a large amount of network bandwidth. One possible workaround to this issue is to select certain traffic for monitoring.

The intelligent flow selection technique uses a coarse-to-fine mode that trades time for space, helping users select desired traffic. They can deploy intelligent flow selection policies based on their intentions. Regardless of whether these policies are performed based on sampling or prediction, the intelligent flow selection technique typically consumes few resources.

As an example, assume that a user wants to monitor the top 100 elephant flows. In this case, the user can define an intelligent flow selection policy based on the Count-Min Sketch technique[10]. This technique performs multiple hash operations and stores only count values instead of flow IDs, achieving a very high level of recognition accuracy while consuming only a small amount of memory. The controller generates ACLs based on the intelligent flow selection result and delivers them to devices to monitor elephant flows.

Figure 2.6 shows the following components that interwork to implement intelligent flow selection:

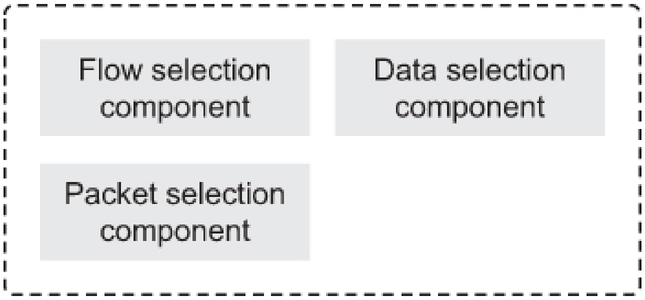


FIGURE 2.6 Components of the intelligent flow selection function. ⏎

- Flow selection component defines a policy for selecting target flows to be monitored. Flows are typically defined using the 5-tuple of the IP header and can be aggregated based on dimensions such as interface, tunnel, and protocol.
- Packet selection component defines a policy for selecting packets from target flows. This policy can be implemented based on a sampling interval, a sampling probability, or certain packet characteristics.

- Data selection component defines the data set to be collected, which can be changed for each packet or flow.

2.2.2.2 Efficient Data Reporting

Per-packet on-path measurement can capture subtle dynamic changes on a network, but the involved packets will inevitably contain a lot of redundant information. Likewise, a lot of network bandwidth will be consumed if all the information is directly reported. This also imposes an extremely heavy burden on data analysis, especially if an analyzer needs to manage tens of thousands of network nodes.

One way to report data more efficiently is to use binary data transmission encoding. The XML format is typically used today to encode NETCONF-based network management information, but this type of encoding consumes a lot of network bandwidth and is not suitable for reporting flow-based on-path network measurement information. Binary encoding, such as GPB, offers an effective approach to reduce the amount of data to be reported.

Another way to minimize the amount of data to be reported is to use data filtering. This approach enables a network device to filter data based on certain criteria and then convert the data into events to notify upper-layer applications. Take flow path tracing as an example. Flow-based load balancing is typically performed on live networks, where flow paths seldom change — a path change usually indicates an abnormality. It is normal to have a large amount of duplicate path data (including data about the node and inbound and outbound interfaces of each hop). Such data does not need to be reported and can therefore be filtered out by network devices. In this way, only data about newly discovered or changed flow paths is reported, thereby reducing the amount of reported data.

Network devices can also cache data that is generated during a specified period of time and does not have high requirements on timeliness, and then compress and report it in batches. This approach reduces not only the amount of data to be reported but also the frequency at which it is reported, lowering the pressure of data collection.

Figure 2.7 shows the following components that interwork to implement efficient data reporting:

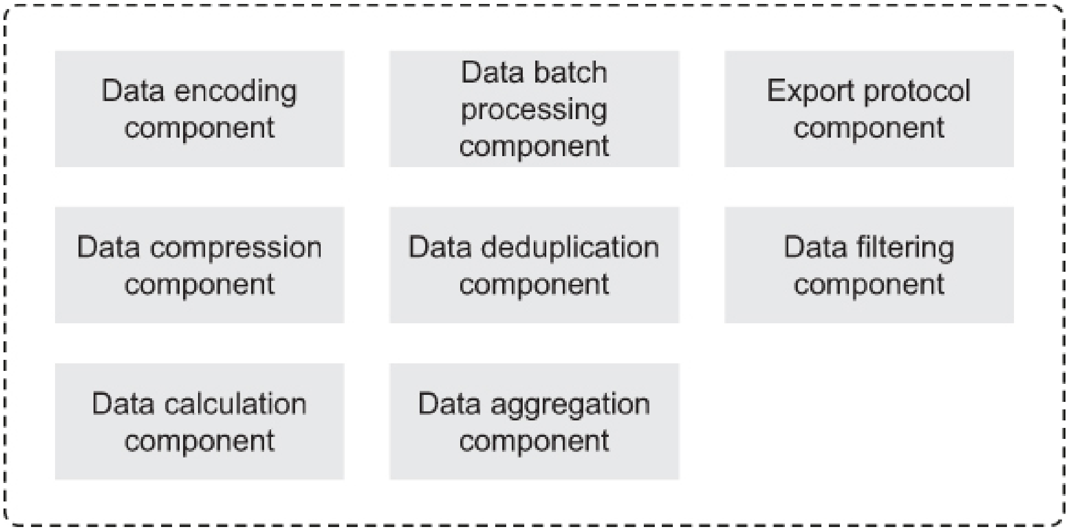

Data encoding component
Data batch processing component
Export protocol component
Data compression component
Data deduplication component
Data filtering component
Data calculation component
Data aggregation component

FIGURE 2.7 Components of the efficient data reporting function. ⏎

- Data encoding component defines a method for encoding telemetry data.
- Data batch processing component defines the size of batch data buffered on the device side before it is exported.
- Export protocol component defines the protocol used for exporting telemetry data.
- Data compression component defines the algorithm for compressing raw data.
- Data deduplication component defines the algorithm for deleting redundant data from raw data.
- Data filtering component defines the policy for filtering required data.
- Data calculation component defines the policy for preprocessing raw data and generating new data.
- Data aggregation component defines the process of combining and synthesizing data.

2.2.2.3 DNP

Limited data plane resource spaces (e.g., data storage space and instruction space) make it difficult to provide continuous and complete data monitoring and reporting. At the same time, the amount of data that applications need changes dynamically. For example, an application may require only intermittent inspections during normal operation, but it requires precise real-time monitoring once a risk is detected. Deploying and running all network measurement functions in the data plane consumes a lot of resources and adversely affects data forwarding while offering only limited benefits. It is therefore necessary to provide a dynamic loading mechanism through which network measurement functions can be loaded on demand to meet various service requirements while using limited resources.

As its name suggests, DNP is a dynamically loadable network measurement technique. It supports on-demand loading and unloading of network measurement functions on devices. In this way, it minimizes resource consumption in the data plane while meeting service requirements. For example, to measure the performance of a flow in hop-by-hop mode, a user can load the corresponding measurement function onto the involved device through either configuration or dynamic programming. If the measurement function is no longer needed, the user can unload the application from the device to release the occupied instruction space and data storage space.

Figure 2.8 shows the following components that interwork to implement the DNP function:

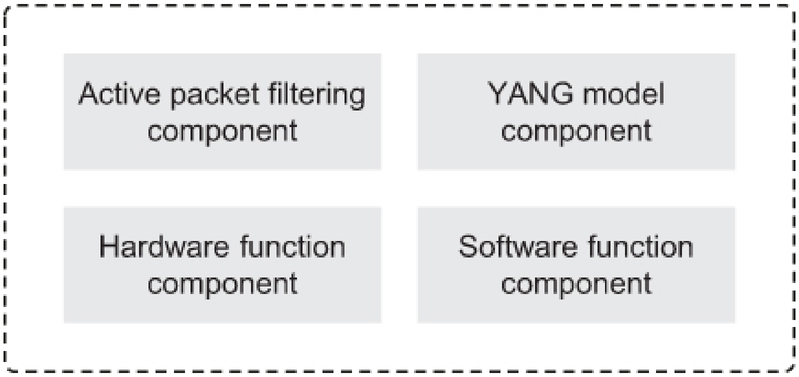

Active packet filtering component
YANG model component
Hardware function component
Software function component

FIGURE 2.8 Components of the DNP function. ⏎

- Active packet filtering component defines DNP by dynamically updating the packet filtering policy (including flow selection and related actions). It is suitable for most hardware.
- YANG model component implements different data processing and filtering functions through dynamic deployment.
- Hardware function component dynamically loads hardware-based functions into the forwarding path at run time through mechanisms such as reserved pipelines and function stubs. This is supported by only some hardware.
- Software function component enables dynamically loadable software functions to be implemented in a suitable CPU.

The DNP technique provides sufficient flexibility and extensibility for IFIT, allowing intelligent flow selection and efficient data reporting to be dynamically loaded to devices as policies.

2.2.2.4 On-Demand Underlying Technique Selection

IFIT can flexibly adapt to different network conditions and application requirements thanks to its support for multiple underlying data collection and export techniques. For example, it can collect data in either the passport or postcard mode according to the desired data type. And it can switch from the passport mode to the postcard mode if an application needs to trace the location where data packets are lost. IFIT can further integrate multiple data plane monitoring and measurement techniques and provide a comprehensive data plane telemetry solution. It also supports the deployment of new configurations and operations according to application requirements and real-time telemetry data analysis results.

Figure 2.9 shows how the on-demand underlying technique selection function is implemented and lists the candidate on-path telemetry techniques. This function is located in a logically centralized controller, from where it dynamically distributes all control and configuration instructions to suitable nodes in the domain. It makes configuration and operation decisions based on application requirements and real-time telemetry data analysis results.

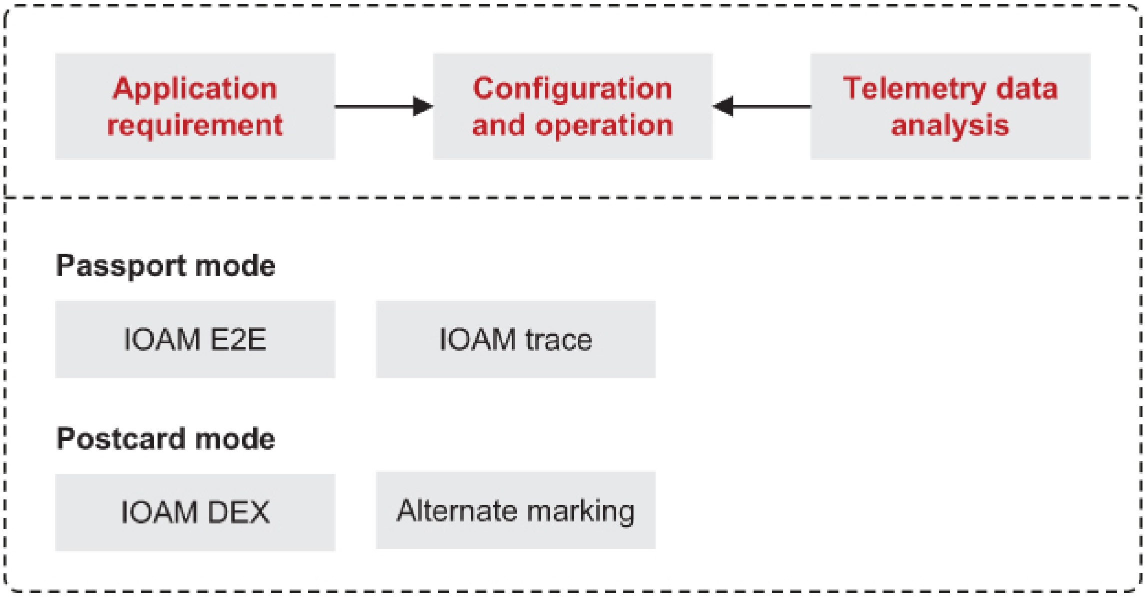


FIGURE 2.9 On-demand underlying technique selection function. ⏎

2.2.3 IFIT Deployment Automation

IFIT applications can be flexibly and dynamically deployed, and telemetry information can be exchanged between network nodes and controllers to enable closed-loop automation. As shown in Figure 2.10, the IFIT deployment automation solution combines YANG Push with IGP, Path Computation Element Communication Protocol (PCEP), and BGP extensions.

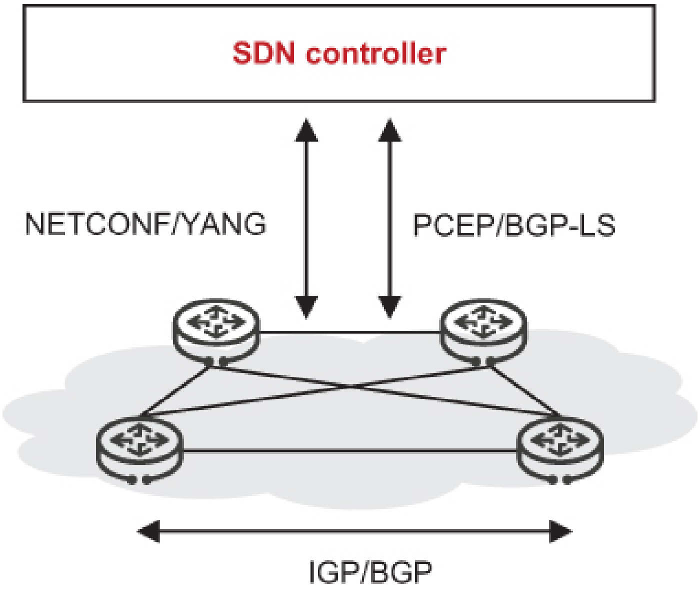


FIGURE 2.10 IFIT deployment automation. ⏎

YANG Push[11, 12, 13, 14] is a Model Driven Telemetry (MDT) technique that supports closed-loop automation. Applications can acquire specific data by subscribing to the standard YANG data model based on NETCONF, or they can configure network nodes through NETCONF, RESTCONF, and other network management protocols to implement on-path measurement. In addition, *draft-ydt-ippm-alt-mark-yang*[15] and *draft-ietf-ippm-ioam-yang*[16] respectively define data models for alternate marking (and allows this function to be applied to the specified flow through NETCONF) and the IOAM function (with support for all IOAM options) using the YANG data modeling language.

To facilitate on-path telemetry deployment automation in IFIT, it is necessary to obtain the on-path telemetry capabilities of nodes and links on the network, including the supported on-path measurement methods and options. IGP, Border Gateway Protocol-Link State (BGP-LS), and BGP extensions for on-path telemetry capability advertisement are respectively defined in *draft-wang-lsr-igp-extensions-ifit*[17], *draft-wang-idr-bgpls-extensions-ifit*[18], and *draft-ietf-idr-bgp-ifit-capabilities*[19]. IGP and BGP-LS extensions are respectively used to distribute and report the on-path telemetry capability information about nodes or links, and BGP extensions are used to advertise the on-path telemetry capability information about the next-hop node of a BGP route.

IFIT automation requires the use of a controller to enable SLA monitoring for existing tunnels. This allows for the quick detection of SLA violations and performance deterioration so that tunnel deployments can be changed accordingly. BGP Segment Routing (SR) Policy and PCEP extensions are respectively defined in *draft-ietf-idr-sr-policy-ifit*[20] and *draft-ietf-pce-pcep-ifit*[21] to help distribute on-path telemetry capability information. Thus, the on-path telemetry function can be automatically enabled in tunnel path instantiation.

2.3 STORIES BEHIND IPV6 ON-PATH TELEMETRY

2.3.1 Origin of NTF

Our research into telemetry techniques started back in 2018. In my view, the difficulties involved in IP network O&M are largely related to O&M data due to the following three problems:

- The amount of data reported by network devices is insufficient.
- The speed at which network devices report data is low.
- The types of data reported by network devices are incomplete.

When telemetry was first introduced, it was seen as a promising technique that could provide high-speed data reporting channels. We therefore studied it as a key solution to the preceding problems. However, a major issue facing research into telemetry is the wide range of complex techniques involved, making its working principles unclear and confusing. For example, some people mistakenly consider telemetry to be equivalent to gRPC or In-band Network Telemetry (INT). To ensure that research could proceed in a smooth and organized manner, it was necessary to establish a systematic technical framework. We therefore thoroughly reviewed telemetry techniques based on different planes and dimensions, and then formulated the NTF and submitted a draft[22] to the IETF Operations and Management Area Working Group (OPSAWG). The NTF defines telemetry in three planes as follows:

- Management plane telemetry collects network management data through protocols such as gRPC and NETCONF.
- Data plane telemetry collects data plane data obtained through mechanisms such as alternate marking and IOAM and reports the data through protocols such as IPFIX.
- Control plane telemetry reports control protocol data through protocols such as BMP.

The draft was later updated according to OPSAWG suggestions to include information related to external telemetry, enriching the NTF. In addition to defining related telemetry terms, this draft classifies involved techniques. It helps provide a clear understanding of the telemetry framework, as well as the differences and relationships between related techniques promoted by different IETF WGs.

The IETF prefers designing and standardizing protocols required for interworking over standardizing drafts about use cases and frameworks. This was brought about partially by the Source Packet Routing in Networking (SPRING) WG. In the early stage, the SPRING WG defined many use case and requirement drafts, but they were of limited use. As a result, the IETF Routing Domain held an open meeting to discourage writing such drafts. But because the IETF is likely to divide an overall solution into different parts and distribute them to different WGs, people unfamiliar with how these parts relate to each other find it hard to understand the whole picture. Framework drafts are, therefore, a necessity for solving such problems. This is why we promoted the NTF draft. Recognizing the importance of telemetry techniques and the need to systematize them, the WG adopted the NTF draft, which eventually evolved into an RFC.

2.3.2 Origin of IFIT

A key point of reference during our study of data plane telemetry was the IOAM draft that defined tracing, proof of transit, and edge-to-edge options[23]. During the research, we found that IOAM tracing options have many limitations, all of which require solutions. For example, the IOAM trace option causes the packet size to increase, and IOAM causes devices to report large amounts of data to the analyzer. Techniques such as on-path measurement in postcard mode, alternate marking, intelligent flow selection, and efficient data reporting were therefore introduced to address these limitations, extending our research on data plane telemetry techniques far beyond IOAM itself. At this stage, the on-path telemetry solution that was formed using these techniques and could be put into commercial use had not been given a name yet. Initially, our colleagues were not aware of the problem, but discussions soon became complex and even chaotic due to the lack of clear and well-accepted definitions for terms. Sometimes, after heated debates, we found that the discussed solution had nothing to do with the original IOAM. In the end, I called a dedicated meeting to discuss how we should name this solution. Because several members in the research team were either losing weight or had already done so successfully, IFIT quickly became a popular choice. At that time, many members in the research team were either losing weight or had already done so successfully. Because the word "fit" has the same meaning as "healthy," IFIT quickly became a popular choice. IP network O&M has always been a pain point and is a major cause of complaints in the industry: Although network services are becoming more and more advanced, network O&M modes are still very traditional and inefficient. Through our research on IFIT, we hoped to offer better IP O&M techniques and solutions in order to make networks healthier.

Establishing common terms helped improve the efficiency of subsequent discussions as we all shared the same understanding. Later, I took a Massachusetts Institute of Technology (MIT) artificial intelligence course offered by the NetEase open course app. The professor mentioned the Rumpelstiltskin principle: Once you can name something, you get power over it. This resonated with me after my experience with IFIT.

2.3.3 POF and DNP

The idea of DNP, a special technique in the IFIT framework, came from Dr. Song Haoyu at Futurewei. Because the information obtained through on-path measurement is flexible and may be extended regardless of how techniques and standards are defined, a mechanism is needed to change the logic of the data forwarding plane and obtain newly defined information. During the research on Software Defined Network (SDN) technology, Dr. Song published a paper[24] about Protocol Oblivious Forwarding (POF) in Association for Computing Machine Special Interest Group on Data Communication (ACM SIGCOMM), a professional forum for discussing topics related to communications and computer networks. This paper is an important development in SDN. It defines the POF mechanism, which implements data plane programming more flexibly than OpenFlow-based flow table programming[25], thereby better meeting requirements on network function extension. In the POF mechanism, POF switches can be oblivious to protocols. Specifically, they only need to locate data based on one or more {offset, length} tuples, conduct table lookups, and then perform associated operations under the instructions of the POF controller. This mechanism enables switches to support new protocols without the need to be upgraded or replaced, making it possible to realize network innovations much faster. The POF mechanism has also had a tremendous impact on both the Protocol Independent Forwarding (PIF) mechanism[26] and Programming Protocol independent Packet Processors (P4) mechanism[27] proposed later. To implement the POF mechanism, it is necessary to find good use cases. Among the limited use cases requiring flexible data plane changes, data plane telemetry is a promising one. While researching data plane telemetry with us, Dr. Song proposed the DNP technique to support flexible customization of an on-path measurement mechanism without the need to replace or upgrade hardware and chips. DNP can therefore be considered an application of the POF mechanism in data plane telemetry to some extent. Later, during the development of IFIT, we identified many scenarios that required on-path measurement information and encapsulation modes to be changed. Thanks to Huawei's long-term investment in researching network processor chips, routers on the live network can meet such requirements after a software upgrade. This significantly reduces the cost and time required for deploying on-path telemetry and fully demonstrates the value of the DNP technique.

REFERENCES

1. Song H, Qin F, Martinez-Julia P et al. Network telemetry framework [EB/OL]. (2022-05-27) [2024-09-30]. RFC 9232. ⏎
2. Presuhn R, Case J, Mccloghrie K et al. Version 2 of the protocol operations for the simple network management protocol (SNMP) [EB/OL]. (2002-10) [2024-09-30]. RFC 3416. ⏎
3. Gerhards R. The syslog protocol [EB/OL]. (2009-03) [2024-09-30]. RFC 5424. ⏎
4. Mizrahi T, Sprecher N, Bellagamba E et al. An overview of operations, administration, and maintenance (OAM) tools [EB/OL]. (2014-06) [2024-09-30]. RFC 7276. ⏎
5. Kavasseri R, Stewart B. Event MIB [EB/OL]. (2000-10) [2024-09-30]. RFC 2981. ⏎
6. Chisholm S, Romascanu D. Alarm management information base (MIB) [EB/OL]. (2004-09) [2024-09-30]. RFC 3877. ⏎
7. Phaal P, Panchen S, Mckee N. InMon corporation’s s flow: A method for monitoring traffic in switched and routed networks [EB/OL]. (2001-09) [2024-09-30]. RFC 3176. ⏎
8. ITU-T. Operation, administration and maintenance (OAM) functions and mechanisms for ethernet-based networks [EB/OL]. (2019-08-29) [2024-09-30]. G.8013/Y.1731. ⏎
9. Wu Q, Bryskin I, Birkholz H et al. A YANG data model for ECA policy management [EB/OL]. (2021-02-19) [2024-09-30]. Draft-ietf- netmod-eca-policy-01. ⏎
10. Cormode G, Muthukrishnan S. An improved data stream summary: The count-min sketch and its applications [EB/OL]. (2021-02-19) [2024-09-30]. ⏎
11. Voit E, Clemm A, Gonzalez Prieto A et al. Subscription to YANG notifications [EB/OL]. (2019-09) [2024-09-30]. RFC 8639. ⏎
12. Voit E, Clemm A, Gonzalez Prieto A et al. Dynamic subscription to YANG events and datastores over netconf [EB/OL]. (2019-09) [2024-09-30]. RFC 8640. ⏎
13. Clemm A, Voit E. Subscription to YANG notifications for datastore updates [EB/OL]. (2005-04) [2024-09-30]. RFC 8641. ⏎
14. Voit E, Rahman R, Nilsen-Nygaard E et al. Dynamic subscription to YANG events and datastores over restconf [EB/OL]. (2019-11) [2024-09-30]. RFC

8650. ⏎

15. Graf T, Wang M, Fioccola G et al. A YANG data model for the alternate marking method [EB/OL]. (2024-09-02) [2024-09-30]. draft-ydt-ippm-alt-mark-yang-03. ⏎
16. Zhou T, Guichard J, Brockners F et al. A YANG data model for in-situ OAM [EB/OL]. (2024-03-01) [2024-09-30]. Draft-ietf-ippm-ioam-yang-13. ⏎
17. Wang Y, Zhou T, Qin F et al. IGP extensions for in-situ flow information telemetry (IFIT) capability advertisement [EB/OL]. (2021-01-29) [2024-09-30]. Draft-wang-lsr-igp-extensions-ifit-01. ⏎
18. Wang Y, Zhou T, Liu M et al. BGP-LS extensions for in-situ flow information telemetry (IFIT) capability advertisement [EB/OL]. (2021-01-14) [2024-09-30]. Draft-wang-idr-bgpls-extensions-ifit-00. ⏎
19. Fioccola G, Pang R, Wang S et al. Advertising in-situ flow information telemetry (IFIT) capabilities in BGP [EB/OL]. (2024-07-05) [2024-09-30]. Draft-ietf-idr-bgp-ifit-capabilities-05. ⏎
20. Qin F, Yuan H, Yang S et al. BGP SR policy extensions to enable IFIT [EB/OL]. (2024-04-19) [2024-09-30]. Draft-ietf-idr-sr-policy-ifit-08. ⏎
21. Yuan H, Wang X, Yang P et al. Path computation element communication protocol (PCEP) extensions to enable IFIT [EB/OL]. (2024-07-05) [2024-09-30]. draft-ietf-pce-pcep-ifit-05. ⏎
22. Song H, Li ZQ, Martinez-Julia P et al. Network telemetry framework [EB/OL]. (2019-09-07) [2024-09-30]. Draft-song-opsawg-ntf-03. ⏎
23. Brockners F, Bhandari S, Pignataro C et al. Data fields for in-situ OAM [EB/OL]. (2018-01-03) [2024-09-30]. Draft-brockners-inband-oam-data-07. ⏎
24. Song H. Protocol-oblivious forwarding: Unleash the power of SDN through a future-proof forwarding plane [EB/OL]. (2013-08) [2024-09-30]. ⏎
25. ONF. Open flow switch specification [EB/OL]. (2015-03-26) [2024-09-30]. ⏎
26. ONF. Protocol independent forwarding [EB/OL]. (2014-09-15) [2024-09-30]. ⏎
27. The Linux Foundation Projects. P4 open source programming language [EB/OL]. (2024-09) [2024-09-30]. ⏎

CHAPTER 3

Data Plane of IPv6 On-Path Telemetry

DOI: 10.1201/9781003677901-5

IPV6 ON-PATH TELEMETRY UTILIZES methods including alternate marking and IOAM to implement on-path measurement. This chapter describes how these two methods work in the data plane.

3.1 ALTERNATE MARKING METHOD

In IPv6 on-path telemetry, the alternate marking method inserts alternate marking instructions into IPv6 extension headers to instruct nodes through which a service flow passes to collect and report telemetry statistics, thereby enabling on-path measurement of packet loss, delay, and jitter[1].

3.1.1 How the Alternate Marking Method Works

Figure 3.1 provides a visual representation of how the alternate marking method works. As shown in the figure, service packets enter the network through a Provider Edge (PE, PE1 in this case) and leave the network through PE2. To measure the packet loss and delay of the network, the alternate marking method periodically marks the packets by alternately setting the packet loss marking bit (L) and delay marking bit (D) in the measurement packet header to 0 or 1.

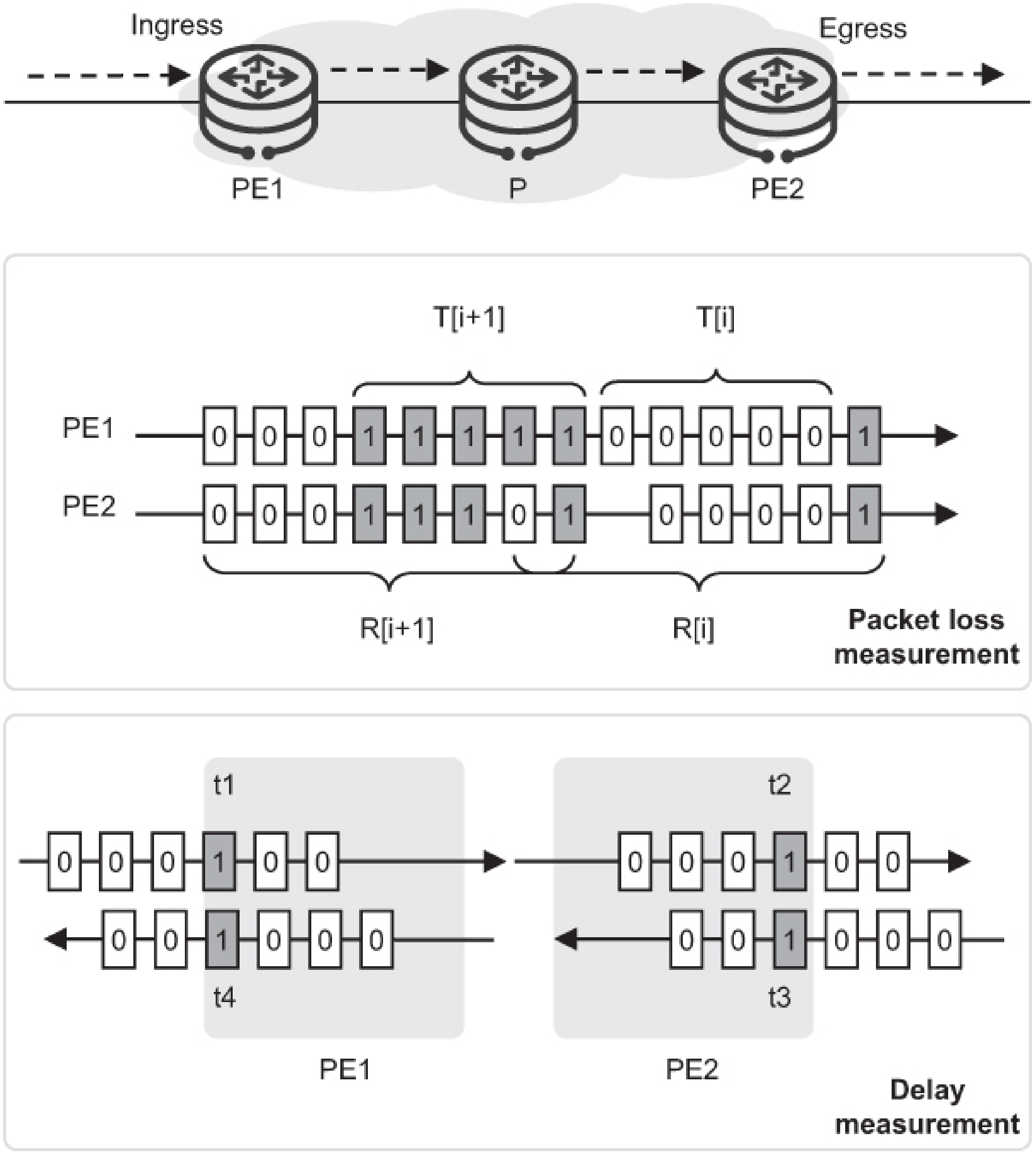


FIGURE 3.1 How the alternate marking method works. ⏎

Alternate marking measures packet loss from PE1 to PE2 as follows:

1. PE1 sets the L bit of each service packet to 0 or 1 on the ingress. It then flips the L bit value at the beginning of each measurement period *i* and calculates the number of packets (T[*i*]) with either a 0 or 1 bit value in that period.
2. PE2 calculates the number of packets (R[*i*]) with the bit value of 0 or 1 in each measurement period *i* on the egress. To prevent packet disorder from affecting the measurement result, the measurement needs to be performed

over a longer period. For details about how to select a measurement period, see Section 3.1.2.

3. In period *i*, the number of lost packets is calculated using the formula T[*i*] − R[*i*], and the packet loss rate is calculated using the formula (T[*i*] − R[*i*])/T[*i*].

Alternate marking measures the delay between PE1 and PE2 as follows:

1. PE1 sets the D bit of a packet to 1 on the ingress and records the timestamp t1.
2. PE2 receives the service packet with the D bit set to 1 and records the timestamp t2.
3. The one-way delay from PE1 to PE2 is calculated as t2 − t1, and that from PE2 to PE1 is calculated as t4 − t3. The two-way delay is calculated as (t2 − t1) + (t4 − t3).

This implementation ensures that the alternate marking method can proactively detect minor network and application changes and then reflect the real packet loss and delay of the network and applications.

The alternate marking method supports two measurement modes: E2E and hop-by-hop (trace). The E2E mode is used to monitor the overall service quality, whereas the hop-by-hop mode is used for hop-by-hop demarcation of low-quality services or on-demand hop-by-hop monitoring of Very Important Person (VIP) services.

In E2E mode, alternate marking needs to be enabled on both the ingress and egress, and an alternate marking measurement instance needs to be deployed on the ingress to trigger measurement. When this mode is used, only the ingress and egress are aware of alternate marking and report measurement data, and transit nodes do not participate, as shown in Figure 3.2.

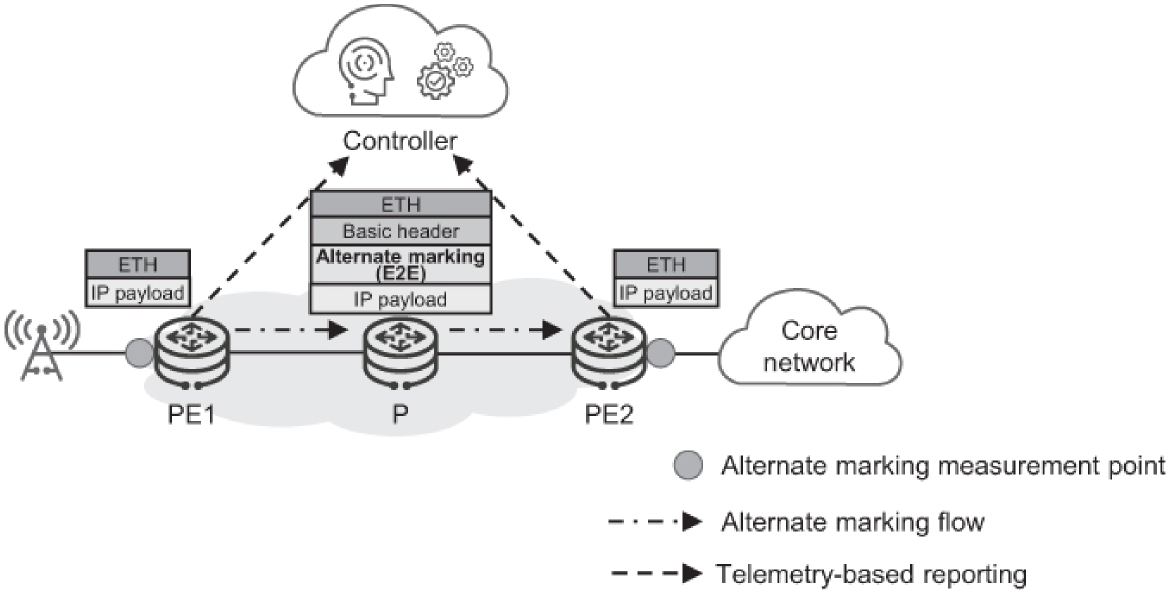


FIGURE 3.2 E2E mode. ⏎

In hop-by-hop mode, an alternate marking measurement instance needs to be deployed on the ingress to trigger measurement, and alternate marking needs to be enabled on all nodes along the service flow path, as shown in Figure 3.3.

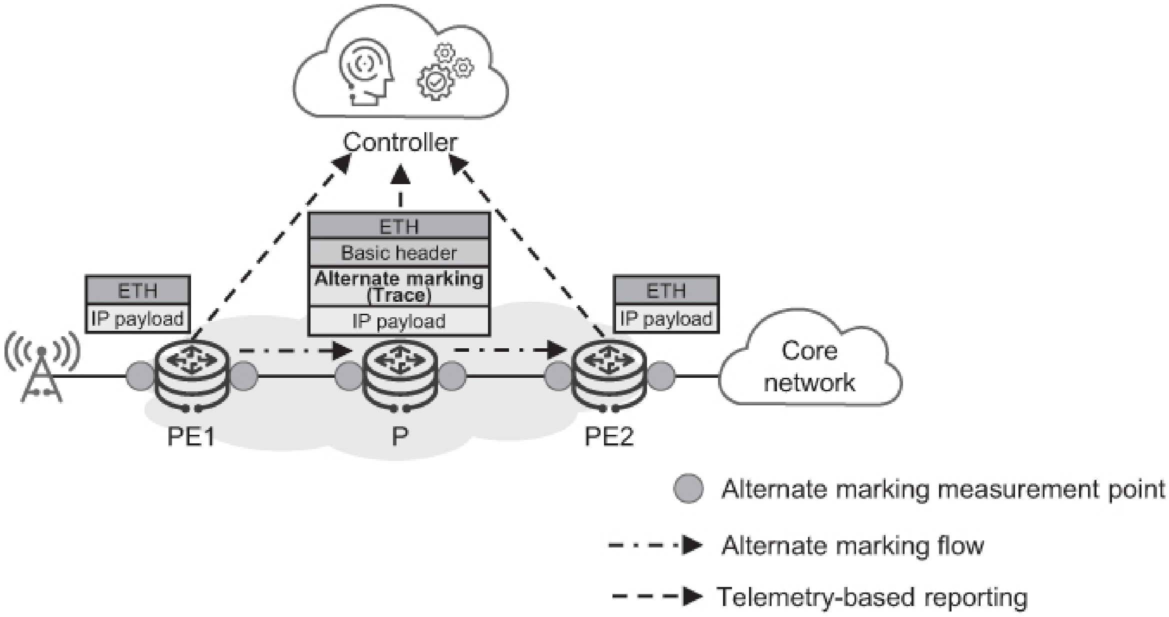


FIGURE 3.3 Hop-by-hop mode. ⏎

In most cases, E2E and hop-by-hop modes are used together, whereby hop-by-hop measurement is automatically triggered if the E2E measurement result reaches a specified threshold. This enables the forwarding path of the service flow to be traced and faults to be quickly demarcated and located. This also helps reduce the consumption of measurement resources, avoid reporting large amounts of data, and minimize deployment overheads.

3.1.2 Measurement Period Selection for Alternate Marking

The alternate marking method requires time synchronization among participating nodes and an appropriate measurement period to ensure that all packets within the measurement period are counted.

Time synchronization can be implemented by using the Network Time Protocol (NTP) or Precision Time Protocol (PTP). NTP offers a synchronization precision of 50 ms, which is sufficient for measurement periods within seconds. However, for nanosecond-level delay measurement, high-precision PTP is required. For details about these time synchronization technologies, see Appendix B.

To ensure measurement accuracy, R_x counters must be obtained after the last packet within the current period arrives and before the first packet in the next period arrives. The time at which R_x counters are obtained depends on the specific implementation. The following implementation serves as a reference only. Assume that the time synchronization error of nodes along the on-path telemetry path is Δ, the network transmission delay is D_t, the transmission out-of-order jitter is D_j, the time of the transmit end is T_Time, the time of the receive end is R_Time, the standard time is Time, and the counters on the transmit and receive ends are T_x and R_x, respectively.

Figure 3.4 shows a best-case scenario in which the time synchronization error between the transmit and receive ends is 0, and all time axes are identical with the standard time. The transmit end obtains the number of sent packets (T_x[N]) when the Nth period (T_Time[N] = Time [N]) ends. However, due to network delay, the receiving time of packets is $D_t + D_j$ later than the sending time of packets on the standard time axis. Assume that R_x[N] is obtained at R_Time[N] + 2T/3. R_x[N] is accurate as long as the time when R_x[N] is obtained (t2) is later than the arrival time (t1) of the last packet in this period. In other words, $D_t + D_j < 2T/3$.

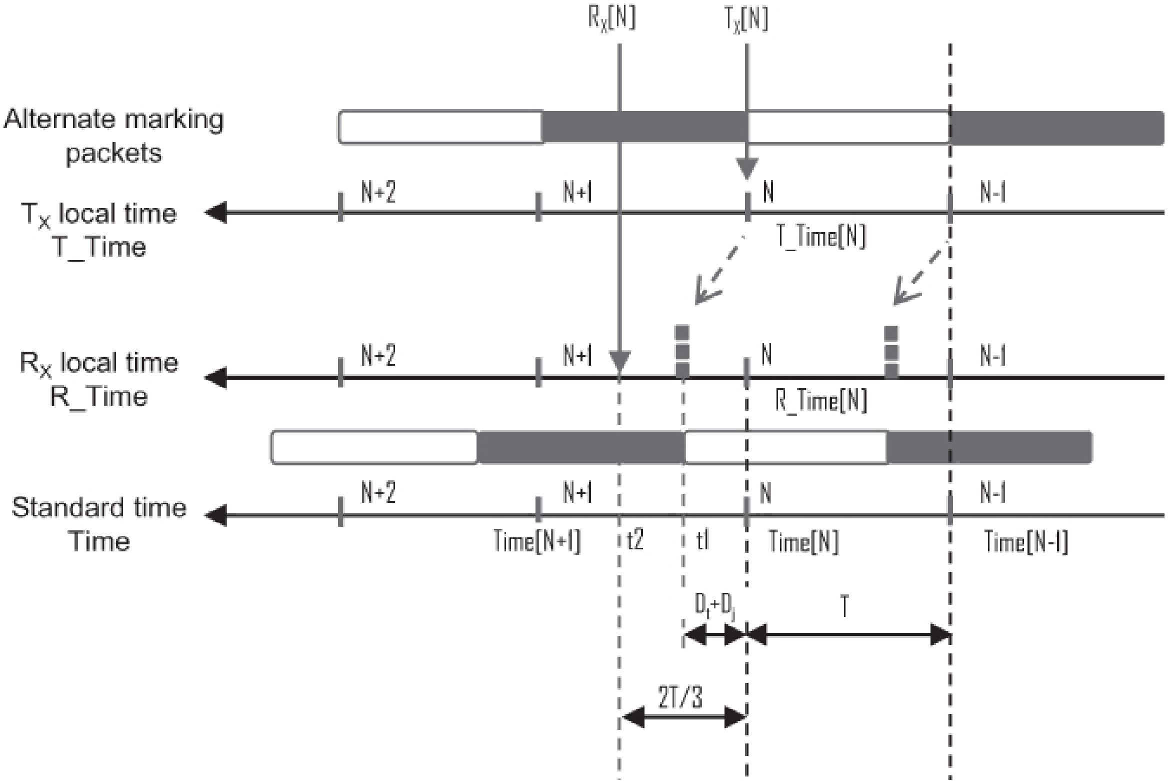


FIGURE 3.4 Measurement periods and synchronization precision analysis in a best-case scenario. ⏎

Figure 3.5 shows a worst-case scenario in which, with reference to the standard time, the local time at the transmit end is Δ later and that at the receive end is Δ earlier. In the Nth period, the transmit end obtains $T_x[N]$ at the local time T_Time[N], which is Time[N] + Δ. R_Time[N] at the receive end is Time[N] − Δ and is 2Δ earlier than T_Time[N]. In other words, the Nth period of the receive end is moved 2Δ rightwards on the time axis. This might cause packet counters to be obtained before all packets in the current period are received, resulting in inaccurate measurement.

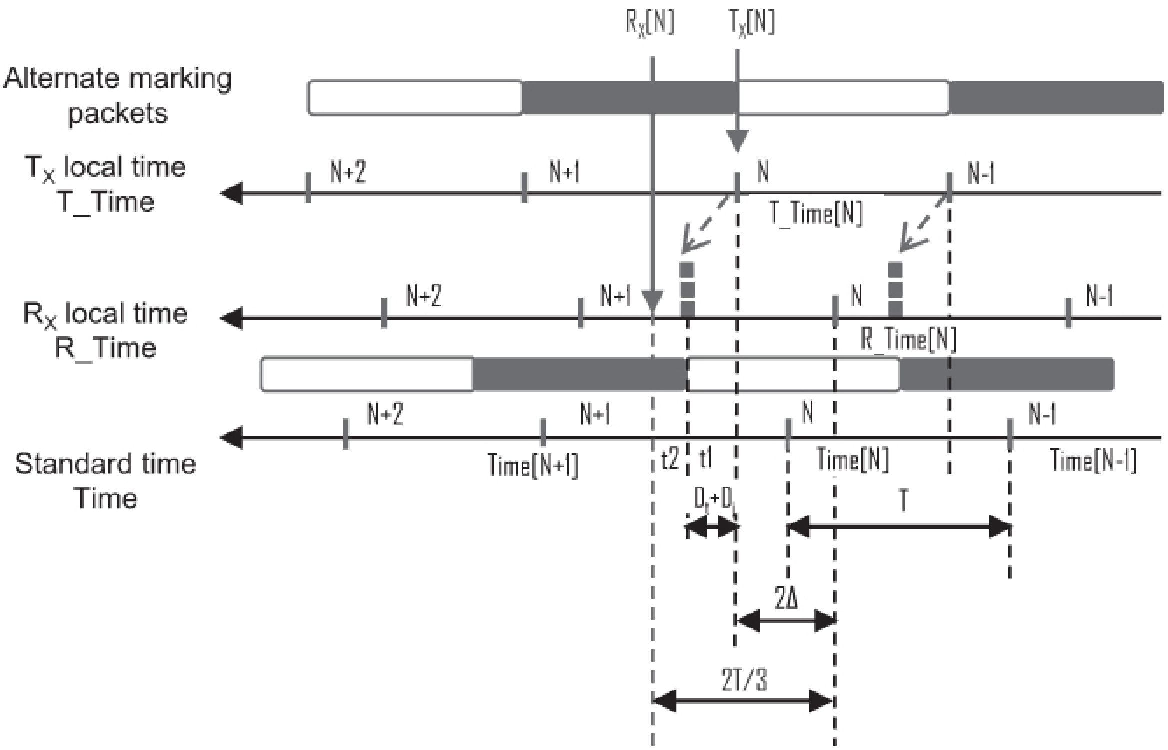


FIGURE 3.5 Measurement periods and synchronization precision analysis in a worst-case scenario (1). ⏎

Assume that the receive end obtains $R_x[N]$ at R_Time[N] + 2T/3. $R_x[N]$ is accurate as long as the time when $R_x[N]$ is obtained (t2) is later than the arrival time (t1) of the last packet in this period. In other words, $2\Delta + D_t + D_j < 2T/3$.

Figure 3.6 shows another worst-case scenario, but in this case, the local time at the transmit end is Δ earlier and that at the receive end is Δ later than the standard time. In the Nth period, the transmit end obtains $T_x[N]$ at the local time *T*_Time[N], which is Time[N] − Δ. R_Time[N] at the receive end is Time[N] + Δ and is 2Δ later than T_Time[N]. In other words, the Nth period of the receive end is moved 2Δ leftwards on the time axis. This might cause data in the current period to arrive before the receive end finishes obtaining packet counters in the previous period, resulting in counting errors.

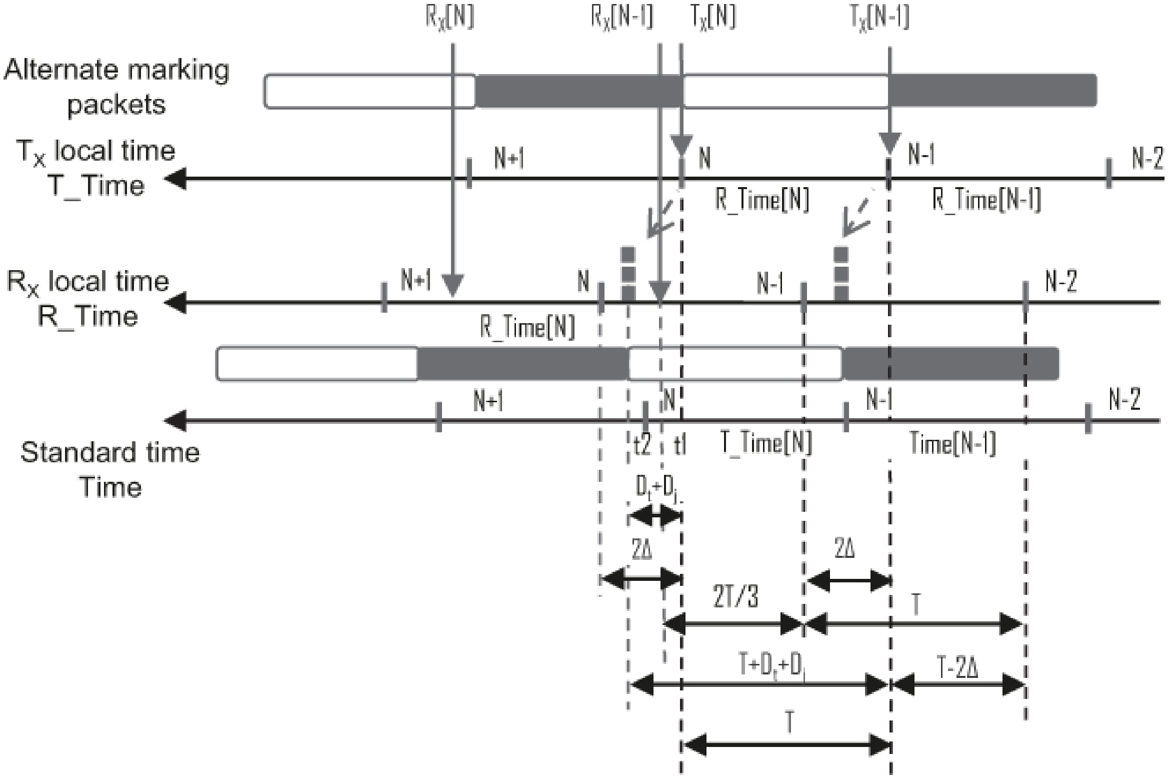


FIGURE 3.6 Measurement periods and synchronization precision analysis in a worst-case scenario (2). ⏎

Assume that the receive end obtains $R_x[N]$ at R_Time[N] + 2T/3 and obtains $R_x[N - 1]$ at R_Time[N] + 2T/3. R_x counters are accurate as long as the following condition is met: $(T - 2\Delta) + (T + D_t + D_j) > T + 2T/3$ or $2\Delta - (D_t + D_j) < T/3$ (that is, the time when the first packet in the current period arrives ($t2$) is later than the time when $R_x[N - 1]$ is obtained ($t1$)).

The time synchronization error of NTP is 50 ms, and the transmission delay of service flows between measurement nodes ranges from 5 to 20 ms. Assume that the maximum out-of-order delay is 200 ms. Considering the restrictions on period T in the preceding worst-case scenarios, measurement accuracy can be ensured as long as period T is set to 1s or longer. For the performance measurement period, Table 3.1 provides the recommended values, which take into account the software scheduling delay and appropriate processing interval. Setting the default measurement period to 3s is recommended.

TABLE 3.1 Performance Measurement Periods ⏎

Maximum Time Synchronization Error (ms)	Maximum Tolerable Out-of-Order Delay (ms)	Maximum Transmission Delay (ms)	Maximum Software Scheduling Delay (ms)	Minimum Appropriat Processing Interval (ms)
50	200	20	100	100
50	200	20	1000	100
50	200	20	1500	100
50	200	20	3000	100

3.1.3 Measurement Information Encapsulation for Alternate Marking

IPv6 defines a flexible extension header mechanism[2] that allows optional network layer information to be encoded into separate extension headers, each of which is identified by a unique Next Header (NH) value and placed between the IPv6 header and the upper-layer header. Alternate marking information can be encapsulated into one of the following IPv6 extension headers as an option or Type-Length-Value (TLV): IPv6 Hop-by-Hop Options Header (HBH), Destination Options Header (DOH), and Segment Routing Header (SRH). The following describes each of these methods.

3.1.3.1 Alternate Marking Information Encapsulated into an IPv6 HBH or DOH

RFC 9343 defines how to use the alternate marking method in hop-by-hop and E2E modes for measuring performance indicators[3] (i.e., packet loss, delay, and jitter) of an IPv6 network, enabling accurate fault locating. This RFC also defines an IPv6 Alternate Marking (AltMark) Option, which can be encapsulated into an HBH or DOH. Specifically, an HBH is used to encapsulate alternate marking information that needs hop-by-hop processing, whereas a DOH is used to encapsulate alternate marking information that needs only destination node processing. Figure 3.7 shows the format of IPv6 AltMark Option.

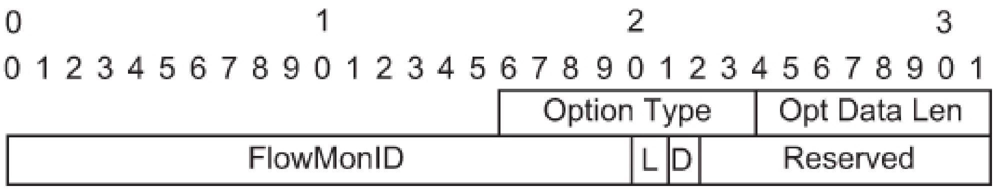


FIGURE 3.7 Format of IPv6 AltMark Option. ⏎

Table 3.2 describes the fields in IPv6 AltMark Option.

TABLE 3.2 Fields in IPv6 AltMark Option ⏎

Field	**Length**	**Description**
Option Type	8 bits	Option type. The type value of the AltMark Option defined in RFC 9343 is 0 × 12. As defined in RFC 8200, the two highest-order bits of the Option Type field are set to 00, indicating that an IPv6 node processing this option should skip it and continue to process the header if the node does not support this option. The third-highest-order bit of this option is set to 0, indicating that option data cannot be modified during forwarding[2].
Opt Data Len	8 bits	Length (in bytes) of the data fields (excluding Option Type and Opt Data Len fields) in this option.
FlowMonID	20 bits	Unique ID of a flow. The value is generated by the measurement ingress or centralized controller and must be unique.
L	1 bit	Loss measurement flag.
D	1 bit	Delay measurement flag.
Reserved	10 bits	Reserved field.

In addition, *draft-zhou-ippm-enhanced-alternate-marking* defines an enhanced alternate marking method[4]. It uses part of the Reserved field of IPv6 AltMark Option to introduce a 5-bit NextHeader (NH) field and adds optional extended data fields. This enables the alternate marking method to better adapt to large-scale networks and improves its future-oriented extensibility. Figure 3.8 shows the format of the enhanced IPv6 AltMark Option.

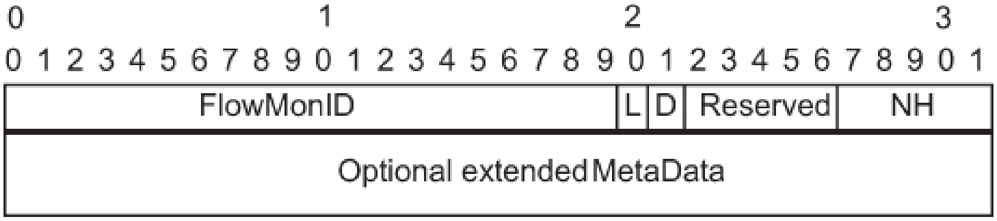


FIGURE 3.8 Format of the enhanced IPv6 AltMark Option.

Table 3.3 describes the fields in the enhanced IPv6 AltMark Option.

TABLE 3.3 Fields in the Enhanced IPv6 AltMark Option

Field	**Length**	**Description**
FlowMonID	20 bits	Unique ID of a flow. The value is generated by the measurement ingress or centralized controller and must be unique.
L	1 bit	Loss measurement flag.
D	1 bit	Delay measurement flag.
Reserved	5 bits	Reserved field.
NH	5 bits	The NH value of 0 is reserved for backward compatibility. It is recommended that values 1–15 be reserved for private use or for experimentation, and values 16–31 be defined by the IETF.
Optional extended MetaData	Variable	Optional extended metadata.

Currently, *draft-zhou-ippm-enhanced-alternate-marking* defines the format of optional extended data fields when the NH value is 16, as shown in Figure 3.9.

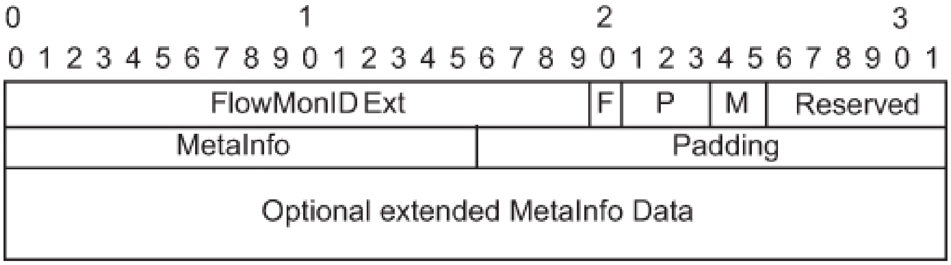


FIGURE 3.9 Format of optional extended data fields when the NH value is 16.

Table 3.4 describes the optional extended data fields when the NH value is 16.

TABLE 3.4 Optional Extended Data Fields When the NH Value Is 16 ⏎

Field	Length	Description
FlowMonID Ext	20 bits	Extended flow identifier. It is used to reduce FlowMonID conflict. Generally, it represents the node ID.
F	1 bit	Flow direction ID. The value 1 indicates a forward flow from the ingress to the egress. (Conversely, a flow from the egress to the ingress is a reverse flow.)
P	3 bits	Measurement period flag. The values are as follows: • 0b000: 1 s. • 0b001: 10 s. • 0b010: 30 s. • 0b011: 60 s. • 0b100: 300 s.
M	2 bits	Measurement mode flag. In cases where the IPv6 HBH cannot be used, the DOH can be used to carry alternate marking information, and this field can be used to identify the hop-by-hop or E2E mode. The values are described as follows: • 0b00: reserved field. • 0b01: E2E mode. • 0b10: hop-by-hop mode. • 0b11: reserved field.
Reserved	6 bits	Reserved field.
MetaInfo	16 bits	Bitmap used to indicate the extended metadata.
Padding	16 bits	Padding bit.
Optional extended MetaInfo Data	Variable	Extended data identified by MetaInfo. The format depends on the type of optional data (e.g., timestamp data or control information used for the automatic setup of reverse flow monitoring rules), as shown later.

MetaInfo is used as a bitmap to indicate whether more metadata is attached for the enhanced function. Table 3.5 describes the flag bits involved in the MetaInfo field.

TABLE 3.5 Flag Bits in the MetaInfo Field ⏎

Flag Bit	Description
Bit 0	When this bit is set to 1, a 6-byte timestamp is attached to MetaInfo and will overwrite the Padding field. Timestamp is used for measuring per-packet delay. Its first 2 bytes indicate the number of seconds, and the last 4 bytes indicate the number of nanoseconds.
Bit 1	When this bit is set to 1, a more detailed 4-byte control field is attached after MetaInfo. For details about the format, see "Format of the fields carrying control information used for the automatic setup of reverse flow monitoring rules."
Bit 2	When this bit is set to 1, a 4-byte sequence number is attached after MetaInfo. The sequence number is used to detect out-of-order packets.

If multiple flag bits are set in the MetaInfo field, the corresponding extended MetaInfo data must be placed in a packet in the same order as these flag bits. The formats of the data fields for the enhanced functions corresponding to the three preceding flags are as follows:

- When Bit 0 is set to 1, the timestamp data of when measurement information was encapsulated is attached after MetaInfo. Figure 3.10 shows the format of the timestamp data field.

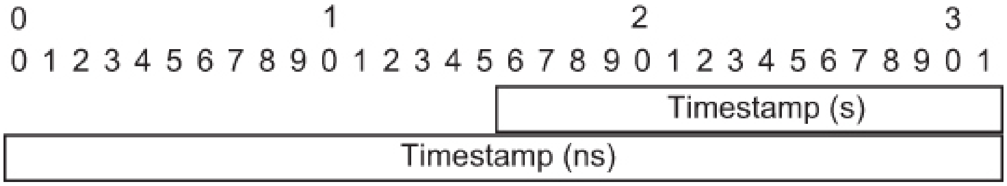


FIGURE 3.10 Timestamp data field. ⏎

Timestamp (s) indicates the seconds part of the timestamp and overwrites the Padding field after MetaInfo. Timestamp (ns) indicates the nanoseconds part of the timestamp. The node that encapsulates the alternate marking information generates the timestamp, which is then carried all the way to the decapsulating node. In this way, each transit node can compare the timestamp with its local time to measure the one-way delay.

- When Bit 1 is set to 1, control information about the automatic setup of reverse flow monitoring rules is attached after MetaInfo. Figure 3.11 shows the format of the control information data field.

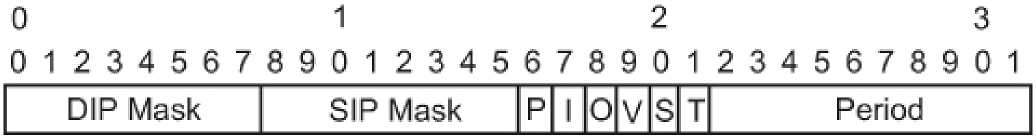


FIGURE 3.11 Format of the fields carrying control information used for the automatic setup of reverse flow monitoring rules. ⏎

Alternate marking instances need to be configured only on a centralized device (e.g., an aggregation device). After receiving a packet carrying alternate marking information, the edge device on which alternate marking is enabled extracts information such as the IP address and port number from the packet if the V field in the control information is set to 1. It then automatically sets up a reverse alternate marking instance based on (1) the IP address mask lengths and measurement period in the control information data field, and (2) whether information such as the protocol number and port number needs to be matched. This greatly simplifies deployment and configuration.

Table 3.6 describes the fields carrying control information used for the automatic setup of reverse flow monitoring rules.

TABLE 3.6 Fields Carrying Control Information Used for the Automatic Setup of Reverse Flow Monitoring Rules ⏎

Field	**Length**	**Description**
DIP Mask	8 bits	Mask length of the forward flow's destination IP address.
SIP Mask	8 bits	Mask length of the forward flow's source IP address.
P	1 bit	Protocol number matching flag. If the value is set to 1, a reverse flow setup rule must include matching the protocol number of the corresponding forward flow.
I	1 bit	Source port number matching flag. If the value is set to 1, the source port number in a reverse flow setup rule must match the destination port number of the corresponding forward flow.

Field	Length	Description
O	1 bit	Destination port number matching flag. If the value is set to 1, the destination port number in a reverse flow setup rule must match the source port number of the corresponding forward flow.
V	1 bit	Reverse flow monitoring flag. If the value is set to 1, reverse flow monitoring rules need to be set up. A node on which the automatic setup of reverse flow monitoring rules is enabled proactively creates a reverse flow setup rule based on the forward flow and carried control information, and automatically allocates a reverse flow ID if the V field in the forward flow is set to 1.
S	1 bit	Differentiated services code point (DSCP) matching flag. If the value is set to 1, a reverse flow setup rule must include matching the DSCP value of the corresponding forward flow.
T	1 bit	If the value is set to 1, measurement paths are established between Network-to-Network Interfaces (NNIs). If the value is set to 0, measurement paths are established between User-to-Network Interfaces (UNIs).
Period	10 bits	Measurement period, in seconds.

In the following example, assume that the forward flow setup rule is as follows: source IP address (SIP) = 2001:DB8:1:1:1:1::1, source IP address mask length (SIP Mask) = 96, destination IP address (DIP) = 2001:DB8:2:2:2::1, destination IP address mask length (DIP Mask) = 80, protocol number = 16, source port number = 100, and destination port number = 200. If the rule for the automatic setup of reverse flow monitoring needs to match the IP addresses and corresponding masks, protocol number, source port number, and destination port number of the forward flow, the control information used for the automatic setup of reverse flow monitoring rules is as follows: DIP Mask = 80, SIP Mask = 96, P = 1, I = 1, O = 1, and V = 1.

In this case, a reverse flow setup rule automatically created by the device is as follows: SIP = 2001:DB8:2:2:2::1, SIP Mask = 80, DIP = 2001:DB8:1:1:1:1::1, DIP Mask = 96, protocol number = 16, source port number = 200, and destination port number = 100.

- When Bit 2 is set to 1, a 4-byte sequence number is attached after MetaInfo. The sequence number can be used to detect packet loss and out-of-order packets. Figure 3.12 shows the format of the sequence number field.

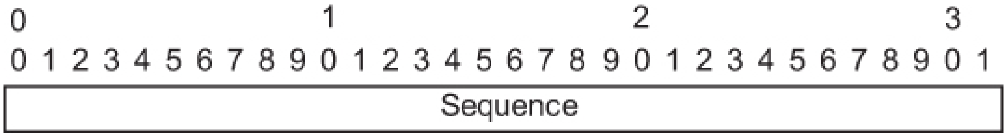


FIGURE 3.12 Format of the sequence number field. ⏎

Figure 3.13 shows the format of an IPv6 packet in which the DOH carries enhanced alternate marking information with the NH value being 16.

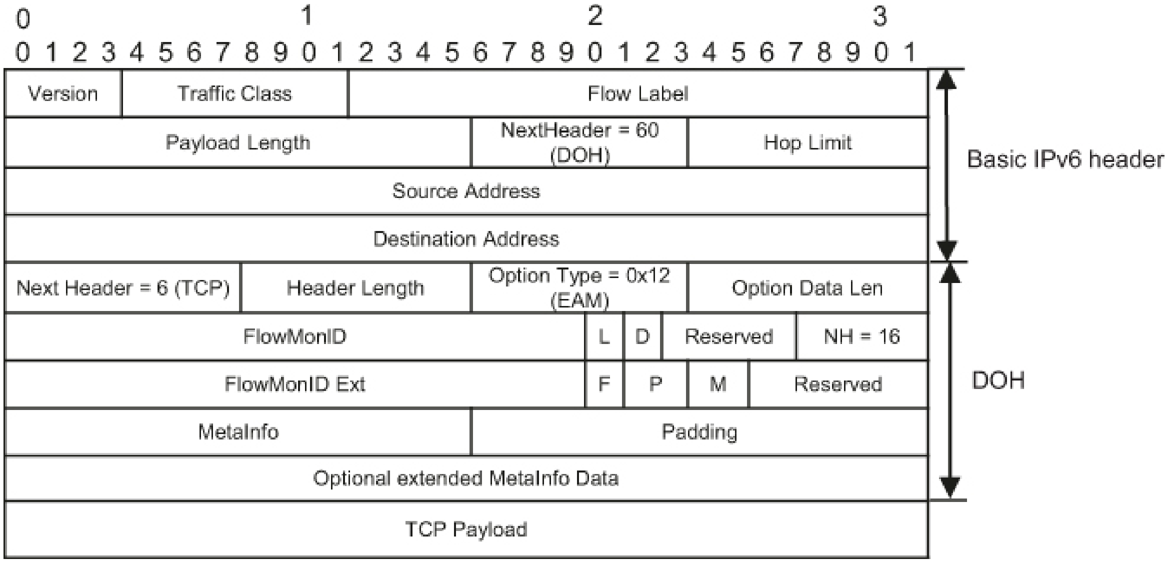


FIGURE 3.13 Format of an IPv6 packet in which the DOH carries enhanced alternate marking information with the NH value being 16. ⏎

3.1.3.2 Alternate Marking Information Encapsulated into an SRH

On an SRv6 network[5], an SRH specifies the path through which a data packet needs to pass and related additional information[6]. Figure 3.14 shows the format of an IPv6 packet carrying an SRH.

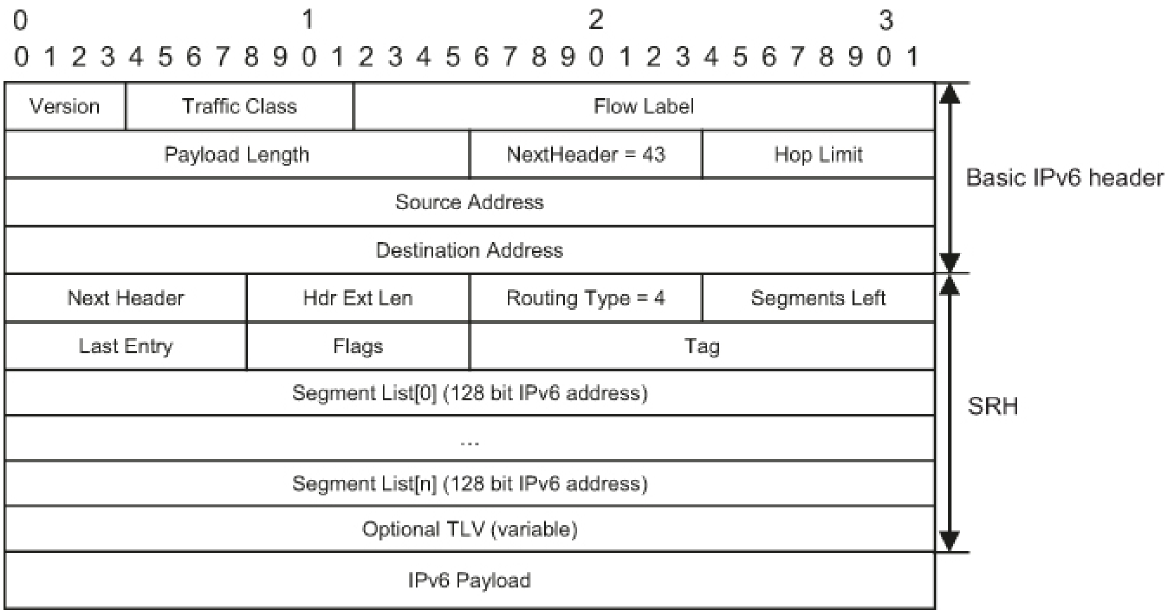


FIGURE 3.14 Format of an IPv6 packet carrying an SRH. ⏎

Table 3.7 describes the fields in the SRH.

TABLE 3.7 Fields in the SRH ⏎

Field	**Length**	**Description**
Next Header	1 byte	Type of the header immediately following the SRH. Common header types are as follows: • 4: IPv4. • 6: TCP. • 17: UDP. • 41: IPv6. • 43: IPv6-Route (IPv6 Routing Header (RH)). • 58: Internet Control Message Protocol Version 6 (ICMPv6). • 59: no next header.
Hdr Ext Len	1 byte	Length of an SRH, excluding the first 8 bytes, in multiples of 8 bytes.
Routing Type	1 byte	Type of the RH. SRHs have a value of 4.
Segments Left	1 byte	Number of remaining segments. It is called SL for short.

Field	Length	Description
Last Entry	1 byte	Index of the last element in a segment list.
Flags	1 byte	Flags reserved for special processing, such as OAM.
Tag	2 bytes	Tag indicating whether a packet is part of a group of packets, such as those sharing the same set of properties.
Segment List[n]	128 bits	The nth segment in a segment list, expressed using an IPv6 address.
Optional TLV	Variable	TLVs, such as padding TLVs and Hash-Based Message Authentication Code (HMAC) TLVs.

The variable-length SRH TLV (Optional TLV) can be used to carry alternate marking information. This approach allows a specified SRv6 node to process alternate marking information, unlike the approach described in RFC 9343, whereby alternate marking information is encapsulated into the HBH or DOH. The use of the SRH TLV enables the alternate marking method to be deployed on demand and nodes to support it through on-demand upgrades. An optional SRH TLV (SRH AltMark TLV)[7] is defined in *draft-fz-spring-srv6-alt-mark* for carrying alternate marking information in an SRv6 SRH. Figure 3.15 shows its format. Many vendors have already implemented this approach, which can be used as a reference for ensuring interoperability between vendors.

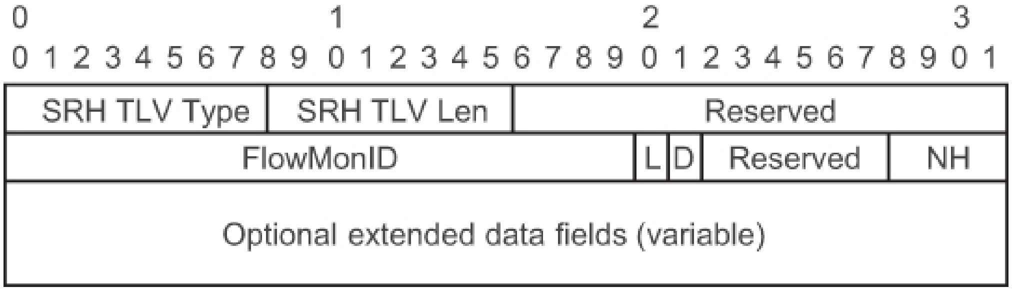


FIGURE 3.15 Format of the SRH AltMark TLV. ⏎

Table 3.8 describes the fields in the SRH AltMark TLV.

TABLE 3.8 Fields in the SRH AltMark TLV. ⏎

Field	Length	Description

Field	Length	Description
SRH TLV Type	8 bits	SRH TLV type. Setting this field to 130 is recommended.
SRH TLV Len	8 bits	Length (in bytes) of the data fields (excluding the SRH TLV Type and SRH TLV Len fields) in this TLV.
Reserved	16 bits	Reserved field.
FlowMonID	20 bits	Unique ID of a flow. The value is generated by the measurement ingress or centralized controller and must be unique.
L	1 bit	Packet loss measurement flag.
D	1 bit	Delay measurement flag.
Reserved	6 bits	Reserved field.
NH	4 bits	Used to indicate the presence of extended data fields. The value 0 indicates that no optional extended data fields are carried.
Optional extended data fields	Variable	Variable-length optional extended data fields. Their presence is determined by the setting of the NH field.

Currently, *draft-fz-spring-srv6-alt-mark* defines the format of optional extended data fields when the NH value is 9, as shown in Figure 3.16.

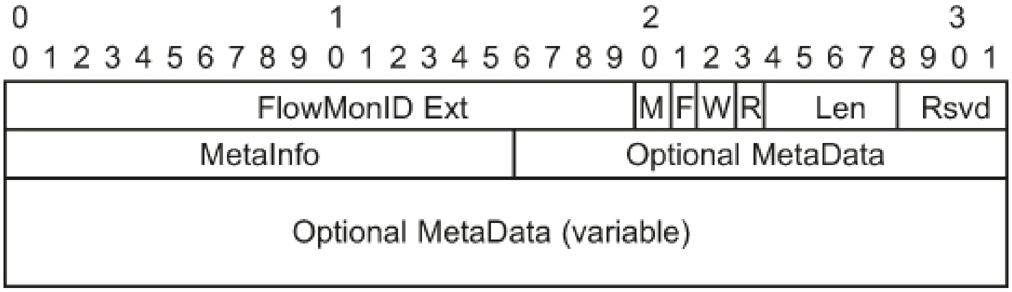


FIGURE 3.16 Format of the optional extended data fields when the NH value is 9. ⏎

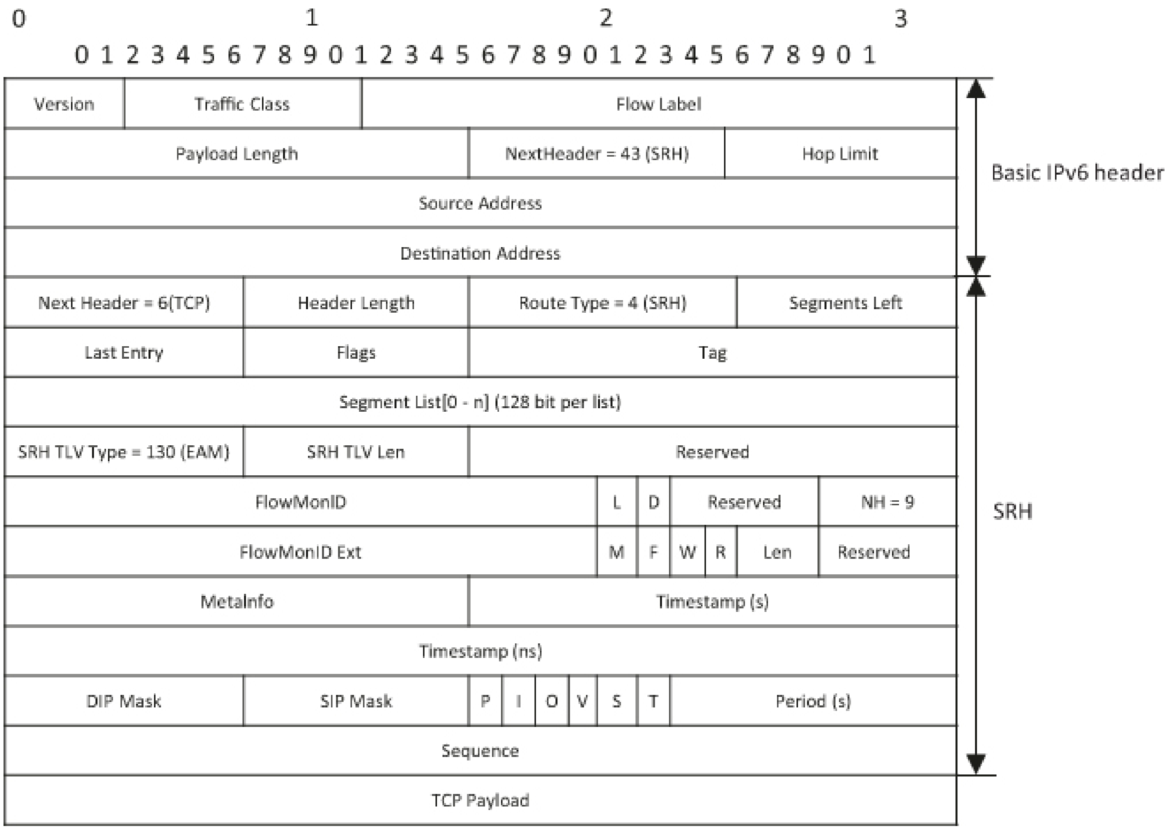
Table 3.9 describes the optional extended data fields when the NH value is 9.

TABLE 3.9 Optional Extended Data Fields When the NH Value Is 9 ⏎

Field	Length	Description
FlowMonID Ext	20 bits	Extended flow identifier. It is used to reduce FlowMonID conflict. Generally, it represents the node ID.
M	1 bit	Measurement mode. Value 1 indicates end-to-end measurement, and value 0 indicates hop-by-hop measurement.
F	1 bit	The value 1 indicates that the original packet to be measured is fragmented.
W	1 bit	The value 1 indicates a forward flow, and the value 0 indicates a reverse flow.
R	1 bit	Reserved field.
Len	4 bits	Length (in bytes) of all optional extended data fields when the NH value is 9.
Rsvd	4 bits	Reserved field.
MetaInfo	16 bits	Bitmap used to indicate more extended metadata for enhanced functions.
Optional MetaData	Variable	Optional metadata.

The MetaInfo and Optional MetaData fields are the same as the MetaInfo and Optional extended MetaInfo Data fields described in *draft-zhou-ippm-enhanced-alternate-marking*.

Figure 3.17 shows the format of an IPv6 packet in which the SRH TLV carries enhanced alternate marking information.



Version	Traffic Class	Flow Label						
Payload Length	NextHeader = 43 (SRH)	Hop Limit						
Source Address								
Destination Address								
Next Header = 6(TCP)	Header Length	Route Type = 4 (SRH)	Segments Left					
Last Entry	Flags	Tag						
Segment List[0 - n] (128 bit per list)								
SRH TLV Type = 130 (EAM)	SRH TLV Len	Reserved						
FlowMonID	L	D	Reserved	NH = 9				
FlowMonID Ext	M	F	W	R	Len	Reserved		
MetaInfo	Timestamp (s)							
Timestamp (ns)								
DIP Mask	SIP Mask	P	I	O	V	S	T	Period (s)
Sequence								
TCP Payload								

FIGURE 3.17 Format of an IPv6 packet in which the SRH TLV carries enhanced alternate marking information. ⏎

3.2 IOAM METHOD

The IOAM method of IPv6 on-path telemetry defines a series of measurement instructions and data formats to record and collect measurement information during packet forwarding, implementing on-path measurement.

3.2.1 How the IOAM Method Works

As indicated by “I” (referring to “In-Situ”) in IOAM, OAM measurement data is carried in service data packets rather than in OAM-specific data packets. In addition to achieving higher measurement precision than active measurement mechanisms, IOAM can be used for the following purposes:

- Prove that a group of packets is forwarded along a predefined path.
- Collect detailed statistics about traffic forwarded over multiple paths.
- Prevent a network device from processing separate measurement packets and conventional data packets differently.

IOAM defines a series of measurement instructions and data formats to implement on-path measurement. Figure 3.18 shows the IOAM reference model[8].

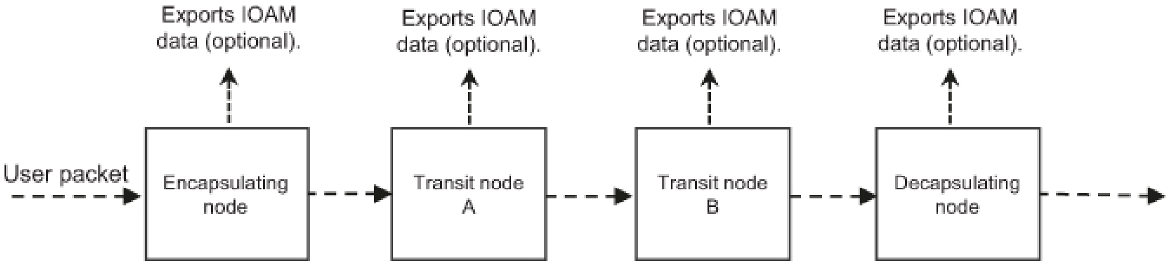

Exports IOAM data (optional).
Exports IOAM data (optional).
Exports IOAM data (optional).
Exports IOAM data (optional).
User packet
Encapsulating node
Transit node A
Transit node B
Decapsulating node

FIGURE 3.18 IOAM reference model for implementing on-path measurement. ⏎

IOAM focuses on limited domains[9], referred to as IOAM-Domains. An IOAM-Domain consists of encapsulating nodes, transit nodes, and decapsulating nodes.

A node plays a defined role (encapsulating, transit, or decapsulating) within an IOAM-Namespace and can play different roles in different IOAM-Namespaces. An encapsulating node encapsulates one or more IOAM Option-Types into specified data packets, a transit node updates IOAM Data-Fields according to these Option-Types, and a decapsulating node removes all these Option-Types from data packets before sending the packets to the destination. In addition, each node reports IOAM data to the control and analysis device as required based on IOAM Option-Types.

Similar to the alternate marking method, IOAM also requires time synchronization among participating devices. For details about time synchronization technology, see Appendix B.

IOAM defines the following options for different application scenarios:

- Per-hop Tracing Option-Type: records information about the path through which a measurement data packet passes on a per-hop basis, including information about the delay and jitter. There are two option types for per-hop tracing: Pre-allocated Trace Option-Type and Incremental Trace Option-Type. The former refers to a mode in which the IOAM encapsulating node pre-allocates space for IOAM Data-Fields of all nodes along the path, whereas the latter refers to a mode in which space for IOAM Data-Fields is applied for hop by hop.
- Proof of Transit (POT) Option-Type: used to verify whether a data packet passes through a predetermined path.

- Edge-to-Edge Option-Type: used to carry the E2E performance data of an IOAM-Domain.
- Direct Export Option-Type: used to instruct each node to directly report IOAM data. This prevents data packets from increasing in size as they accumulate IOAM data added at each hop, thereby reducing measurement bandwidth and processing overheads.

3.2.2 Encapsulation of IOAM Data-Fields

RFC 9486 defines how IOAM Data-Fields are encapsulated in IPv6. They can be encapsulated into the HBH or DOH[10] of an IPv6 packet. Figure 3.19 shows the IPv6 IOAM Option format.

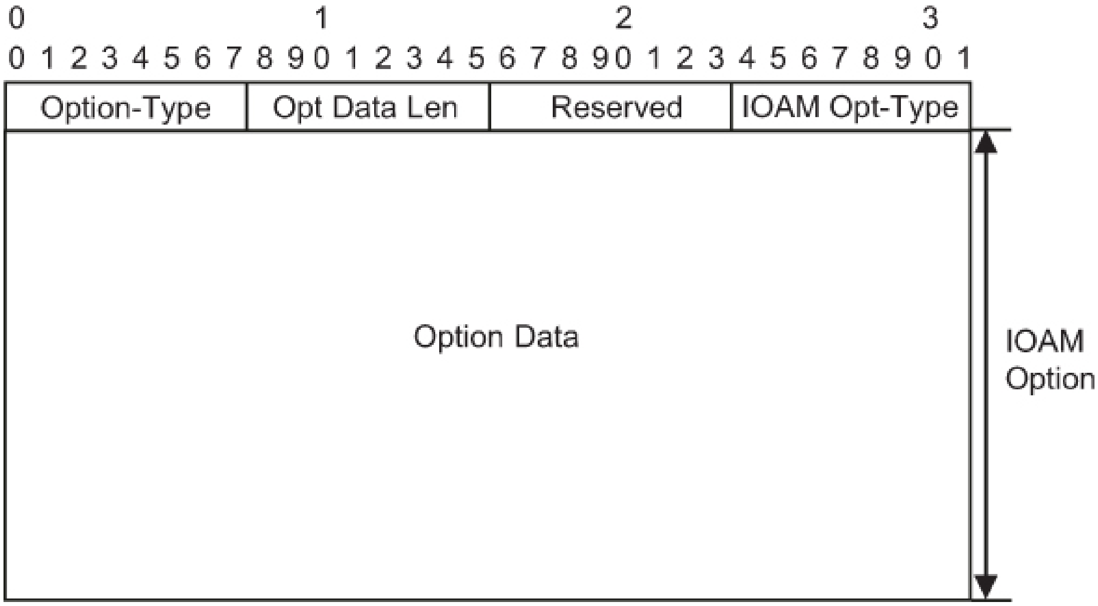


FIGURE 3.19 IPv6 IOAM Option format. ⏎

Table 3.10 describes the fields in IPv6 IOAM Option.

TABLE 3.10 Fields in IPv6 IOAM Option ⏎

Field	Length	Description
Option-type	1 byte	Option-Type. The values of this field are as follows: • 0 × 11: indicates that option data cannot be changed. This value is used when IOAM Opt-Type is set to Edge-to-Edge or Direct Export. • 0 × 31: indicates that option data can be changed. This value is used when IOAM Opt-Type is set to Pre-allocated Trace, Incremental Trace, or POT.

Field	Length	Description
Opt Data Len	1 byte	Length (in bytes) of Option Data (excluding the Option-Type and Opt Data Len fields).
Reserved	1 byte	Reserved field.
IOAM Opt-Type	1 byte	IOAM option type. RFC 9197 defines the IOAM option types of value ranging from 0 to 3[11], and RFC 9326 defines the IOAM option type of value 4[12] as follows: • 0: IOAM Pre-allocated Trace Option-Type • 1: IOAM Incremental Trace Option-Type • 2: IOAM POT Option-Type • 3: IOAM Edge-to-Edge Option-Type • 4: IOAM Direct Export Option-Type.
Option Data	Variable	Option data specific to Option-Types.

The following describes the preceding IOAM Option-Types.

3.2.2.1 IOAM Per-hop Tracing Option-Type

With IOAM Per-hop Tracing Option-Type, OAM information about each IOAM node through which a data packet passes is collected and stored in the data packet. IOAM Per-hop Tracing Option-Type can be used for the following purposes:

- Collect information about different paths that different data packets traverse between an IOAM encapsulating node and an IOAM decapsulating node on a network that implements load balancing. This information can be used to optimize the load balancing algorithm in order to enhance network resource utilization.
- Collect information about the path through which a particular data packet or a set of data packets pass between the IOAM encapsulating node and the IOAM decapsulating node, and collect the delay and jitter data in data packets on different nodes along the path.

There are two option-types for per-hop tracing: IOAM Pre-allocated Trace Option-Type and IOAM Incremental Trace Option-Type. The former refers to a mode in which the IOAM encapsulating node pre-allocates space for IOAM

Data-Fields of all nodes along the path, whereas the latter refers to a mode in which space for IOAM Data-Fields is applied for hop-by-hop. In general, IOAM Pre-allocated Trace Option-Type is used for software-based IOAM, and IOAM Incremental Trace Option-Type is used for hardware-based IOAM in order to achieve better performance. The implementation depends on user needs. The two Option-Types have the same format, as shown in Figure 3.20.

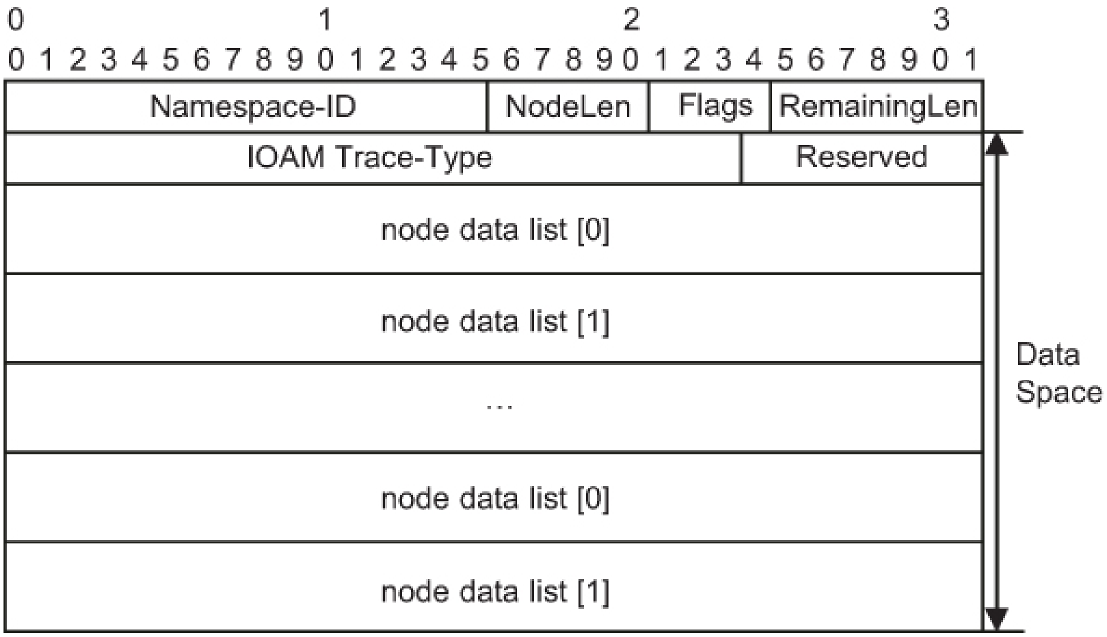


FIGURE 3.20 Format of IOAM Per-hop Tracing Option-Type. ⏎

Table 3.11 describes the fields in IOAM Per-hop Tracing Option-Type.

TABLE 3.11 Fields in IOAM Per-Hop Tracing Option-Type ⏎

Field	Length	Description
Namespace-ID	16 bits	Identifier of an IOAM-Namespace. IOAM processing is performed only when the locally configured Namespace-ID is the same as the Namespace-ID carried in the packet. 0×0000 is the default Namespace-ID and must be supported by all IOAM nodes.
NodeLen	5 bits	Length (in multiples of 4 bytes) of the data added to the option by each IOAM node.
Flags	4 bits	Flag bits. When Bit 0 is set to 1, the node data list has overflowed, and subsequent IOAM nodes do not need to process the data space.

Field	Length	Description
RemainingLen	7 bits	Remaining length. This field is used to identify the remaining length of the pre-allocated data space that can be filled in by an IOAM node.
IOAM Trace-Type	24 bits	Types of data contained in the node data list. Each bit in this field represents a data type. The packing sequence of the data fields in each node data element is the same as the bit sequence of the IOAM Trace-Type field. The bits are defined as follows: • Bit 0: If this bit is set to 1, Hop_Lim (1-byte hop count, copied from the Hop Limit field in the IPv6 header of a service packet) and node_id (3-byte node ID) in short format are present in the node data list. • Bit 1: If this bit is set to 1, ingress_if_id (2-byte ingress interface identifier) and egress_if_id (2-byte egress interface identifier) in short format are present in the node data list. • Bit 2: If this bit is set to 1, the 4-byte timestamp seconds field is present in the node data list. • Bit 3: If this bit is set to 1, the 4-byte timestamp fraction field is present in the node data list. • Bit 4: If this bit is set to 1, the 4-byte transit delay field is present in the node data list. • Bit 5: If this bit is set to 1, the 4-byte user-defined IOAM-Namespace-specific data field (short format) is present in the node data list. • Bit 6: If this bit is set to 1, the 4-byte queue depth field is present in the node data list. • Bit 7: If this bit is set to 1, the 4-byte Checksum Complement field is present in the node data list. • Bit 8: If this bit is set to 1, 1-byte hop count and 7-byte node ID (wide format) are present

Field	Length	Description
		in the node data list. • Bit 9: If this bit is set to 1, 4-byte ingress interface identifier and 4-byte egress interface identifier (wide format) are present in the node data list. • Bit 10: If this bit is set to 1, the 8-byte user-defined IOAM-Namespace-specific data field (wide format) is present in the node data list. • Bit 11: If this bit is set to 1, 4-byte buffer occupancy is stored in the node data list. • Bits 12 to 21: reserved. • Bit22: If this bit is set to 1, variable-length user-defined data is stored in the node data list. • Bit23: reserved. • Bits 12–21: reserved. • Bit 22: If this bit is set to 1, variable-length user-defined data is stored in the node data list. • Bit 23: reserved.
Reserved	8 bits	Reserved field whose value is fixed at 0.
node data list [n]	Variable	Area for storing node data defined by IOAM Trace-Type.

3.2.2.2 IOAM POT Option-Type

In scenarios where a user subscribes to different value-added services (e.g., traffic cleaning and traffic acceleration), the user needs to confirm the path through which data packets pass in order to verify the validity of these services. The IOAM POT Option-Type defines data packet identifiers and a group of information about node-by-node iteration operations, making it possible to determine whether a data packet passes through a defined path. Figure 3.21 shows the format of IOAM POT Option-Type.

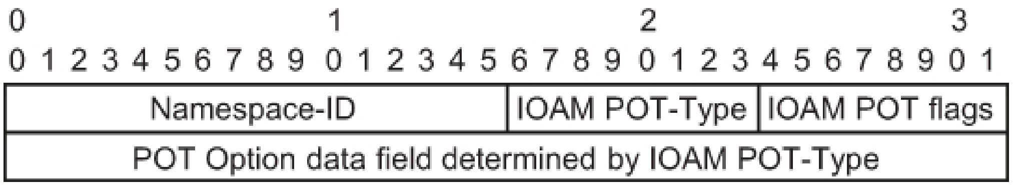


FIGURE 3.21 Format of IOAM POT Option-Type. ⏎

Table 3.12 describes the fields in IOAM POT Option-Type.

TABLE 3.12 Fields in IOAM POT Option-Type ⏎

Field	Length	Description
Namespace-ID	2 bytes	The definition of this field is the same as that of the Namespace-ID field in "Fields in IOAM Per-hop Tracing Option-Type."
IOAM POT-Type	1 byte	IOAM POT Option-Type.
IOAM POT flags	1 byte	Flag bits, which are not currently defined.
POT Option data field determined by IOAM POT-Type	4 bytes	Optional POT Option data. Its type is determined by IOAM POT-Type.

RFC 9197 defines IOAM POT-Type 0, indicating that POT data is a 16-byte field used to carry data associated with POT procedures[11]. Figure 3.22 shows the format of IOAM POT Option-Type with IOAM POT-Type being 0.

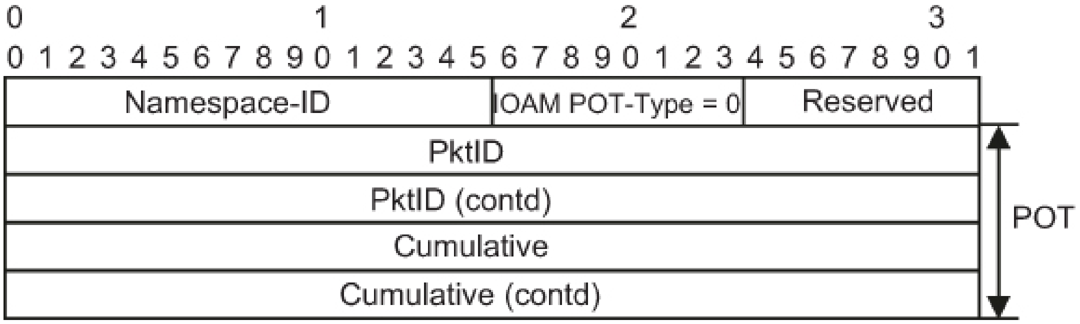


FIGURE 3.22 Format of IOAM POT Option-Type with IOAM POT-Type being 0. ⏎

Table 3.13 describes the fields in IOAM POT Option-Type with IOAM POT-Type being 0.

TABLE 3.13 Fields in IOAM POT Option-Type with IOAM POT-Type Being 0. ⏎

Field	Length	Description
Namespace-ID	2 bytes	The definition of this field is the same as that of the Namespace-ID field in “Fields in IOAM Per-hop Tracing Option-Type.”
IOAM POT-Type	1 byte	IOAM POT Option-Type. The value is 0.
Reserved	1 byte	Reserved field.
PktID	8 bytes	Packet identifier. The allocation mechanism is implementation-specific.
Cumulative	8 bytes	Path proof information, which is implementation-specific.

Path verification involves checking whether the value of the Cumulative field is the same as the expected value on the IOAM egress.

3.2.2.3 IOAM Edge-to-Edge Option-Type

The Edge-to-Edge Option-Type is used to carry the E2E performance data of service traffic. IOAM option data is added by the IOAM encapsulating node and parsed by the IOAM decapsulating node to obtain the E2E performance data of the involved IOAM-Domain. Figure 3.23 shows the format of IOAM Edge-to-Edge Option-Type.

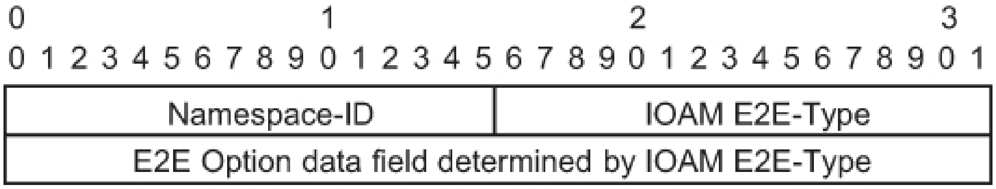


FIGURE 3.23 Format of IOAM Edge-to-Edge Option-Type. ⏎

Table 3.14 describes the fields in IOAM Edge-to-Edge Option-Type.

TABLE 3.14 Fields in IOAM Edge-to-Edge Option-Type ⏎

Field	Length	Description
Namespace-ID	2 bytes	The definition of this field is the same as that of the Namespace-ID field in “Fields in IOAM Per-hop Tracing Option-Type.”

Field	Length	Description
IOAM E2E-Type	2 bytes	Types of optional Edge-to-Edge Option data. Each bit in this field represents a data type. The packing sequence of the data fields in each node data element is the same as the bit sequence of the IOAM Trace-Type field. The bits are defined as follows: • Bit 0 (most significant bit): If this bit is set to 1, the 8-byte sequence number field is present in the E2E option data field determined by IOAM E2E-Type (Edge-to-Edge Option data for short). • Bit 1: If this bit is set to 1, a 4-byte sequence number field is present in Edge-to-Edge Option data. • Bit 2: If this bit is set to 1, the 4-byte timestamp seconds field is present in Edge-to-Edge Option data. This field represents the time at which the packet entered the IOAM-Domain. • Bit 3: If this bit is set to 1, the 4-byte timestamp fraction field is present in Edge-to-Edge Option data. This field represents the time at which the packet entered the IOAM-Domain. • Bits 4–15: reserved fields. The value 0 indicates that the corresponding packet field is invalid.
E2E Option data field determined by IOAM E2E-Type	Variable	Optional Edge-to-Edge Option data. This field is determined by IOAM E2E-Type. The sequence number field can be used to collect statistics on E2E packet loss and disorder, and the timestamp seconds and timestamp fraction fields can be used to collect statistics on E2E delay.

3.2.2.4 IOAM Direct Export Option-Type

The IOAM Option-Types described earlier incur relatively high bandwidth and processing overheads. This is because an IOAM header is encapsulated by the IOAM encapsulating node, and IOAM Option data is only reported by the IOAM decapsulating node, leading to IOAM Option data being accumulated in data packets.

RFC 9326 defines a method of implementing on-path measurement in postcard mode, that is, IOAM Direct Export Option-Type[12]. In this mode, each IOAM measurement node directly reports IOAM data, thereby preventing hop-by-hop accumulation of IOAM Option data, reducing measurement bandwidth and processing overheads, and achieving better deployability. Figure 3.24 shows the architecture of IOAM Direct Export Option-Type.

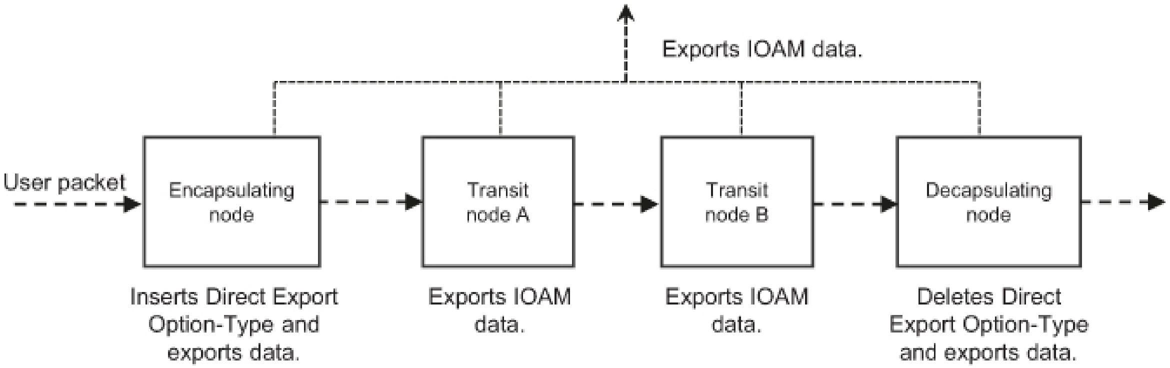
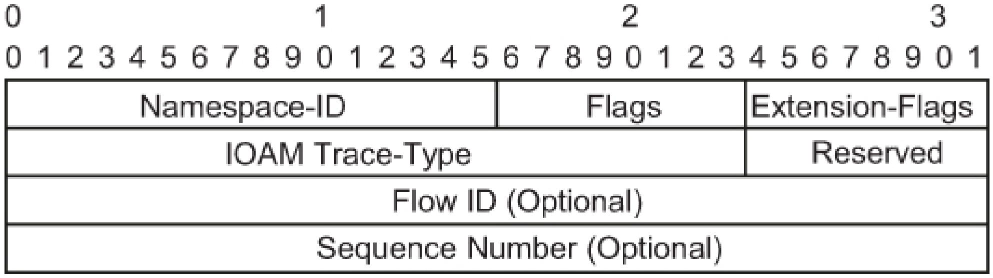


FIGURE 3.24 IOAM Direct Export Option-Type architecture. ⏎

The IOAM encapsulating node encapsulates Direct Export Option-Type information into a data packet. Each IOAM transit node through which the data packet passes exports IOAM Data-Fields according to Direct Export Option-Type and reports them to the control and analysis device for centralized analysis and display. Figure 3.25 shows the format of IOAM Direct Export Option-Type.



```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+---------------+---------------+
|         Namespace-ID          |     Flags     |Extension-Flags|
+-------------------------------+---------------+---------------+
|               IOAM Trace-Type                 |   Reserved    |
+-----------------------------------------------+---------------+
|                       Flow ID (Optional)                      |
+---------------------------------------------------------------+
|                  Sequence Number (Optional)                   |
+---------------------------------------------------------------+
```

FIGURE 3.25 Format of IOAM Direct Export Option-Type ⏎

Table 3.15 describes the fields in IOAM Direct Export Option-Type.

TABLE 3.15 Fields in IOAM Direct Export Option-Type ⏎

Field	Length	Description
Namespace-ID	2 bytes	The definition of this field is the same as that of the Namespace-ID field in "Fields in IOAM Per-hop Tracing Option-Type."
Flags	1 byte	Flag bits, which are not currently defined.
Extension-Flags	1 byte	Indicates the bitmap that carries extension information. The values of this field are as follows: • Bit 0: If this field is set to 1, the optional Flow ID field is present. • Bit 1: If this bit is set to 1, the optional Sequence Number field is present.
IOAM Trace-Type	3 bytes	Specifies which IOAM trace data needs to be exported. The definition of this field is the same as that of the IOAM-Trace-Type field in "Fields in IOAM Per-hop Tracing Option-Type."
Reserved	1 byte	Reserved field.
Flow ID	4 bytes	Optional flow ID. When multiple IOAM measurement nodes report data, the involved controller calculates measurement data based on the flow ID.
Sequence Number	4 bytes	Optional packet sequence number. The value of this field starts from 0 and increases by 1 each time a packet is counted.

3.3 STORIES BEHIND IPV6 ON-PATH TELEMETRY

3.3.1 Why Is MPLS Outdated?

MPLS has been a significant innovation in IP technology and played a pivotal role in the IP-based transformation of telecom networks. By supporting important features like VPN, TE, and Fast Reroute (FRR) through extensions, MPLS meets carrier-grade network requirements such as multi-service transport, QoS guarantee, and high reliability. To enhance OAM functionality, MPLS — MPLS-TP in particular — has introduced many innovations, including performance measurement techniques. However, because MPLS-TP uses out-of-band measurement methods, the flexibility of IP networks makes the use of these techniques challenging. To address these challenges, certain restrictions have been imposed on MPLS-TP, such as prohibiting the use of PHP and load balancing for MPLS LSPs. These restrictions limit the application scenarios of MPLS-TP.

The optimal solution to these problems lies in on-path measurement, which is the only technique capable of overcoming the various limitations encountered thus far. MPLS has faced unprecedented challenges related to its extensibility. Before the introduction of on-path measurement, MPLS used structured data (labels within the label stack) to extend the information space of the data plane. However, carrying the metadata involved in on-path measurement using structured labels is difficult. We realized this issue early in our research into on-path measurement and filed a patent application for MPLS extension headers, subsequently submitting a draft[13] to the MPLS WG. Our intention was to draw inspiration from the IPv6 extension header mechanism and place MPLS extension headers between the MPLS label stack and payload in order to carry on-path measurement information and other metadata. Given our concurrent SRv6-related research and development activities, we compared MPLS extension headers with IPv6 extension headers in order to make an informed choice about future technology. Following our thorough comparison and analysis, we concluded that although the MPLS extension header mechanism can offer flexible extensibility, its introduction came too late. MPLS was not originally designed with such an extension mechanism, and introducing MPLS extension headers at this stage posed a critical issue — poor compatibility. The implementation of MPLS assumes that an MPLS label stack precedes the payload; using MPLS extension headers disrupts this assumption. Originally, MPLS packet forwarding could rely solely on the topmost label of the label stack. But in order to enhance load balancing and achieve other functions, most device vendors implemented solutions that read the entire MPLS label stack and even the IPv4 and IPv6 headers in the payload. This means that the MPLS extension header mechanism would affect existing MPLS forwarding — its introduction would require network-wide upgrades due to compatibility issues, posing significant challenges given the numerous MPLS networks already in existence. In contrast, adopting the IPv6 extension header mechanism, which is incompatible with IPv4, enables the deployment of an additional IPv6/SRv6 (virtual) plane alongside the IPv4/MPLS (virtual) plane within the same network, as these two planes barely affect each other. Moreover, the forward compatibility mechanism was defined for IPv6 extension headers at the beginning of IPv6 design. This means that legacy IPv6 devices unable to identify option information in IPv6 extension headers of IPv6 packets can still forward the packets — these devices treat them as common IPv6 packets if a specific bit is set in the option-type definition. As a result, IPv6 packets with new extensions can traverse existing IPv6 devices, enabling incremental deployment and evolution of new extended functions. This makes it far easier to introduce new technologies and facilitate their development.

IPv6 extension headers offer significantly more extensibility compared to the MPLS label stack. Nevertheless, the IPv6 extension header mechanism, which was introduced at the outset of IPv6, has seen limited adoption over the past several decades. This has largely been due to the constraints imposed by network forwarding hardware capabilities. Hardware-based forwarding systems can process packets with fixed-length headers more easily, achieving optimal forwarding performance. In contrast, handling packets with variable-length headers is more challenging, ultimately resulting in a substantial decrease in packet forwarding performance. Significant breakthroughs have been made in network hardware capabilities over the past few years, and the development of technologies — notably programmable chip technology — has facilitated optimal forwarding performance even for packets with variable-length headers. This has overcome the technical challenges involved in extending network functions using the IPv6 extension header mechanism.

The evolution from MPLS to IPv6 extension headers highlights the difficulties encountered in extending IP technologies. While MPLS's hardware-friendly nature facilitated the development of new network services through its extensions, it faced bottlenecks when further advancements became necessary. IPv6 extensions have become increasingly competitive thanks to the advances made in network software and hardware technologies. To build high-quality networks that continuously deliver exceptional user experiences, it is essential that we gain in-depth insights into industry changes, adjust our development strategies accordingly, and adapt to emerging trends.

3.3.2 Why Are TLVs Defined for the SRH?

IPv6 on-path measurement is closely related to the SRv6 SRH standard[14], which defines segment lists used to indicate SRv6 paths and optional TLVs. During the SRv6 SRH standardization process, some technical experts believed that SRH TLVs were useless and insisted on removing them. However, to perform on-path measurement on specified nodes and links, the corresponding instructions must be encapsulated using SRH TLVs. On-path measurement is therefore an important use case, requiring that SRH TLVs be retained. Traditional IPv6 experts also questioned this. They said that the IPv6 standard[2] has already defined DOH1 and DOH2, located respectively before and after the RH. DOH1 carries instructions that need to be processed by the nodes or links specified in the

RH, and DOH2 carries the instructions that need to be processed by the destination node. This means that DOH1 can carry node- and link-specific instructions, making it unnecessary to introduce SRH TLVs for this purpose. We countered this by highlighting that, in order to support the IOAM Trace Option, the length of DOH1 (placed before the SRH) increases as IOAM information of each involved node and link is recorded. The result is that to guide packet forwarding, network devices have to obtain the segments indicating nodes or links from the segment list in the SRH, which continuously moves rightwards in the IPv6 header. This will severely degrade forwarding performance or make it impossible to obtain segments from the SRH to guide packet forwarding. Such issues can be avoided by using SRH TLVs to carry IOAM instructions and recorded IOAM information. In this way, the position of segment lists in the SRH remains essentially unchanged within the IPv6 header, allowing critical information used for path indication to be preferentially processed, while IOAM information of specified nodes or links can be recorded in SRH TLVs. If a network device lacks sufficient forwarding capabilities to support IOAM information recording, or a problem such as a Maximum Transmission Unit (MTU)-crossing event occurs, IOAM information recording can be stopped. This was a compelling use case — after efforts from all parties, SRH TLVs were retained and eventually published as an RFC.

REFERENCES

1. Fioccola G, Cociglio M, Mirsky G et al. Alternate-marking method [EB/OL]. (2022-12) [2024-09-30]. RFC 9341. ⏎
2. Deering S, Hinden R. Internet Protocol, Version 6 (IPv6) specification [EB/OL]. (2017-07) [2024-09-30]. RFC 8200. ⏎
3. Fioccola G, Zhou T, Cociglio M et al. IPv6 application of the alternate-marking method [EB/OL]. (2022-12) [2024-09-30]. RFC 9343. ⏎
4. Zhou T, Fioccola G, Liu Y et al. Enhanced alternate marking method [EB/OL]. (2024-05-27) [2024-09-30]. Draft-zhou-ippm-enhanced-alternate-marking-15. ⏎
5. Filsfils C, Camarillo P, Leddy J et al. Segment Routing over IPv6 (SRv6) network programming [EB/OL]. (2021-02) [2024-09-30]. RFC 8986. ⏎
6. Filsfils C, Dukes D, Previdi S et al. IPv6 segment routing header (SRH) [EB/OL]. (2020-03) [2024-09-30]. RFC 8754. ⏎
7. Fioccola G, Zhou T, Cociglio M et al. Application of the alternate marking method to the segment routing header [EB/OL]. (2024-08-09) [2024-09-30]. Draft-fz-spring-srv6-alt-mark-09. ⏎

8. Brockners F, Bhandari S, Bernier D et al. In situ operations, administration, and maintenance (IOAM) deployment [EB/OL]. (2023-04) [2024-09-30]. RFC 9378. ⏎
9. Carpenter B, Liu B. Limited domains and internet protocols [EB/OL]. (2020-07) [2024-09-30]. RFC 8799. ⏎
10. Bhandari S, Brockners F. IPv6 options for in situ operations, administration, and maintenance (IOAM) [EB/OL]. (2023-09) [2024-09-30]. RFC 9486. ⏎
11. Brockners F, Bhandari S, Mizrahi T. Data fields for in situ operations, administration, and maintenance (IOAM) [EB/OL]. (2022-05) [2024-09-30]. RFC 9197. ⏎
12. Song H, Gafni B, Brockners F et al. In situ operations, administration, and maintenance (IOAM) direct exporting [EB/OL]. (2022-11-15) [2024-09-30]. RFC 9326. ⏎
13. Song H, Zhou T, Andersson L et al. MPLS network actions using post-stack extension headers [EB/OL]. (2023-10-11) [2024-09-30]. Draft-song-mpls-extension-header-13. ⏎
14. Filsfils C, Dukes D, Previdi S et al. IPv6 segment routing header (SRH) [EB/OL]. (2020-03) [2024-09-30]. RFC 8754. ⏎

CHAPTER 4

Control Plane of IPv6 On-Path Telemetry

DOI: 10.1201/9781003677901-6

DURING IPV6 ON-PATH TELEMETRY deployment, a control plane protocol is used to advertise and negotiate the measurement capabilities (such as supported measurement modes) of nodes and different links. This allows participating nodes to communicate with each other based on agreed-upon information, such as the measurement mode. In addition, on-path telemetry can be automatically instantiated through control plane protocol extensions, making deployment far easier. This chapter describes the control plane protocol extensions involved in IPv6 on-path telemetry, namely, IGP, BGP, BGP-LS, BGP SR Policy, and PCEP extensions.

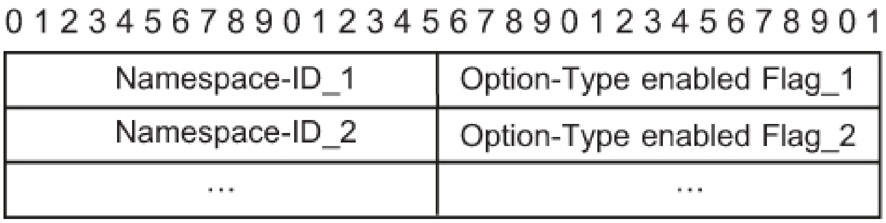
The IFIT framework is defined in the IETF for the implementation of on-path telemetry. For simplicity in the discussions of protocol extensions, IFIT is used in the following sections to refer to on-path telemetry.

4.1 IGP EXTENSIONS FOR IFIT

IGPs are routing protocols used within an AS. Through an IGP extension, nodes can transmit and synchronize on-path telemetry capability information, enabling them to negotiate information (e.g., IFIT capabilities and measurement mode) for IFIT instance deployment. Common IPv6 IGPs include IS-IS[1] and Open Shortest Path First version 3 (OSPFv3)[2], and *draft-wang-lsr-igp-extensions-ifit* defines IGP extensions for IFIT capability advertisement[3]. The following describes the IFIT Capability TLV and then how IGPs are used to advertise IFIT capabilities.

4.1.1 IGP-Based IFIT Capability TLV

The IGP-based IFIT Capability TLV consists of one or more 2-tuples. Figure 4.1 shows the corresponding format.



0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Namespace-ID_1	Option-Type enabled Flag_1
Namespace-ID_2	Option-Type enabled Flag_2
…	…

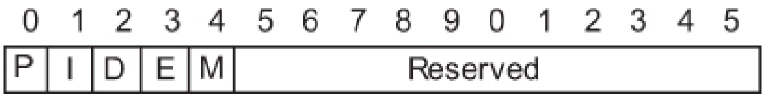
FIGURE 4.1 Format of the IGP-based IFIT Capability TLV. ⏎

Table 4.1 describes the fields in the IGP-based IFIT Capability TLV.

TABLE 4.1 Fields in the IGP-based IFIT Capability TLV ⏎

Field	Length	Description
Namespace-ID	2 bytes	In-situ operations, administration, and maintenance (IOAM) namespace-ID. IOAM processing is performed only when the locally configured namespace-ID is the same as the one in the packet. 0 × 0000 is the default namespace-ID, which must be supported by all IOAM nodes.
Option-Type enabled Flag	2 bytes	For details, see the following format.

Figure 4.2 shows the format of the Option-Type enabled Flag in the IGP-based IFIT capability.



0	1	2	3	4	5 6 7 8 9 0 1 2 3 4 5
P	I	D	E	M	Reserved

FIGURE 4.2 Format of the Option-Type Enabled Flag in the IGP-based IFIT capability. ⏎

Table 4.2 describes the fields in the IGP-based IFIT capability Option-Type enabled Flag.

TABLE 4.2 Fields in the IGP-based IFIT Capability Option-Type Enabled Flag ⏎

Field	Length	Description
P	1 bit	IOAM Pre-allocated Trace Option flag. The value 1 indicates that the network node supports IOAM Pre-Allocated Trace.
I	1 bit	IOAM Incremental Trace Option Type flag. The value 1 indicates that the network node supports IOAM Incremental Trace.
D	1 bit	IOAM DEX Option-Type flag. The value 1 indicates that the network node supports IOAM DEX.
E	1 bit	IOAM E2E Option-Type flag. The value 1 indicates that the network node supports IOAM E2E processing.
M	1 bit	Alternate marking flag. The value 1 indicates that the network node can process packets that are marked using the alternate marking method defined in RFC 9341.
Reserved	11 bits	Reserved flags: These flags must be set to 0 upon transmission and ignored upon receipt.

4.1.2 IFIT Capability Advertisement Using IS-IS

RFC 7981 defines the IS-IS Router Capability TLV, which routers use to advertise their capabilities[4]. This TLV can be used to advertise the IFIT capability of routers. It can also be used to encapsulate and advertise the IS-IS Node IFIT Capability sub-TLV, which is defined in *draft-wang-lsr-igp-extensions-ifit*. Figure 4.3 shows the format of this sub-TLV.

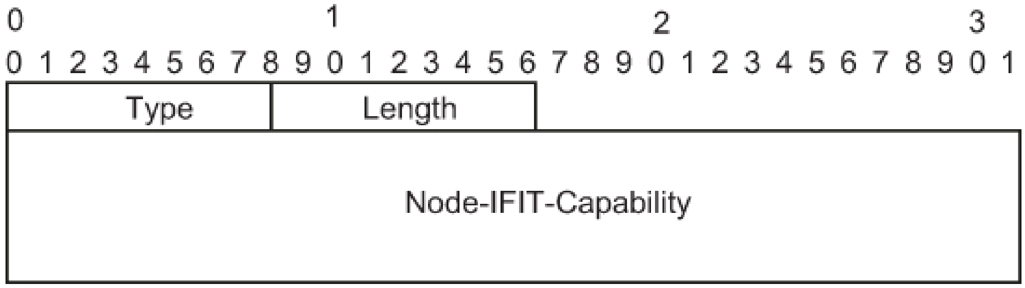


FIGURE 4.3 Format of the IS-IS Node IFIT Capability sub-TLV. ⏎

Table 4.3 describes the fields in the IS-IS Node IFIT Capability sub-TLV.

TABLE 4.3 Fields in the IS-IS Node IFIT Capability Sub-TLV ⏎

Field	Length	Description
Type	1 byte	Type of the IS-IS node IFIT capability sub-TLV.
Length	1 byte	Length of the IS-IS node IFIT capability sub-TLV.
Node-IFIT-Capability	Variable	IS-IS Node IFIT capability. The length of this field is a multiple of 4 bytes. For details about the format, see the format of the IGP-based IFIT Capability TLV described in Section 4.1.1.

If the links of a node have different IFIT capabilities, the IFIT capabilities of each link need to be advertised. The IS-IS Link IFIT Capability sub-TLV defined in *draft-wang-lsr-igp-extensions-ifit* is used for this purpose[3]. Figure 4.4 shows the format of this sub-TLV.

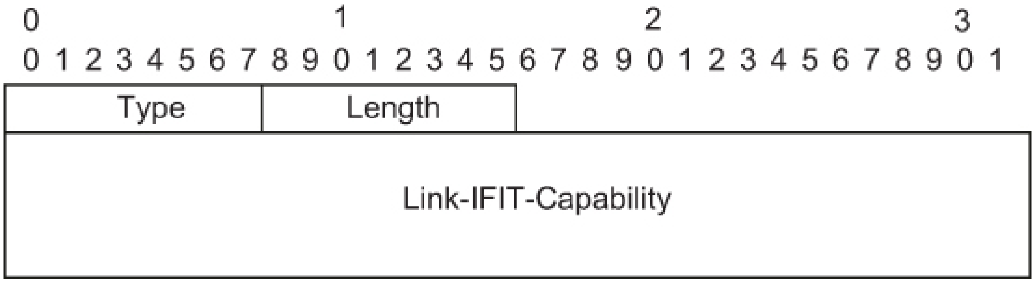


FIGURE 4.4 Format of the IS-IS Link IFIT Capability sub-TLV. ⏎

Table 4.4 describes the fields in the IS-IS Link IFIT Capability sub-TLV.

TABLE 4.4 Fields in the IS-IS Link IFIT Capability Sub-TLV ⏎

Field	Length	Description
Type	1 byte	Type of the IS-IS Link IFIT Capability sub-TLV.
Length	1 byte	Length of the IS-IS Link IFIT Capability sub-TLV.
Link-IFIT-Capability	Variable	IS-IS Link IFIT capability. The length of this field is a multiple of 4 bytes. For details about the format, see the format of the IGP-based IFIT Capability TLV described in Section 4.1.1.

The Link IFIT Capability sub-TLV can be transmitted as a sub-TLV of IS-IS TLVs 22, 23, 25, 141, 222, and 223. Table 4.5 describes these TLVs.

TABLE 4.5 IS-IS TLVs Related to the Link IFIT Capability Sub-TLV ⏎

TLV Type Value	Description	Standard
22	Extended Intermediate Systems (IS) Reachability TLV	RFC 5305[5]
23	IS Neighbor Attribute TLV	RFC 5311[6]
25	L2 Bundle Member Attributes TLV	RFC 8668[7]
141	Inter-AS Reachability Information TLV	RFC 9346[8]
222	Multi-Topology-IS Neighbor (MT-ISN) TLV. Unlike TLV 22, this TLV contains a topology ID.	RFC 5120[9]
223	MT IS Neighbor Attribute TLV. Unlike TLV 23, this TLV contains a topology ID.	RFC 5311[6]

4.1.3 IFIT Capability Advertisement Using OSPFv3

OSPFv3 uses Router Information Link State Advertisements (RI LSAs) to advertise optional router capability information[10]. An OSPFv3 RI LSA can be used to advertise the IFIT capability of a node. Figure 4.5 shows the format of the OSPFv3 Node IFIT Capability TLV.

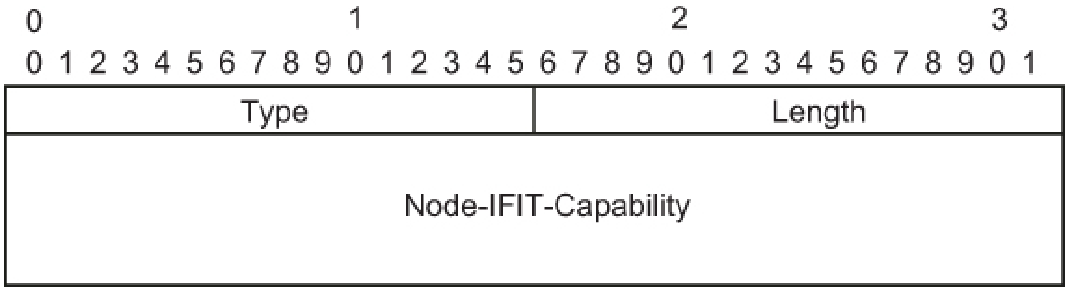


FIGURE 4.5 Format of the OSPFv3 Node IFIT Capability TLV. ⏎

Table 4.6 describes the fields in the OSPFv3 Node IFIT Capability TLV.

TABLE 4.6 Fields in the OSPFv3 Node IFIT Capability TLV ⏎

Field	Length	Description

Field	Length	Description
Type	2 bytes	Type of the OSPFv3 Node IFIT Capability TLV.
Length	2 bytes	Length of the OSPFv3 Node IFIT Capability TLV.
Node-IFIT-Capability	Variable	OSPFv3 Node IFIT capability. The length of this field is a multiple of 4 bytes. For details about the format, see the format of the IGP-based IFIT Capability TLV described in Section 4.1.1.

If the links of a node have different IFIT capabilities, the IFIT capabilities of each link need to be advertised. The Link IFIT sub-TLV is used for this purpose. It is a sub-TLV of the extensible Router-Link TLV, which is used by the E-Router-LSA (an OSPFv3 extended LSA) to transmit information about router links in an OSPFv3 domain[11]. Figure 4.6 shows the format of the OSPFv3 Link IFIT Capability sub-TLV.

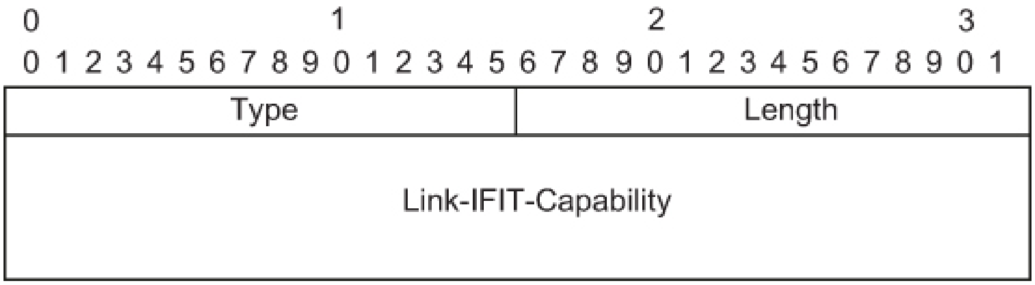


FIGURE 4.6 Format of the OSPFv3 Link IFIT Capability sub-TLV. ⏎

Table 4.7 describes the fields in the OSPFv3 Link IFIT Capability sub-TLV.

TABLE 4.7 Fields in the OSPFv3 Link IFIT Capability Sub-TLV ⏎

Field	Length	Description
Type	2 bytes	Type of the OSPFv3 Link IFIT Capability sub-TLV.
Length	2 bytes	Length of the OSPFv3 Link IFIT Capability sub-TLV.
Link-IFIT-Capability	Variable	OSPFv3 Link IFIT capability. The length of this field is a multiple of 4 bytes. For details about the format, see the format of the IGP-based IFIT Capability TLV described in Section 4.1.1.

4.2 BGP-LS EXTENSIONS FOR IFIT

BGP-LS defines a mechanism for reporting network link states and traffic engineering (TE) information through BGP extensions[12]. IFIT capability information of nodes and links can be reported to external entities (such as controllers) through BGP-LS.

The BGP-LS-based IFIT Capability TLV consists of one or more 2-tuples, and Figure 4.7 shows the corresponding format. The fields in the BGP-LS-based IFIT Capability TLV are the same as those in the IGP-based IFIT Capability TLV. For details, see the description of the fields in the IGP-based IFIT Capability TLV described in Section 4.1.1.

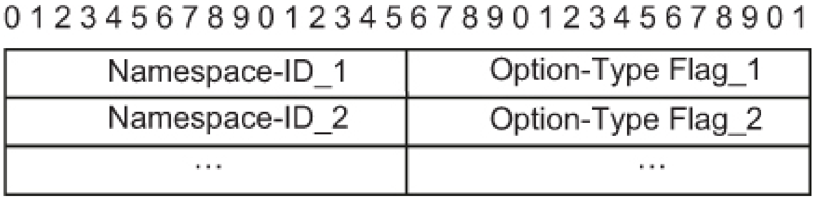


FIGURE 4.7 Format of the BGP-LS-based IFIT Capability TLV. ⏎

Node and link Capability TLVs of BGP-LS are extended in *draft- wang-idr-bgpls-extensions-ifit* to report IFIT capabilities[13]. Figure 4.8 shows the format of the BGP-LS Node IFIT Capability TLV.

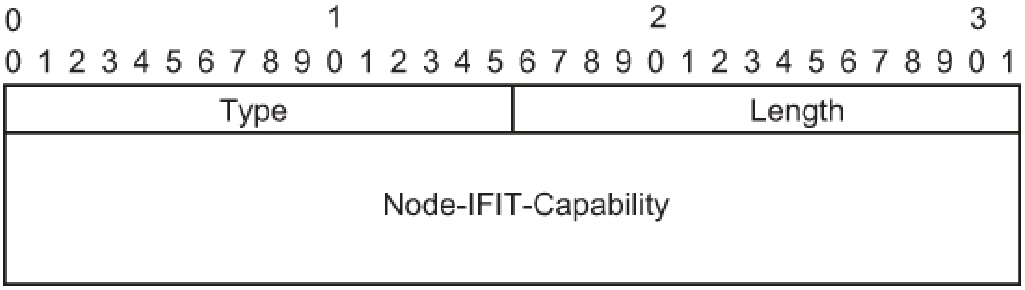


FIGURE 4.8 Format of the BGP-LS Node IFIT Capability TLV. ⏎

Table 4.8 describes the fields in the BGP-LS Node IFIT Capability TLV.

TABLE 4.8 Fields in the BGP-LS Node IFIT Capability TLV ⏎

Field	**Length**	**Description**
Type	2 bytes	Type of the BGP-LS Node IFIT Capability TLV.
Length	2 bytes	Length of the BGP-LS Node IFIT Capability TLV.

Field	Length	Description
Node-IFIT-Capability	Variable	BGP-LS Node IFIT capability. The length of this field is a multiple of 4 bytes. For details about the format, see the format of the BGP-LS-based IFIT Capability TLV described earlier.

Figure 4.9 shows the format of the BGP-LS Link IFIT Capability TLV.

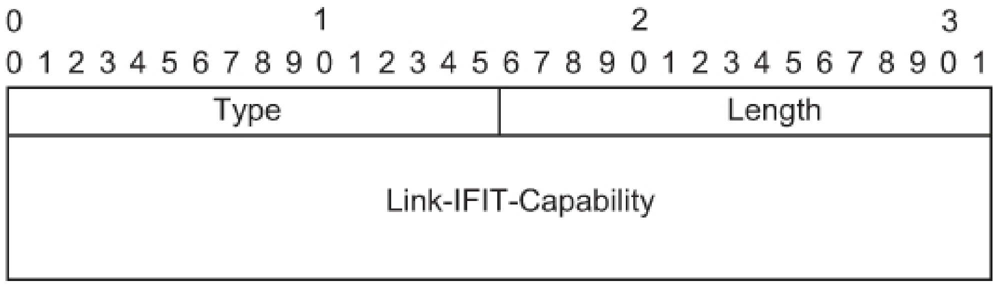


FIGURE 4.9 Format of the BGP-LS Link IFIT Capability TLV. ⏎

Table 4.9 describes the fields in the BGP-LS Link IFIT Capability TLV.

TABLE 4.9 Fields in the BGP-LS Link IFIT Capability TLV ⏎

Field	Length	Description
Type	2 bytes	Type of the BGP-LS Link IFIT Capability TLV.
Length	2 bytes	Length of the BGP-LS Link IFIT Capability TLV.
Link-IFIT-Capability	Variable	BGP-LS link IFIT capability. The length of this field is a multiple of 4 bytes. For details about the format, see the format of the BGP-LS-based IFIT Capability TLV described earlier.

4.3 BGP EXTENSIONS FOR IFIT

The way BGP extension-based IFIT capabilities are advertised is defined in *draft-ietf-idr-bgp-ifit-capabilities*[14]. When BGP is deployed on both the ingress and egress of IFIT and used by network nodes to advertise routes, it can also be used to advertise IFIT capabilities.

Figure 4.10 provides an example showing how IFIT and its capability advertisement are implemented in a BGP VPN scenario. CE1 and CE2 are customer edge nodes, PE1 and PE2 are provider edge nodes, and the Route

Reflector (RR)/P is a provider core node. BGP is deployed between PE1 and PE2, which use BGP extensions for IFIT to negotiate IFIT capabilities and mode. Assume that a service flow is transmitted from CE1 to CE2. Upon receiving a packet, PE1 locates an IFIT instance according to the packet characteristics. It then encapsulates the corresponding IFIT information for the packet based on the deployment policy and capability negotiation result before forwarding the packet toward the destination. After receiving the packet, PE2 decapsulates the IFIT information based on the negotiated measurement mode and then forwards the packet to CE2.

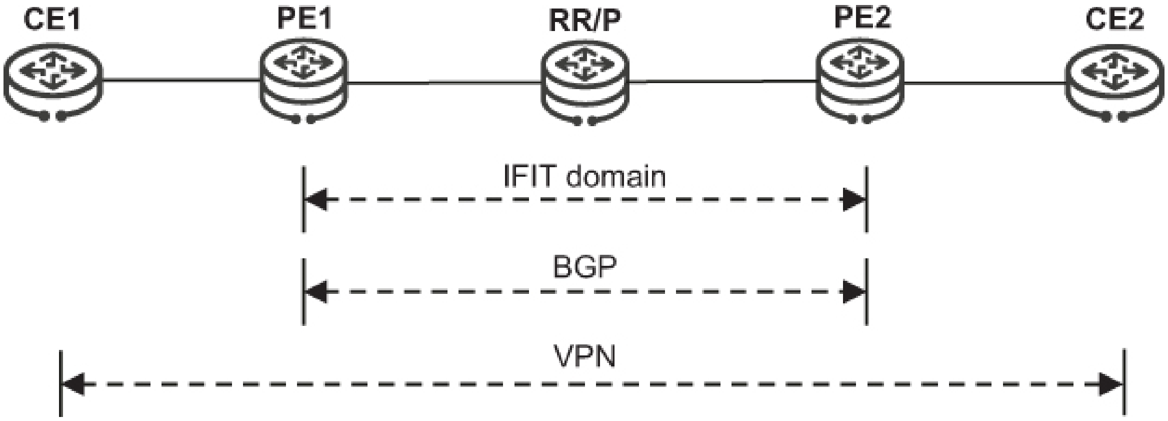
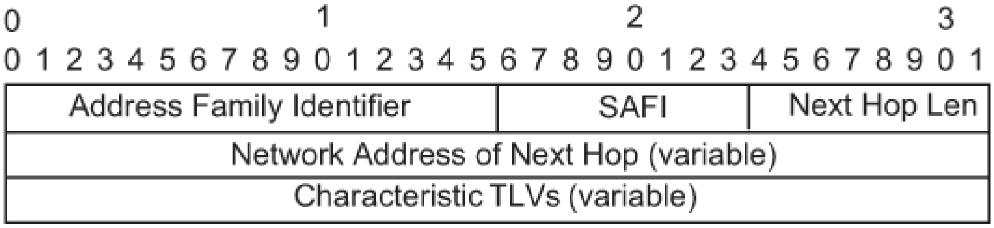


FIGURE 4.10 Example of implementing IFIT and capability advertisement in a BGP VPN scenario. ⏎

BGP IFIT capabilities are carried in the BGP Next Hop Dependent Characteristics (NHC) attribute, which is an optional transitive attribute defined in *draft-ietf-idr-entropy-label*[15]. Figure 4.11 shows its format.



```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+---------------+---------------+
|   Address Family Identifier   |     SAFI      | Next Hop Len  |
+-------------------------------+---------------+---------------+
|             Network Address of Next Hop (variable)            |
+---------------------------------------------------------------+
|                Characteristic TLVs (variable)                 |
+---------------------------------------------------------------+
```

FIGURE 4.11 Format of the BGP NHC attribute. ⏎

Table 4.10 describes the fields in the BGP NHC attribute.

TABLE 4.10 Fields in the BGP NHC Attribute ⏎

Field	Length	Description

Field	Length	Description
Address Family Identifier	2 bytes	The Address Family Identifier (AFI) field and the next field are used in combination to identify the network layer protocol of the next hop address, the way in which the next hop address is encoded, and the semantics of Network Layer Reachability Information (NLRI).
SAFI	1 byte	It is short for Subsequent Address Family Identifier.
Next Hop Len	1 byte	Length of the next hop address, in bytes.
Network Address of Next Hop	Variable	Next hop IP address.
Characteristic TLVs	Variable	Characteristic TLVs. For details, see the format of Characteristic TLVs in the BGP NHC attribute.

Characteristic TLVs in the BGP NHC attribute define specific characteristics. The BGP NHC attribute is associated with the next hop of a route. When the next hop changes, the Characteristic TLVs in the BGP NHC attribute are modified or deleted to reflect the new next hop capabilities. Figure 4.12 shows the format of the Characteristic TLVs in the BGP NHC attribute.

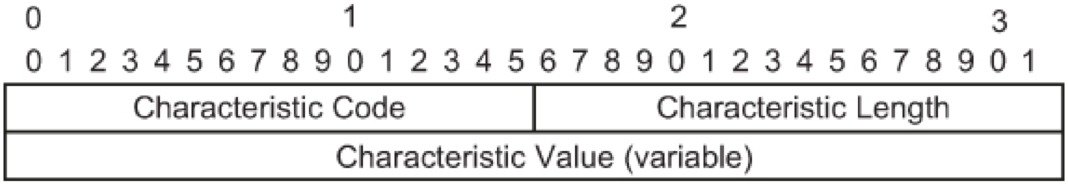


FIGURE 4.12 Format of the Characteristic TLVs in the BGP NHC attribute. ⏎

Table 4.11 describes the fields in Characteristic TLVs in the BGP NHC attribute.

TABLE 4.11 Fields in Characteristic TLVs in the BGP NHC attribute ⏎

Field	Length	Description

Field	Length	Description
Characteristic Code	2 bytes	To be allocated by the Internet Assigned Numbers Authority (IANA).
Characteristic Length	2 bytes	Length (in bytes) of the Characteristic Value field.
Characteristic Value	Variable	It varies according to Characteristic Code.

As one of the Characteristic TLVs in the BGP NHC attribute, the BGP IFIT Characteristic TLV is included in BGP update messages. This TLV indicates that the IFIT capabilities carried in such messages are supported by the BGP next hop that advertises the messages. The BGP next hop is identified by the IPv6 address of the IFIT decapsulation node. Figure 4.13 shows the format of the BGP IFIT Characteristic TLV.

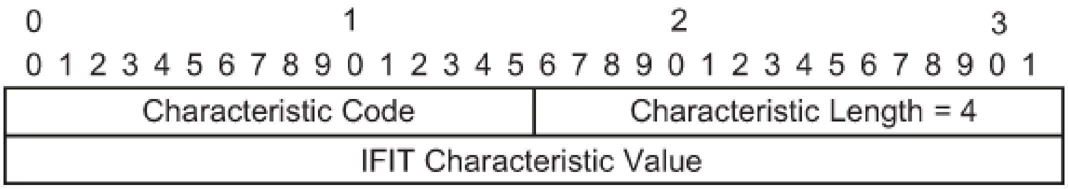


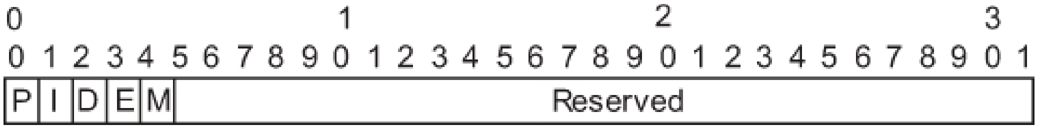
FIGURE 4.13 Format of the BGP IFIT Characteristic TLV. ⏎

Table 4.12 describes the fields in the BGP IFIT Characteristic TLV.

TABLE 4.12 Fields in the BGP IFIT Characteristic TLV ⏎

Field	Length	Description
Characteristic Code	2 bytes	IFIT characteristic code. The value is to be allocated by the IANA.
Characteristic Length	2 bytes	Length (in bytes) of the characteristic. The value is 4.
IFIT Characteristic Value	4 bytes	For details, see the format of BGP IFIT Characteristic. The length of this field must be 4 bytes (otherwise, this TLV is ignored).

Figure 4.14 shows the format of BGP IFIT Characteristic.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-----------------------------------------------------+
|P|I|D|E|M|                     Reserved                        |
+-+-+-+-+-+-----------------------------------------------------+
```

FIGURE 4.14 Format of BGP IFIT Characteristic. ⏎

Table 4.13 describes the fields in BGP IFIT Characteristic.

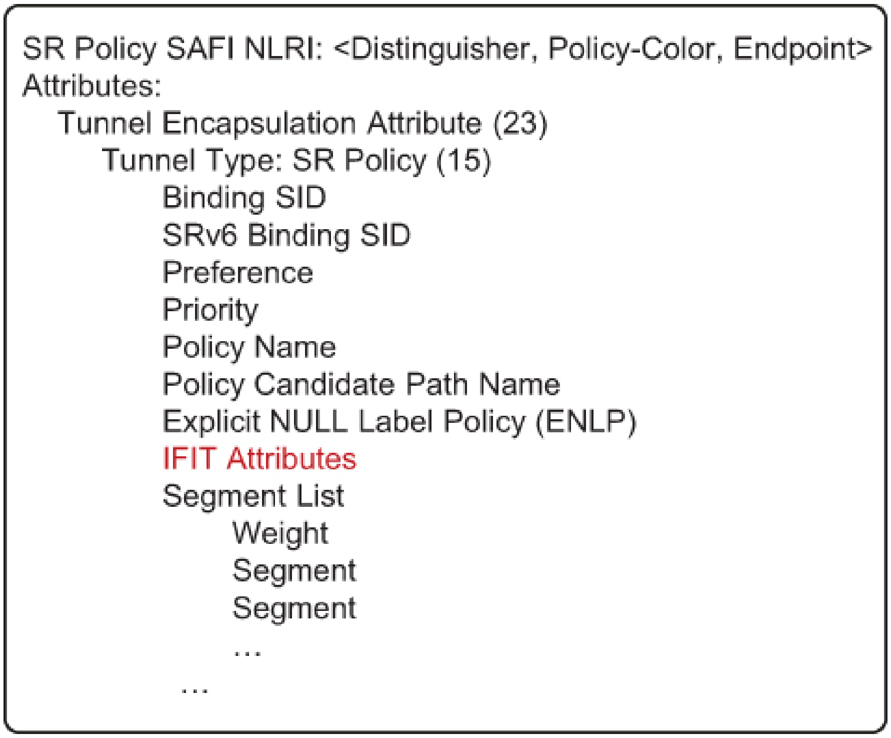
TABLE 4.13 Fields in BGP IFIT Characteristic ⏎

Field	Length	Description
P	1 bit	For details, see the P field in “Fields in the Option-Type enabled Flag in the IGP-based IFIT capability” in Section 4.1.1.
I	1 bit	For details, see the I field in “Fields in the Option-Type enabled Flag in the IGP-based IFIT capability” in Section 4.1.1.
D	1 bit	For details, see the D field in “Fields in the Option-Type enabled Flag in the IGP-based IFIT capability” in Section 4.1.1.
E	1 bit	For details, see the E field in “Fields in the Option-Type enabled Flag in the IGP-based IFIT capability” in Section 4.1.1.
M	1 bit	For details, see the M field in “Fields in the Option-Type enabled Flag in the IGP-based IFIT capability” in Section 4.1.1.
Reserved	27 bits	Reserved flags: These flags must be set to 0 upon transmission and ignored upon receipt.

4.4 BGP SR POLICY EXTENSIONS FOR IFIT

An SR Policy[16] is a set of candidate SR paths consisting of one or more Segment Lists (SLs) and necessary path attributes. It allows instantiation of an ordered list of segments with specific traffic diversion intents. To ensure fast detection of SR Policy service performance, prompt adjustment of service deployment, and automatic closed-loop control of services, IFIT can be deployed automatically when an SR Policy is delivered. The attributes related to IFIT instance deployment are defined by *draft-ietf-idr-sr-policy-ifit*[17].

IFIT attributes are control information that contains IFIT methods and deployment parameters. These attributes are encapsulated into the tunnel encapsulation attribute in the SR Policy SAFI defined in *draft-ietf-idr-segment-routing-te-policy*[18]. Figure 4.15 shows the structure of the BGP SR Policy information that carries IFIT attributes.



```
SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:
   Tunnel Encapsulation Attribute (23)
      Tunnel Type: SR Policy (15)
           Binding SID
           SRv6 Binding SID
           Preference
           Priority
           Policy Name
           Policy Candidate Path Name
           Explicit NULL Label Policy (ENLP)
           IFIT Attributes
           Segment List
                 Weight
                 Segment
                 Segment
                 ...
           ...
```

FIGURE 4.15 Structure of the BGP SR Policy information that carries IFIT attributes. ⏎

Figure 4.16 shows the format of the BGP SR Policy IFIT Attribute sub-TLV.

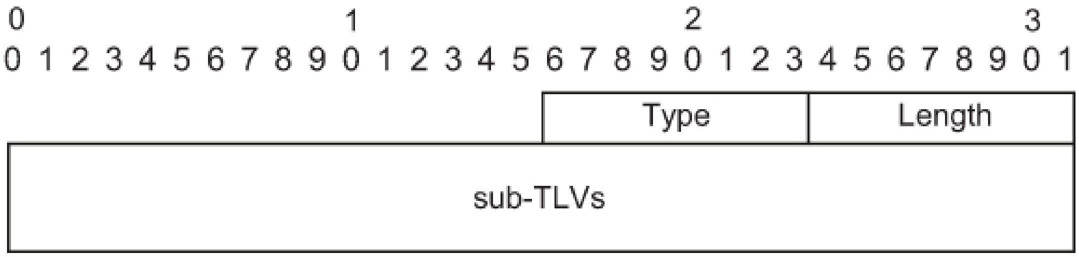


FIGURE 4.16 Format of the BGP SR Policy IFIT Attribute sub-TLV. ⏎

Table 4.14 describes the fields in the BGP SR Policy IFIT Attribute sub-TLV.

TABLE 4.14 Fields in the BGP SR Policy IFIT Attribute Sub-TLV ⏎

Field	**Length**	**Description**

Field	Length	Description
Type	1 byte	Type of the BGP SR Policy IFIT attribute.
Length	1 byte	Length (in bytes) of the BGP SR Policy IFIT attribute, that is, the length of sub-TLVs. The value 0 indicates that no sub-TLV field exists and that IFIT is not enabled.
sub-TLVs	Variable	Sub-TLV of the BGP SR Policy IFIT attribute. Sub-TLVs may be separately defined by the IFIT method that is used, which can be IOAM or alternate marking. The sub-TLVs are as follows: • IOAM Pre-allocated Trace Option Sub-TLV • IOAM incremental Trace Option Sub-TLV • IOAM Direct Export Option Sub-TLV • IOAM Edge-To-Edge Option Sub-TLV • Enhanced Alternate Marking Sub-TLV

One or more BGP SR Policy IFIT Attribute sub-TLV fields can be encapsulated at a time. If there are two conflicting sub-TLVs (e.g., IOAM Pre-allocated Trace Option Sub-TLV and IOAM Incremental Trace Option Sub-TLV) or if there is more than one sub-TLV of the same type, none of the corresponding methods can be used. IFIT can be disabled by setting an empty IFIT attribute sub-TLV. Any IFIT attribute sub-TLVs that cannot be recognized can be ignored, thereby ensuring compatibility.

The following describes the sub-TLVs of the BGP SR Policy IFIT attribute.

4.4.1 IOAM Pre-Allocated Trace Option Sub-TLV

This sub-TLV indicates that the IOAM pre-allocated trace function is enabled for the corresponding SR Policy. IOAM Pre-allocated Trace Option Sub-TLV is used to instruct the IOAM encapsulation node to pre-allocate the data space required for E2E recording. It also instructs the IOAM nodes through which a data packet passes to collect and process measurement data, ensuring that the path through which the packet passes in the IOAM domain can be traced and visualized. Figure 4.17 shows the format of this sub-TLV.

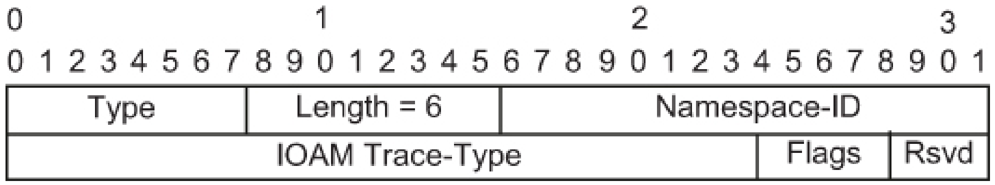


FIGURE 4.17 Format of the IOAM pre-allocated trace option sub-TLV. ⏎

Table 4.15 describes the fields in the IOAM Pre-allocated Trace Option Sub-TLV.

TABLE 4.15 Fields in the IOAM Pre-Allocated Trace Option Sub-TLV ⏎

Field	Length	Description
Type	8 bits	Type of the IOAM Pre-Allocated Trace Option Sub-TLV. The value is to be allocated.
Length	8 bits	Length (in bytes) of the IOAM Pre-Allocated Trace Option Sub-TLV. The value is 6.
Namespace-ID	16 bits	For details, see the Namespace-ID field description in "IOAM Per-hop Tracing Option" in Section 3.2.2.
IOAM Trace-type	24 bits	For details, see the IOAM Trace-Type field description in "IOAM Per-hop Tracing Option" in Section 3.2.2.
Flags	4 bits	For details, see the Flags field description in "IOAM Per-hop Tracing Option" in Section 3.2.2.
Rsvd	4 bits	Reserved field whose value is fixed at 0.

4.4.2 IOAM Incremental Trace Option Sub-TLV

This sub-TLV is similar to IOAM Pre-allocated Trace Option Sub-TLV. However, it differs in that it instructs each IOAM node to incrementally allocate data space for IOAM tracing information to be written.

4.4.3 IOAM Direct Export Option Sub-TLV

If this sub-TLV is carried in the BGP SR Policy IFIT attribute, it indicates that direct export of the IOAM data of the corresponding SR Policy is enabled. This sub-TLV allows the IOAM data of the corresponding SR Policy to be directly exported to the collector (without being encapsulated into data packets that are being transmitted). Figure 4.18 shows the format of this sub-TLV.

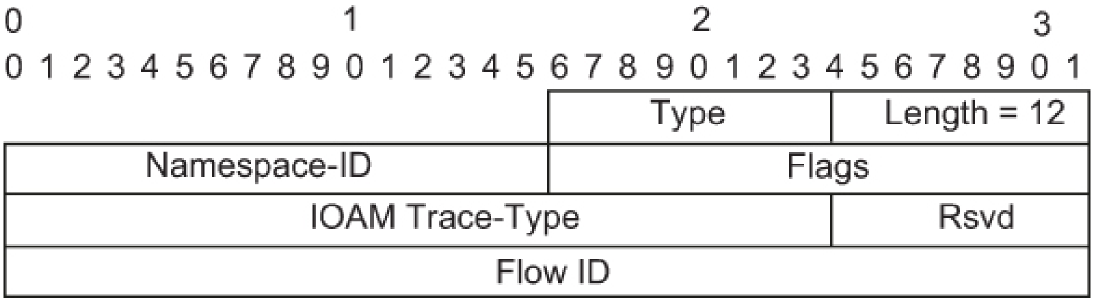


FIGURE 4.18 Format of the IOAM direct export option sub-TLV. ⏎

Table 4.16 describes the fields in the IOAM Direct Export Option Sub-TLV.

TABLE 4.16 Fields in the IOAM Direct Export Option Sub-TLV ⏎

Field	Length	Description
Type	1 byte	Type of the IOAM Direct Export Option Sub-TLV. The value is to be allocated.
Length	1 byte	Length (in bytes) of the IOAM Direct Export Option Sub-TLV. The value is 12.
Namespace-ID	2 bytes	For details, see the namespace-ID field description in “IOAM Direct Export Option” in Section 3.2.2.
Flags	2 bytes	For details, see the Flags field description in “IOAM Direct Export Option” in Section 3.2.2.
IOAM Trace-Type	3 bytes	For details, see the IOAM Trace-Type field description in “IOAM Direct Export Option” in Section 3.2.2.
Rsvd	1 byte	Reserved field whose value is fixed at 0.
Flow ID	4 bytes	For details, see the Flow ID field description in “IOAM Direct Export Option” in Section 3.2.2.

4.4.4 IOAM Edge-to-Edge Option Sub-TLV

This sub-TLV indicates that IOAM E2E measurement is enabled for the corresponding SR Policy. In E2E measurement, an IOAM encapsulating node (ingress node in an IOAM domain) encapsulates IOAM option information into a data packet, and an IOAM decapsulating node (egress node in the IOAM domain) parses and processes the information. Transit nodes do not participate in IOAM. Figure 4.19 shows the format of this sub-TLV.

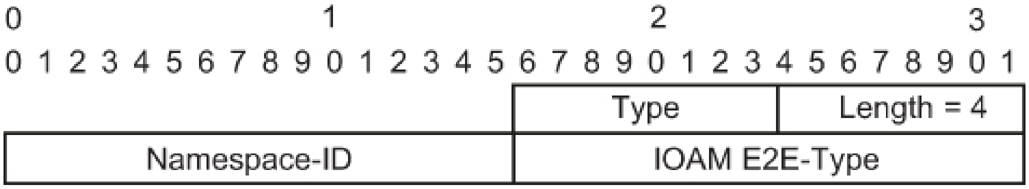


FIGURE 4.19 Format of the IOAM edge-to-edge option sub-TLV. ⏎

Table 4.17 describes the fields in the IOAM Edge-to-Edge Option Sub-TLV.

TABLE 4.17 Fields in the IOAM Edge-to-Edge Option Sub-TLV ⏎

Field	**Length**	**Description**
Type	1 byte	Type of the IOAM Edge-to-Edge Option Sub-TLV. The value is to be allocated.
Length	1 byte	Length (in bytes) of the IOAM Edge-to-Edge Option Sub-TLV. The value is 4.
Namespace-ID	2 bytes	For details, see the Namespace-ID field description in “IOAM Edge-to-Edge Option” in Section 3.2.2.
IOAM E2E-type	2 bytes	For details, see the IOAM E2E-Type field description in “IOAM Edge-to-Edge Option” in Section 3.2.2.

4.4.5 Enhanced Alternate Marking Sub-TLV

This sub-TLV indicates that enhanced alternate marking measurement is enabled for the corresponding SR Policy. Figure 4.20 shows the format of this sub-TLV.

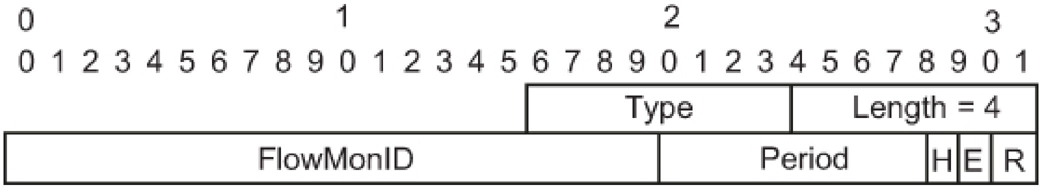


FIGURE 4.20 Format of the Enhanced Alternate Marking Sub-TLV. ⏎

Table 4.18 describes the fields in the Enhanced Alternate Marking Sub-TLV.

TABLE 4.18 Fields in the Enhanced Alternate Marking Sub-TLV ⏎

Field	Length	Description
Type	8 bits	Type of the Enhanced Alternate Marking Sub-TLV. The value is to be allocated.
Length	8 bits	Length (in bytes) of the Enhanced Alternate Marking Sub-TLV. The value is 4.
FlowMonID	20 bits	For details, see the FlowMonID field description in Section 3.1.3.
Period	8 bits	Measurement period, in seconds.
H	1 bit	Hop-by-hop measurement flag. The value 1 indicates hop-by-hop measurement.
E	1 bit	E2E measurement flag. The value 1 indicates E2E measurement.
R	2 bits	Reserved field whose value is fixed at 0.

4.5 PCEP EXTENSIONS FOR IFIT

PCEP[19] is a communication protocol used between a Path Computation Client (PCC) and a Path Computation Element (PCE). It defines a group of messages and objects used for computing and managing E2E optimal paths.

A PCE can work in stateful or stateless mode[20]. Regardless of the mode, a PCE can provide path computation services. However, in stateless mode, a PCE does not save or maintain path information, whereas in stateful mode, a PCE maintains path information and provides path creation, deletion, and optimization functions for PCCs.

PCEP extensions for IFIT attributes are defined in *draft-ietf- pce-pcep-ifit*[21]. Such extensions allow a PCC to indicate its supported IFIT capabilities. And in stateful mode, a PCE can use the extensions to deploy IFIT for a specific path on a PCC.

4.5.1 PCEP Extensions for IFIT Capabilities

During the PCEP initialization phase, a PCEP speaker (PCE or PCC) should advertise its supported IFIT capabilities by encapsulating the IFIT capability information in the optional TLV of the OPEN object in Open messages[22]. Figure 4.21 shows the format of the IFIT Capability TLV of PCEP.

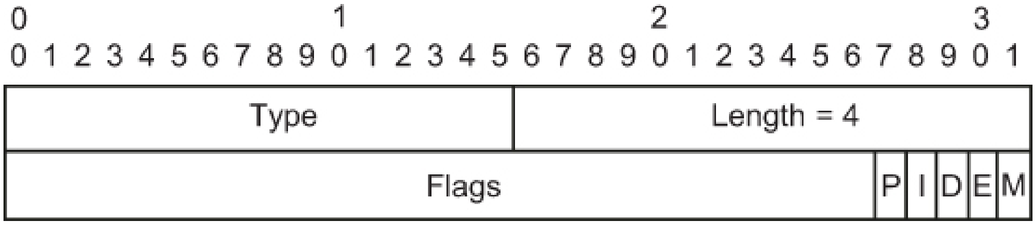


FIGURE 4.21 Format of the IFIT Capability TLV of PCEP. ⏎

Table 4.19 describes the fields in the IFIT Capability TLV of PCEP.

Table 4.19 Fields in the IFIT Capability TLV of PCEP ⏎

Field	Length	Description
Type	16 bits	Type of the IFIT Capability TLV of PCEP. The value is to be allocated.
Length	16 bits	Length (in bytes) of the IFIT Capability TLV. The value is 4.
Flags	27 bits	Reserved flags. These flags must be set to 0 upon transmission and ignored upon receipt.
P	1 bit	IOAM Pre-allocated Trace Option Type-enabled flag. A PCC or PCE indicates its support for this option by setting the value of this flag to 1. The P flag must be set by both the PCC and PCE in order to complete the IOAM Pre-allocated Trace function instantiation (allowing the PCE to deploy this function on the PCC).
I	1 bit	IOAM Incremental Trace Option Type-enabled flag. A PCC or PCE indicates its support for this option by setting the value of this flag to 1. The I flag must be set by both the PCC and PCE in order to complete the IOAM incremental trace function instantiation (allowing the PCE to deploy this function on the PCC).
D	1 bit	IOAM DEX Option Type-enabled flag. A PCC or PCE indicates its support for this option by setting the value of this flag to 1. The D flag must be set by both the PCC and PCE in order to complete the IOAM DEX function instantiation (allowing the PCE to deploy this function on the PCC).

Field	Length	Description
E	1 bit	IOAM E2E Option Type-enabled flag. A PCC or PCE indicates its support for this option by setting the value of this flag to 1. The E flag must be set by both the PCC and PCE in order to complete the IOAM E2E function instantiation (allowing the PCE to deploy this function on the PCC).
M	1 bit	Alternate Marking-enabled flag. A PCC or PCE indicates its support for this option by setting the value of this flag to 1. The M flag must be set by both the PCC and PCE in order to complete the alternate marking function instantiation (allowing the PCE to deploy this function on the PCC).

PCEP-based IFIT capability advertisement complies with the following conventions:

- PCEP extensions cannot be used to obtain IFIT capability information of the PCC if one or both PCEP speakers do not encapsulate the IFIT Capability TLV in the OPEN object in Open messages (messages used to establish a PCEP session).
- A PCEP speaker that does not recognize the IFIT Capability TLV ignores it.

4.5.2 PCEP Extensions for IFIT Attributes

In stateful mode, a PCE deploys IFIT for a specified path on the PCC using PCEP extensions for IFIT attributes. These attributes are carried as TLVs of the label switched path attributes (LSPA) object, enable the specified IFIT feature, and instruct the PCC to instantiate a path (an LSP). They can be carried in the following three types of PCEP messages:

- Path Computation Initialization (PCInitiate) message: sent by a PCE to a PCC to trigger LSP instantiation or deletion. For a PCE-initiated LSP with IFIT enabled, the IFIT-Attributes TLV must be included in the LSPA object within the PCInitiate message.
- Path Computation LSP Update Request (PCUpd) message: sent by a PCE to a PCC to update LSP parameters. For a PCE-initiated LSP with IFIT enabled, the IFIT-Attributes TLV must be included in the LSPA object within the PCUpd message, instructing the PCC to update IFIT parameters.

- Path Computation LSP State Report (PCRpt) message: sent by a PCC to a PCE to report the state of one or more paths.

Figure 4.22 shows the format of the IFIT-Attributes TLV of PCEP.

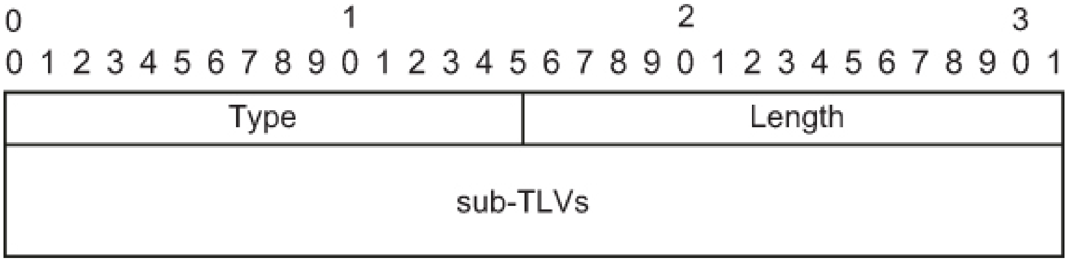


FIGURE 4.22 Format of the IFIT-Attributes TLV of PCEP. ⏎

Table 4.20 describes the fields in the IFIT-Attributes TLV of PCEP.

Table 4.20 Fields in the IFIT-Attributes TLV of PCEP ⏎

Field	**Length**	**Description**
Type	2 bytes	Type of the IFIT-Attributes TLV of PCEP. The value is to be allocated.
Length	2 bytes	Length (in bytes) of the IFIT-Attributes sub-TLVs of PCEP.
sub-TLVs	Variable	IFIT-attributes sub-TLVs in a PCEP message. Sub-TLVs can be separately defined by the IFIT method that is used, which can be IOAM or alternate marking. The sub-TLVs are as follows: • IOAM Pre-allocated Trace Option Sub-TLV • IOAM Incremental Trace Option Sub-TLV • IOAM Direct Export Option Sub-TLV • IOAM Edge-To-Edge Option Sub-TLV • Enhanced Alternate Marking Sub-TLV

IFIT-Attributes Sub-TLVs in a PCEP message are similar to those of the BGP SR Policy IFIT attribute. For details, see Section 4.4.

4.6 STORIES BEHIND IPV6 ON-PATH TELEMETRY

4.6.1 Centralized or Distributed Protocol Selection

Compatibility issues arose when the IFIT alternate marking solution was deployed. If it is not supported by nodes or links on a service flow path, IPv6 on-path telemetry may fail. This is especially problematic if the solution is not supported by the egress, as it may discard packets carrying IFIT alternate marking information, leading to network faults.

To address such issues, an automatic IPv6 on-path telemetry deployment solution is required. One simple and effective approach is to obtain the IPv6 on-path telemetry capability information of nodes and links along the SRv6 path through suitable protocols, enabling such information to be checked during the deployment of IPv6 on-path telemetry. During our discussions, two solutions were proposed.

Solution 1: Use protocols such as gRPC or NETCONF to report the IPv6 on-path telemetry capability information of nodes or links to the controller.

Solution 2: Use extensions of protocols, such as IGP and BGP extensions, to advertise IPv6 on-path telemetry capability information of nodes and links to other nodes and report the information to the controller.

Solution 1 was widely accepted due to it being easy to implement. However, I had reservations about it for the following reasons:

- Solution 1 requires a model to be defined for the data transmitted through gRPC or NETCONF. However, standardization of YANG models in the IETF is currently inadequate, hindering interconnection between network devices and third-party controllers. In contrast, BGP-LS extensions are better standardized for reporting network device capabilities and have been successfully used in the interconnection between network devices and third-party controllers.
- If solution 1 is used, a controller must be deployed to collect information in a unified manner. However, there may be cases where no controller is available, devices cannot connect to a third-party controller, or a controller cannot connect to devices due to some faults. In such cases, distributed protocols (such as IGP and BGP extensions) must be used to advertise IPv6 on-path telemetry capability information of nodes and links. This allows the capability information to be checked during IPv6 on-path telemetry deployment, even if no controller is available or the connection between a controller and devices is faulty.

Considering these reasons, we submitted drafts to the IETF for advertising IPv6 on-path telemetry capability information of nodes and links through IGP and BGP extensions and reporting such information through BGP-LS extensions. However, standardizing the use of IGP extensions for our purpose was opposed by traditional IGP experts. They believed that information extended for IGPs should be used for route calculation, and that the IPv6 on-path telemetry capability information of nodes and links does not serve this purpose. Consequently, they deemed IGP extensions unsuitable in this case. Through extensive discussions with these experts, I learned that they objected to the extensions of BGP beyond the routing function and did not want the same to happen to IGPs. As requirement-driven development and traditional border protection are inextricably intertwined in the IETF's protocol standardization process, the standardization of IPv6 on-path telemetry control plane protocols still has a long way to go.

REFERENCES

1. Hopps C. Routing IPv6 with IS-IS [EB/OL]. (2008-10) [2024-09-30]. RFC 5308. ⏎
2. Coltun R, Ferguson D, Moy J et al. OSPF for IPv6 [EB/OL]. (2008-07) [2024-09-30]. RFC 5340. ⏎
3. Wang Y, Zhou T, Qin F et al. IGP extensions for in-situ flow information telemetry (IFIT) capability advertisement [EB/OL]. (2021-01-29) [2024-09-30]. draft-wang-lsr-igp-extensions-ifit-01. ⏎
4. Ginsberg L, Previdi S, Chen M. IS-IS extensions for advertising router information [EB/OL]. (2016-10) [2024-09-30]. RFC 7981. ⏎
5. Li T, Smit H. IS-IS extensions for traffic engineering [EB/OL]. (2008-10) [2024-09-30]. RFC 5305. ⏎
6. Mcpherson D, Ginsberg L, Previdi S et al. Simplified extension of link state PDU (LSP) space for IS-IS[EB/OL]. (2009-02) [2024-09-30]. RFC 5311. ⏎
7. Ginsberg L, Bashandy A, Filsfils C et al. Advertising layer 2 bundle member link attributes in IS-IS[EB/OL]. (2019-12) [2024-09-30]. RFC 8668. ⏎
8. Chen M, Ginsberg L, Previdi S et al. IS-IS extensions in support of inter-autonomous system (AS) MPLS and GMPLS traffic engineering [EB/OL]. (2023-02) [2024-09-30]. RFC 9346. ⏎
9. Przygienda T, Shen N, Sheth N. M-ISIS: Multi topology (MT) routing in intermediate system to intermediate systems (IS-ISs) [EB/OL]. (2008-02) [2024-09-30]. RFC 5120. ⏎
10. Lindem A, Vasseur JP, Aggarwal R et al. Extensions to OSPF for advertising optional router capabilities [EB/OL]. (2016-02) [2024-09-30]. RFC 7770. ⏎
11. Lindem A, Roy A, Goethals D et al. OSPFv3 link state advertisement (LSA) extensibility [EB/OL]. (2018-04) [2024-09-30]. RFC 8362. ⏎

12. Gredler H, Medved J, Previdi S et al. North-bound distribution of link-state and traffic engineering (TE) information using BGP [EB/OL]. (2016-03) [2024-09-30]. RFC 7752. ⏎
13. Wang Y, Zhou T, Liu M et al. BGP-LS extensions for in-situ flow information telemetry (IFIT) capability advertisement [EB/OL]. (2021-01-14) [2024-09-30]. Draft-wang-idr-bgpls-extensions-ifit-00. ⏎
14. Fioccola G, Pang R, Wang S et al. Advertising in-situ flow information telemetry (IFIT) capabilities in BGP[EB/OL]. (2024-07-05) [2024-09-30]. Draft-ietf-idr-bgp-ifit-capabilities-05. ⏎
15. Decraene B, Scudder JG, Kompella K et al. BGP next hop dependentcharacteristics attribute [EB/OL]. (2024-09-26) [2024-09-30]. Draft-ietf-idr-entropy-label-16. ⏎
16. Filsfils C, Talaulikar K, Voyer D et al. Segment routing policy architecture [EB/OL]. (2022-07) [2024-09-30]. RFC 9256. ⏎
17. Qin F, Yuan H, Yang S et al. BGP SR policy extensions to enable IFIT [EB/OL]. (2024-04-19) [2024-09-30]. Draft-ietf-idr-sr-policy-ifit-08. ⏎
18. Previdi S, Filsfils C, Talaulikar K et al. Advertising segment routing policies in BGP [EB/OL]. (2023-10-23) [2024-09-30]. Draft-ietf-idr- segment-routing-te-policy-26. ⏎
19. Vasseur JP, Le Roux JL. Path computation element (PCE) communication protocol (PCEP) [EB/OL]. (2009-03) [2024-09-30]. RFC 5440. ⏎
20. Farrel A, Vasseur JP, Ash J. A path computation element (PCE)-based architecture [EB/OL]. (2006-08) [2024-09-30]. RFC 4655. ⏎
21. Yuan H, Wang X, Yang P et al. Path computation element communication protocol (PCEP) extensions to enable IFIT [EB/OL]. (2024-07-05) [2024-09-30]. Draft-ietf-pce-pcep-ifit-05. ⏎
22. IANA. Path computation element protocol (PCEP) numbers [EB/OL]. (2024-09) [2024-09-30]. ⏎

CHAPTER 5

IPv6 On-Path Telemetry Information Reporting

DOI: 10.1201/9781003677901-7

THIS CHAPTER **DESCRIBES THREE methods** used for reporting IPv6 on-path telemetry information[1,2]: gRPC, UDP telemetry, and IPFIX.

5.1 GRPC-BASED INFORMATION REPORTING

5.1.1 gRPC Protocol

gRPC[3] is an open-source high-performance RPC framework initially created by Google. It can be used as a data transmission protocol and work with telemetry to enable precise and efficient monitoring of network devices in real time. gRPC implements the communication mechanism related to RPC, improving development efficiency by enabling communication parties to focus on services. Figure 5.1 shows the layered structure of the gRPC protocol stack.

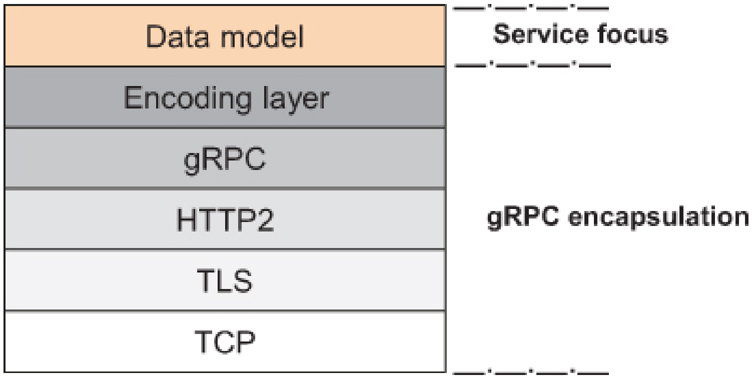


FIGURE 5.1 Layered structure of the gRPC protocol stack.

Table 5.1 describes each layer.

TABLE 5.1 Layers of the gRPC Protocol Stack

Layer	Description
TCP	Communication protocol layer. gRPC establishes connections based on TCP.
TLS	Optional communication encryption layer. gRPC uses transport layer security (TLS) to encrypt communication channels.
HTTP2	gRPC is carried over hypertext transfer protocol version 2 (HTTP/2) and uses HTTP/2 features such as bidirectional streaming, flow control, header compression, and multiplexing multiple requests over a single connection.
gRPC	RPC layer, which defines the protocol interaction format for RPCs.
Encoding layer	gRPC uses GPB or JSON for encoding data. • GPB[4]: a language-neutral, platform-neutral, and extensible format for serializing binary structured data in communications protocols and data storage. During a gRPC connection, .proto files are required to define gRPC subscription parameters and describe messages carried by gRPC. Such files are also used by GPB to describe the encoding dictionaries, which record data structures. And based on such files, a controller can use software such as the **protoc-3.0.2-windows-x86_64.exe** file to automatically generate code for interconnecting with devices. • JSON[5,6]: a lightweight data exchange format derived from the European Computer Manufacturers Association Script (ECMAScript). It stores and represents data in a text format independent of programming languages. JSON has a straightforward structure, making it easy for humans to read and write, and easy for machines to parse and generate. JSON is not as efficient or performant as GPB for reporting large volumes of telemetry data. Therefore, GPB is recommended for IPv6 on-path telemetry, and JSON is not described in this book.

Layer	Description
Data model	Data model of the service module. To correctly invoke information, communication parties need to define and exchange each other’s data model using a .proto file.

gRPC works in server/client mode and transmits packets over HTTP/2. Figure 5.2 shows its deployment model.

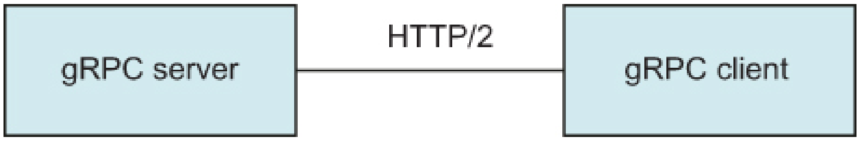


FIGURE 5.2 gRPC deployment model. ⏎

gRPC works as follows:

1. The server listens on the specified port, waiting for connection requests from the client.
2. A user logs in to the server from the client.
3. The client sends a request by invoking the gRPC method provided by a .proto file.
4. The server returns a response message.

gRPC can be deployed on a network device in dial-in or dial-out mode.

1. Dial-in mode: A network device functions as the gRPC server, and a controller functions as the gRPC client. The controller sends a request to the device to establish a gRPC connection and obtain required data or deliver configurations. This mode is suitable for small-scale networks and scenarios where the controller needs to deliver configurations to devices.

   In dial-in mode, the controller can perform the following operations:

   - Subscribe: Collects interface traffic statistics, CPU usage, and memory usage of a device at high speed.
   - Get: Obtains the running status and configuration of a device.
   - Capabilities: Obtains the capabilities of a device.
   - Set: Delivers configurations to a device.
2. Dial-out mode: A network device functions as a gRPC client, and a controller functions as a gRPC server. In this mode, a device proactively establishes a gRPC connection with the controller and then pushes subscribed data to it. This mode is suitable for large-scale networks.

The following describes the gRPC-based telemetry process, using Huawei products and .proto files with the prefix **huawei-** as an example.

1. Configure subscription: Subscribe to data sources on network devices for data collection. Two subscription modes are available: static and dynamic.
   - Static subscription: Subscription data sources are configured on network devices through commands for data collection, and the transmission protocol is set to gRPC. This subscription mode defines the RPC interface using the **huawei-grpc-dialout.proto** file.
   - Dynamic subscription: gRPC service functions are configured on network devices through commands, and then the collector is used to deliver dynamic configurations to the devices for data collection. This subscription mode defines the RPC interface using the **huawei-grpc-dialin.proto** file.
2. Push sampled data: Network devices report sampled data to the collector (based on the controller-delivered configuration), which then stores the received data.

   Specifically, the network devices encode collected information in the GPB format and push it to the controller. This is done according to the data encoding format (GPB), sampling path, sampling timestamp, and other information defined in the **huawei-telemetry.proto** file and the encoding data structure of the specific service (e.g., **huawei-ifit.proto**).
3. Read and process the data: The controller reads and processes the reported telemetry data.

   Specifically, the controller decodes and processes the data based on the encoding format defined in the **huawei-telemetry.proto** file. If the encoding format is GPB, the controller first determines which data structure is used based on the **sensor_path** field in the **huawei-telemetry.proto** file. The value of this field identifies a specific service .proto file — for example, **huawei-ifit:ifit/huawei-ifit-statistics:flow-statistics/flow-statistic** indicates that the data structure is defined in the **huawei-ifit.proto** file. The controller then parses the **data_gpb** field defined in the **huawei-telemetry.proto** file to obtain the device measurement information.

5.1.2 Telemetry Information Reporting Using gRPC

The GPB format is recommended for reporting IPv6 on-path telemetry data. In this format, three types of .proto files are required between the controller and network devices to facilitate decoding: RPC interface definition file (e.g., **huawei-grpc-dialout.proto**), telemetry header definition file (e.g., **huawei-**

telemetry.proto), and service data file (e.g., **huawei-ifit.proto**). These files are user-defined and must be agreed upon by both communication parties. The following uses the static subscription mode as an example to explain these files.

5.1.2.1 huawei-grpc-dialout.proto

This file defines an RPC interface. Example 5.1 shows its content, with explanations of key parts.

Example 5.1 huawei-grpc-dialout.proto ⏎

```
syntax = "proto3";   //The proto version is v3.
package xxx_dialout;   //The package name is xxx_dialout.
service gRPCDataservice {   //The service name is
gRPCDataservice.
    rpc dataPublish(stream serviceArgs) returns(stream
serviceArgs) {};   //The data push method is
dataPublish. It uses the bidirectional streaming mode,
with the input parameter being the serviceArgs data
flow.
}
message serviceArgs {   //Description of the message
format.
    int64 ReqId = 1;   //Request ID.
    oneof MessageData {
        bytes data = 2;   //Sampled data in the GPB
encoding format.
        string data_json = 4;   //Sampled data in the
JSON encoding format.
        bytes packed_data = 5;   //Sampled data in the
GPB encoding format for packing GPB data.
        string packed_data_json = 6;   //Sampled data
in the JSON encoding format for packing JSON data.
    }
    string errors = 3;   //Description when an error
occurs.
}
```

5.1.2.2 huawei-telemetry.proto

This file defines the data header used in reporting data sampled through telemetry. It contains the sampling path, sampling timestamp, and other important information. Example 5.2 shows its content, with explanations of key parts.

Example 5.2 huawei-telemetry.proto ⏎

```
syntax = "proto3";   //The proto version is v3.
package telemetry;   //The package name is telemetry.
message TelemetryPacked {
   repeated Telemetry telemetry = 1;
}
```

```
message Telemetry {   //Definition of the telemetry
message structure.
   string node_id_str = 1;   //Device name.
   string subscription_id_str = 2;   //Name of the
static subscription.
   string sensor_path = 3;   //Subscription path.
   string proto_path = 13;   //Message path
corresponding to the sampling path in the .proto file.
   uint64 collection_id = 4;   //Sampling round.
   uint64 collection_start_time = 5;  //Start time of
a sampling round.
   uint64 msg_timestamp = 6;    //Timestamp when the
message was generated.
   TelemetryGPBTable data_gpb = 7;   //Data defined by
TelemetryGPBTable.
   uint64 collection_end_time = 8;   //End time of a
sampling round.
   uint32 current_period = 9; //Sampling period, in
ms. Value 0 indicates OnChange sampling for real-time
reporting of changes.
   string except_desc = 10;    //Exception
description, which is reported when a sampling
exception occurs.
   string product_name = 11;   //Product name.
   enum Encoding {
    Encoding_GPB = 0;   //GPB encoding format.
   Encoding_JSON = 1;   //JSON encoding format.
  };
  Encoding encoding = 12;   //Encoding format. If GPB
is used, the data_gpb field is used to carry message
content. Otherwise, the data_str field is used.
  string data_str = 14;   //Valid only when a non-GPB
encoding format is used. Otherwise, this field's value
is empty.
  string ne_id = 15;   //Unique NE ID, identifying the
NE to which data belongs in gateway scenarios.
  string software_version = 16;   //Software version
number.
  string mac_address = 17;    //System Media Access
Control (MAC) address.
  string esn = 18; //Equipment Serial Number (ESN).
}
message TelemetryGPBTable {   //Definition of the
TelemetryGPBTable message structure.
   repeated TelemetryRowGPB row = 1;   //Array
definition. Each array element uses the
TelemetryRowGPB structure.
  repeated DataPath delete = 2;      //Deletes one or
more data paths.
   Generator generator = 3;   //Data source
description, which applies to the OnChange+ service
that requires high reliability.
}
message Generator {
   uint64 generator_id = 1;   //Data source ID.
Multiple data sources can provide data concurrently
while maintaining their own reliability.
   uint32 generator_sn = 2;   //Message sequence
number, ranging from 0x0 to 0xFFFFFFFF. The sequence
numbers of messages sent by each data source must be
consecutive. Otherwise, data out-of-synchronization
occurs.
   bool generator_sync = 3;   //Data source
synchronization flag. Value true indicates that full
OnChange data is being synchronized. If the value is
true but no data is contained, the synchronization is
complete.
}
message TelemetryRowGPB {
   uint64 timestamp = 1;   //Timestamp of sampling the
current instance.
  Path path = 2;   //Data tree node, which contains
only the data path and information of key fields.
   bytes content = 11;   //Sampling instance data. The
sensor_path field is required to determine which
.proto file is used for data encoding.
}
message DataPath {
   uint64 timestamp = 1;   //Timestamp of sampling the
current instance.
   Path path = 2;   //Data tree node, which contains
only the data path and information of key fields.
}
message Path {
   repeated PathElem node = 1;   //Data tree node,
which contains only the data path and information of
key fields.
}
message PathElem {
  string name = 1;   //Name of the data tree node.
  map<string, string> key = 2;   //Key field
name-value mapping table of the data tree node.
}
```

Table 5.2 shows an example of parsing the sampled data.

TABLE 5.2 Example of Parsing the Sampled Data ⏎

GPB Encoding	GPB Decoding

GPB Encoding

```
{
   1:"HUAWEI"
   2:"s4"

   3:"huawei-ifit:ifit/
huawei-ifit-statistics:flow-
statistics/flow-statistic"
   4:46
   5:1515727243419
   6:1515727243514
   7{
     1[{
       1: 1515727243419
       2{
         ...
       }
     }]
   }
   8:1515727243419
   9:10000
   10:"OK"
   11:"Product",
   12:0 }
```

GPB Decoding

```
{
   "node_id_str":"HUAWEI",
   "subscription_id_str":"s4",
   "sensor_path":"huawei-
ifit:ifit/huawei-ifit-
statistics:flow-statistics/
flow-statistic",
  "collection_id":46,
  "collection_start_
time":"2018/1/12
11:20:43.419",
  "msg_timestamp":"2018/1/12
11:20:43.514",
  "data_gpb":{
    "row":[{
      "timestamp":"2018/1/12
11:20:43.419",
      "content":{
        ...
      }
   }]
 },
 "collection_end_
time":"2018/1/12
11:20:43.419",
  "current_period":10000,
  "except_desc":"OK",
  "product_name":"Product",
  "encoding":Encoding_GPB
}
```

5.1.2.3 huawei-ifit.proto

This file describes the data format of a specific service, such as the on-path telemetry service. Example 5.3 shows its content, with explanations of key parts.

Example 5.3 huawei-ifit.proto

```
syntax = "proto3";   //The proto version is v3.
package huawei_ifit;   //The package name is huawei_
ifit.
message Ifit {   //Definition of the IFIT message
structure.
  message Global {    //Definition of the global
message structure.
    bool enable = 1 [json_name = "enable"];   //IFIT
enabling flag.
    uint32 node_id = 2 [json_name = "node-id"];   //
Node ID.
  }
  Global global = 1 [json_name = "global"];   //Global
message data.
  message FlowStatistics {   //Definition of flow
statistics messages' structure.
    message FlowStatistic {   //Definition of a flow
statistics message structure.
      uint64 flow_id = 1 [json_name = "flow-id"];   //
Flow identifier.
      enum Direction {   //Enumeration definition of
flow directions.
        INVALID_ENUM_VALUE_Direction = 0;
        Direction_INGRESS = 1;
        Direction_TRANSITX_INPUT = 2;
        Direction_TRANSITX_OUTPUT = 3;
        Direction_EGRESS = 4;
        Direction_EGRESSX_TOX_CPU = 5;
        Direction_EGRESSX_NORMALX_DROP = 6;
        Direction_INGRESSX_OUTPUT = 7;
        Direction_EGRESSX_INPUT = 8;
        Direction_EGRESSX_BUM = 9;
     };
     Direction direction = 2 [json_name = "direction"];
//Flow direction.
      enum AddressFamily {   //Enumeration definition
of IP address families.
        AddressFamily_IPV4 = 0;
        AddressFamily_IPV6 = 1;
      };
      AddressFamily address_family = 3 [json_name =
"address-family"];   //IP address family.
      string source_ip = 4 [json_name = "source-ip"];
//Source IP address.
      string destination_ip = 5 [json_name =
"destination-ip"];   //Destination IP address.
      uint32 source_mask = 6 [json_name =
"source-mask"];   //Mask of the source IP address.
      uint32 destination_mask = 7 [json_name =
"destination-mask"];   //Mask of the destination IP
address.
      uint32 source_port = 8 [json_name =
"source-port"];   //Source port number.
      uint32 destination_port = 9 [json_name =
"destination-port"];   //Destination port number.
      uint32 protocol = 10 [json_name = "protocol"];
//Protocol type.
      string vpn_name = 11 [json_name = "vpn-name"];
//VPN name.
      uint32 if_index = 12 [json_name = "if-index"];
//Interface index.
      uint32 error_info = 13 [json_name = "error-info"];
//Error code
      uint32 interval = 14 [json_name = "interval"];
//Measurement period, in seconds.
      uint64 period_id = 15 [json_name = "period-id"];
//Measurement period ID. The value is the rounded-up
result of the following formula: Current time (UTC +
leap second compensation time) - January 1, 1970,
00:00:00/Configured measurement period (ms). Here, UTC
is short for Coordinated Universal Time.
      uint64 packet_count = 16 [json_name =
"packet-count"];   //Number of packets.
      uint64 byte_count = 17 [json_name =
"byte-count"];   //Number of bytes.
      uint32 timestamp_second = 18 [json_name =
"timestamp-second"];   //Seconds part of the
timestamp.
      uint32 timestamp_nanosecond = 19 [json_name =
"timestamp-nanosecond"];   //Nanoseconds part of the
timestamp.
      string tunnel_if_index = 20 [json_name =
"tunnel-if-index"];   //Tunnel interface index.
      uint32 ttl = 21 [json_name = "ttl"];   //TTL of
the measured packet.
      uint32 dscp = 22 [json_name = "dscp"];   //DSCP
of the measured packet.
      ...
    }
    repeated FlowStatistic flow_statistic = 1 [json_
name = "flow-statistic"];   //Flow statistics record.
Multiple records can be included.
  }
  FlowStatistics flow_statistics = 2 [json_name =
"flow-statistics"];   //Flow statistics records.
  ...
}
```

Example 5.4 describes the parsing process using an IP 5-tuple-based on-path telemetry message instance (excluding the RPC header).

Example 5.4 Parsing of an IP 5-Tuple-based On-Path Telemetry Message Instance ⏎

```
{
  "node_id_str":"HUAWEI",
  "subscription_id_str":"subscript",
  "sensor_path":"huawei-ifit:ifit/
huawei-ifit-statistics:flow-statistics/
flow-statistic",
  "proto_path":"huawei_ifit.Ifit",
  "collection_id":29,
  "collection_start_time":"2020-02-20 23:41:16.647",
  "msg_timestamp":"2020-02-20 23:41:16.721",
  "data_gpb":{
    "row":[{
      "timestamp":"2020-02-20 23:41:16.647",
      "content":"{
        "flow-statistics":{
      "flow-statistic":[{
         "flow-id":"1179649",
         "direction":"Direction_INGRESS",
         "address-family":"AddressFamily_IPV4",
         "source-ip":"10.1.1.1",
         "destination-ip":"10.1.1.2",
         "source-mask":24,
         "destination-mask":24,
         "source-port":0,
         "destination-port":0,
         "protocol":255,
         "vpn-name":"vpn1",
         "if-index":8,
         "error-info":0,
         "interval":10,
         "period-id":"158221327",
         "packet-count":"82237",
         "byte-count":"9046070",
         "timestamp-second":1582213260,
         "timestamp-nanosecond":43014419,
         "ttl":255,
         "dscp":255
       }]
      }
    }"
  }],
  "generator":{
    "generator_id":0,
    "generator_sn":0,
    "generator_sync":false
   }
 },
 "collection_end_time":"2020-02-20 23:41:16.647",
 "current_period":0,
 "encoding":"Encoding_GPB",
}
```

Table 5.3 describes the key fields.

TABLE 5.3 Key Fields in an IP 5-Tuple-based on-Path Telemetry Message Instance ⏎

Field	Data Type	Description
node_id_str	String	Device name.
subscription_id_str	String	Telemetry subscription name.
sensor_path	String	Subscription path, which specifies a specific sampling service.
proto_path	String	Path of the .proto file.
collection_id	Integer	Sampling round.
collection_start_time	YYYY-MM-DD, HH:MM:SS	Collection start timestamp.
msg_timestamp	YYYY-MM-DD, HH:MM:SS	Message timestamp.
data_gpb	–	Message content. When GPB is used for encoding, the **data_gpb** field is used to carry message content.

Field	Data Type	Description
flow-id	Integer	ID of a flow to be measured.
direction	String	Instance direction.
address-family	String	Address type.
source-ip	Dotted decimal notation	Source IP address.
destination-ip	Dotted decimal notation	Destination IP address.
source-mask	Integer	Mask of the source IP address.
destination-mask	Integer	Mask of the destination IP address.
source-port	Integer	Source port number.
destination-port	Integer	Destination port number.
protocol	Integer	Protocol type.
vpn-name	String	VPN name.
if-index	Integer	Interface index.
error-info	Integer	Error code corresponding to the error information.
interval	Integer	Measurement period, in seconds.
period-id	Integer	Measurement period ID.
packet-count	Integer	Number of packets.
byte-count	Integer	Number of bytes.
timestamp-second	Integer	Seconds part of the timestamp.
timestamp-nanosecond	Integer	Nanoseconds part of the timestamp.
ttl	Integer	Time to Live (TTL) value.

Field	Data Type	Description
dscp	Integer	Differentiated services code point (DSCP).
generator	–	Data source information, including: • generator_id: data source ID, which is an integer. • generator_sn: message sequence number, which is an integer. • generator_sync: data source synchronization flag, which is a Boolean value.
collection_end_time	YYYY-MM-DD, HH:MM:SS	Collection end timestamp.
current_period	Integer	Sampling period, in ms. Value 0 indicates OnChange sampling for real-time reporting of changes.
encoding	Enumerated value	Encoding format. This example uses GPB encoding.

5.2 UDP TELEMETRY-BASED INFORMATION REPORTING

5.2.1 UDP Telemetry Protocol

The UDP telemetry protocol[7] provides continuous and high-speed information reporting based on the YANG Push[8,9] model subscription mechanism. Compared with gRPC, UDP telemetry offers the following advantages:

- Eliminates the need to maintain numerous TCP connections, reducing the burden on data collectors. This significantly improves performance, especially on large-scale networks.
- Eliminates the need to maintain connection status, making it possible to encapsulate and send UDP messages through hardware. This helps achieve further performance gains.

Figure 5.3 shows the layered structure of the UDP telemetry protocol stack.

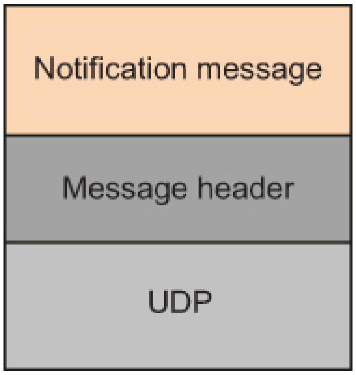


FIGURE 5.3 Layered structure of the UDP telemetry protocol stack. ⏎

Table 5.4 describes each layer.

TABLE 5.4 Layers of the UDP Telemetry Protocol Stack ⏎

Layer	Description
UDP	UDP is used to establish communication connections.
Message header	UDP telemetry message header layer.
Notification message	Layer for the encoded message content. Common encoding formats include GPB and JSON.

5.2.1.1 UDP Telemetry Message Header

Figure 5.4 shows the format of the UDP telemetry message header.

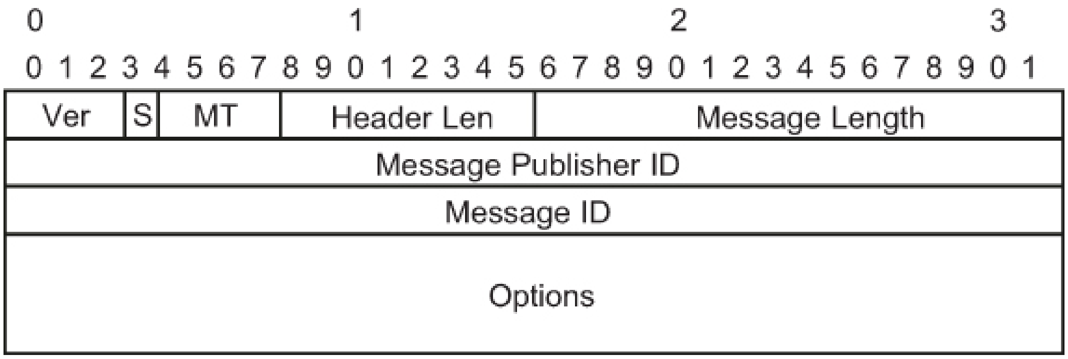


FIGURE 5.4 Format of the UDP telemetry message header. ⏎

Table 5.5 describes the fields in the UDP telemetry message header.

TABLE 5.5 Fields in the UDP Telemetry Message Header ⏎

Field	Length	Description
Ver	3 bits	Protocol version number. The current version number is 1.
S	1 bit	Encoding format in the MT field. • When S is not set, an IETF-defined encoding format is specified by the MT field. • When S is set, the MT field is a private field for freely specifying a non-standard encoding format.
MT	4 bits	Media type, which identifies the encoding format of notification messages. When S is not set, the values and meanings of the MT field are as follows: • 0: reserved • 1: application/yang-data+json[10] • 2: application/yang-data+xml[10] • 3: application/yang-data+cbor[11] When S is set, MT represents a private space to be freely used for non-standard encodings.
Header Len	8 bits	Length (in bytes) of the message header. The message header includes the fixed header and options.
Message Length	16 bits	Total length (in bytes) of the UDP telemetry message containing the message header. When a notification message is segmented using the Segmentation option, the length refers only to the segment length.
Message Publisher ID	32 bits	Message publisher’s identifier, which is unique for the publisher node. This identifier, combined with Message ID, ensures that each message is unique.

Field	Length	Description
Message ID	32 bits	Message identifier, which is generated by the publisher of UDP telemetry messages and used to reconstruct segmented messages. To ensure uniqueness, the publisher uses consecutive Message ID values for each message generated from the same Message Publisher ID. Specifically, the Message ID increases by 1 for each such message, allowing message loss to be detected. Once the Message ID reaches the last value ($2^{32} - 1$), it resets to 0. Different subscribers can share the same Message ID sequence.
Options	Variable	Options in the TLV format.

Figure 5.5 shows the format of the Options field in the message header.

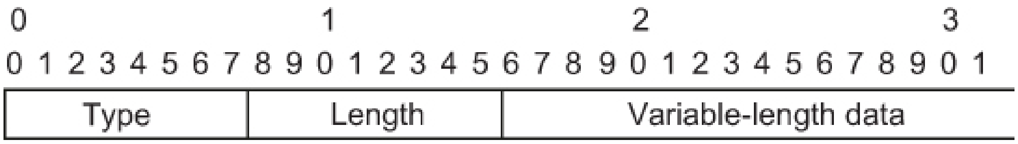


FIGURE 5.5 Format of the options field in the message header. ⏎

Table 5.6 describes the fields in the Options field in the message header.

TABLE 5.6 Fields in the Options Field in the Message Header ⏎

Field	Length	Description
Type	8 bits	Option type.
Length	8 bits	Total length (in bytes) of the TLV, including the Type and Length fields.
Variable-length data	Variable	Value of the TLV.

The options currently defined are Segmentation and Private Encoding.

5.2.1.1.1 Segmentation Option

The maximum length of a UDP message is 65,535 bytes, with a payload up to 65,527 bytes (which excludes the 8-byte UDP header). The Segmentation option is included in a UDP message if the message content exceeds this payload length and needs to be fragmented into multiple segments. Figure 5.6 shows the format of the Segmentation option.

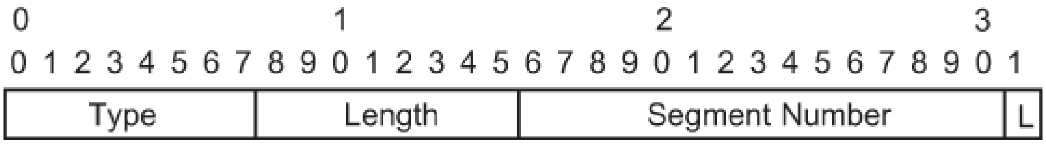


FIGURE 5.6 Format of the Segmentation option. ⏎

Table 5.7 describes the fields in the Segmentation option in the message header.

TABLE 5.7 Fields in the Segmentation Option ⏎

Field	**Length**	**Description**
Type	8 bits	The option type is Segmentation.
Length	8 bits	Length of the Segmentation option. The value is fixed to 4 bytes.
Segment Number	15 bits	Sequence number. The value is 0 for the first segment and increases by 1 for each new segment. Segment Number cannot be reset to zero.
L	1 bit	Flag indicating whether the current segment is the last one in a segmented message. • 0: indicates the current segment is not the last one. • 1: indicates the current segment is the last one, meaning the total number of segments used to transmit this entire message is Segment Number + 1.

If a message is segmented and contains multiple options, the first segment must contain all these options, but subsequent segments may not. The receiver must support receiving out-of-order segments. If the receiver cannot reassemble the entire message because it has not received all segments within the specified time, it should discard all received segments.

5.2.1.1.2 Private Encoding Option

The Private Encoding option is used for a text-based description of the user-defined encoding formats, as the MT field in the message header has limited space. Figure 5.7 shows the format of the Private Encoding option.

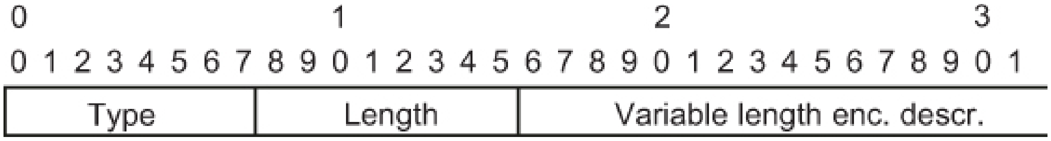


FIGURE 5.7 Format of the Private Encoding option. ⏎

Table 5.8 describes the fields in the Private Encoding option.

TABLE 5.8 Fields in the Private Encoding Option ⏎

Field	**Length**	**Description**
Type	1 byte	The option type is Private Encoding.
Length	1 byte	Length (in bytes) of the Private Encoding option.
Variable length enc. descr.	Variable	User-defined description for the Private Encoding option.

5.2.1.2 UDP Telemetry Notification Message

UDP telemetry notification messages can be encoded in Concise Binary Object Representation (CBOR), XML, or JSON format. For details about the data content related to these formats, see the YANG model definition of the related service.

5.2.2 Telemetry Information Reporting Using UDP

IPv6 on-path telemetry information is carried in notification messages. The specific data model is defined in *draft-fz-ippm-on-path-telemetry-yang*[12], as shown in the following:

```
module: on-path-telemetry
   +--ro on-path-telemetry-data
     +--ro timestamp?               yang:date-and-time
     +--ro acquisition-method?       identityref
     +--ro emission-type?           identityref
     +--ro interface*              [if-name]
       +--ro if-name               if:interface-ref
       +--ro profile-name           string
       +--ro filter
       |  +--ro filter-type?     telemetry-filter-type
       |  +--ro ace-name?  -> /acl:acls/acl/aces/ace/name
       +--ro protocol-type?    telemetry-protocol-type
       +--ro node-action         telemetry-node-action
       +--ro period?              uint64
       +--ro period-number?       uint64
       +--ro flow-mon-id?         uint32
       +--rw method-type?         altmark-method-type
       +--ro altmark-loss-measurement?
       |  +--ro in-traffic-pkts?    yang:counter64
       |  +--ro out-traffic-pkts?   yang:counter64
       |  +--ro in-traffic-bytes?         uint64
       |  +--ro out-traffic-bytes?        uint64
       +--ro altmark-delay-measurement?
       |  +--ro pkts-timestamps?    yang:date-and-time
       |     +--ro pkt-timestamp?   yang:date-and-time
       +--ro path-delay?
       |  +--ro path-delay-mean           uint32
       |  +--ro path-delay-min            uint32
       |  +--ro path-delay-max            uint32
       |  +--ro path-delay-sum            uint64
       +--ro ioam-incremental-tracing
ioam-trace-data
       +--ro ioam-preallocated-tracing  oam-trace-data
       +--ro ioam-direct-export        ioam-trace-data
       +--ro ioam-proof-of-transit      ioam-pot-data
       +--ro ioam-edge-to-edge         ioam-e2e-data
```

The alternate marking- and IOAM-based telemetry data information defined in *draft-fz-ippm-on-path-telemetry-yang* includes the following:

- timestamp: timestamp of a message.
- acquisition-method: method of acquiring (e.g., subscription or query) telemetry updates.
- emission-type: how data packets are sent, for example, on-change or periodic.
- interface: a list of interfaces to which on-path telemetry is applied, including the following detailed information:
  - if-name: interface name.
  - profile-name: identifier of the on-path telemetry configuration profile.
  - filter: filter, such as an ACL or Access Control Entry (ACE), used to identify a flow.
  - protocol-type: type (such as IPv6 or SFC-NSH) of protocol used to encapsulate telemetry data. (SFC-NSH is short for Service Function Chaining Network Service Header.)
  - node-action: action to be taken on a flow, such as marking the alternate marking header, reading the alternate marking data, or unmarking the alternate marking header.

- period: measurement period for alternate marking.
- period-number: number of the alternate marking measurement period.
- flow-mon-id: identifier of the monitored flow. It correlates the exported data of the same flow from multiple nodes and from multiple packets.
- method-type: type of alternate marking.
- altmark-loss-measurement: alternate marking-based packet loss measurement result information, including the numbers of incoming packets, outgoing packets, and bytes.
- altmark-delay-measurement: contains the timestamp information (pkt-timestamp) of packets in the current delay measurement period.
- path-delay: includes the average, minimum, maximum, and total delay of the measured paths.
- Ioam-incremental-tracing: IOAM incremental tracing data.
- Ioam-preallocated-tracing: IOAM pre-allocated tracing data.
- Ioam-direct-export: directly exported IOAM data.
- Ioam-proof-of-transit: IOAM POT data.
- Ioam-edge-to-edge: IOAM E2E data.

5.3 IPFIX-BASED INFORMATION REPORTING

5.3.1 IPFIX Protocol

IPFIX[13,14] is an IETF protocol used to measure IP data flows on a network. It provides a standardized method for collecting and reporting flow information through unified and extensible templates.

With this protocol, network administrators can easily extract and view measurement information.

IPFIX is implemented based on flows, which are streams of packets that come from the same sub-interface or that have the same 5-tuple information (source and destination IP addresses, source and destination port numbers, and protocol type), ToS, and other information. This protocol records various flow statistics, including timestamps, number of packets, and number of bytes.

5.3.1.1 Typical IPFIX Networking

Figure 5.8 shows the typical IPFIX networking, which involves the following three roles:

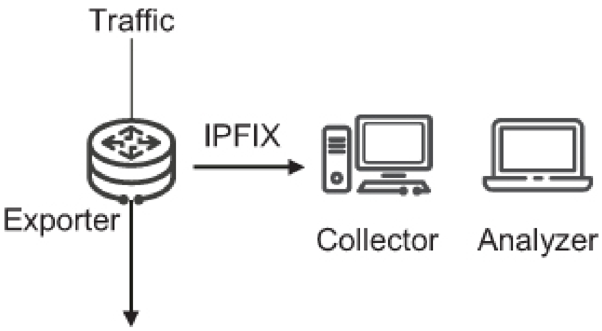


FIGURE 5.8 Typical IPFIX networking. ⏎

- Exporter: analyzes and processes network traffic, extracts flow statistics that meet specific conditions, and exports the statistics to the collector.
- Collector: parses the collected data packets and saves the statistics to the database for the analyzer to parse.
- Analyzer: extracts statistics from the collector, provides support for subsequent service processing, and converts the statistics into graphical representations.

Simply put, IPFIX measurement involves a group of metering processes that observe data packets at one or more observation points. The exporter sends information about all observation points to the collector through IPFIX. The analyzer then analyzes the information and implements network optimization, security monitoring, traffic accounting, and other functions based on the analysis result.

5.3.1.2 Basic Concepts in IPFIX

IPFIX involves the following basic concepts:

- Observation point and observation domain: places where IP packets are observed. An observation point can be specified on each interface of a device. An observation domain is a set of observation points and can be a line card or a hardware module.
- Flow record: contains information about a specific flow observed at an observation point. A flow record contains measured properties (e.g., total number of bytes in all packets of the flow) and usually characteristic properties (e.g., source IP address) of a flow.
- Metering process: processes IP data packets at one or more observation points and generates flow records.
- Exporting process: processes flow records, converts them into IPFIX messages, and outputs the messages.

- Sampling functions: sample service data flows in N:1 mode in the metering process. This reduces the number of data flows to be processed, easing the processing load in scenarios that do not require high precision.
- Filter functions: filter service data flows in the metering process. They filter out traffic that administrators are not interested in.
- Collecting process: receives and processes IPFIX messages on the collector.
- Template: an ordered sequence of <type length> pairs used to specify the structure and semantics of a particular set of information that needs to be transmitted from an IPFIX device to a collector. Each template is uniquely identified by a template ID.
- Template record: defines the structure and interpretation of fields in a data record.
- Data record: contains the values of parameters defined by a template record.
- Options template record: defines the structure and interpretation of fields in a data record and the context information (e.g., line card and output process) of the data record.
- Set: a collection of the same type of records. There are three types of sets: template set, options template set, and data set.
- Template set: a set of one or more template records.
- Options template set: a set of one or more options template records.
- Data set: a set of one or more data records of the same type. Each data record is first defined by a template record or an options template record.
- Information Element (IE): a description of a protocol- and encoding-independent attribute that may be present in an IPFIX record. IEs are defined in IANA's "IPFIX Information Elements" registry[15]. The type associated with an IE indicates constraints on what the IE may contain and also determines the valid encoding mechanisms for use in IPFIX.

Figure 5.9 shows a summary of IPFIX terms. A data set consists of data records, a template set consists of template records, and an options template set consists of option template records.

FIGURE 5.9 IPFIX terms. ⏎

Set	Contents	
	Template	Record
Data Set	/	Data Record(s)
Template Set	Template Record(s)	/
Options Template Set	Options Template Record(s)	/

5.3.1.3 IPFIX Message

An IPFIX message is composed of a message header and zero or more sets (data, template, or options template sets), as shown in Figure 5.10.

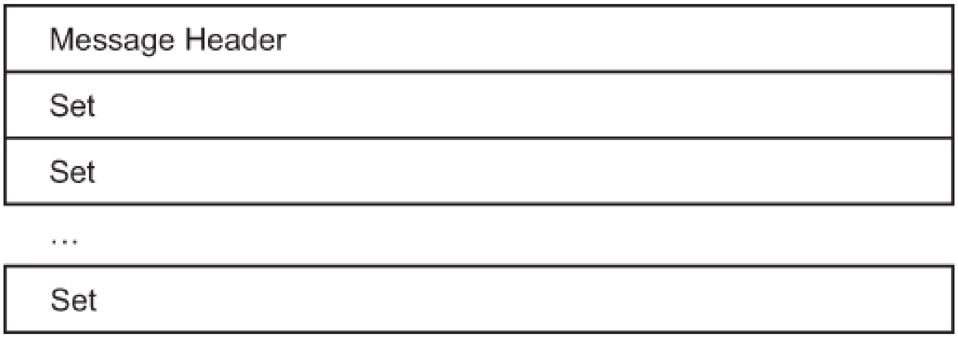


FIGURE 5.10 IPFIX message structure. ⏎

Figure 5.11 shows an example of an IPFIX message consisting of a template set and a data set (containing three data records).

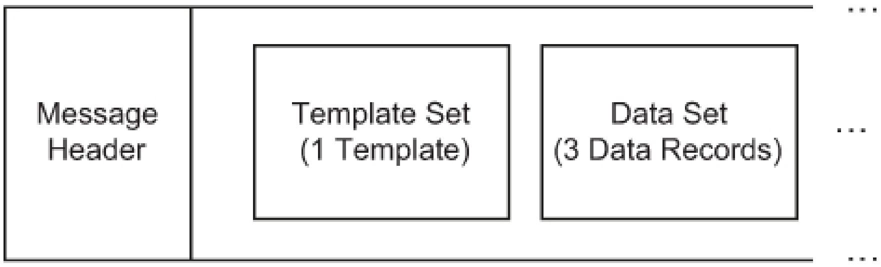


FIGURE 5.11 Example of an IPFIX message. ⏎

The following provides details about each part of the message format.

5.3.1.3.1 IPFIX Message Header Figure 5.12 shows the format of the IPFIX message header.

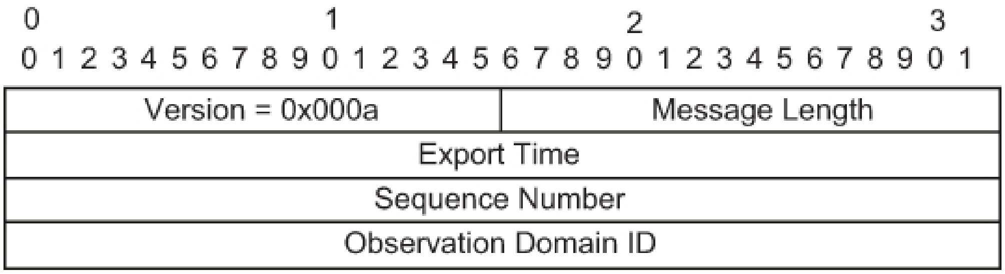


FIGURE 5.12 Format of the IPFIX message header. ⏎

Table 5.9 describes the fields in the IPFIX message header.

TABLE 5.9 Fields in the IPFIX Message Header ⏎

Field	Length	Description
Version	2 bytes	Version number. The IPFIX version number is 0x000a, which is 1 greater than NetFlow version 9[16].
Message Length	2 bytes	Total length (in bytes) of the IPFIX message, including the message header and sets.
Export Time	4 bytes	Time when the IPFIX message header left the exporter. The value is the number of seconds elapsed since 00:00:00 on January 1, 1970 (UTC), encoded as an unsigned 32-bit integer.
Sequence Number	4 bytes	Incremental sequence counter for all IPFIX data records. This field can be used by the collecting process to identify whether any IPFIX data record is lost. The sequence number does not increase for template records or options template records.
Observation Domain ID	4 bytes	ID of an observation domain. Observation Domain ID should be 0 when no specific observation domain ID is relevant for the entire IPFIX message (e.g., when exporting the aggregated data records).

5.3.1.3.2 IPFIX Set Figure 5.13 shows the IPFIX set structure.

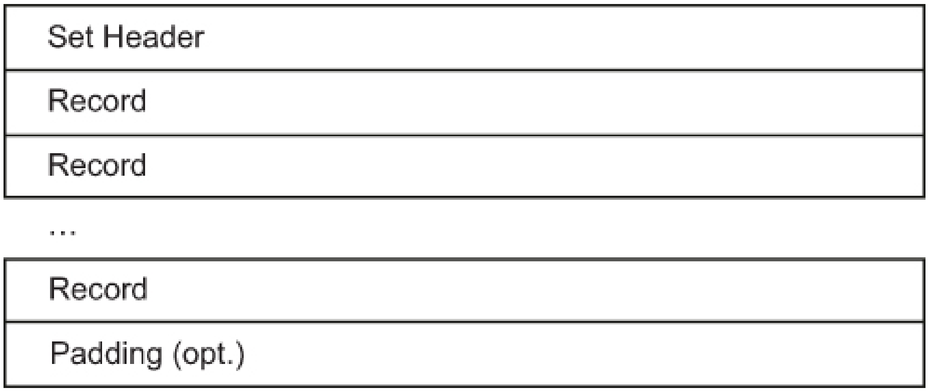


FIGURE 5.13 IPFIX set structure. ⏎

Figure 5.14 shows the format of the set header.

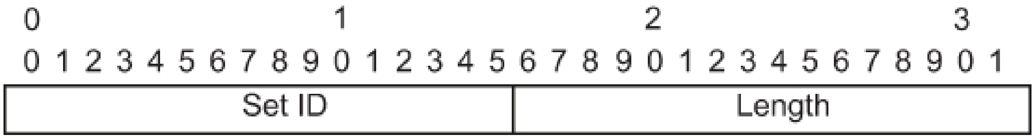


FIGURE 5.14 Format of the IPFIX set header. ⏎

Table 5.10 describes the fields in the IPFIX set header.

TABLE 5.10 Fields in the IPFIX Set Header ⏎

Field	**Length**	**Description**
Set ID	2 bytes	Set identifier. IDs 0 and 1 are not used. IDs 2 and 3 are reserved for template sets and options template sets, respectively. IDs 4–255 are reserved for future use. IDs 256 and above are used for data sets.
Length	2 bytes	Total length (in bytes) of a set, including the set header, all records, and optional padding. Because a single set may contain multiple records, this field must be used to determine the location of the next set.

Figure 5.15 shows the format of an IPFIX template record.

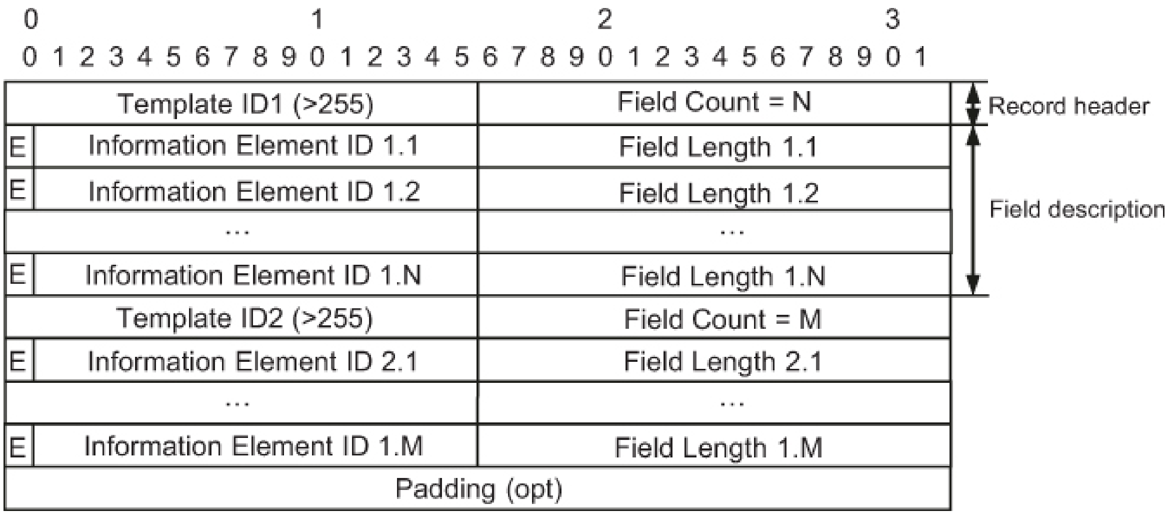


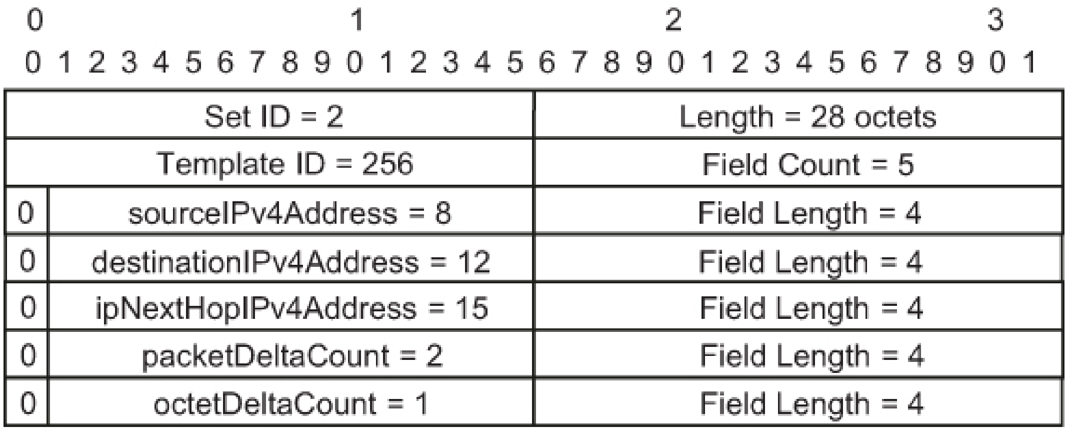
FIGURE 5.15 Format of an IPFIX template record. ⏎

Table 5.11 describes the fields in an IPFIX template record.

TABLE 5.11 Fields in an IPFIX Template Record ⏎

Field	Length	Description
Template ID	16 bits	Identifier of a template. Each template record has a unique template ID ranging from 256 to 65535. Values 0 to 255 are reserved for special set types (e.g., template sets themselves).
Field Count	16 bits	Number of fields in the template record.
E	1 bit	Enterprise bit. If this bit is 1, the Information Element ID identifies an enterprise-specific IE. If this bit is 0, the Information Element ID identifies an IETF-defined IE.
Information Element ID	Variable	Identifier of an IE.
Field Length	16 bits	Length (in bytes) of the value of the corresponding IE.
Padding	Variable	Optional padding bytes, which must be set to 0

Figure 5.16 shows an example of an IPFIX template set.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Set ID = 2		Length = 28 octets
Template ID = 256		Field Count = 5
0	sourceIPv4Address = 8	Field Length = 4
0	destinationIPv4Address = 12	Field Length = 4
0	ipNextHopIPv4Address = 15	Field Length = 4
0	packetDeltaCount = 2	Field Length = 4
0	octetDeltaCount = 1	Field Length = 4

FIGURE 5.16 Example of an IPFIX template set. ⏎

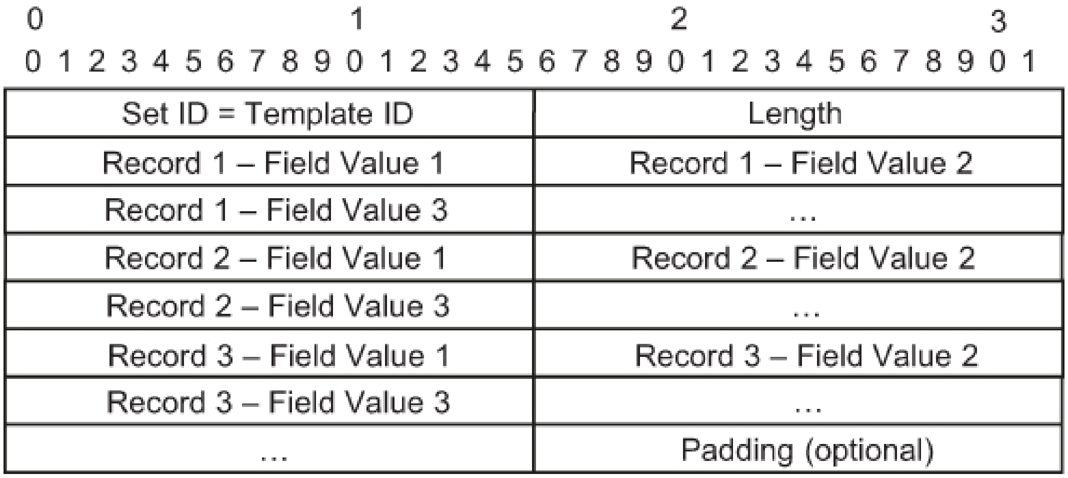
Table 5.12 describes the fields in the IPFIX template set.

TABLE 5.12 Fields in the IPFIX Template Set ⏎

Field	Length	Description
Set ID	16 bits	The template set ID is 2

Field	Length	Description
Length	16 bits	The length of the template set is 28 bytes
Template ID	16 bits	The template ID is 256 (corresponding to the set ID in the subsequent data set).
Field Count	16 bits	The number of fields in the template is 5, including the subsequent fields such as the source IPv4 address.
0	1 bit	The IE is defined by the IETF. The meanings of the subsequent 0 fields are the same.
sourceIPv4Address = 8	15 bits	Source IPv4 address IE, with value 8.
Field Length	16 bits	Length of the value of the corresponding IE. Field Length in this row indicates that the length of the value of the source IPv4 address IE is 4. The meanings of the subsequent Field Length fields are the same.
destinationIPv4Address = 12	15 bits	Destination IPv4 address IE, with value 12.
ipNextHopIPv4Address = 15	15 bits	Next hop IPv4 address IE, with value 15.
packetDeltaCount = 2	15 bits	Packet increment count IE, with value 2.
octetDeltaCount = 1	15 bits	Byte increment count IE, with value 1.

An IPFIX data set consists of a set header and one or more field values. The template ID to which the field values belong is encoded in the Set ID field (i.e., Set ID in a data set = Template ID in the corresponding template set). Figure 5.17 shows the format of an IPFIX data set.

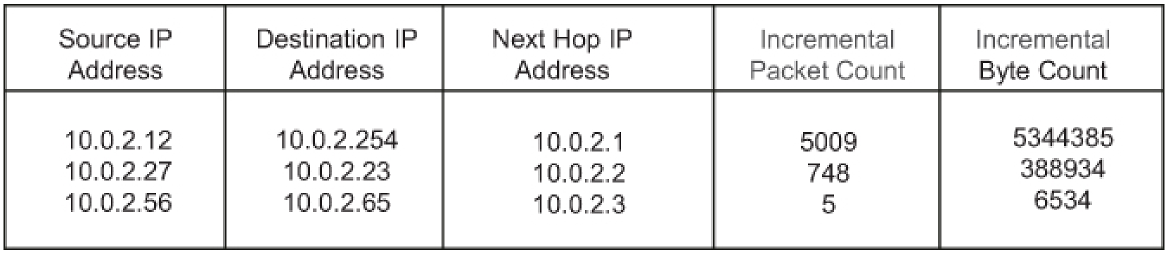


```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Set ID = Template ID	Length
Record 1 – Field Value 1	Record 1 – Field Value 2
Record 1 – Field Value 3	…
Record 2 – Field Value 1	Record 2 – Field Value 2
Record 2 – Field Value 3	…
Record 3 – Field Value 1	Record 3 – Field Value 2
Record 3 – Field Value 3	…
…	Padding (optional)

FIGURE 5.17 Format of an IPFIX data set. ⏎

Figure 5.18 shows an example of statistics from three IPFIX messages that are collected based on the source, destination, and next hop IP addresses.



Source IP Address	Destination IP Address	Next Hop IP Address	Incremental Packet Count	Incremental Byte Count
10.0.2.12	10.0.2.254	10.0.2.1	5009	5344385
10.0.2.27	10.0.2.23	10.0.2.2	748	388934
10.0.2.56	10.0.2.65	10.0.2.3	5	6534

FIGURE 5.18 Example of IPFIX message statistics. ⏎

When an IPFIX template set uses the template set example provided earlier, the format of the corresponding IPFIX data set is shown in Figure 5.19.

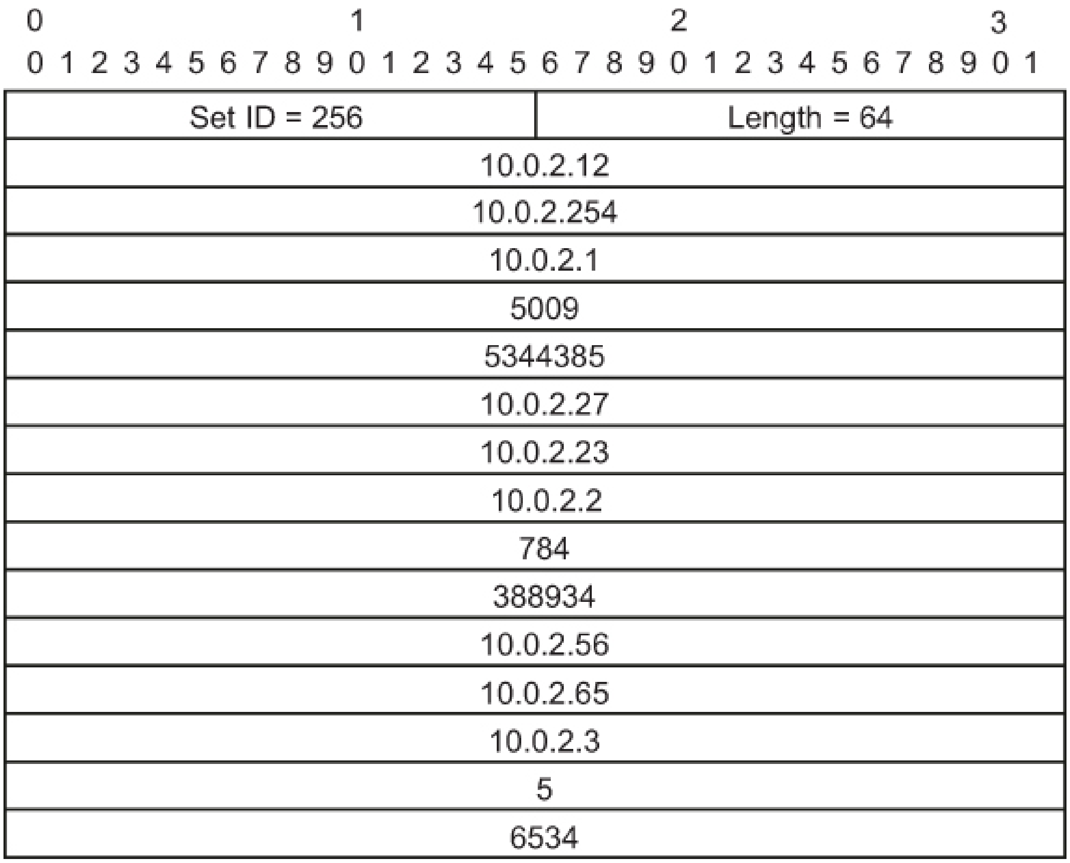


FIGURE 5.19 Format of an example IPFIX data set. ⏎

Table 5.13 describes the fields in the IPFIX data set.

TABLE 5.13 Fields in the IPFIX Data Set ⏎

Field	Length	Description
Set ID	2 bytes	The value is 256, which is the same as the value of Template ID in the IPFIX template set example.
Length	2 bytes	The length of the data set is 64 bytes.
10.0.2.12	4 bytes	Source IP address of the first data record.
10.0.2.254	4 bytes	Destination IP address of the first data record.
10.0.2.1	4 bytes	Next hop IP address of the first data record.
5009	4 bytes	Number of incremental packets in the first data record.
5344385	4 bytes	Number of incremental bytes in the first data record.

Field	Length	Description
10.0.2.27	4 bytes	Source IP address in the second data record. The remaining fields in the second and third data records have similar meanings to those in the first data record.

5.3.2 Telemetry Information Reporting Using IPFIX

IPFIX can be used for reporting on-path telemetry information, including alternate marking and IOAM information.

5.3.2.1 Reporting Alternate Marking-Based Telemetry Information

IPFIX IEs[17] for reporting alternate marking-based telemetry information are defined in *draft-ietf-opsawg-ipfix-alt-mark*. For IPFIX, telemetry data can be decomposed by a data collector or by an alternate marking-aware node (exporter) that is used to output data.

When a data collector is used to decompose telemetry data, the following IEs[18] are used to report alternate marking information such as FlowMonID, the L flag, and D flag encapsulated in the HBH, DOH, or SRH.

- ipPayloadPacketSection (IE 314): carries data (the length of which is specified by sectionExportedOctets) from the IP payload of a measured packet. If sectionOffset exists, the data starts from the specified offset of n bytes after the first byte of the IP payload. If sectionOffset does not exist, the data starts from the first byte of the IP payload. The IPv6 payload is the rest of the packet following the 40-byte IPv6 header, including all IPv6 extension headers.
- sectionOffset (IE 409): specifies an offset of a data packet section (e.g., ipHeaderPacketSection or ipPayloadPacketSection). If sectionOffset does not exist, the data starts from the first byte of the corresponding packet section.
- sectionExportedOctets (IE 410): Data length in ipPayloadPacketSection.

When an exporter is used to decompose telemetry data, the following IEs[17] (whose IDs are to be defined) are used to report alternate marking information such as FlowMonID, the L flag, and D flag.

- FlowMonID: identifier of a measured flow

- L flag: packet loss measurement flag
- D flag: delay measurement flag
- PeriodID: identifier of a measurement period

The following IEs[18] (whose IDs are to be defined) are used to report the one-way path delay information:

- pathDelayMeanDeltaMicroseconds: average one-way path delay.
- pathDelayMinDeltaMicroseconds: minimum one-way path delay.
- pathDelayMaxDeltaMicroseconds: maximum one-way path delay.
- pathDelaySumDeltaMicroseconds: total one-way path delay. To reduce the processing load of measurement devices, the analyzer calculates the average delay based on the total delay reported by the measurement devices and the total number of packets.

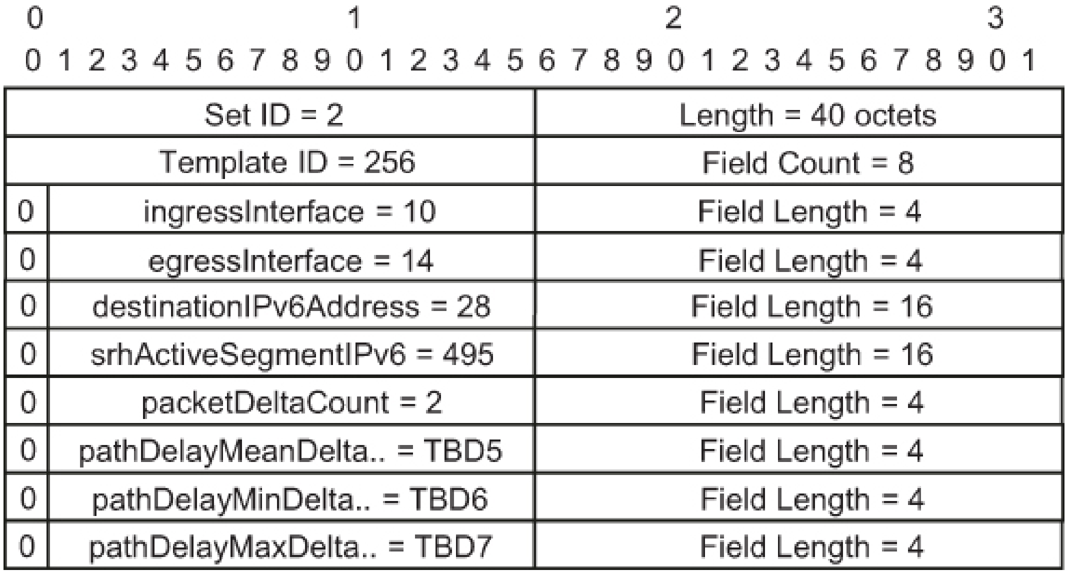
The following IEs[19] are used to report the measurement statistics:

- octetDeltaCount (IE 1) and packetDeltaCount (IE 2): numbers of bytes and packets since the last report, used to count packet loss statistics.
- flowStartSeconds (IE 150), flowStartMilliseconds (IE 152), flowStartMicroseconds (IE 154), flowStartNanoseconds (IE 156), flowEndSeconds (IE 151), flowEndMilliseconds (IE 153), flowEndMicroseconds (IE 155), and flowEndNanoseconds (IE 157): measurement result information used for flow delay calculation.

The following IEs[19, 20, 21] are used to report the relationship between the forwarding topology and control plane:

- ingressInterface (IE 10) and egressInterface (IE 14): a node's logical inbound and outbound interfaces used to forward a data packet.
- egressPhysicalInterface (IE 253): a node's physical outbound interface used to forward a data packet.
- srhActiveSegmentIPv6 (IE 495): active segment on an SRv6 forwarding path.
- destinationIPv6Address (IE 28), destinationTransportPort (IE 11), protocolIdentifier (IE 4), and sourceIPv6Address (IE 27): a flow's IPv6 destination address, destination port number, protocol type, and IPv6 source address.

Figure 5.20 shows the format of an IPFIX template set message used to report an alternate marking-based aggregated delay measurement result that contains egressInterface and srhActiveSegmentIPv6.

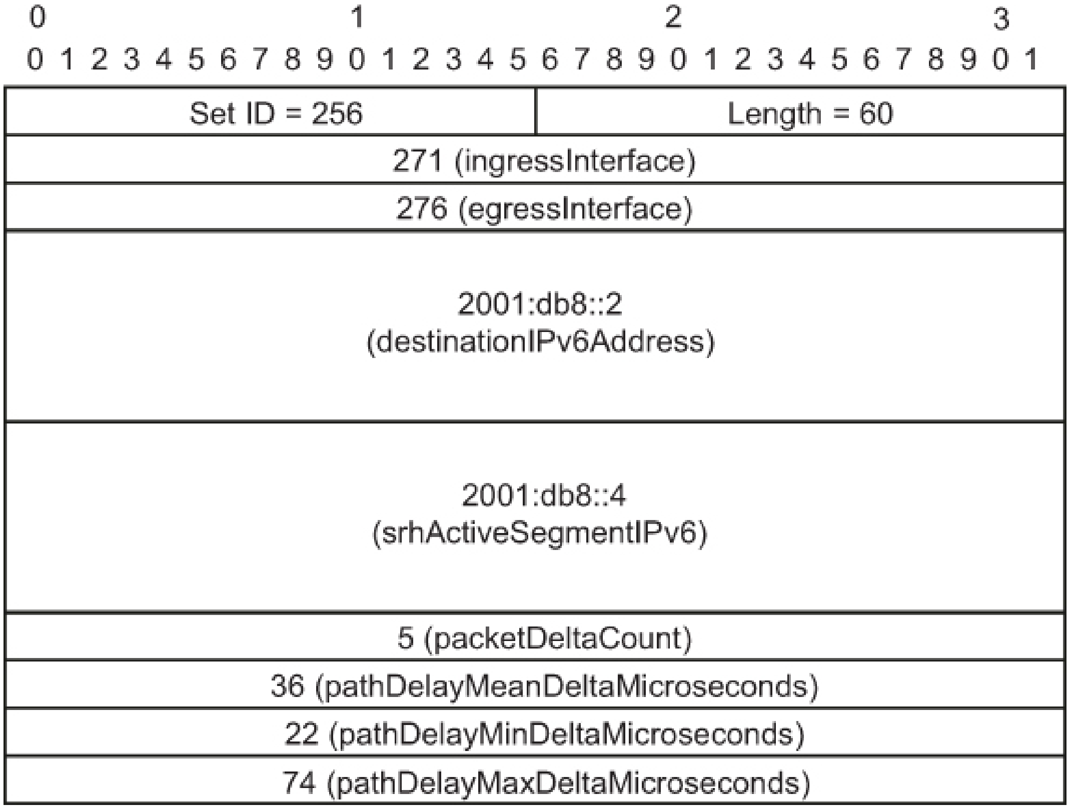


```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Set ID = 2		Length = 40 octets
Template ID = 256		Field Count = 8
0	ingressInterface = 10	Field Length = 4
0	egressInterface = 14	Field Length = 4
0	destinationIPv6Address = 28	Field Length = 16
0	srhActiveSegmentIPv6 = 495	Field Length = 16
0	packetDeltaCount = 2	Field Length = 4
0	pathDelayMeanDelta.. = TBD5	Field Length = 4
0	pathDelayMinDelta.. = TBD6	Field Length = 4
0	pathDelayMaxDelta.. = TBD7	Field Length = 4

FIGURE 5.20 Format of an IPFIX template set message based on alternate marking. ⏎

Figure 5.21 shows the format of an IPFIX data set message.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Set ID = 256	Length = 60
271 (ingressInterface)	
276 (egressInterface)	
2001:db8::2 (destinationIPv6Address)	
2001:db8::4 (srhActiveSegmentIPv6)	
5 (packetDeltaCount)	
36 (pathDelayMeanDeltaMicroseconds)	
22 (pathDelayMinDeltaMicroseconds)	
74 (pathDelayMaxDeltaMicroseconds)	

FIGURE 5.21 Format of an IPFIX data set message based on alternate marking. ⏎

5.3.2.2 Reporting IOAM-Based Telemetry Information

Exporting IOAM data in raw (i.e., uninterpreted) format from network devices to analytics systems using IPFIX is defined in *draft-spiegel-ippm-ioam-rawexport*[22]. It specifies that the exporter does not interpret, aggregate, or reformat IOAM data before export. Instead, the IOAM node that implements IOAM data plane operations encapsulates, updates, and decapsulates the IOAM data, while the IOAM data processing system (collector/analyzer) interprets the data. This allows the IOAM node to focus on data plane operations, thereby reducing the processing load and improving the overall efficiency.

The following describes the IPFIX IEs involved in IOAM raw data export. These IEs can be used to carry and report the IOAM data in measured data packets. For further details about the IEs, see the IANA "IPFIX Information Elements" registry.

- ipHeaderPacketSection (IE 313): carries data (the length of which is specified by sectionExportedOctets) starting from the IP header of a measured packet, for example, the IP 5-tuple information of a monitored flow.
- dataLinkFrameSection (IE 315): carries data (the length of which is specified by sectionExportedOctets) starting from the data link frame of a measured packet.
- dataLinkFrameType (IE 408): specifies the type of data link frames.
- sectionOffset (IE 409): specifies an offset of a data packet section (e.g., ipHeaderPacketSection or ipPayloadPacketSection). If sectionOffset does not exist, the data starts from the first byte of the corresponding packet section.
- sectionExportedOctets (IE: 410): specifies the length of the exported data segment.

The following IEs can be used to report necessary IOAM information, so that the analyzer can restore complete telemetry data.

- forwardingStatus (IE 89): forwarding status of the measured flow (e.g., normal forwarding, ACL-based packet loss, and packet loss caused by table lookup errors).

- ioamReportFlags (IE ID to be defined): an 8-bit IOAM report flag that describes an attribute associated with an IOAM report. The bits are as follows:
  - Bit 0: dropped association, indicating that the data packet has been discarded.
  - Bit 1: congested queue association, indicating that congestion occurs in the monitored queue.
  - Bit 2: tracked flow association, indicating that a desired flow is matched.
  - Bits 3 to 7: reserved.
- ioamEncapsulationType (IE ID to be defined): IOAM encapsulation type. This IE specifies the encapsulation type of ioamPreallocatedTraceData, ioamIncrementalTraceData, ioamE2EData, ioamPOTData, and ioamDirectExportData. The IOAM encapsulation types currently defined are as follows:
  - 0: None. IOAM data complies with the format defined in RFC 9197[23].
  - 1: Generic Routing Encapsulation (GRE). IOAM data complies with the format defined in *draft-weis-ippm-ioam-eth*[24].
  - 2: IPv6. IOAM data complies with the format defined in RFC 9486[25].
  - 3: Generic Protocol Extension for VXLAN (VXLAN-GPE). IOAM data complies with the format defined in *draft-brockners- ippm-ioam-vxlan-gpe*[26].
  - 4: Generic Network Virtualization Encapsulation (GENEVE) Option. IOAM data complies with the format defined in *draft-brockners-ippm-ioam-geneve*[27].
  - 5: GENEVE Next Protocol. IOAM data complies with the format defined in *draft-weis-ippm-ioam-eth*[24].
  - 6: Network Service Header (NSH). IOAM data complies with the format defined in RFC 9452[28].
- ioamPreallocatedTraceData (IE ID to be defined): IOAM pre-allocated trace data. This IE carries the IOAM pre-allocated trace data field defined in RFC 9197. The data format is determined by the ioamEncapsulationType IE.
- ioamIncrementalTraceData (IE ID to be defined): IOAM incremental trace data. This IE carries the IOAM incremental trace data field defined in RFC 9197. The data format is determined by the ioamEncapsulationType IE.
- ioamE2EData (IE ID to be defined): IOAM E2E data. This IE carries the IOAM E2E data field defined in RFC 9197. The data format is determined by the ioamEncapsulationType IE.

- ioamPOTData (IE ID to be defined): IOAM proof of transit data. This IE carries the IOAM proof of transit data field defined in RFC 9197. The data format is determined by the ioamEncapsulationType IE.
- ioamDirectExportData (IE ID to be defined): IOAM direct export data. This IE carries the IOAM direct export data field defined in RFC 9326[29]. The data format is determined by the ioamEncapsulationType IE.
- ipHeaderPacketSectionWithPadding (IE ID to be defined): IP data section to be supplemented. The sectionExportedOctets (IE 410) field specifies the number of bytes to be intercepted from the raw packet. Padding is added if the raw packet contains fewer bytes than the number specified by sectionExportedOctets (IE 410). If there is no sectionOffset field corresponding to the ipHeaderPacketSectionWithPadding IE, the offset is 0.

Figure 5.22 shows the format of an IOAM-based IPFIX template set message, where IOAM data is exported as a part of ipHeaderPacketSection.

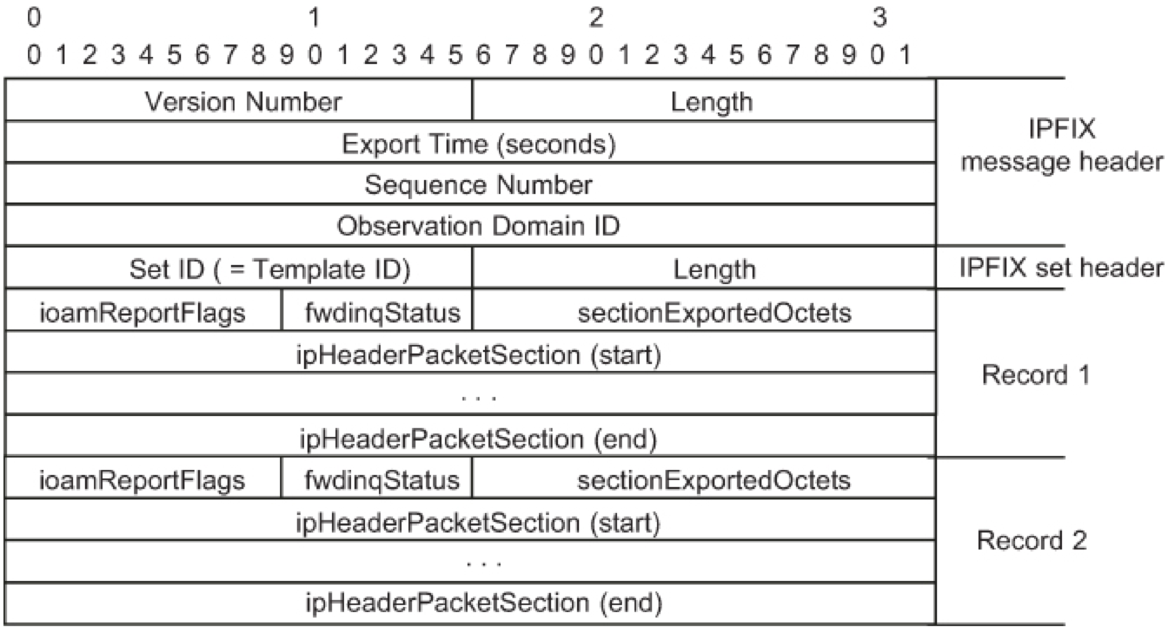


FIGURE 5.22 Format of an IPFIX template set message based on IOAM. ⏎

The fields involved in this message are explained earlier. The associated data set message contains data specified by this template set, that is, content (the length of which is specified by the sectionExportedOctets field) starting from the IP header of the measured traffic packet.

5.4 STORIES BEHIND IPV6 ON-PATH TELEMETRY

5.4.1 From SNMP to Telemetry

In my view, the challenges involved in IP network O&M are closely related to O&M data. One of the major problems is the lack of specific types of O&M data. A typical example is the lack of on-path measurement information, making it difficult to demarcate and locate network faults. This section focuses on the issues related to the volume and performance of data reporting.

IP network O&M used to rely heavily on SNMP[30]. However, SNMP typically takes minutes to report information and consumes significant CPU resources due to its pull mode approach, which requires initiating a request to the involved device for each data retrieval operation. This severely limits its ability to support high-speed data collection. Furthermore, SNMP describes data by using structure-agnostic MIBs, making standardization difficult. For example, IP networks have statistical multiplexing characteristics that can lead to temporary traffic surges, known as microbursts, which negatively affect user experience. But SNMP cannot detect this issue. Specifically, the average bandwidth calculated at minute-level intervals may be significantly less than the available link bandwidth of network devices, leading to inconsistencies in network O&M. This means that although the link bandwidth of network devices appears adequate to meet service bandwidth demands, service deterioration has already occurred.

The emergence of telemetry technologies, like gRPC, has made reporting network device information far more performant and allowed far more data to be reported. Such technologies report binary data and support the push mode. Compared with SNMP, they achieve over a tenfold increase in performance, enable information reporting within subseconds, and can be used to detect microbursts. Furthermore, gRPC supports a wide range of programming features, making it easy to customize reported information through structured data descriptions and to subscribe to information. The shift from the pull mode used by SNMP/MIB to the push mode used by telemetry is an important evolutionary step in IP network O&M and can better meet the need for on-path measurement information reporting.

5.4.2 What Is the Appropriate Cost for O&M?

Our research into the technologies used in on-path measurement has shown that handling reported data is an important task that places significant demands on the system performance of the controller and network devices as a whole. After calculating the volumes of reported data in typical scenarios using passport- and postcard-based on-path measurement, we were shocked at the results — they were far higher than we initially anticipated, revealing the huge amount of information to be reported.

Take IOAM technology as an example. OAM information needs to be output on a per-packet basis at a specific linear ratio. Assuming that each packet has a payload of 512 bytes and records four types of information at each node, with each type being 4 bytes in size, the data collected at each node would be 4 × 4 = 16 bytes. For a path consisting of 10 hops, the additional OAM information received by the egress would be 16 × 10 = 160 bytes according to IOAM Trace Option, and the linear proportion k would be 160/512 = 5/16.

Routers today typically use 400 Gbps links. According to the aforementioned k value, network devices must use links with a rate greater than 125 Gbps to report OAM information. This means that the controller, analyzer, and collector would need to store and process 1,350,000 GB (125 Gbps/8 × 24 × 3600) of data within a day, a truly astonishing amount. Network operators naturally want more bandwidth to carry monetizable services, but they would be unlikely to sacrifice links with a rate greater than 125 Gbps just to report OAM information, nor would they be willing to store and process the massive amount of such information.

IPv6 on-path telemetry will have a significant impact on how network operators view O&M. However, this does not mean that solving network OAM problems should be done at any cost. The greatest challenge in researching IPv6 on-path telemetry lies in finding ways to minimize the implementation cost. To address this challenge, IFIT based on the alternate marking method was developed. By implementing fault demarcation and locating through on-path measurement while efficiently controlling the volume of reported data, it brings the cost of network O&M to within an acceptable level.

IP network O&M has long been a challenge, and O&M personnel have invested significant resources into it. This has led many to instinctively accept that low IP O&M efficiency is the norm and, therefore, invest more in developing IP services rather than enhancing network O&M capabilities. In the past, the difficulties in IP network O&M were mainly due to the lack of effective technical means. But with the development of IPv6 Enhanced and IFIT technologies, on-path telemetry has become feasible. This raises a practical question for network operators: Are they willing to invest enough to deploy these technologies? Even if IFIT is feasible, it incurs costs associated with things like requiring new network hardware or chips for support, deployment of a network controller, clock synchronization, and additional link bandwidth for information reporting. While network operators are accustomed to the existing operations model and can mark these OAM costs as expenses, the introduction of new technology is undoubtedly a challenge.

During the promotion of IPv6 Enhanced innovation, we often face challenges that arise from habitual thinking. We have therefore proposed the concept of "Making-up Missed IP Lessons"[31]: IP's simplicity can be seen as a product of its time. When IP was first developed, it could only support simple features due to the limited network software and hardware capabilities at that time. Yet this should not be taken to mean that IP is inherently limited in this regard. Today, with breakthroughs in network software and hardware capabilities, many technologies that were once considered impossible have now become possible. IP technologies and networks need to be upgraded to possess even stronger capabilities for resolving O&M issues and meeting service requirements. On the premise of technical feasibility, I believe that the following two aspects are vital for the healthy development of IP network O&M: First, network technology innovators need to minimize the cost incurred by technology innovation. And second, network operators need to balance their investment in network O&M while developing services.

REFERENCES

1. Fioccola G, Zhu K, Graf T et al. Alternate marking deployment framework [EB/OL]. (2024-07-03) [2024-09-30]. Draft-ietf-ippm-alt-mark-deployment-01. ⏎
2. Brockners F, Bhandari S, Bernier D et al. In situ operations, administration, and maintenance (IOAM) deployment [EB/OL]. (2023-04) [2024-09-30]. RFC 9378. ⏎
3. Google. What is gRPC? [EB/OL]. (2024-09) [2024-09-30]. ⏎
4. Google. Protocol buffers documentation [EB/OL]. (2024-09) [2024-09-30]. ⏎
5. Bray T. The javascript object notation (JSON) data interchange format [EB/OL]. (2014-03) [2024-09-30]. RFC 7159. ⏎
6. ECMA. The JSON data interchange syntax [EB/OL]. (2017-12) [2024-09-30]. ECMA-404. ⏎
7. Zheng G, Zhou T, Graf T et al. UDP-based transport for configured subscriptions [EB/OL]. (2024-07-04) [2024-09-30]. Draft-ietf-netconf-udp-notif-14. ⏎
8. Voit E, Clemm A, Gonzalez Prieto A et al. Subscription to YANG notifications [EB/OL]. (2019-09) [2024-09-30]. RFC 8639. ⏎
9. Clemm A, Voit E. Subscription to YANG notifications for datastore updates [EB/OL]. (2019-09) [2024-09-30]. RFC 8641. ⏎

10. Bierman A, Bjorklund M, Watsen K. RESTCONF protocol [EB/OL]. (2017-01) [2024-09-30]. RFC 8040. ⏎
11. Veillette M, Petrov I, Pelov A et al. Encoding of data modeled with YANG in the concise binary object representation (CBOR) [EB/OL]. (2022-07-18) [2024-09-30]. RFC 9254. ⏎
12. Fioccola G, Zhou T. On-path telemetry YANG data model [EB/OL]. (2024-06-19) [2024-09-30]. draft-fz-ippm-on-path-telemetry-yang-00. ⏎
13. Claise B, Trammell B, Aitken P. Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information [EB/OL]. (2013-09) [2024-09-30]. RFC 7011. ⏎
14. Sadasivan G, Brownlee N, Claise B. Architecture for IP flow information export [EB/OL]. (2009-03) [2024-09-30]. RFC 5470. ⏎
15. IANA. IP flow information export (IPFIX) entities [EB/OL]. (2024-09) [2024-09-30]. ⏎
16. Claise B. Cisco systems netflow services export version 9 [EB/OL]. (2004-10) [2024-09-30]. RFC 3954. ⏎
17. Graf T, Fioccola G, Zhou T et al. IPFIX alternate-marking information [EB/OL]. (2024-07-08) [2024-09-30]. Draft-ietf-opsawg-ipfix-alt-mark-00. ⏎
18. Graf T, Claise B, Huang Feng A. Export of delay performance metrics in IP Flow information export (IPFIX) [EB/OL]. (2024-09-25) [2024-09-30]. Draft-ietf-opsawg-ipfix-on-path-telemetry-13. ⏎
19. Quittek J, Bryant S, Bryant B et al. Information model for IP flow information export [EB/OL]. (2008-01) [2024-09-30]. RFC 5102. ⏎
20. Graf T, Claise B, Francois P. Export of segment routing IPv6 information in IP flow information export (IPFIX) [EB/OL]. (2022-07-24) [2024-09-30]. Draft-tgraf-opsawg-ipfix-srv6-srh-05. ⏎
21. Filsfils C, Previdi S, Ginsberg L et al. Segment routing architecture [EB/OL]. (2018-07) [2024-09-30]. RFC 8402. ⏎
22. Spiegel M, Brockners F, Bhandari S et al. In-situ OAM raw data export with IPFIX [EB/OL]. (2024-08-15) [2024-09-30]. Draft-spiegel- ippm-ioam-rawexport-07. ⏎
23. Brockners F, Bhandari S, Mizrahi T. Data fields for in situ operations, administration, and maintenance (IOAM) [EB/OL]. (2022-05) [2024-09-30]. RFC 9197. ⏎
24. Weis B, Brockners F, Hill C et al. EtherType protocol identification of in-situ OAM data [EB/OL]. (2022-02-21) [2024-09-30]. Draft-weis-ippm-ioam-eth-05. ⏎

25. Bhandari S, Brockners F. IPv6 options for in situ operations, administration, and maintenance (IOAM) [EB/OL]. (2023-09) [2024-09-30]. RFC 9486. ⏎
26. Brockners F, Bhandari S, Govindan V et al. VXLAN-GPE encapsulation for in-situ OAM data [EB/OL]. (2020-05-07) [2024-09-30]. Draft-brockners-ippm-ioam-vxlan-gpe-03. ⏎
27. Brockners F, Bhandari S, Govindan V et al. Geneve encapsulation for in-situ OAM data [EB/OL]. (2021-05-23) [2024-09-30]. Draft-brockners-ippm-ioam-geneve-05. ⏎
28. Brockners F, Bhandari S. Network service header (NSH) encapsulation for In Situ OAM (IOAM) data [EB/OL]. (2023-08-23) [2024-09-30]. RFC 9452. ⏎
29. Song H, Gafni B, Brockners F et al. In situ operations, administration, and maintenance (IOAM) direct exporting [EB/OL]. (2022-11-15) [2024-09-30]. RFC 9326. ⏎
30. Harrington D, Presuhn R, Wijnen B. An architecture for describing simple network management protocol (SNMP) management frameworks [EB/OL]. (2002-12) [2024-09-30]. RFC 3411. ⏎
31. Li Z, Dong J, Zhang Y, etc. IPv6 network slicing: offering new experience for industries. *People's Posts and Telecommunications Press*. (2023-06) [2024-09-30]. ⏎

CHAPTER 6

IPv6 On-Path Telemetry Controller

DOI: 10.1201/9781003677901-8

In addition to processing and displaying telemetry information, the IPv6 on-path telemetry controller enables fine-grained management and optimized use of network resources by adjusting network behavior based on predefined policies and rules. This chapter describes the architecture and core functions of the IPv6 on-path telemetry controller, its IP Network Digital Map technology (implemented based on functions such as on-path telemetry), and its northbound and southbound interface capabilities. Specifically, it comprehensively explains the key technical principles and working processes of the IPv6 on-path telemetry controller.

6.1 CONTROLLER ARCHITECTURE

6.1.1 Typical Architecture of a Network Controller

Traditional network O&M mainly depends on the NMS for managing faults, configurations, accounting, performance, and security. Thanks to the transition from SNMP to telemetry in network monitoring technology, the NMS and network devices can now obtain monitoring data proactively instead of passively. Detecting service loss in a passive manner cannot achieve multi-service transport or multi-capability convergence required by today's IP networks. Consequently, attention is shifting toward a new system that integrates management, control, and analysis capabilities, one that is gradually becoming the mainstream approach for future network O&M.

In this section, "network controller" refers to such an integrated management, control, and analysis system. This section explores its design, first discussing the challenges and opportunities it faces, and then describing its overall architecture and key functions.

6.1.1.1 Opportunities and Challenges Faced by Network Controllers

With 5G rapidly developing and the cloud era unfolding, new business models are emerging one after the other, and enterprises are beginning to embrace cloudification and digitalization. As an enabler of digital and cloud transformation in various industries, the telecom industry faces many new opportunities and challenges brought about by massive connections, all-cloud, and the intelligence of everything.

Opportunities:

- Emergence of new business models: 5G and cloud technologies drive the development of new business models, providing enterprises with more opportunities for innovation.
- Enhanced service capabilities: With intelligent technologies, the telecom industry can offer differentiated products and services, enhancing customer satisfaction and market competitiveness.
- Improved operational efficiency: Automation and intelligent technologies help reduce operational costs and improve network resource utilization.

Challenges:

- Increasingly complex networks: 5G doubles or even triples the density of base stations in comparison with 4th Generation of Mobile Communication Technology (4G), requiring many sites to be built or reconstructed. In addition, complex cross-domain and cross- network service scenarios lead to high planning and deployment costs.
- High O&M costs: The costs associated with site visits remain stubbornly high. And the manual efforts and independent tools required in analysis and decision-making not only add to the costs but also increase the risk of errors.
- Difficult business monetization: As enterprises undergo cloudification, carriers lack the capability to deliver differentiated products and services, therefore having difficulty in effectively monetizing SLAs. In addition, service provisioning takes too long due to a low level of automation.

Facing the preceding challenges, carriers must fundamentally change the way they operate networks and solve network O&M issues through architectural innovation rather than just tinkering with the traditional architecture.

To achieve this, carriers must transform their networks toward automation and intelligence. Autonomous Networks has become an industry consensus and is developing quickly. In 2021, nine standards organizations jointly promoted the industry's recognition of the vision, target architecture, and level standards of Autonomous Networks with carriers and vendors through the TeleManagement (TM) Forum Multiple Standards Developing Organization (M-SDO) platform, leveraging industry standards, white papers, Autonomous Network summits, Catalyst projects, and other means. These organizations include the TM Forum, CCSA, Global System for Mobile Communications Association (GSMA), 3rd Generation Partnership Project (3GPP), IETF, and ETSI.

Autonomous Driving Network (ADN) is a solution developed by Huawei for the Autonomous Networks industry. It aims to leverage connectivity and intelligence to build a self-fulfilling, self-healing, and self-optimizing autonomous network. ADN enables single-domain autonomy and cross-domain collaboration to develop self-configuration, self-healing, and self-optimizing network capabilities for carriers and enterprises, delivering a zero-wait, zero-touch, and zero-fault experience to consumers, public sectors, and enterprises. Under the guidance of the TM Forum's Autonomous Networks Framework

(ANF), Huawei has made significant strides in multiple key technologies, including converged sensing, digital twin, intelligent decision-making, and human–machine symbiosis, to build a high-level Autonomous Networks foundation and expedite the evolution toward higher-level Autonomous Networks.

ADN consists of three layers: smart devices, management and control platform, and cloud-based applications. It transforms the existing network architecture in the following aspects:

- Replaces inefficient and repetitive work with automated processes to handle the heavy O&M workload caused by massive connections and large network scale, significantly reducing the network construction and service provisioning time.
- Shifts from complaint-driven O&M to proactive O&M (proactively identifying and resolving problems and then notifying customers), and achieves predictive O&M through in-depth analysis of massive data.
- Leverages the advantages of Artificial Intelligence (AI) machine learning and massive data to provide assistance or even make decisions under human supervision, significantly improving response speed, resource efficiency, and energy efficiency for network services.
- Shares data throughout the planning, construction, maintenance, and optimization phases to achieve closed-loop autonomy. Traditionally, these four phases are relatively independent, with interaction between the upstream and downstream parties relying on processes and manual handoffs. In ADN, SLAs covering network performance, service provisioning time, fault recovery time, network lifecycle, and other aspects are defined in the network planning phase and automatically fulfilled in the construction, maintenance, and optimization phases. This enables business innovation for differentiated network services through guaranteed network and service experience.

In the three-layer architecture, the network controller functions as the management and control platform. The network controller can generate a comprehensive, high-precision network digital map specific to domains by associating discrete network resources, services, status data, and more through massive network data collection and digital network modeling. This enables integrated network data collection, network awareness, network decision-making, and network control.

6.1.1.2 Key Functions of the Network Controller

The network controller provides four types of functions — planning, construction, maintenance, and optimization — to deliver closed-loop operation of network services throughout the lifecycle, as shown in Figure 6.1.

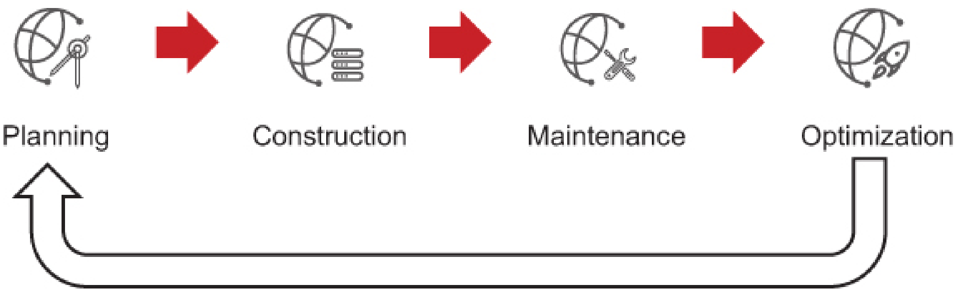


FIGURE 6.1 Key functions of the network controller. ⏎

The four types of functions are described as follows:

- Planning: Design network capabilities based on the network capacity and service characteristics (e.g., design schemes for IP address allocation and IGP domain division).
- Construction: Deploy networks based on construction intents (e.g., configure devices to provide network services and deploy private line services).
- Maintenance: Monitor network status and quickly respond to and rectify faults or errors.
- Optimization: Optimize and adjust a network based on the analysis results of network status, traffic, and quality to ensure that the network runs as expected.

To provide these functions, a network controller typically requires five full-lifecycle service modules: business service, management service, control service, analysis service, and platform service. Figure 6.2 shows a typical network controller architecture consisting of these modules.

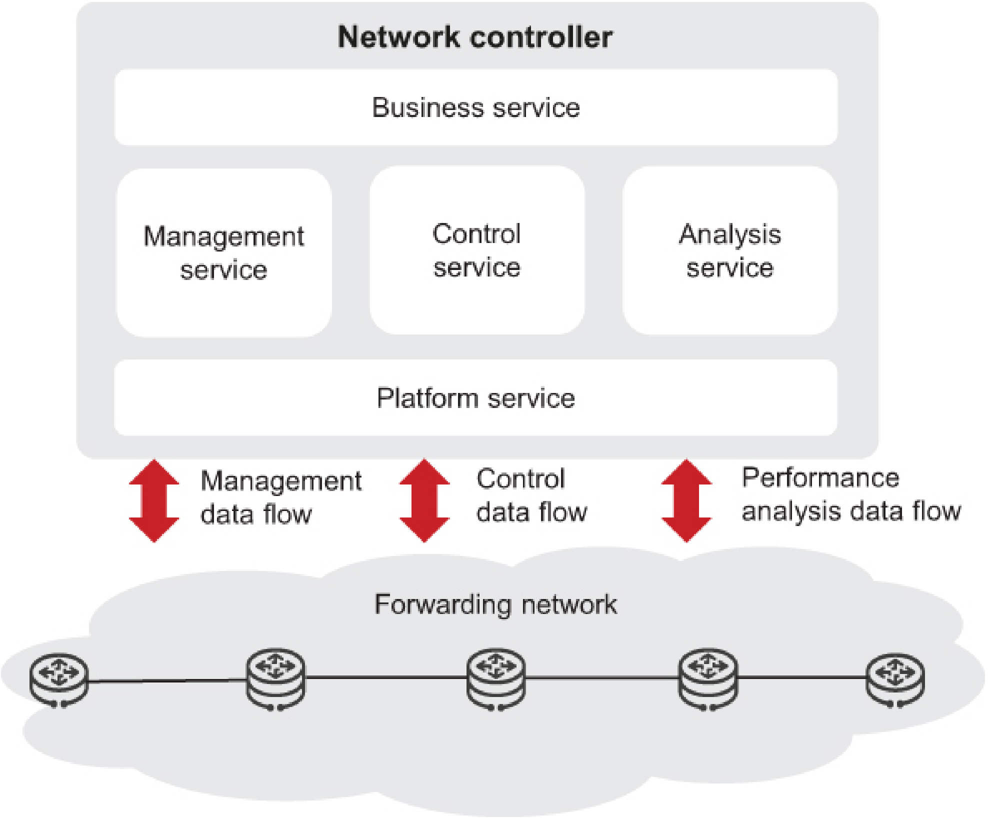


FIGURE 6.2 Typical network controller architecture. ⏎

The five modules are described as follows:

- Business service: It receives service requests from higher-level systems and orchestrates the management, control, and analysis service modules to execute corresponding functions based on service characteristics. It also evaluates the execution results of these functions and directs subsequent function execution to meet service requirements.
- Management service: It provides network management capabilities, including the Fault, Configuration, Accounting, Performance, and Security (FCAPS) capabilities available on a traditional NMS. It also enables external systems to obtain information such as NE data, network data, topology information, and physical inventory data through Application Program Interfaces (APIs). Furthermore, this module provides intent-level interfaces, converts user intents into network functions and deploys them, and maintains intent states after the network topology or configuration changes.
- Control service: It functions as the network-level control plane and works with the control planes of devices to control traffic.
- Analysis service: It displays the network states, predicts network behaviors, analyzes network faults, and provides operation suggestions for fault rectification based on the network performance and running data obtained by various network probes using big data and AI technologies. For example, it can collect network traffic information through telemetry, predict traffic peaks and troughs, and trigger network traffic

optimization. This module can also facilitate fault rectification by determining the root cause of faults based on information such as alarms, service paths, and service traffic.

- Platform service: It provides unified service governance, security, user management, log, and alarm capabilities, and unified data services to enable data flow among the management, control, and analysis service modules. This universal platform for application services associates itself with the control service module to optimize service objects created by the management service module and associates itself with the analysis service module to collect and analyze the performance of these objects.

The preceding service modules can be deployed individually or in combination with each other, meeting the personalized requirements of users.

6.1.2 Architecture of the IPv6 On-Path Telemetry Controller

The IPv6 on-path telemetry controller consists of the configuration, collection, analysis, and visualization units, as shown in Figure 6.3. The configuration and collection units respectively map to the management and platform services in a typical controller, and the analysis and visualization units map to the analysis service.

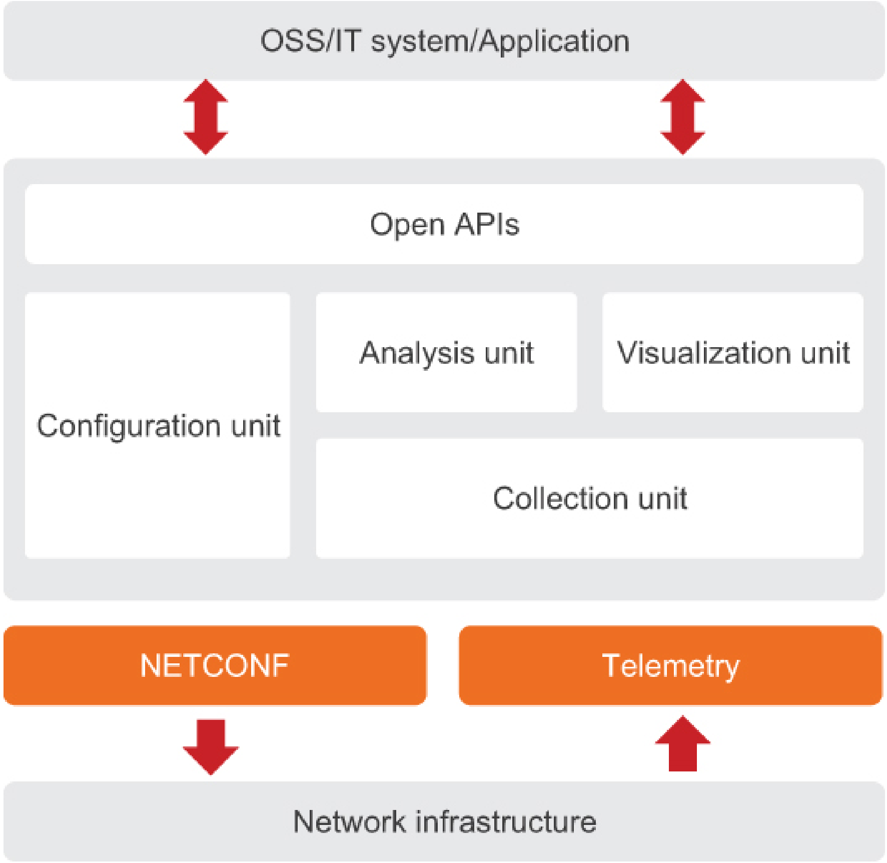


FIGURE 6.3 Components of the IPv6 on-path telemetry controller. ⏎

In the northbound direction, the IPv6 on-path telemetry controller connects to the Operational Support System (OSS)/Information Technology (IT) system/application layer, which serves as a platform for carriers to achieve digital operations transformation. This layer includes the service orchestrator, big data analytics- and intelligence-based policy producer, and self-service e-commerce portal in addition to the traditional OSS. It provides functions such as presentation of network infrastructure resources, presentation of service paths, and management of service policies for E2E network-wide operations. Carriers can monetize their network infrastructure by leveraging this layer to provide users with application services, including traditional services such as broadband, video, and Business to Business (B2B) private lines, and emerging services such as cloud computing and vertical industry Internet of Things (IoT).

In the southbound direction, the IPv6 on-path telemetry controller connects to the network infrastructure layer, which is a physical network that consists of IP devices and provides basic communication connection services. The infrastructure layer of a cloud-based network is a constantly evolving and ubiquitously connected network that consists of both traditional and SDN networks, and provides high-bandwidth and low-delay communication services. The controller abstracts managed devices into network resource pools based on a unified abstraction model so that business intents can be implemented.

Table 6.1 describes the functions of the units that comprise the IPv6 on-path telemetry controller.

TABLE 6.1 Units of the IPv6 On-Path Telemetry Controller and Their Functions ⏎

Unit	**Function**
Configuration unit	This unit enables the deployment of on-path telemetry monitoring instances for measuring network objects. It is divided into two sub-modules: on-path telemetry instance configuration management and hop-by-hop policy configuration management. • On-path telemetry instance configuration management: allows users to enable on-path measurement and add, delete, modify, and query service-, tunnel-, and application-level monitoring instances. • Hop-by-hop policy configuration management: defines criteria to determine poor-QoE events for monitored objects and selects a policy to switch from the E2E mode to the hop-by-hop mode following a poor-QoE event.
Collection unit	This unit collects through telemetry the real-time performance data (e.g., number of packets, number of bytes, and timestamps) of monitored objects in on-path telemetry. It also parses and converts the data (including the data collection NEs, data objects, data values, and data time) into a unified format for the analysis unit to use. The collection unit mainly provides the following functions: • Telemetry Configuration management: deploys key information, such as the collector IP address, telemetry performance data collection protocol, and collected service indicators related to on-path telemetry. • Data collection: collects data from NEs and parses and processes it.

Unit	Function
Analysis unit	This unit provides multidimensional analysis of the service running status. Using the on-path telemetry data obtained from the collection unit, it calculates the SLA data (e.g., packet loss rate, delay, and traffic rate) of the objects measured by on-path telemetry instances managed by the configuration unit.
Visualization unit	This unit provides GUIs for real service path restoration, multidimensional service running status analysis, and running status display based on the results obtained from the analysis unit.

6.2 IPV6 ON-PATH TELEMETRY CONTROLLER

As mentioned in Section 6.1.2, the basic functions of the IPv6 on-path telemetry controller are provided by the configuration, collection, analysis, and visualization units. The following describes the key techniques and interaction processes involved in implementing the related functions of these units.

6.2.1 Configuration Unit

The configuration unit enables configuration management for on-path telemetry instances and hop-by-hop policies.

- Configuration management for on-path telemetry instances: provides different functions based on configuration objects, as described in Table 6.2, for the measurement of VPN services, SRv6 Policies, and application traffic.

TABLE 6.2 Configuration Management Functions for On-Path Telemetry Instances ⏎

Function	Description
Global configuration management	Performs NE-level configuration, such as enabling IFIT, port aggregation, and clock policies.
Custom flow management	Monitors flows by VPN and IP 5-tuple information at the application level.
Auto-identified flow management	Automatically learns the to-be-monitored flows by VPN and UNI.
Whitelist management	Works with auto-identified flow management to define a whitelist for learning the IP addresses of flows that pass through UNIs and match configured rules. The whitelist takes effect globally on a specified device.

Function	Description
VPN monitoring management	Monitors VPN services by VPN and next-hop locator.
Tunnel monitoring management	Monitors SRv6 Policies.

- Configuration management for hop-by-hop policies: sets packet loss, delay, and jitter thresholds and applies them to network objects in batches. If the SLA data of a network object exceeds a threshold, the controller can automatically switch the corresponding on-path telemetry instance from the E2E mode to the hop-by-hop mode in order to demarcate and locate poor-QoE issues.

6.2.2 Collection Unit

In general, IPv6 on-path telemetry information is reported through telemetry. The controller creates an on-path telemetry instance and delivers instance configurations to devices for data collection. The collection process involves two steps: configuration management and data collection.

- Configuration management: After the types of data and the devices from which they are to be collected are selected on the GUI, the controller delivers the destination IP address, protocol type, port number, and other information used by the collector to these devices and specifies the objects to be collected. In an on-path telemetry instance, either the E2E or hop-by-hop mode can be specified for statistics collection.
- Data collection: Devices collect telemetry data according to the specified configurations and then encapsulate it into telemetry packets for reporting. Such data includes flow IDs, flow characteristics, and flow statistics (e.g., number of packets and timestamp). The controller receives and stores the data collected from monitored objects for analysis and visualization purposes.

6.2.3 Analysis and Visualization Unit

In addition to collecting and analyzing network quality data in real time, the controller can also determine network status in real time and accurately demarcate and locate faults by using test methods such as ping and traceroute. It can also ensure deterministic and service-level SLA assurance for IP transport services by combining various techniques for measuring network performance with other techniques such as ring discovery, traffic suppression, and path restoration.

The following uses Huawei controller iMaster NCE-IP as an example to describe real-time service SLA visualization, proactive mass fault O&M, and efficient fault demarcation and locating provided by the IFIT Alternate Marking (IFIT-AM) solution:

- Real-time service SLA visualization

  For mobile transport services, an overview of network-wide base stations is displayed, as shown in Figure 6.4.

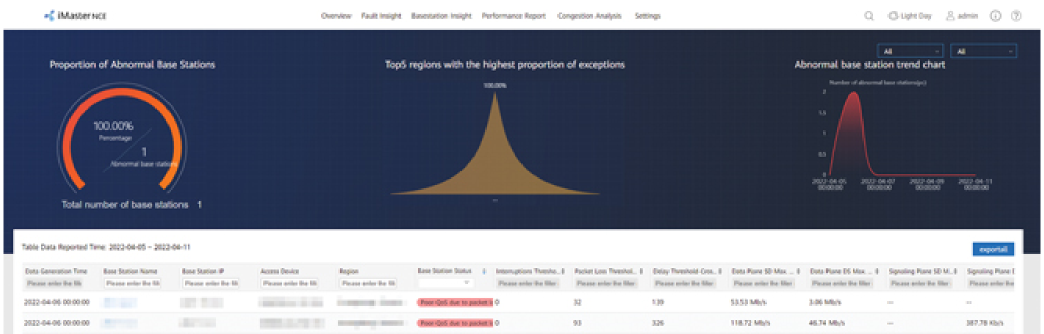
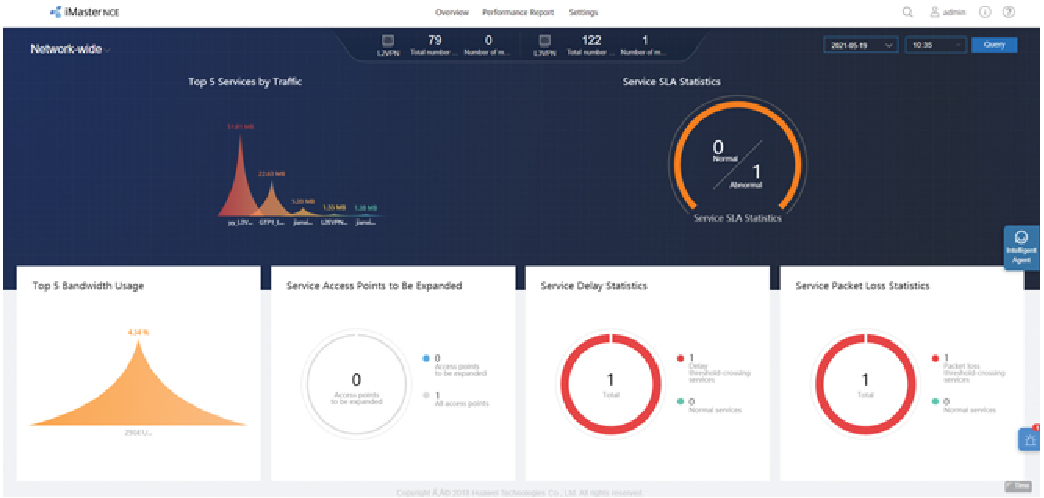


Table Data Reported Time: 2022-04-05 ~ 2022-04-11

Data Generation Time	Base Station Name	Base Station IP	Access Device	Region	Base Station Status	Interruptions Thresho...	Packet Loss Threshol...	Delay Threshold-Cros...	Data Plane SD Max ...	Data Plane DS Max ...	Signaling Plane SD M...	Signaling Plane I
Please enter the filt	Please enter the filt	Please enter the filt	Please enter the filt	Please enter the filt		Please enter the filter	Please enter the filter	Please enter the filter	Please enter the filter	Please enter the filter	Please enter the filter	Please enter the
2022-04-06 00:00:00	[illegible]	[illegible]	[illegible]	[illegible]	Poor QoS due to packet l	0	32	139	53.53 Mb/s	3.06 Mb/s	--	--
2022-04-06 00:00:00	[illegible]	[illegible]	[illegible]	[illegible]	Poor QoS due to packet l	0	93	326	118.72 Mb/s	46.74 Mb/s	--	387.78 Kb/s

FIGURE 6.4 Overview of base stations. ⏎

  For VPN private line services, an overview of VPNs — covering the number of monitored private lines, number of poor-QoE private lines (including those affected by packet loss and delay), and more — is displayed, as shown in Figure 6.5.



Copyright Â,Â© 2018 Huawei Technologies Co., Ltd. All rights reserved.

FIGURE 6.5 Overview of VPNs. ⏎

  For specific private line services, an overview of VPN connections is displayed, as shown in Figure 6.6.

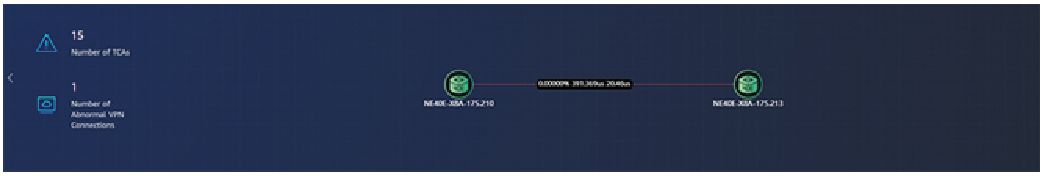

15
Number of TCAs
1
Number of Abnormal VPN Connections
NE40E-X8A-175.210
0.00000% 391.369us 20.46us
NE40E-X8A-175.213

FIGURE 6.6 Overview of VPN connections. ⏎

- Proactive mass fault O&M

- AI-based fault clustering: IFIT-AM is used to restore the paths of flows with packet loss or delay that exceed the threshold, and an AI algorithm is used to calculate the common paths shared by these threshold-crossing flows to cluster common faults into a single one.
- Fast demarcation and locating: IFIT-AM hop-by-hop measurement is used for threshold-crossing flows. Potential root causes are identified through demarcation and locating based on NE KPI analysis results. Then, rectification suggestions are provided, allowing faults to be proactively rectified before users report issues.

Figure 6.7 provides an example of a hop-by-hop path restoration scenario.

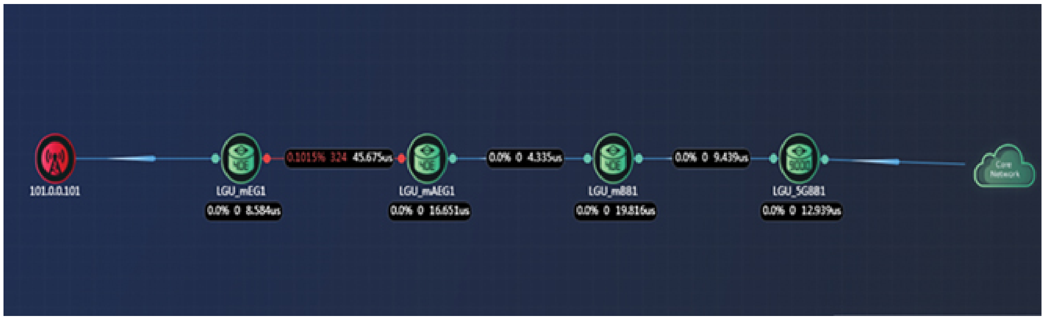
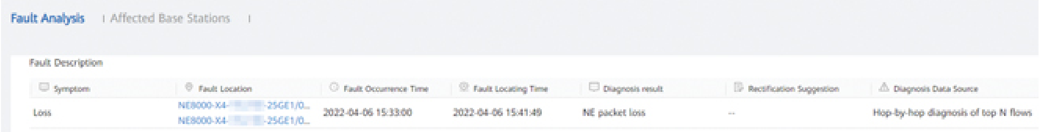


FIGURE 6.7 Hop-by-hop path restoration. ⏎

- Efficient fault demarcation and locating
  - Fault analysis summary: Fault analysis results are summarized by base station, and faults on the wireless and transport networks can be clearly demarcated. Faults on the transport network can be quickly rectified based on their locations, diagnostic results, rectification suggestions, and more, as shown in Figure 6.8.



Fault Analysis | Affected Base Stations |

Fault Description

Symptom	Fault Location	Fault Occurrence Time	Fault Locating Time	Diagnosis result	Rectification Suggestion	Diagnosis Data Source
Loss	NE8000-X4-[redacted]-25GE1/0... NE8000-X4-[redacted]-25GE1/0...	2022-04-06 15:33:00	2022-04-06 15:41:49	NE packet loss	--	Hop-by-hop diagnosis of top N flows

FIGURE 6.8 Fault analysis. ⏎

  - Historical fault playback: Historical fault details, diagnostic results, and hop-by-hop SLA information of all service flows over the last 7 days can be played back to facilitate troubleshooting, as shown in Figure 6.9.

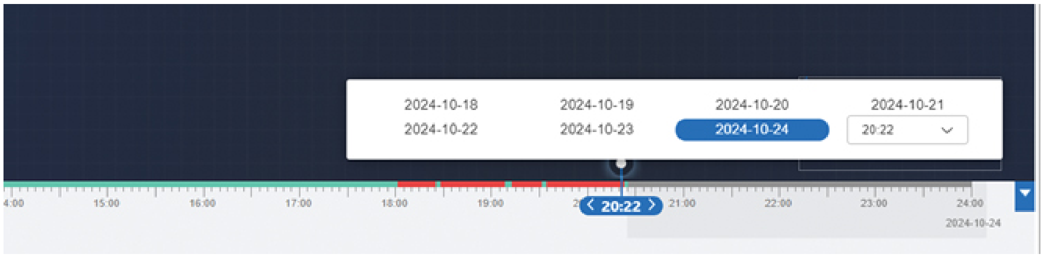


FIGURE 6.9 Historical fault playback. ⏎

6.3 IP NETWORK DIGITAL MAP

The introduction of IPv6 on-path telemetry has significantly enhanced network O&M visualization capabilities, enabling more accurate and efficient perception of SLAs. This, in turn, allows the network to deliver agile and differentiated SLA guarantees. To address the diverse SLA requirements across industries driven by digital transformation, Huawei has developed IP Network Digital Map technology, leveraging comprehensive network state data.

IP Network Digital Map provided by Huawei iMaster NCE-IP serves as a digital foundation for IP network intelligent O&M, as shown in Figure 6.10. It functions like an intelligent, automated navigation map (e.g., Google Maps), offering holographic network visualization, agile service provisioning, automatic traffic optimization, intelligent fault analysis, and eco-friendly operations. This helps to continuously guarantee service SLAs and improve service experience.

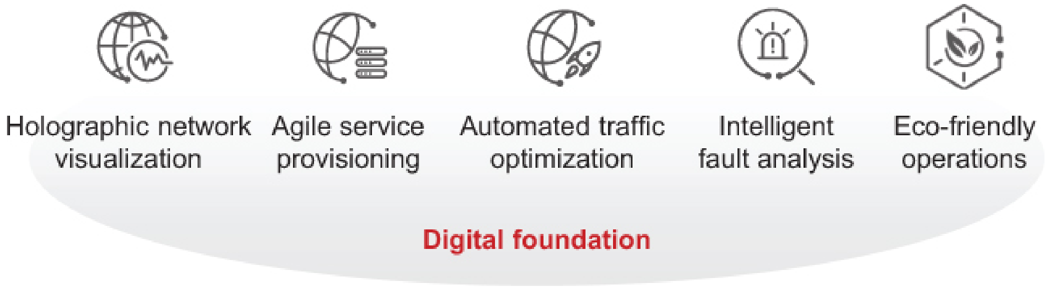

Holographic network visualization
Agile service provisioning
Automated traffic optimization
Intelligent fault analysis
Eco-friendly operations
Digital foundation

FIGURE 6.10 Network digital map. ⏎

6.3.1 Basic Capabilities of IP Network Digital Map

IP Network Digital Map collects data about physical resources, slices, tunnels, routes, VPN services, applications, and more on multi-vendor devices in real time through standard protocols such as BGP-LS, BMP, SNMP, telemetry, and NETCONF. It also collects and presents multidimensional indicators (e.g., delay, bandwidth, packet loss, and energy consumption) for ultra-large networks in real time via a distributed network performance collection framework. This enables users to gain a clear understanding of the entire network and identify service issues.

IP Network Digital Map supports multi-layer visualization, as shown in Figure 6.11.

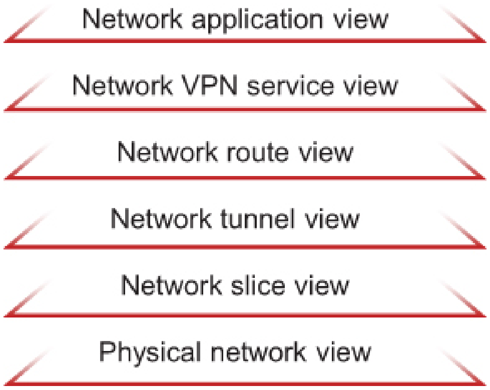


FIGURE 6.11 Multi-layer visualization provided by IP Network Digital Map. ⏎

These visualization layers are described as follows:

- Physical network view: This view displays physical network topologies based on the Geographic Information System (GIS) coordinates and physical connections of NEs. It automatically lays out topologies and supports zooming in and out on multi-level topologies. Furthermore, it displays the statuses and alarms of NEs and Layer 2 links on topologies.
- Network slice view: This view displays the topologies of network slices. If a specific slice is selected, it can display the statuses of all NEs and links on the slice.
- Network tunnel view: This view displays the statuses and paths of SR-TE, SR Policy, and other types of tunnels on physical or slice networks. It also supports functions such as delay circle rendering, path precomputation, and optimization history playback.
- Network route view: This view displays IGP/BGP route information, including route prefix information.
- Network VPN service view: This view displays basic information, peer connections, and forwarding paths about E2E VPN services.
- Network application view: This view provides network traffic composition analysis and visualization functions and supports optimization and scheduling of network traffic for applications.

IP Network Digital Map supports multidimensional network data and can be customized to display various types of data, including:

- State data: The status of each link is displayed on the topology. When a fault occurs on the network, the IGP running on devices immediately performs convergence and updates and floods the updated link state, which is then reported through BGP-LS to the controller for display. After detecting the link fault, the controller sets the link to the down state.
- Bandwidth data: The bandwidth usage of each link is displayed on the topology. Different bandwidth usage ranges can be set for links, with links colored accordingly on the topology.

- Delay data: The delay of each link is displayed on the topology. Different delay ranges can be set for links, with links colored accordingly on the topology.
- Cost data: The TE metric of each link is displayed on the topology. Different TE metric ranges can be set for links, with links colored accordingly on the topology.
- Packet loss rate data: The packet loss rate of each link is displayed.
- Energy data: The energy efficiency and real-time power consumption of NEs are displayed.
- Availability data: Link availability is displayed based on automatic evaluation of link fault frequency, reflecting link stability.

6.3.2 Value-Added Capabilities of IP Network Digital Map

IP Network Digital Map offers the following value-added capabilities based on holographic visualization:

- Navigation-like path computation: IP Network Digital Map uses the intelligent cloud-map algorithm to compute optimal paths matching service intents within seconds based on a random combination of more than 20 factors. It can also detect poor service quality within seconds, locate root causes within minutes, and complete automatic optimization within minutes to meet differentiated SLA assurance requirements.
- IP traffic scheduling: IP Network Digital Map can detect congestion on IP networks in real time and, through BGP Flow Specification (FlowSpec), can automatically optimize and adjust traffic paths to deliver minute-level SLA closed-loop assurance. This is a significant improvement over most IP networks today, as their use of best-effort forwarding makes them prone to congestion during traffic surges, and manual traffic balancing results in poor user experience because it takes an average of 3 hours to implement.
- Highly stable IP network: IP Network Digital Map provides intent verification and BGP route analysis functions to proactively detect incorrect network changes and intercept potential major network accidents. This helps carriers build secure, reliable, and highly stable IP networks.
- Eco-friendly operations: IP Network Digital Map provides an energy analysis function that allows energy consumption on the entire network to be viewed, managed, and optimized.

The following describes IP Network Digital Map's value-added applications that are closely related to IPv6 on-path telemetry in terms of planning, construction, maintenance, and optimization.

6.3.2.1 Planning Phase: Network Configuration Verification

Carriers must be cautious about changing configurations on IP networks, which often carry many cross-city, cross-province, and even cross- country data services — any errors introduced by those changes can lead to huge losses. A report[1] published by the TM Forum reveals that 43% of carriers believe that their service capabilities are severely hampered by manual configuration. Carriers therefore pay a great deal of attention to configuration and want an online configuration verification tool that can assess and verify the impact of network configurations before deployment and intercept incorrect configurations.

As shown in Figure 6.12, IP Network Digital Map provides network configuration verification to meet such requirements. Here, UPE stands for User-End Provider Edge, and ASBR stands for Autonomous System Boundary Router.

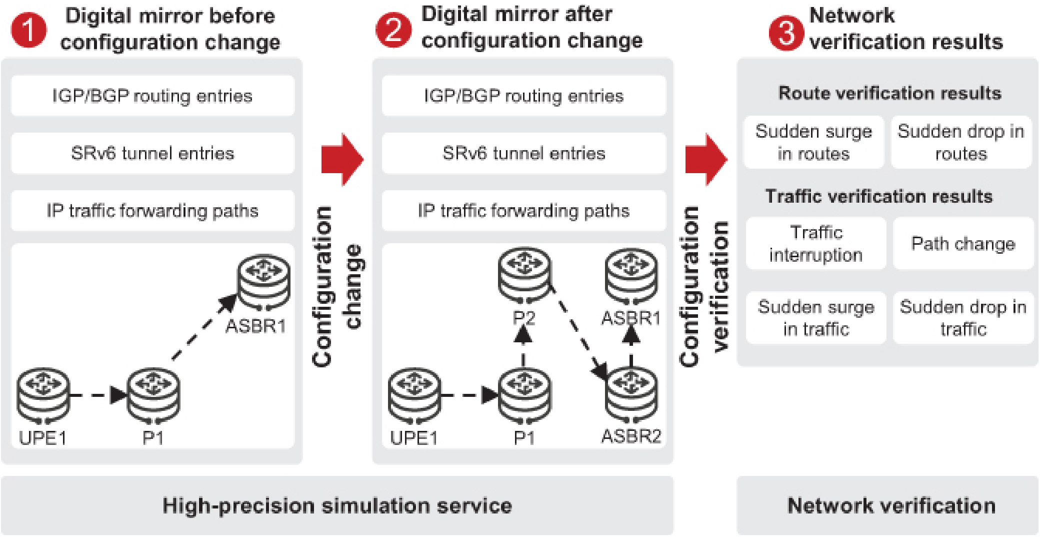

1 Digital mirror before configuration change
IGP/BGP routing entries
SRv6 tunnel entries
IP traffic forwarding paths
ASBR1
UPE1
P1
Configuration change
2 Digital mirror after configuration change
IGP/BGP routing entries
SRv6 tunnel entries
IP traffic forwarding paths
P2
ASBR1
UPE1
P1
ASBR2
Configuration verification
3 Network verification results
Route verification results
Sudden surge in routes
Sudden drop in routes
Traffic verification results
Traffic interruption
Path change
Sudden surge in traffic
Sudden drop in traffic
High-precision simulation service
Network verification

FIGURE 6.12 Network configuration verification. ⏎

The key points involved in the configuration verification process are as follows:

- High-precision simulation: IP Network Digital Map simulates not only the states and behaviors of network protocols and traffic, but also the routing and forwarding tables of network devices, using configuration changes, interconnecting routes, and traffic of network devices as inputs. This provides an authentic and objective basis for assessing the risks associated with network changes.
- Network verification: IP Network Digital Map evaluates network risks through Control Plane Verification (CPV) and Data Plane Verification (DPV) based on the routing tables, forwarding tables, and traffic loads of devices. CPV verifies changes in the number of control plane routes (sudden surges and drops), route reachability, route reliability, and other aspects, whereas DPV verifies network forwarding plane paths. These two techniques complement each other to identify potential risks in network configuration changes, thereby intercepting incorrect configurations.

6.3.2.2 Construction Phase: Service Automation

Carriers and enterprises often deploy devices from multiple vendors on their networks. Quickly adapting to multi-vendor devices and rolling out new services are the core competitive edges of service automation. However, there are challenges involved in adapting to new devices and rolling out new services.

- Adapting to new devices efficiently depends on the capabilities and response speeds of vendors. This results in slow device integration, low automation, and long provisioning periods, which are bottlenecks in E2E service delivery.
- Rolling out new services depends on the updates of OSS and controller versions, creating problems such as insufficient API integration and high customization costs. These problems increase the time taken to roll out new services, meaning that service rollout cannot keep pace with the dynamic changes involved in service scenarios.

IP Network Digital Map addresses these challenges and enables fast service provisioning on multi-vendor networks through a high-performance and high-reliability automation engine. It increases the device adaptation efficiency by about 90%, slashing the device adaptation and management periods from months to days, and speeds up the rollout of new services by about 80%, cutting the time taken from 6 to 9 months to just 1 month.

The automation engine, shown in Figure 6.13, consists of design time and run time. The design time establishes mappings between service YANG models and device YANG models, and the run time uses these mappings to provision services. Specifically, the automation engine writes Specific Service Plugin (SSP) and Specific NE Driver (SND) packages at design time and loads software packages at run time, enabling fast management of new devices and building of new services.

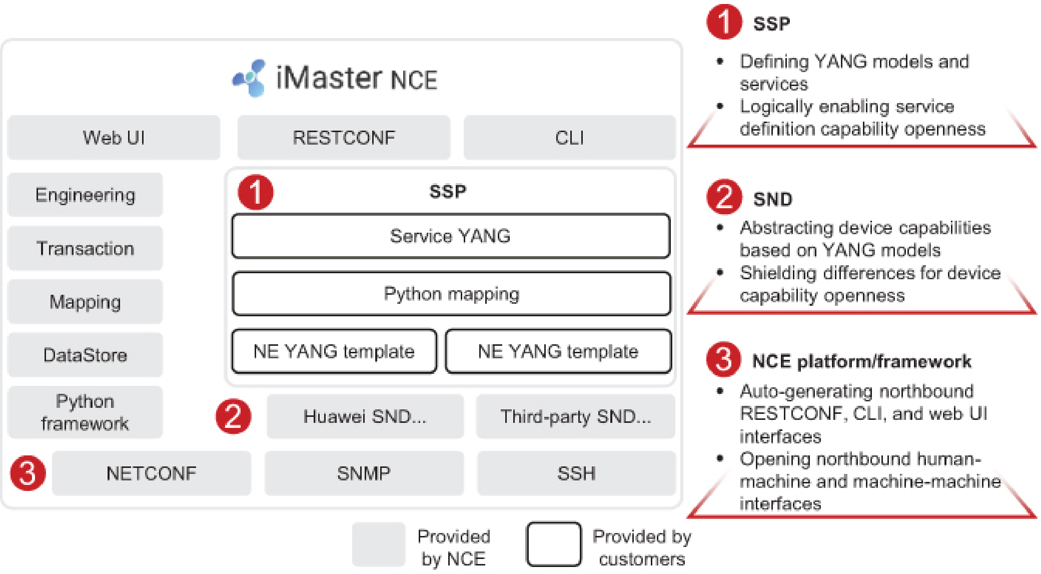

iMaster NCE
Web UI
RESTCONF
CLI
Engineering
Transaction
Mapping
DataStore
Python framework
1
SSP
Service YANG
Python mapping
NE YANG template
NE YANG template
2
Huawei SND...
Third-party SND...
3
NETCONF
SNMP
SSH
Provided by NCE
Provided by customers
1 SSP
Defining YANG models and services
Logically enabling service definition capability openness
2 SND
Abstracting device capabilities based on YANG models
Shielding differences for device capability openness
3 NCE platform/framework
Auto-generating northbound RESTCONF, CLI, and web UI interfaces
Opening northbound human-machine and machine-machine interfaces

FIGURE 6.13 Automation engine. ⏎

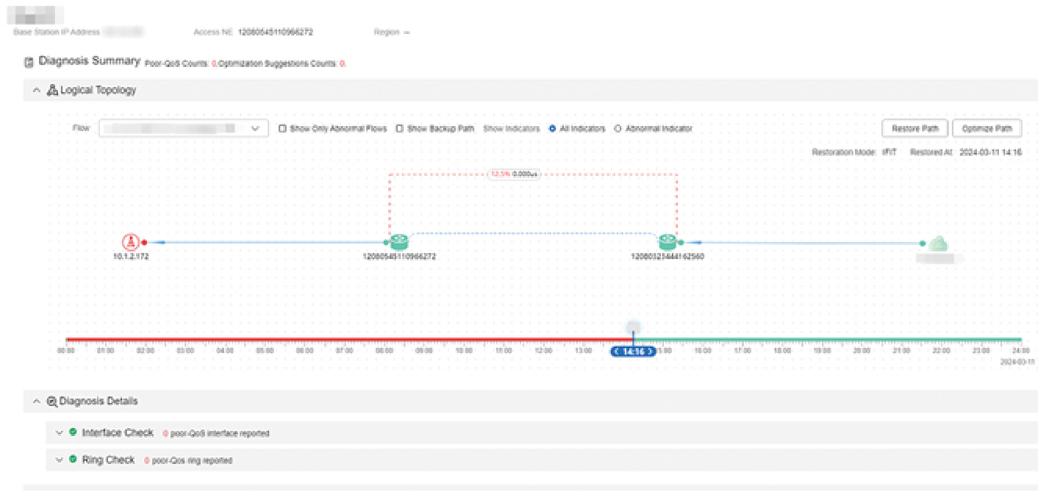
- SSP package: provides the data models required for completing a set of network-level service configurations.
- SND package: provides data models for interacting with network devices. These models contain YANG files that define device information, such as device types, vendors, connections, and features. By loading SND packages, IP Network Digital Map can establish connections with devices, query data, and deliver configurations to manage devices.

6.3.2.3 Maintenance Phase: Intelligent Analysis

Holographic visualization facilitates intelligent analysis, which plays an important role in network maintenance. This section focuses only on intelligent network congestion analysis and fault analysis.

- Intelligent network congestion analysis

  As the packet loss rate increases to a certain level on an IP transport network, the TCP throughput sharply decreases, significantly affecting user experience. The packet loss rate is therefore an important indicator reflecting the transport network performance. Intelligent network congestion analysis focuses on visualizing and troubleshooting base station traffic suppression, offering a complete O&M solution that visualizes network-wide E2E traffic suppression and regional poor-QoE status in addition to quickly demarcating and locating faults.
- Visualization of network-wide E2E traffic suppression: displays the distribution of traffic suppression on a heatmap, visualizes network congestion points, prioritizes high-value areas, drills down to problematic base stations, and provides precise capacity expansion suggestions. Specifically, intelligent network congestion analysis measures base station SLAs through TWAMP and IFIT. It then calculates the suppressed traffic of a base station based on the collected packet loss rate and actual traffic, and determines whether the base station has poor-QoE issues. Based on this analysis, it visualizes E2E traffic suppression on the entire network and displays suppressed traffic in each region.
- Visualization of regional poor-QoE status: enables minute-level automatic troubleshooting and efficient network O&M by leveraging service SLA visualization, including E2E service quality visualization, service path restoration, and hop-by-hop path SLA visualization.
- Fast fault demarcation and locating: displays the packet loss of base station services, restores the hop-by-hop path for a single base station service based on different time points, and analyzes the impact of link SLAs and bandwidth on packet loss, enabling accurate fault demarcation and locating. Figure 6.14 provides an example showing a diagnosis summary.



Base Station IP Address Access NE: 12080545110966272 Region: --

Diagnosis Summary Poor-QoS Counts: 0 Optimization Suggestions Counts: 0

Logical Topology

Flow: ☐ Show Only Abnormal Flows ☐ Show Backup Path Show Indicators ● All Indicators ○ Abnormal Indicator

Restoration Mode: iFIT Restored At: 2024-03-11 14:16

Diagnosis Details

Interface Check 0 poor-QoS interface reported

Ring Check 0 poor-Qos ring reported

FIGURE 6.14 Fault demarcation and locating for base stations. ⏎

- Intelligent fault analysis

  It is currently difficult to clearly perceive hardware, forwarding, and configuration errors on the network, and there are no effective means to locate them after they are perceived. In addition, checking massive alarm data is inefficient and leads to longer service interruptions. To address these issues on IP networks such as 5G transport and intelligent metro networks, IP Network Digital Map builds a fault propagation model based on O&M big data, AI, and expert knowledge, and conducts continuous online self-learning. This reduces O&M costs, improves troubleshooting efficiency, and reduces dependence on experts. Specifically:
- Lower O&M costs: IP Network Digital Map clusters network events and alarms and performs association analysis to reduce redundant alarms, service tickets, and O&M costs.
- Higher troubleshooting efficiency: IP Network Digital Map clusters the events and alarms related to the same fault into one incident through association analysis by time and topology, and identifies the root event. This achieves “one incident, one ticket” and avoids dispatching unnecessary service tickets.
- Less dependence on experts: Event clustering and Root Cause Analysis (RCA) are implemented based on massive O&M data, extensive expert knowledge, and AI algorithms, and work automatically without experts, even for faults that are difficult to handle manually. This helps implement comprehensive and quick troubleshooting.

6.3.2.4 Optimization Phase: Intelligent Network Optimization

IP Network Digital Map also provides automatic optimization capabilities. This section focuses on the value-added capabilities of automatic tunnel path optimization.

Traditional telecom networks provide undifferentiated connection services, resulting in wasted resources on less demanding services and insufficient resources for demanding ones. To address this issue, IPv6 Enhanced networks need to balance resources and quality by providing differentiated network services based on service requirements. Differentiated SLA assurance provides network connections with differentiated bandwidth, delay, and availability for specific services. However, over time, resource imbalance can occur for network services, with some links becoming overloaded while others remaining underutilized. IP Network Digital Map guarantees load balancing and improves network throughput to fulfill SLAs by accurately detecting service quality changes, quickly locating network quality deterioration points, and promptly optimizing service traffic. Figure 6.15 shows the automatic tunnel path adjustment process.

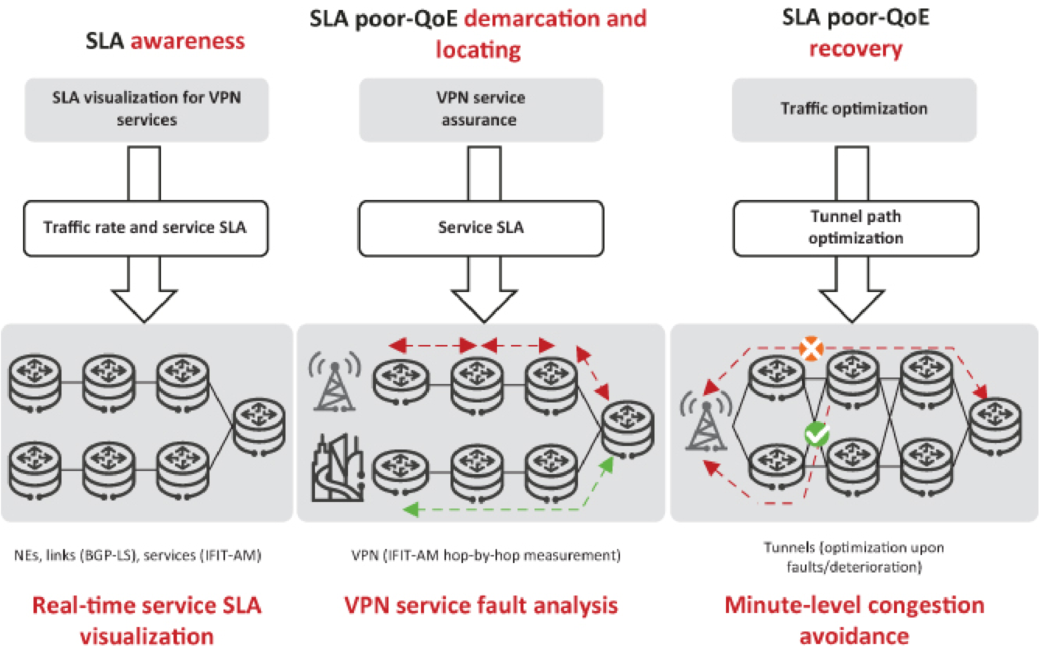

SLA awareness
SLA poor-QoE demarcation and locating
SLA poor-QoE recovery
SLA visualization for VPN services
VPN service assurance
Traffic optimization
Traffic rate and service SLA
Service SLA
Tunnel path optimization
NEs, links (BGP-LS), services (IFIT-AM)
VPN (IFIT-AM hop-by-hop measurement)
Tunnels (optimization upon faults/deterioration)
Real-time service SLA visualization
VPN service fault analysis
Minute-level congestion avoidance

FIGURE 6.15 Automatic tunnel path adjustment. ⏎

- SLA awareness: uses BGP-LS to quickly detect changes in network topology (including faulty NEs and links) and changes in link bandwidth and delay. It also implements in-situ flow measurement through IFIT-AM and reporting through telemetry (within seconds) to accurately measure service SLAs and show network and service quality by layer.
- SLA poor-QoE demarcation and locating: automatically checks each hop through IFIT-AM when service quality deteriorates, identifies faulty points on service forwarding paths, and visualizes demarcation and locating results in the network topology.
- SLA poor-QoE recovery: uses multi-factor cloud-map algorithms to recompute network paths based on SLA deterioration demarcation and locating results. It also re-optimizes network paths using techniques such as SR Policy to bypass the faulty points and ensure service SLAs.

Providing differentiated SLA assurance to different tenants involves providing a tunnel for each service. This results in a hundredfold increase in the number of tunnels. The controller must therefore be able to manage massive numbers of tunnels in order to implement E2E control of network-wide paths. IP Network Digital Map supports millions of tunnels and can compute paths network-wide in minutes, meeting the requirements for managing super-large-scale networks. With just one map, E2E network management can be fully realized.

6.4 EXTERNAL INTERFACES OF THE IPV6 ON-PATH TELEMETRY CONTROLLER

6.4.1 NBIs of the Controller

The controller provides multiple northbound interfaces (NBIs) to quickly interconnect with OSSs. It also supports multiple southbound interfaces (SBIs) to achieve unified management and control over IP devices.

To obtain on-path telemetry SLA data, the controller provides two types of NBIs, Representational State Transfer (REST) and performance text:

- RESTful NBI: provides RESTful query capabilities based on the microservice architecture and uses HTTPS to transmit data.
- Performance text NBI: generates northbound performance text files and integrates with the upper-layer OSS for performance management using File Transfer Protocol (FTP) or Secure File Transfer Protocol (SFTP).

Performance text NBIs function like other NBIs and serve as a bridge between the upper-layer OSS and controller. They enable the upper-layer OSS to obtain the performance data of devices managed by the controller and support the following two file transfer modes:

- Pull mode: The upper-layer OSS functions as an FTP/SFTP client to obtain northbound performance text files from the controller (which functions as the FTP/SFTP server).
- Push mode: The controller functions as an FTP/SFTP client to send northbound performance text files to the user-specified FTP/SFTP server.

IPv6 on-path telemetry provides network topology performance data through the controller’s NBIs. The network topology consists of an underlying network layer (physical network) and an upper VPN service layer (logical network). The YANG data model for network topologies defined in RFC 8345[2] provides standard methods to represent physical and logical network topologies. Based on this model, a performance monitoring model is defined in RFC 9375[3] to monitor and manage the performance of network topologies.

The following provides an excerpt about the YANG model for network and VPN service performance monitoring defined in RFC 9375.

```
augment /nw:networks/nw:network/nt:link:  +--rw perf-mon
 +--rw low-percentile?               percentile
    +--rw intermediate-percentile?    percentile
    +--rw high-percentile?           percentile
    +--rw measurement-interval?       uint32
    +--ro pm* [pm-type]
```

```
|  +--ro pm-type          identityref
|  +--ro pm-attributes
|     +--ro start-time?        yang:date-and-time
|     +--ro end-time?          yang:date-and-time
|     +--ro pm-source?         identityref
|     +--ro one-way-pm-statistics
|     |  +--ro loss-statistics
|     |  |  +--ro packet-loss-count? yang:counter64
|     |  |  +--ro loss-ratio?       percentage
|     |  +--ro delay-statistics
 |     |   |  +--ro unit-value?        identityref
 |  |  |  +--ro min-delay-value?    yang:gauge64
  |  |  |  +--ro max-delay-value?    yang:gauge64
  |  |  |  +--ro low-delay-percentile? yang:gauge64
  |  |  |  +--ro intermediate-delay-percentile?  yang:gauge64
  |  |  |  +--ro high-delay-percentile?
              yang:gauge64
  |  |  +--ro jitter-statistics
  |  |     +--ro unit-value?         identityref
  |  |     +--ro min-jitter-value?   yang:gauge64
  |  |     +--ro max-jitter-value?   yang:gauge64
  |  |     +--ro low-jitter-percentile?
              yang:gauge64
  |  |     +--ro intermediate-jitter-percentile?
               yang:gauge64
  |  |     +--ro high-jitter-percentile?
               yang:gauge64
  |  +--ro one-way-pm-statistics-per-class*
        [class-id]
  |     +--ro class-id               string
  |     +--ro loss-statistics
  |     |  +--ro packet-loss-count? yang:counter64
  |     |  +--ro loss-ratio?    percentage
  |     +--ro delay-statistics
  |     |  +--ro unit-value?         identityref
  |     |  +--ro min-delay-value?    yang:gauge64
  |     |  +--ro max-delay-value?    yang:gauge64
  |     |  +--ro low-delay-percentile?
               yang:gauge64
  |     |  +--ro intermediate-delay-percentile?
               yang:gauge64
  |     |  +--ro high-delay-percentile?
               yang:gauge64
  |     +--ro jitter-statistics
  |     +--ro unit-value?            identityref
  |     +--ro min-jitter-value?      yang:gauge64
  |     +--ro max-jitter-value?      yang:gauge64
  |     +--ro low-jitter-percentile?
            yang:gauge64
   |     +--ro intermediate-jitter-percentile?
            yang:gauge64
   |     +--ro high-jitter-percentile?
            yang:gauge64
   +--rw vpn-pm-type
      +--rw inter-vpn-access-interface
      |  +--rw inter-vpn-access-interface?   empty
         +--rw vpn-tunnel!
```

```
             +--ro vpn-tunnel-type?   identityref
augment /nw:networks/nw:network/nw:node/ nt:
  termination-point:
  +--ro pm-statistics
     +--ro last-updated?          yang:date-and-time
     +--ro inbound-octets?            yang:counter64
     +--ro inbound-unicast?           yang:counter64
     +--ro inbound-broadcast?         yang:counter64
     +--ro inbound-multicast?         yang:counter64
     +--ro inbound-discards?          yang:counter64
     +--ro inbound-errors?            yang:counter64
     +--ro inbound-unknown-protocol?  yang:counter64
     +--ro outbound-octets?           yang:counter64
     +--ro outbound-unicast?          yang:counter64
     +--ro outbound-broadcast?        yang:counter64
     +--ro outbound-multicast?        yang:counter64
     +--ro outbound-discards?         yang:counter64
     +--ro outbound-errors?           yang:counter64
     +--ro vpn-network-access*  [network-access-id]
        +--ro network-access-id    vpn-common:vpn-id
        +--ro last-updated?       yang:date-and-time
        +--ro inbound-octets?         yang:counter64
        +--ro inbound-unicast?        yang:counter64
        +--ro inbound-broadcast?      yang:counter64
        +--ro inbound-multicast?      yang:counter64
        +--ro inbound-discards?       yang:counter64
        +--ro inbound-errors?         yang:counter64
        +--ro inbound-unknown-protocol? yang:counter64
        +--ro outbound-octets?        yang:counter64
        +--ro outbound-unicast?       yang:counter64
        +--ro outbound-broadcast?     yang:counter64
        +--ro outbound-multicast?     yang:counter64
        +--ro outbound-discards?      yang:counter64
        +--ro outbound-errors?        yang:counter64
```

In this YANG model, two types of data nodes are defined: link and termination point.

The link data node involves two types of links: topology links and abstract links (between a pair of VPN PEs). The following link-level performance indicators (all of which are unidirectional) are defined for this data node:

- **percentile**: specifies the percentile for measuring delay and jitter. It is represented at three levels, namely, high, intermediate, and low, with the default values being the 10th, 50th, and 90th percentiles, respectively. Setting a percentile node to 0.000 indicates that the client is not interested in receiving a particular percentile. If all percentile nodes are set to 0.000, no percentile-related nodes will be reported for a given performance metric. For example, if only high percentiles are set, only **high-delay-percentile** and **high-jitter-percentile** data will be collected for a given link at given **start-time**, **end-time**, and **measurement-interval**.
- **measurement-interval**: specifies the performance measurement interval, in seconds.
- **start-time**: specifies the start time for the performance measurement of links.
- **end-time**: specifies the end time for the performance measurement of links.

- **pm-source**: specifies the performance monitoring source. The data for topology links can be obtained through BGP-LS[4]. Statistics about VPN abstract links can be collected through VPN OAM mechanisms, for example, the OAM mechanisms referenced in RFC 9182[5] or the Ethernet service OAM mechanism referenced in RFC 9291[6]. Alternatively, the statistics can be collected through the OAM mechanism of an underlying technique, for example, the on-path telemetry technique described in this book.
- **loss-statistics**: indicates a set of one-way packet loss statistics, which are used to measure E2E packet loss between VPN sites or between any two network nodes. The reported value can be the number of lost packets or the packet loss rate.
- **delay-statistics**: indicates a set of one-way delay statistics, which are used to measure E2E delay between VPN sites or between any two network nodes. The reported values can be the maximum/minimum delay or percentile values.
- **jitter-statistics**: indicates a set of one-way jitter statistics, which are used to measure E2E jitter between VPN sites or between any two network nodes. The reported values can be the maximum/minimum jitter or percentile values.
- **one-way-pm-statistics-per-class**: lists performance statistics for abstract links between VPN PEs or topology links. An abstract link is identified with a unique identifier (**class-id**).
- **vpn-pm-type**: indicates the type of VPN performance measurement, which is typically either **inter-vpn-access-interface** or **vpn-tunnel** (usually, they are not used together). **inter-vpn-access-interface** monitors the performance of a P2P VPN connection between the source and destination nodes, which may be two PEs or the source and destination VPN access interfaces on PEs. **vpn-tunnel** monitors the performance of VPN tunnels.
- **vpn-tunnel-type**: indicates the abstract link protocol type of a VPN, such as GRE or IP in IP Encapsulation (IPinIP). It is an “underlay-transport” identifier defined in RFC 9181[7], which describes the transport technology that carries VPN service traffic.

The termination-point data node defines the following minimal set of statistics:

- **last-updated**: indicates the date and time when the counters were last updated.
- **Inbound statistics**: indicates a set of inbound statistics attributes, which are used to measure the inbound statistics of the termination point (e.g., received packets and received packets with errors).
- **Outbound statistics**: indicates a set of outbound statistics attributes, which are used to measure the outbound statistics of the termination point (e.g., sent packets and packets that could not be sent due to errors).
- **vpn-network-access**: lists counters for VPN access defined in the L3VPN Network Model (L3NM)[5] or L2VPN Network Model (L2NM)[8]. It is not required if a port is associated with only a single VPN. If multiple VPN access connections are created using the same physical port, finer-grained counters can be monitored.

6.4.2 Alternate Marking Interfaces for IPv6 On-Path Telemetry

The following provides an excerpt about the YANG model defined in draft-ydt-ippm-alt-mark-yang[9] for configuring alternate marking.

```
module: ietf-alt-mark
+--ro altmark-info
|  +--ro timestamp-type?
|  +--ro available-interface*     [if-name]
|     +--ro if-name              if:interface-ref
+--rw altmark-profiles
   +--rw admin-config
   |  +--rw enabled?              boolean
   +--rw altmark-profile           [profile-name]
      +--rw profile-name          string
      +--rw filter
      |  +--rw filter-type?   altmark-filter-type
      |  +--rw ace-name?  -> /acl:acls/acl/aces/
                                 ace/name
         +--rw protocol-type?  altmark-protocol-type
         +--rw node-action      altmark-node-action
         +--rw period?            uint64
         +--rw flow-mon-id?       uint32
         +--rw measurement-mode?  altmark-
                                  measurement-mode
         +--rw enable-loss-measurement?   boolean
         +--rw enable-delay-measurement?  boolean
```

Two types of data nodes are defined for the draft-ydt-ippm-alt-mark-yang model: altmark-info and altmark-profile. altmark-info nodes represent information objects used by the monitoring system to parse data, including timestamps and the list of all available interfaces that support alternate marking. altmark-profile nodes represent configuration objects used to enable alternate marking, packet loss measurement, and delay measurement and define flow characteristics. The details are as follows:

- **profile-name**: uniquely identifies an alternate marking profile, which corresponds to an on-path telemetry instance.
- **filter**: identifies a flow where the IOAM profile can apply. There may be multiple types of filters, such as ACL and ACE.
- **protocol-type**: indicates the encapsulation protocol type for the alternate marking application, such as IPv6 or SFC-NSH.
- **node-action**: indicates the action to be taken on a flow, such as marking the alternate marking header, reading the alternate marking data, or unmarking the alternate marking header.
- **period**: specifies the alternate marking period[10].
- **flow-mon-id**: identifies the monitored flow and correlates the exported data of the same flow from multiple nodes and multiple packets.
- **measurement-mode**: specifies the measurement mode, which can be hop-by-hop or E2E.
- **enable-loss-measurement**: enables packet loss measurement when set to **true**.
- **enable-delay-measurement**: enables delay measurement when set to **true**.

6.4.3 IOAM Interfaces for IPv6 On-Path Telemetry

The following provides an excerpt about the YANG model defined in draft-ietf-ippm-ioam-yang[11] for configuring IOAM.

```
module: ietf-ioam
   +--rw ioam  +--ro info
   |  +--ro timestamp-type?      identityref
   |  +--ro available-interface* [if-name]
   |     +--ro if-name    if:interface-ref
   +--rw admin-config
   |  +--rw enabled?   boolean
   +--rw profiles
      +--rw profile* [profile-name]
 +--rw profile-name               string
         +--rw filter
         |  +--rw filter-type?  ioam-filter-type
         |  +--rw ace-name?      -> /acl:acls/acl/aces/ace/name
         +--rw protocol-type? ioam-protocol-type
         +--rw incremental-tracing-profile
            {incremental-trace}?
            +--rw node-action?  ioam-node-action
            +--rw trace-types
            |  +--rw use-namespace? ioam-namespace
            |  +--rw trace-type* ioam-trace-type
            +--rw max-length?             uint32
         +--rw preallocated-tracing-profile
            {preallocated-trace}?
            +--rw node-action?  ioam-node-action
            +--rw trace-types
            |  +--rw use-namespace? ioam-namespace
            |  +--rw trace-type* ioam-trace-type
            +--rw max-length?         uint32
         +--rw direct-export-profile
            {direct-export}?
            +--rw node-action?  ioam-node-action
            +--rw trace-types
            |  +--rw use-namespace?   ioam-namespace
            |  +--rw trace-type*  ioam-trace-type
            +--rw flow-id?            uint32
            +--rw enable-sequence-number? boolean
         +--rw pot-profile {proof-of-transit}?
            +--rw use-namespace?  ioam-namespace
            +--rw pot-type?   ioam-pot-type
         +--rw e2e-profile {edge-to-edge}?
            +--rw node-action?   ioam-node-action
            +--rw e2e-types
               +--rw use-namespace?
                  ioam-namespace
               +--rw e2e-type*     ioam-e2e-type
```

Three types of data nodes are defined for the draft-ietf-ippm-ioam-yang model: ro info, admin-config, and profiles. ro info nodes represent information objects used by the monitoring system to parse data, including the timestamps and list of all interfaces that can use IOAM. admin-config nodes represent configuration objects used to enable IOAM globally. IOAM is enabled when **enabled** is set to **true**. Enabling IOAM for the system is the prerequisite for subsequent configurations in profiles to take effect. profiles nodes represent configuration objects used for detailed configurations and are expressed in lists. Each list contains the following information:

- **filter**: identifies a flow where the IOAM profile can apply. There may be multiple types of filters.
- **protocol-type**: indicates the type of encapsulation protocol used to encapsulate IOAM data, such as IPv6 or SFC-NSH.
- **incremental-tracing-profile**, which further includes the following:
    - **node-action**: indicates the operation applied to a specified flow, such as encapsulating the IOAM header, transmitting IOAM data, or removing the IOAM header.
    - **use-namespace**: indicates the namespace used for the trace types.
    - **trace-type**: indicates the per-hop data to be captured by IOAM-enabled nodes and included in the node data list.
    - **max-length**: specifies the maximum length of the node data list, in bytes. **max-length** is only defined at the encapsulation node.
- **preallocated-tracing-profile**: collects the IOAM data at every node that a packet traverses and preallocates memory to each node to store data, so as to ensure visibility into the entire path that a packet takes within an IOAM domain. Its detailed fields are the same as those for **incremental-tracing-profile**.
- **direct-export-profile**: allows IOAM data to be directly exported or exported after local aggregation. In addition to the same detailed fields as those for **preallocated-tracing-profile**, it also includes the following additional fields:
    - **flow-id**: correlates the exported data of the same flow from multiple nodes and multiple packets.
    - **enable-sequence-number**: indicates whether the sequence number is used in export results.
- **pot-profile**: contains configuration information about verifying packets traversing given paths.
- **e2e-profile**: contains configuration information about E2E performance measurement for service flows.

6.5 STORIES BEHIND IPV6 ON-PATH TELEMETRY

6.5.1 Management, Control, and Analysis Convergence

The development of Huawei network controller products reflects the broader evolution of controllers in the industry, having undergone a somewhat tumultuous journey thus far.

Controllers originated from SDN, which initially aimed to separate forwarding and control through OpenFlow. More specifically, controllers were initially designed to handle all path computation tasks and use OpenFlow to push forwarding entries to forwarding devices. However, SDN's ideal of completely separating forwarding and control proved unrealistic, leading to the eventual failure of OpenFlow. Despite this, centralized controllers emerged and have moved away from strict forwarding-control separation. These controllers utilize existing southbound control protocols like BGP and PCEP to deliver control information to network devices. Such an approach facilitates the deployment of value-added network services that are challenging to achieve through distributed networks, for example, global network optimization. This “centralized + distributed” control architecture is better suited for real-world network conditions and has therefore gained recognition in the industry.

During the development of centralized controllers based on protocols such as BGP and PCEP, it became clear that traditional network management functions like FCAPS are still essential. This means that controllers cannot fully replace NMSs. But if controllers and NMSs coexist and operate independently, conflicting management and control of devices may occur due to a lack of information exchange. For example, PCE information delivery goes hand in hand with PCE configuration — they are inseparable. If a controller frequently delivers path information through PCEP for creation and deletion while a network management system frequently enables and disables PCE configurations, disruption of network operations and unexpected issues are likely to occur. In addition, traditional NMSs use protocols like SNMP to collect network topology information, while controllers obtain this data through BGP-LS for path computation. This not only leads to information redundancy but also creates discrepancies in topology information due to performance differences between SNMP and BGP-LS. An easy way to solve these issues is to integrate controllers with traditional network management functions.

Another important development in the evolution of controllers is the incorporation of network analysis functions. During the early days of SDN, network vendors acquired companies like Wandl, Cariden, and OPNET, which specialized in network simulation, planning, and optimization. This resulted in the functions of these acquired companies’ products being integrated into controllers. Huawei initially relied on third-party products for IP network planning and optimization, but after those products were acquired, Huawei had to develop its own IP network simulation, network planning and optimization, and network analysis capabilities. These capabilities were initially offered as network services but were gradually integrated into controller products.

The OpenFlow-based SDN controller was initially only a concept. Developing commercially viable controller products is a complex process influenced by various factors, including technology, industry trends, and customer needs. Eventually, the SDN controller concept has been turned into a network controller product that integrates multiple functions, such as management, control, and analysis. Its southbound protocols now include PCEP, BGP, NETCONF/YANG, and gRPC, while OpenFlow has been completely excluded.

6.5.2 Reflections on Digital Maps

Since 2022, Huawei has been actively promoting the concept of digital maps for controller products. Such maps are similar to applications like Google Maps and AutoNavi Map. They offer intuitive functions and have been well-received by users. Reflecting on the development history of controllers and digital maps, two key aspects stand out.

First, since the inception of controllers, many seemingly impressive features have emerged, aiming to showcase cutting-edge capabilities. Initially influenced by Google's B4 network[12], there was a strong focus on automated network optimization. Then, with the rise of Industry 4.0, concepts like digital twins and network twins gained popularity. Later, the boom in AI led to an abundance of use cases for intelligent applications in controllers. In contrast, the concept of digital maps seems traditional, but has gained greater user acceptance, highlighting the fundamental importance of IP network visualization. If this foundational demand is not addressed properly, discussions of advanced network functions are of little use. Throughout the development of IP network technology, visualization features have often been overlooked, intentionally or not. One of the major reasons is the complexity of IP features and the substantial efforts required for visualization, often yielding underwhelming results after many features are provided. Few are willing to undertake this job, resulting in subpar outcomes.

Second, visualization of IP networks requires technological expertise and breakthroughs, along with effective support from network device capabilities. During the MPLS era, traditional telecom networks successfully transitioned to IP, with key features like VPN, TE, and FRR playing crucial roles. VPN addressed the need for multi-service transport and isolation, TE ensured service quality along paths, and FRR met demands for high reliability. These key features facilitated the shift from traditional telecom private lines to IP-based private lines. However, IP/MPLS has fallen short in terms of the manageability and maintainability inherent in telecom networks. Thanks to the advancement of technologies like controllers, IPv6 Enhanced, and on-path telemetry, the ability to visualize networks completely has been basically achieved. This led to the emergence of digital maps, which have helped boost efficiency in network O&M.

REFERENCES

1. TM Forum. Network automation using machine learning and AI[R]. (2020-04-01) [2024-09-30]. ⏎
2. Clemm A, Medved J, Varga R et al. A YANG data model for network topologies [EB/OL]. (2018-03) [2024-09-30]. RFC 8345. ⏎
3. Wu B, Wu Q, Boucadair M et al. A YANG data model for network and VPN service performance monitoring [EB/OL]. (2023-04) [2024-09-30]. RFC 9375. ⏎
4. Ginsberg L, Previdi S, Wu Q et al. BGP-link state (BGP-LS) advertisement of IGP traffic engineering performance metric extensions [EB/OL]. (2019-03-15) [2024-09-30]. RFC 8571. ⏎
5. Barguil S, Gonzalez De Dios O, Boucadair M et al. A YANG network data model for layer 3 VPNs [EB/OL]. (2022-02-15) [2024-09-30]. RFC 9182. ⏎
6. Sajassi A, Thoria S, Mishra M et al. Internet group management protocol (IGMP) and multicast listener discovery (MLD) proxies for Ethernet VPN (EVPN) [EB/OL]. (2023-06) [2024-09-30]. RFC 9251. ⏎
7. Barguil S, Gonzalez de Dios O, Boucadair M et al. A common YANG data model for layer 2 and layer 3 VPNs [EB/OL]. (2022-02-15) [2024-09-30]. RFC 9181. ⏎

8. Boucadair M, Gonzalez de Dios O, Barguil S et al. A YANG network data model for layer 2 VPNs [EB/OL]. (2022-09) [2024-09-30]. RFC 9291. ⏎
9. Graf T, Wang M, Fioccola G et al. A YANG data model for the alternate marking method [EB/OL]. (2024-02-29) [2024-09-30]. Draft-ydt-ippm-alt-mark-yang-01. ⏎
10. Fioccola G, Zhu K, Graf T et al. Alternate marking deployment framework [EB/OL]. (2024-07-03) [2024-09-30]. Draft-ietf-ippm-alt-mark-deployment-01. ⏎
11. Zhou T, Guichard J, Brockners F et al. A YANG data model for in-situ OAM [EB/OL]. (2024-03-01) [2024-09-30]. Draft-ietf-ippm-ioam-yang-13. ⏎
12. Jain S, Kumar A, Mandal S et al. B4: Experience with a globally-deployed software defined wan [EB/OL]. (2013-08-27) [2024-09-30]. ⏎

CHAPTER 7

Deployment of IPv6 On-Path Telemetry

DOI: 10.1201/9781003677901-9

IPV6 ON-PATH TELEMETRY ENABLES users to measure and observe network quality in real time and is already in use on multiple carrier and enterprise networks. This chapter describes the application and deployment of IPv6 on-path telemetry on several typical networks using the alternate marking-based IFIT solution — referred to as IFIT-AM for short — as an example.

7.1 APPLICATION OF IFIT-AM

Currently, IFIT-AM is an IPv6 on-path telemetry solution that has been put into large-scale commercial use. Various mainstream network solutions already adopt IFIT-AM, including IP Radio Access Network (IP RAN), premium IP private line service, and financial Wide Area Network (WAN) solutions. The following sections describe the successful application of IFIT-AM in these solutions.

7.1.1 IP RAN Deployment

The IP RAN solution helps carriers maximize their Return On Investment (ROI), reduce network construction costs, and ensure smooth network evolution. In this solution, the large-scale mobile transport network supports various access modes and carries various mobile transport services (e.g., HD video). These services pose high requirements on link connectivity and performance indicators.

To meet such requirements, the E2E IFIT-AM + hop-by-hop IFIT-AM solution can be used. It helps quickly demarcate and locate faults and enables them to be replayed on demand, thereby improving SLA experience and O&M efficiency. Figure 7.1 shows how this solution is applied on an IP RAN.

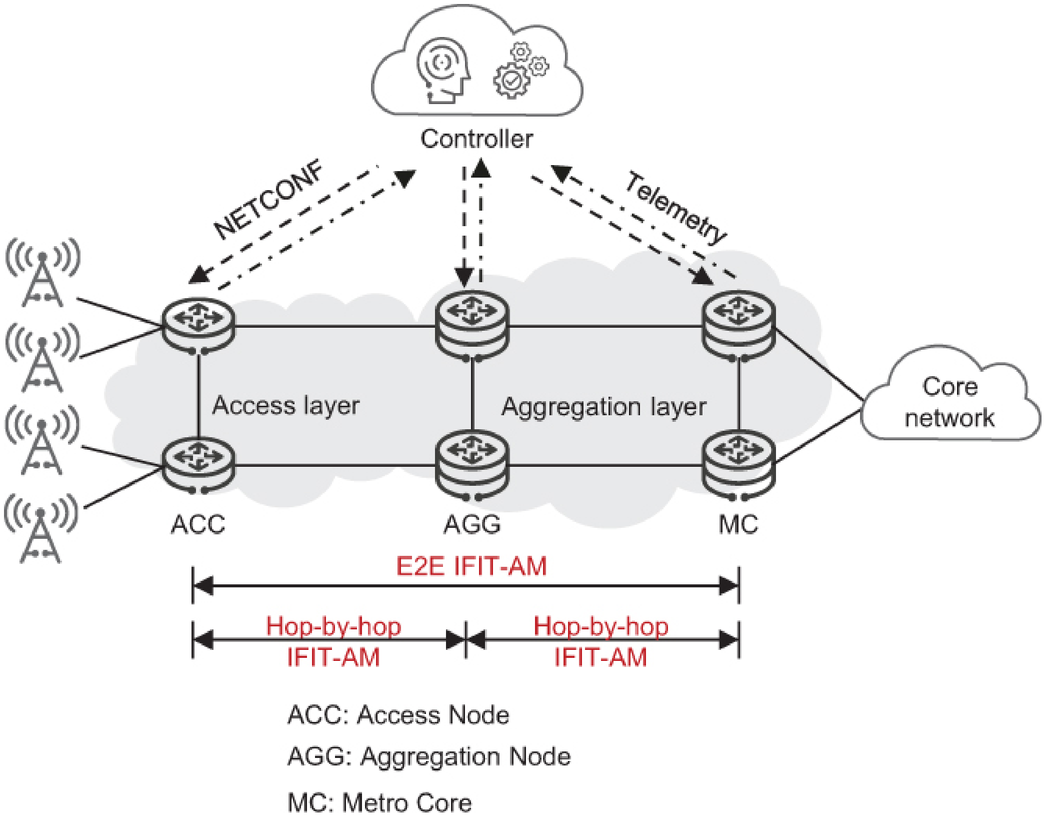


FIGURE 7.1 Application of IFIT-AM on an IP RAN. ⏎

In this scenario, E2E performance measurement is performed using E2E IFIT-AM, and hop-by-hop IFIT-AM performance measurement is performed if the performance indicator of base station flows exceeds the specified threshold. The controller then summarizes the reported hop-by-hop measurement data for path restoration and fault locating. This solution has the following characteristics:

1. The solution enables monitoring of detailed service flow indicators from different dimensions, such as base station, data, and signaling flows. It also supports clustering to process base station flow faults and quickly demarcate poor-QoS services, ensuring that the number of hop-by-hop IFIT-AM instances triggered when multiple faults occur in multiple base station flows does not exceed the system's maximum capacity.
2. For faults outside the IP RAN, the solution helps quickly and accurately prove that the network is not the cause. And for faults inside the IP RAN, the solution helps quickly locate faulty NEs and links, boosting efficiency in network O&M.
3. The real-time performance data of base stations throughout the network can be used to build a big data-based intelligent O&M system. With such a system, it is possible to implement precise service-level SLA awareness in real time and multi-dimensional visualization for base station services. This system can also analyze and evaluate potential network risks and optimize network resources to implement automatic and intelligent O&M.

7.1.2 Premium IP Private Line Service Deployment

The premium IP private line solution enables enterprises to obtain private line services more conveniently, thanks to it leveraging the wide coverage offered by the mobile transport network. E2E collaborative management for private line services implements automatic and intelligent O&M, improves network deployment, operation, and O&M efficiency, and supports digital transformation of various services, including carriers' services for businesses and enterprises' services such as government and healthcare services.

This section uses the cloud private line as an example to describe how the E2E IFIT-AM + hop-by-hop IFIT-AM solution provides VPN service analysis and assurance for premium IP private line services, as shown in Figure 7.2. This solution is used to ensure E2E high reliability and implement minute-level fault locating through visualized O&M.

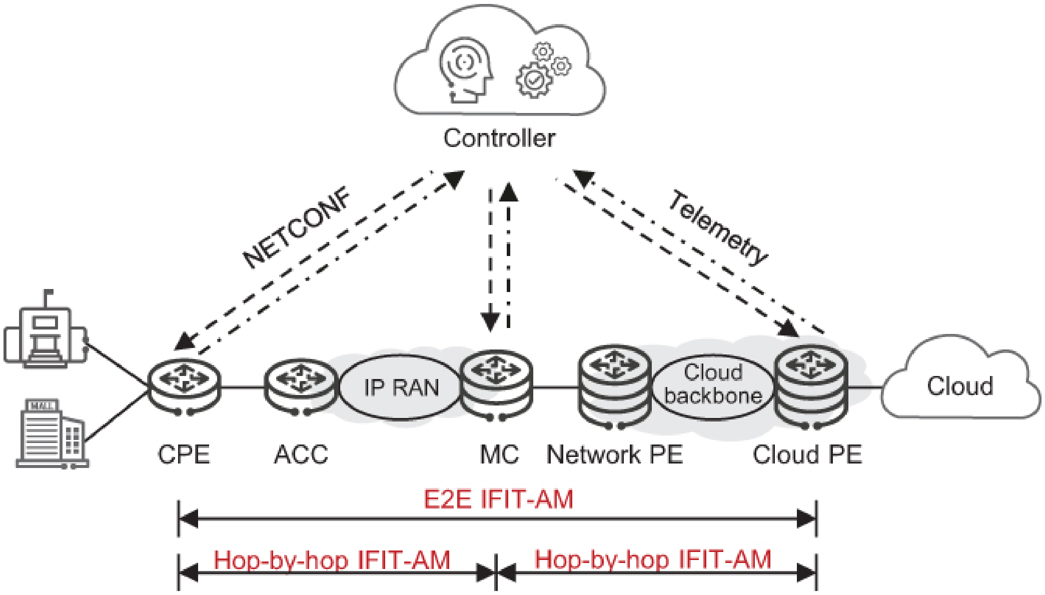


FIGURE 7.2 Application of IFIT-AM for premium IP private line services. ⏎

In this scenario, E2E performance measurement is performed using E2E IFIT-AM, and hop-by-hop IFIT-AM performance measurement is performed if the performance indicator of VPN flows exceeds the specified threshold. The controller then summarizes the reported hop-by-hop measurement data for path restoration and fault locating. This solution has the following characteristics:

1. Analyzes and locates faults of a VPN flow and queries its E2E performance indicators — including the maximum traffic rate, maximum one-way delay, and maximum packet loss rate — with a granularity ranging from minutes to years.
2. Queries E2E VPN service information based on characteristics such as the VPN name, VPN type, and service status. If multiple segments of service flows exist, the status of the segment with the lowest quality is used.
3. Implements E2E multi-dimensional exception identification, network health visualization, intelligent fault diagnosis, fault self-healing in a closed-loop manner, and

more.

7.1.3 Financial WAN Deployment

The financial WAN provides cross-domain network services by coordinating different networks. In the financial industry, tier-2 branches, outlets, subsidiaries, and external organizations connect to tier-1 branches, which aggregate service traffic and then connect to the bank core network, implementing mutual access between them and the head office data center. In this case, the concept of the financial WAN's centralized management is particularly important.

Leveraging SRv6 technology, the financial WAN quickly and easily establishes basic network connections between the cloud and various access points, making service provisioning far more efficient. In the financial industry, requirements for SLA assurance are extremely high. Furthermore, as banking services continue to evolve and give rise to a diverse array of outlet service types, the financial WAN faces high requirements on O&M capabilities. For example, services such as security protection, IoT, and public cloud services, in addition to traditional production and office services, are now prevalent. To address such requirements, the IFIT-AM tunnel-level measurement solution can be used. In Figure 7.3, this solution is used to simplify the O&M process and optimize the O&M experience.

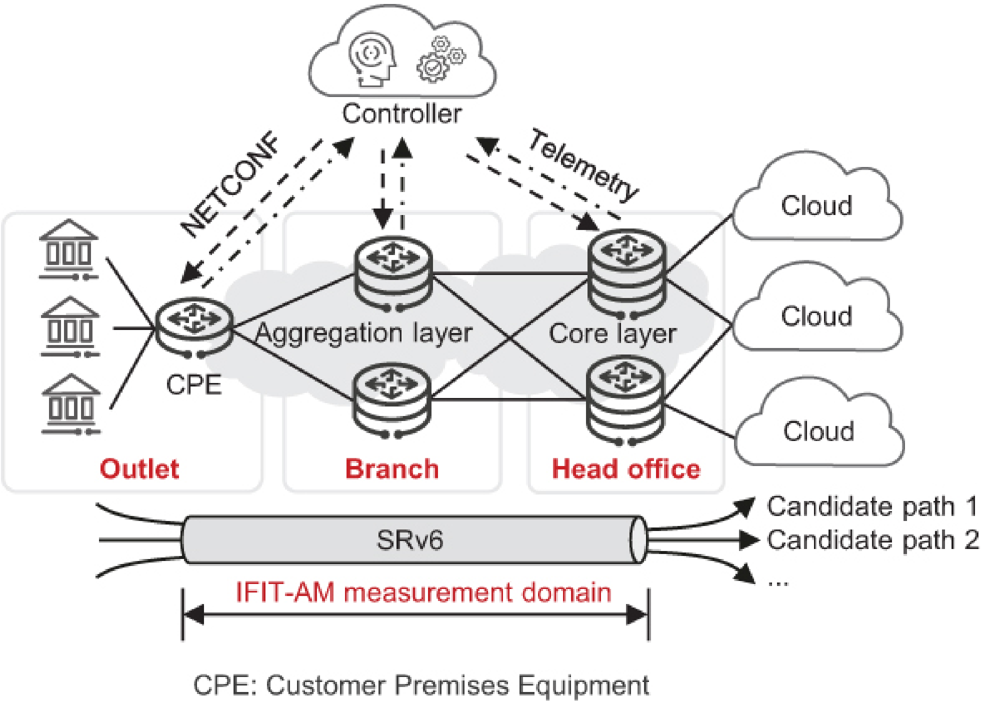


FIGURE 7.3 Application of IFIT-AM on the financial WAN. ⏎

This solution has the following characteristics:

1. In SRv6 scenarios, tunnel-level measurement based on IFIT-AM can be used to measure the quality of each SRv6 Policy segment list and select the optimal link. The

link currently in use is periodically compared with the optimal link for path selection and optimization, implementing intelligent traffic steering.
2. One core controller is deployed to perform centralized O&M on the entire financial network and implement E2E management and scheduling.

7.2 IFIT-AM DEPLOYMENT ON DEVICES

This section describes how to deploy IFIT-AM on devices, covering time synchronization, subscription, and measurement instance deployments. According to the application scenario and measurement granularity, IFIT-AM can be classified into the following types: dynamic learning-based IFIT-AM, static IP 5-tuple-based IFIT-AM, VPN + peer-based IFIT-AM, MAC address-based IFIT-AM, and tunnel-based IFIT-AM. The different measurement types apply to different scenarios and provide reference for deploying and using the IFIT-AM solution on live networks.

7.2.1 Time Synchronization Deployment

To accurately measure packet loss and delay, the IFIT-AM solution requires time synchronization to be deployed between NEs. This can be achieved using a time synchronization protocol such as NTP or PTP. For details about these protocols, see Appendix B. The following describes how to configure them.

7.2.1.1 NTP Synchronization Configuration

NTP synchronization is accurate to milliseconds and can be used for IFIT-AM packet loss measurement. However, it is not suitable for delay measurement due to its low synchronization precision, which introduces large errors in the delay measurement result. The following uses the client/server model as an example to describe how to configure NTP synchronization. In this example, DeviceA and DeviceB are the server and client, respectively, as shown in Figure 7.4.

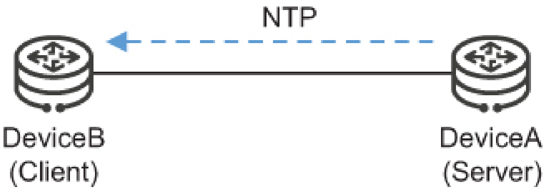


FIGURE 7.4 NTP synchronization example. ⏎

The configuration procedure is as follows:

1. Enable the NTP service and configure listening interfaces on the server.

```
<DeviceA> system-view
[~DeviceA] ntp-service refclock-master 2   //Configure
the local clock as the master clock and set the clock
stratum to 2 so that time is output to DeviceB.
[*DeviceA] ntp-service ipv6 server source-interface
```

```
all enable   //Enable NTP IPv6 listening.
[*DeviceA] undo ntp-service server disable   //Enable
the NTP server function.
[*DeviceA] commit
```

2. Specify the address of the NTP server on the client to synchronize the time from the server.

```
<DeviceB> system-view
[~DeviceB] ntp-service unicast-server 2001:DB8:11::1
//Specify DeviceA as the NTP server.
[*DeviceB] ntp-service ipv6 server source-interface
gigabitethernet 1/0/1   //Specify the interface
connected to DeviceA as a listening interface.
[*DeviceB] commit
```

3. Check the NTP status of the client, including whether the clock has been synchronized, the clock stratum, and the reference clock source.

```
[~DeviceB] display ntp-service status
 clock status: synchronized   //The clock status is
synchronized.
 clock stratum: 3   //The clock stratum is 3 (higher
than the clock stratum 2 of the clock source),
indicating that the clock is directly obtained from
the clock source.
 reference clock ID: 2001:DB8:11::1   //Identifier of
the clock source.
 nominal frequency: 64.0029 Hz   //Nominal clock
frequency.
 actual frequency: 64.0029 Hz   //Actual clock
frequency.
 clock precision: 2^7   //Clock synchronization
precision.
 clock offset: 0.0000 ms   //Clock synchronization
offset.
 root delay: 62.50 ms   //Round Trip Time (RTT) delay
to the primary reference clock.
 root dispersion: 0.20 ms   //Reference error to the
primary reference clock.
 peer dispersion: 0.20 ms   //Reference error to the
peer clock.
 reference time: 06:52:33.465 UTC Feb 7
2020(C7B7AC31.773E89A8)   //Synchronization time.
 synchronization state: clock synchronized   //
Synchronization status of the clock.
```

7.2.1.2 PTP Synchronization Configuration

PTP provides microsecond- or nanosecond-level high-precision time synchronization, enabling IFIT-AM to implement accurate packet loss measurement and microsecond-level delay measurement. Common PTPs include Institute of Electrical and Electronics Engineers (IEEE) 1588v2 and G.8275.1 — the former is used here as an example. In PTP synchronization, an external time source must be configured, and PTP must be enabled on interfaces. This is necessary for time to be received from upstream devices or synchronized to downstream devices. Figure 7.5 shows a PTP synchronization example. The Building Integrated Timing Supply (BITS) device functions as an external clock source and outputs time information to the core device, Metro Core (MC), through a clock interface. After the MC synchronizes the time, it uses PTP to synchronize the clock to other devices, such as the Aggregation Node (AGG) and Access Node (ACC), hop by hop.

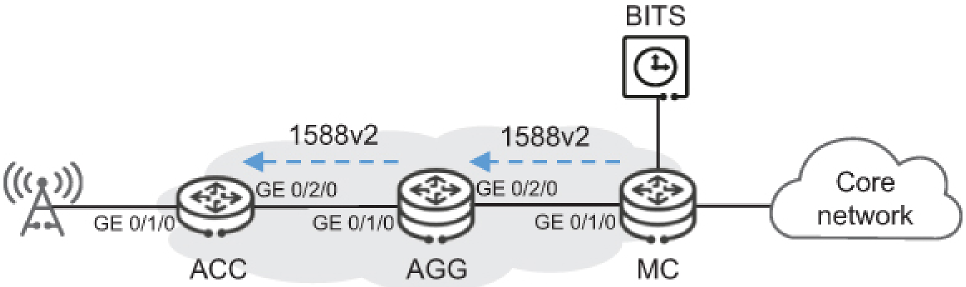


FIGURE 7.5 PTP synchronization example. ⏎

The configuration procedure is as follows:

1. Configure the time source device MC, including enabling PTP and clock synchronization.

```
<MC> system-view
[~MC] ptp enable   //Enable PTP globally.
[*MC] ptp device-type bc   //Set the PTP clock type to
Boundary Clock.
[*MC] ptp domain 1   //Configure a clock domain.
[*MC] clock bits-type bits1 1pps input   //Specify
BITS1 as the external time input source.
[*MC] ptp clock-source bits1 on   //Configure the
device to use BITS1 clock signals to calculate the
clock source.
[*MC] ptp clock-source bits1 priority1 2   //Configure
a time source priority.
[*MC] ptp source-switch ptsf enable   //Enable Packet
Timing Signal Fail (PTSF)-triggered time source
switching.
[*MC] commit
[~MC] interface gigabitethernet 0/1/0
[*MC-GigabitEthernet0/1/0] clock synchronization
enable   //Enable clock synchronization.
[*MC-GigabitEthernet0/1/0] ptp port-state primary
master   //Configure the PTP interface as the master
to output time to downstream devices.
[*MC-GigabitEthernet0/1/0] ptp enable   //Enable PTP
on the interface.
[*MC-GigabitEthernet0/1/0] commit
```

2. Configure the AGG to synchronize the IEEE 1588v2 time from the MC and output the synchronized time to the ACC.

```
<AGG> system-view
[~AGG] ptp enable
```

```
[*AGG] ptp device-type bc
[*AGG] ptp domain 1
[*AGG] commit
[~AGG] interface gigabitethernet 0/1/0
[~AGG-GigabitEthernet0/1/0] clock synchronization
enable
[*AGG-GigabitEthernet0/2/0] ptp port-state primary
master
[*AGG-GigabitEthernet0/1/0] ptp enable
[*AGG-GigabitEthernet0/1/0] commit
[~AGG-GigabitEthernet0/1/0] quit
[~AGG] interface gigabitethernet 0/2/0
[~AGG-GigabitEthernet0/2/0] clock synchronization
enable
[*AGG-GigabitEthernet0/2/0] ptp enable
[*AGG-GigabitEthernet0/2/0] commit
```

3. Configure the ACC to synchronize the IEEE 1588v2 time from the AGG and output the synchronized time to the base station.

```
<ACC> system-view
[~ACC] ptp enable
[*ACC] ptp device-type bc
[*ACC] ptp domain 1
[*ACC] commit
[~ACC] interface gigabitethernet 0/1/0
[~ACC-GigabitEthernet0/1/0] clock synchronization
enable
[*ACC-GigabitEthernet0/1/0] ptp port-state primary
master
[*ACC-GigabitEthernet0/1/0] ptp enable
[*ACC-GigabitEthernet0/1/0] commit
[~ACC-GigabitEthernet0/1/0] quit
[~ACC] interface gigabitethernet 0/2/0
[~ACC-GigabitEthernet0/2/0] clock synchronization
enable
[*ACC-GigabitEthernet0/2/0] ptp enable
[*ACC-GigabitEthernet0/2/0] commit
```

The following deployment examples assume that the preceding time synchronization configurations have been completed on related devices.

7.2.2 Subscription Deployment

IFIT-AM deployment requires telemetry collection instances to be subscribed to for data collection. The collected measurement data is then reported to the analyzer through telemetry for analysis in real time.

Telemetry subscription is classified as either static or dynamic. In static subscription mode, data is continuously reported after subscription data sources are statically configured using commands. In dynamic subscription mode, gRPC is configured to provide services, and the collector dynamically delivers collection tasks based on collection requirements. This section describes how to configure a static subscription, which involves specifying key information such as the collector IP address and port number, transmission protocol type

(TCP or UDP), and collection objects. Figure 7.6 provides an example of subscription deployment.

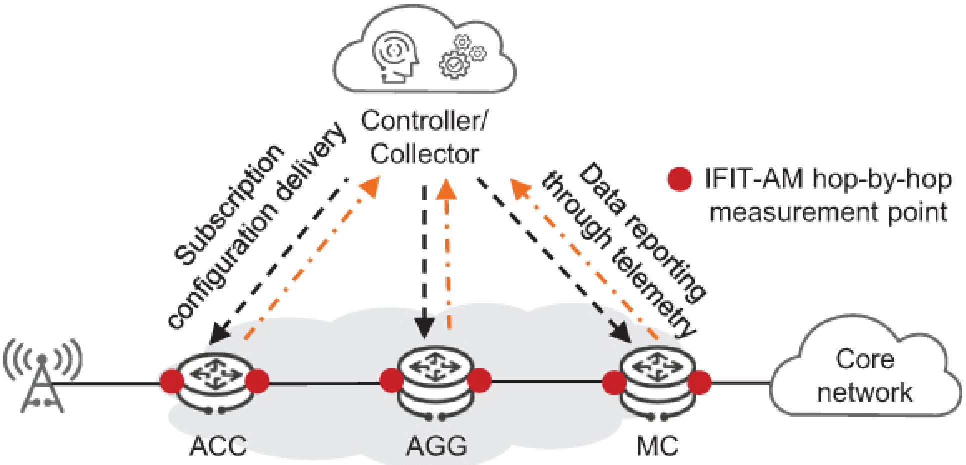

Controller/
Collector
Subscription configuration delivery
Data reporting through telemetry
IFIT-AM hop-by-hop measurement point
Core network
ACC
AGG
MC

FIGURE 7.6 Subscription deployment example. ⏎

The configuration procedure on the MC is as follows (the configurations on the ACC and AGG are similar to the configuration on the MC):

1. Configure the IPv4 or IPv6 address, protocol type, and port number of the collector.

```
<MC> system-view
[~MC] telemetry
[~MC-telemetry] destination-group destination1   //
Create a destination group to which sampled data is
reported.
[*MC-telemetry-destination-group-destination1]
ipv4-address 10.20.2.1 port 10001 protocol udp   //
Specify the destination IP address, protocol type, and
port number of the collector.
[*MC-telemetry-destination-group-destination1] commit
[~MC-telemetry-destination-group-destination1] quit
```

2. Configure collection objects, which are typically represented as a path. The collection content can be flexibly controlled. The following two sampling paths are used as an example.
   - The sampling path **huawei-ifit:ifit/huawei-ifit-statistics:flow-statistics/flow-statistic** indicates the statistic of the IFIT-AM ingress node.
   - The sampling path **huawei-ifit:ifit/huawei-ifit-statistics:flow- hop-statistics/flow-hop-statistic** indicates the statistic of the IFIT-AM hop-by-hop nodes.

```
[~MC-telemetry] sensor-group sensor1   //Create a
sampling sensor group.
[*MC-telemetry-sensor-group-sensor1]  sensor-path
huawei-ifit:ifit/
huawei-ifit-statistics:flow-statistics/flow-statistic
//Specify a sampling path.
```

```
[*MC-telemetry-sensor-group-sensor1-path] quit
[*MC-telemetry-sensor-group-sensor1] sensor-path
huawei-ifit:ifit/
huawei-ifit-statistics:flow-hop-statistics/
flow-hop-statistic
[*MC-telemetry-sensor-group-sensor1-path] quit
[*MC-telemetry-sensor-group-sensor1] commit
[~MC-telemetry-sensor-group-sensor1] quit
```

3. Configure a subscription instance to specify the destination group to which network devices' subscription data is reported and both the source IP address and port number used for reporting packets.

```
[~MC-telemetry]  subscription subscription1   //Create
a subscription to associate the destination group with
the sampling sensor group.
[*MC-telemetry-subscription-subscription1]
sensor-group sensor1
[*MC-telemetry-subscription-subscription1]
destination-group destination1
[*MC-telemetry-subscription-subscription1] commit
```

The following deployment examples assume that the preceding subscription configurations have been completed on related devices.

7.2.3 Dynamic Learning-Based IFIT-AM Deployment

After dynamic learning-based IFIT-AM is enabled on a device's inbound interface of service flows, the device dynamically learns the corresponding flow information and generates IFIT-AM measurement instances. This eliminates the need to specify characteristics (e.g., IP 5-tuple) for service flows.

For example, dynamic learning-based IFIT-AM can be used to monitor SLAs of many base station services. After being configured, it can automatically generate and age measurement instances based on real-time traffic, achieving the optimal utilization of device and network resources.

In the example shown in Figure 7.7, monitoring is performed on control- and data-plane N2/N3 traffic between a base station and 5G Core Network (5GC). Dynamic learning-based IFIT-AM is deployed on an MC (core device). Specifically, bidirectional dynamic flow learning is enabled based on user-side interfaces. After the configuration is complete, the MC automatically identifies service flows from the base station and generates an E2E IFIT-AM measurement instance. And, based on IFIT-AM information carried in downstream traffic, the corresponding ACC can automatically generate a reverse measurement instance from itself to the MC.

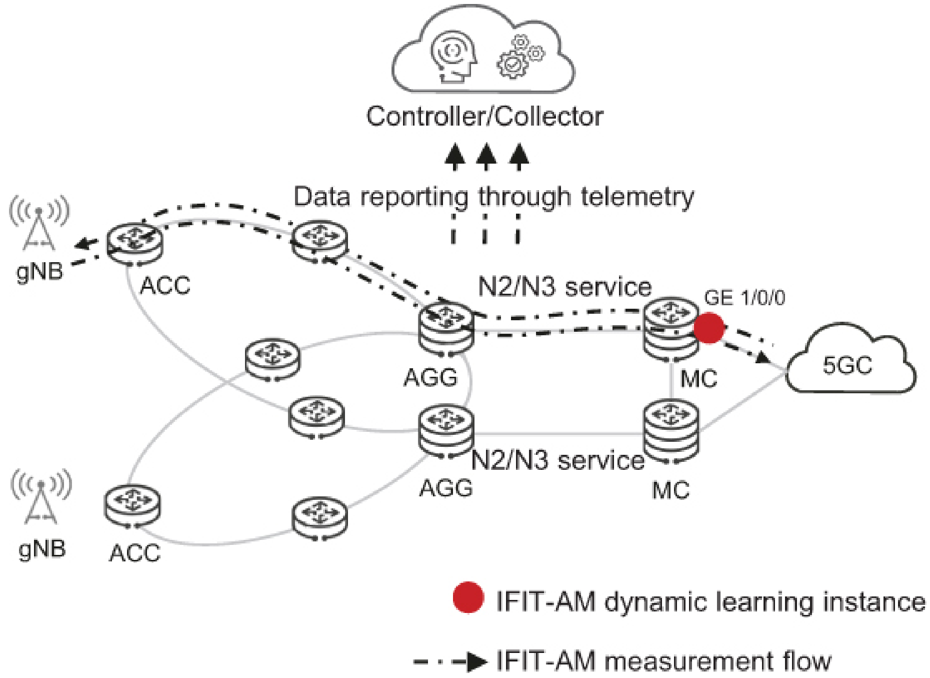


FIGURE 7.7 Deploying dynamic learning-based IFIT-AM for N2/N3 services on a mobile transport network. ⏎

The configuration procedure is as follows:

1. Enable IFIT-AM on the devices on the IFIT-AM measurement path. The following uses the ACC as an example.

```
<ACC> system-view
[~ACC] ifit   //Enable IFIT globally and enter the
IFIT view.
[*ACC-ifit] node-id 1   //Specify the NE ID.
[*ACC-ifit] commit
```

2. Configure dynamic flow learning for IFIT-AM on the corresponding MC.

```
<MC> system-view
[~MC] ifit
[*MC-ifit] node-id 2
[*MC-ifit] flow-learning vpn-instance vpna   //Enable
dynamic flow learning for IFIT-AM for a VPN.
[*MC-ifit-vpn-instance-vpna] flow-learning
bidirectional   //Enable bidirectional flow learning.
[*MC-ifit-vpn-instance-vpna] flow-learning interface
gigabitethernet 1/0/0   //Enable dynamic flow learning
on an interface.
[*MC-ifit-vpn-instance-vpna] commit
[~MC-ifit-vpn-instance-vpna] quit
[~MC-ifit] quit
```

3. Query information about dynamic flows of IFIT-AM to obtain dynamic IFIT-AM measurement instance data generated in real time.

```
[~MC] display ifit dynamic
-------------------------------------------------------
-------------------
Flow Classification                       : dynamic   //
The flow type is dynamic flow.
Instance Id                               : 10   //ID of
the measurement instance.
Instance Type                             : instance
//Type of the measurement instance.
Flow Id                                   : 1572865   //
Dynamically generated flow ID, which is the
combination of the Flow Monitor Id and Flow Node Id
fields.
Flow Monitor Id                           : 524289   //
Dynamically generated flow monitor ID, which is
obtained from the FlowMonID field in the IFIT-AM
header.
Flow Node Id                              : 1   //NE ID,
which is obtained from the FlowMonID Ext field in the
IFIT-AM header.
Flow Type                                : bidirectional
//The dynamic flow is a bidirectional one, meaning
that a reverse flow for this one is automatically
generated on the egress.
Source IP Address/Mask Length             : 10.11.1.1/32
//Source IP address and mask of the learned flow.
Destination IP Address/Mask Length        : 10.22.2.2/32
//Destination IP address and mask of the learned flow.
Protocol                                  : any   //The
protocol type of the dynamic flow is not specified.
Source Port                               : any   //The
source port number of the dynamic flow is not
specified.
Destination Port                          : any   //The
destination port number of the dynamic flow is not
specified.
Dscp                                      : --   //DSCP
is not configured for the dynamic flow.
Interface                                 :
GigabitEthernet1/0/0   //Interface corresponding to
the measurement instance.
vpn-instance                              : vpna   //VPN
instance corresponding to the measurement instance.
Measure State                             : enable   //
Measurement is enabled.
Loss Measure                              : enable   //
Packet loss measurement is enabled.
Delay Measure                             : enable   //
Delay measurement is enabled.
Measure Mode                              : e2e   //E2E
measurement is used.
Interval                                  : 30(s)   //
The measurement interval is 30s.
```

In dynamic learning-based IFIT-AM, policies can be used to flexibly control learning rules. For example, specific service flows can be monitored through a whitelist group, and specific or network segment IP addresses can be specified for dynamically generated IFIT-AM instances. Furthermore, in scenarios where dynamic learning-based IFIT-AM is used to monitor N2/N3 services on a mobile transport network, base station addresses can be set to specific IP addresses in order to distinguish each independent base station. And addresses for the User Plane Function (UPF)/Access and Mobility Management Function (AMF) on a 5GC can be set to network segment IP addresses in order to reduce the number of monitored flows for each base station.

```
<MC> system-view
[~MC] ifit
[*MC-ifit] node-id 2
[*MC-ifit]  whitelist-group 1   //Create a whitelist
group.
[*MC-ifit-whitelist-group-1]  rule rule1 ipv4
source 10.11.1.0 24 destination 10.22.2.0 24   //Set a
dynamic learning whitelist rule.
[*MC-ifit-whitelist-group-1] commit
[~MC-ifit-whitelist-group-1] quit
[~MC-ifit] flow-learning vpn-instance vpna
[*MC-ifit-vpn-instance-vpna]  learning-mode
sip-mask-dip-exact   //Configure a dynamic learning
policy to match the source IP address with the mask
(network segment IP address) and exactly match the
destination IP address (specific IP address).
[*MC-ifit-vpn-instance-vpna] flow-learning
bidirectional
[*MC-ifit-vpn-instance-vpna] flow-learning interface
all whitelist-group 1   //Apply the whitelist rule to
all interfaces in the VPN.
[*MC-ifit-vpn-instance-vpna] commit
```

7.2.4 Static IP 5-Tuple-Based IFIT-AM Deployment

Static IP 5-tuple-based IFIT-AM performs fine-grained SLA measurement for IP service flows. It can be used for on-demand monitoring in scenarios such as key service assurance and service fault demarcation. It can also be used in Hierarchy VPN (HVPN) over SRv6 scenarios, where because VPNs are terminated on Superstratum Provider Edges (SPEs), peer-based IFIT-AM (described in Section 7.2.5) cannot implement expected E2E monitoring.

Figure 7.8 depicts a site-to-cloud private line scenario in which E2E Ethernet Virtual Private Network (EVPN) Layer 3 Virtual Private Network (L3VPN) over SRv6 is deployed between an enterprise branch and enterprise cloud. Static IP 5-tuple-based IFIT-AM can be used.

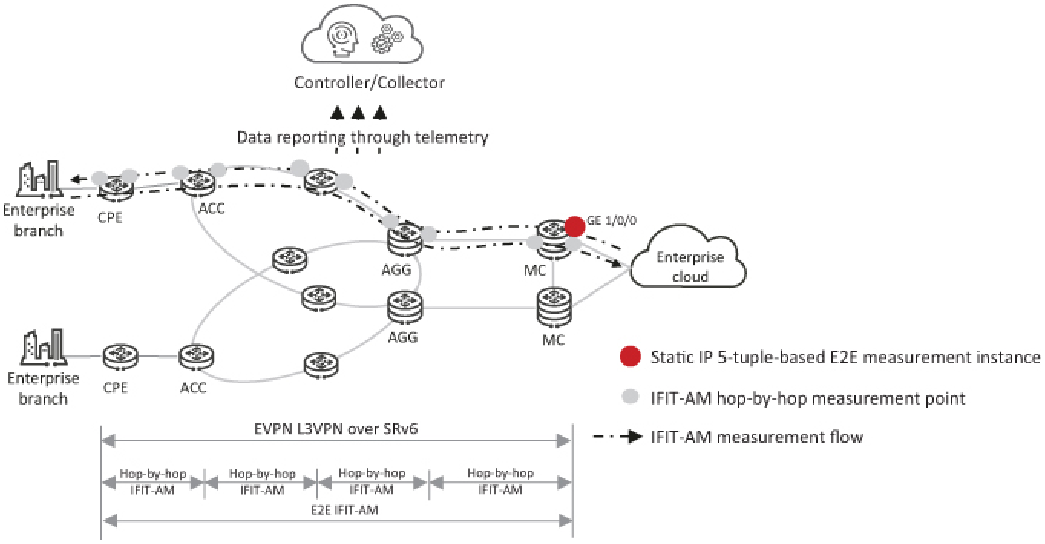


FIGURE 7.8 Deploying static IP 5-tuple-based IFIT-AM in a site-to-cloud private line scenario. ⏎

The configuration procedure is as follows:

1. Enable IFIT-AM on the devices on the IFIT-AM measurement path. The following uses the Customer Premises Equipment (CPE) as an example.

```
<CPE> system-view
[~CPE] ifit
[*CPE-ifit] node-id 1
[*CPE-ifit] commit
```

2. Configure a static IP 5-tuple-based IFIT-AM measurement instance on the corresponding MC.

```
<MC> system-view
[~MC] ifit
[*MC-ifit] node-id 2
[*MC-ifit] instance 1
[*MC-ifit-instance-1] flow bidirectional source-ipv6
2001:db8:1::1 destination-ipv6 2001:db8:2::1
vpn-instance vpna   //Create a measurement flow by
specifying the source and destination IP addresses in
the IP 5-tuple and the VPN instance.
[*MC-ifit-instance-1] binding interface
gigabitethernet 1/0/0   //Bind the measurement flow to
a specified user-side interface.
[*MC-ifit-instance-1] commit
```

Static IP 5-tuple-based IFIT-AM hop-by-hop measurement can be enabled if service exceptions occur. Enabling it requires only the corresponding configuration to be added to the original IFIT-AM measurement instance. No additional configuration is needed because the transit and egress nodes automatically learn and perform measurement based on the IFIT-AM header information.

```
[~MC-ifit-instance-1] measure-mode trace   //Enable
hop-by-hop measurement in the configured measurement
instance.
[*MC-ifit-instance-1] commit
```

7.2.5 VPN + Peer-Based IFIT-AM Deployment

VPN + peer-based IFIT-AM provides SLA measurement between VPN peers in E2E SRv6 EVPN scenarios. It supports multiple service scenarios, such as EVPN L3VPN, EVPN Virtual Private Wire Service (VPWS), and EVPN Virtual Private LAN Service (VPLS). The measurement objects are service flows of a specified VPN between two peers.

Figure 7.9 depicts a site-to-cloud private line scenario in which VPN + peer-based IFIT-AM is deployed on the user-side interface of a VPN to provide E2E measurement for services in the VPN. Hop-by-hop measurement can be enabled if the service SLA is abnormal or fault demarcation is required.

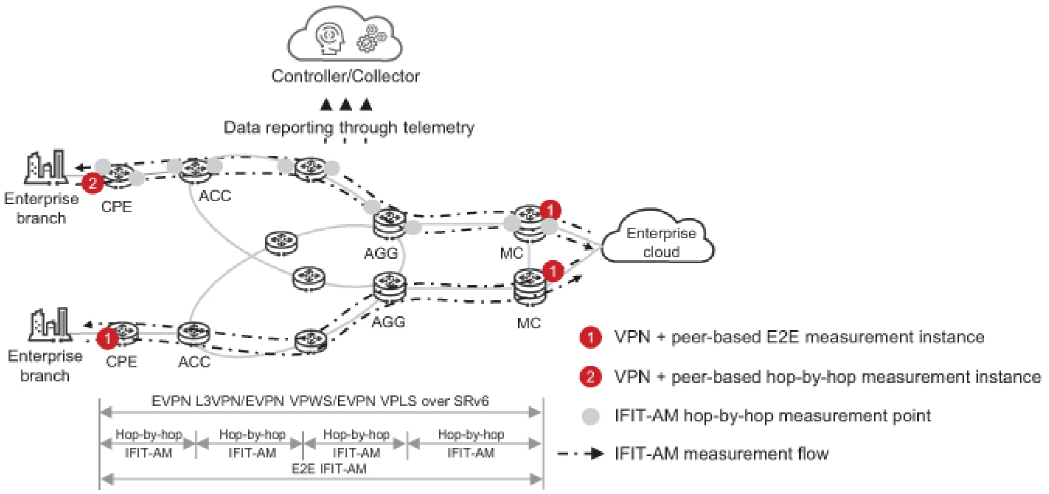

Controller/Collector
Data reporting through telemetry
Enterprise branch
CPE
ACC
AGG
MC
Enterprise cloud
AGG
MC
Enterprise branch
CPE
ACC
1 VPN + peer-based E2E measurement instance
2 VPN + peer-based hop-by-hop measurement instance
IFIT-AM hop-by-hop measurement point
IFIT-AM measurement flow
EVPN L3VPN/EVPN VPWS/EVPN VPLS over SRv6
Hop-by-hop IFIT-AM
Hop-by-hop IFIT-AM
Hop-by-hop IFIT-AM
Hop-by-hop IFIT-AM
E2E IFIT-AM

FIGURE 7.9 Deploying VPN + peer-based IFIT-AM in a site-to-cloud private line scenario. ⏎

The configuration procedure is as follows:

1. Enable IFIT-AM on the devices on the IFIT-AM measurement path. The following uses the CPE as an example.

```
<CPE> system-view
[~CPE] ifit
[*CPE-ifit] node-id 1
[*CPE-ifit] commit
```

2. Configure a VPN + peer-based IFIT-AM measurement instance on the corresponding MC. The configuration process varies according to scenarios.

   The configuration in the EVPN L3VPN scenario is as follows:

```
<MC> system-view
[~MC] ifit
[*MC-ifit] node-id 2
```

```
    [*MC-ifit] instance 1
    [*MC-ifit-instance-1] flow unidirectional source-ipv6
    any destination-ipv6 any vpn-instance vpna
    peer-locator 2001:DB8:60::1 64   //Create a
    measurement flow by specifying the VPN instance and
    peer locator address.
    [*MC-ifit-instance-1] binding interface
    gigabitethernet 1/0/0   //Bind the user-side interface
    to the instance.
    [*MC-ifit-instance-1] commit
```

The configuration in the EVPN VPWS scenario is as follows:

```
<MC> system-view
[~MC] ifit
[*MC-ifit] node-id 2
[*MC-ifit] instance 1
[*MC-ifit-instance-1] flow unidirectional
evpl-instance 1 peer-locator 2001:DB8:40::1 64   //
Create a measurement flow by specifying the Ethernet
Virtual Private Line (EVPL) instance and peer locator
address.
[*MC-ifit-instance-1] binding interface
gigabitethernet 1/0/1   //Bind the user-side interface
to the instance.
[*MC-ifit-instance-1] commit
```

The configuration in the EVPN VPLS scenario is as follows:

```
<MC> system-view
[~MC] ifit
[*MC-ifit] node-id 2
[*MC-ifit] instance 1
[*MC-ifit-instance-1] flow unidirectional evpn
vpn-instance evrf1 peer-locator 2001:DB8:40::1 64   //
Create a measurement flow by specifying the EVPN VPLS
instance and peer locator address.
[*MC-ifit-instance-1] binding interface
gigabitethernet 1/0/2   //Bind the user-side interface
to the instance.
[*MC-ifit-instance-1] commit
```

Hop-by-hop measurement can be enabled if a service exception occurs. The method for configuring it is similar to that for configuring static IP 5-tuple-based IFIT-AM. For details, see Section 7.2.4.

7.2.6 MAC Address-Based IFIT-AM Deployment

MAC address-based IFIT-AM is mainly used in EVPN VPLS scenarios to perform fine-grained SLA measurement for service flows or users identified by MAC addresses. It can be used for on-demand monitoring in scenarios such as key service assurance and service fault demarcation.

For example, in an enterprise site-to-Internet private line scenario, Layer 2 VPLS is typically used to transmit enterprise site-to-Internet services on the link from an Optical

Line Terminal (OLT) to a Broadband Remote Access Server (BRAS). The carrier can identify users based on MAC addresses, provide real-time SLAs for the users, and quickly demarcate and locate user service faults. This helps to improve the user experience. MAC address-based IFIT-AM is deployed on a VPLS user-side interface to provide E2E or hop-by-hop measurement at the service flow level based on the VPN, inbound interface, and source/destination MAC address, as shown in Figure 7.10.

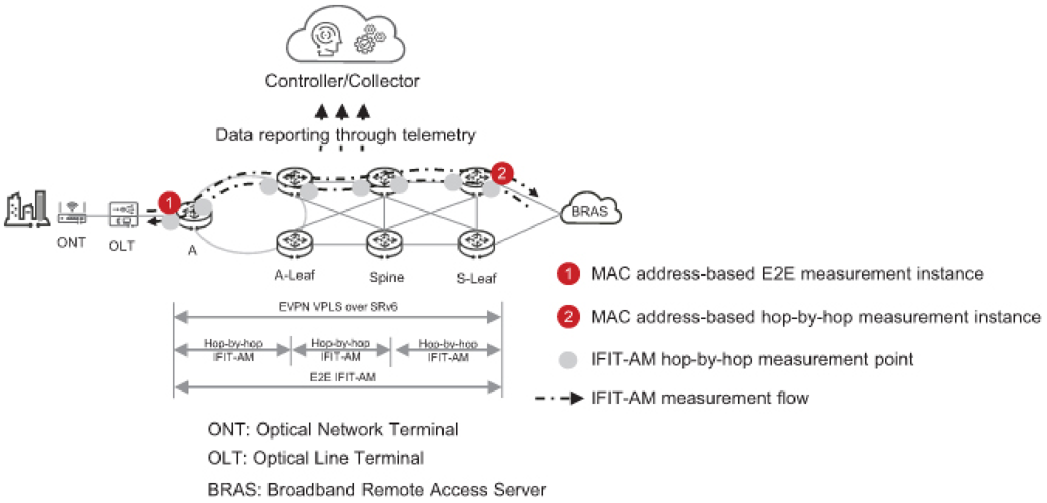

Controller/Collector
Data reporting through telemetry
ONT
OLT
A
A-Leaf
Spine
S-Leaf
BRAS
1
2
EVPN VPLS over SRv6
Hop-by-hop IFIT-AM
Hop-by-hop IFIT-AM
Hop-by-hop IFIT-AM
E2E IFIT-AM
1 MAC address-based E2E measurement instance
2 MAC address-based hop-by-hop measurement instance
● IFIT-AM hop-by-hop measurement point
IFIT-AM measurement flow
ONT: Optical Network Terminal
OLT: Optical Line Terminal
BRAS: Broadband Remote Access Server

FIGURE 7.10 Deploying MAC address-based IFIT-AM in an enterprise site-to-Internet private line scenario.

The configuration procedure is as follows:

1. Enable IFIT-AM on the devices on the IFIT-AM measurement path. The following uses device A as an example.

```
<A> system-view
[~A] ifit
[*A-ifit] node-id 1
[*A-ifit] commit
```

2. Configure a MAC address-based IFIT-AM measurement instance on the corresponding MC.

```
<MC> system-view
[~MC] ifit
[*MC-ifit] node-id 2
[*MC-ifit] instance 1
[*MC-ifit-instance-1]  flow unidirectional evpn
vpn-instance evrf1 destination-mac 00e0-fc12-3456   //
Create a measurement flow by specifying the MAC
address and EVPN VPLS instance.
[*MC-ifit-instance-1] binding interface
gigabitethernet 1/0/0
[*MC-ifit-instance-1] commit
```

Hop-by-hop measurement can be enabled if a service exception occurs. The method for configuring it is similar to that for configuring static IP 5-tuple-based IFIT-AM. For details, see Section 7.2.4.

7.2.7 Tunnel-Based IFIT-AM Deployment

Tunnel-based IFIT-AM (hop-by-hop) measurement can be enabled to demarcate a failure point if a tunnel-level SLA exception occurs. Fine-grained flow-level measurement (e.g., static IP 5-tuple- or MAC address-based IFIT-AM) can also be enabled to diagnose services carried over the tunnel. Tunnel-based IFIT-AM reduces the number of IFIT-AM measurement instances required for network monitoring and can be flexibly used together with flow-level measurement to improve network resource utilization. An SRv6 Policy is used as an example to describe tunnel-based IFIT-AM. SRv6 Policy-based IFIT-AM can measure traffic on each segment list of an SRv6 Policy and visualize tunnel-level SLAs.

Figure 7.11 depicts a financial private network scenario in which SRv6 Policy-based IFIT-AM is deployed on the ingress PE of a tunnel to provide E2E or hop-by-hop measurement at the segment list level of the tunnel.

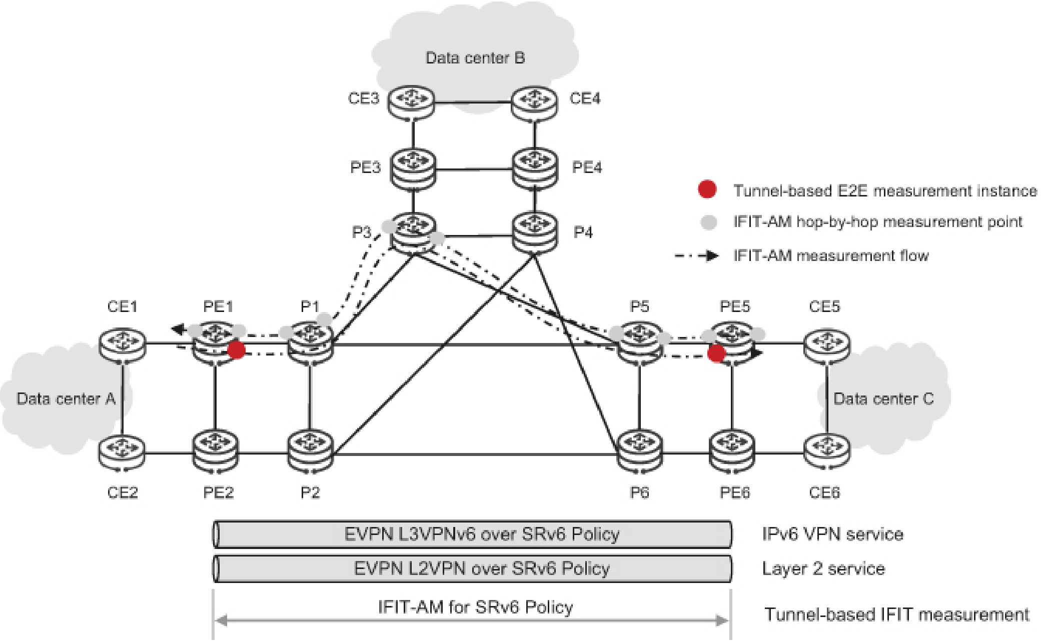


FIGURE 7.11 Deploying SRv6 Policy-based IFIT-AM in a financial private network scenario. ⏎

The configuration procedure is as follows:

1. Enable IFIT-AM on the devices on the IFIT-AM measurement path. The following uses P1 as an example.

```
<P1> system-view
[~P1] ifit
[*P1-ifit] node-id 1
[*P1-ifit] commit
```

2. Configure tunnel-based IFIT-AM on the headend of the SRv6 Policy. In this example, PE1 and PE5 are the headend and endpoint, respectively.

   ```
   <PE1> system-view
   [~PE1] segment-routing ipv6
   [*PE1-segment-routing-ipv6] srv6-te policy policy1
   endpoint 2001:DB8:3::5 color 10   //Configure an SRv6
   Policy destined for PE5.
   [*PE1-segment-routing-ipv6-policy-policy1] ifit
   loss-measure enable   //Enable tunnel-based IFIT-AM
   packet loss measurement.
   [*PE1-segment-routing-ipv6-policy-policy1] ifit
   delay-measure enable   //Enable tunnel-based IFIT-AM
   delay measurement.
   [*PE1-segment-routing-ipv6-policy-policy1] commit
   [~PE1-segment-routing-ipv6-policy-policy1] quit
   [~PE1-segment-routing-ipv6] quit
   ```

3. Query the IFIT-AM measurement instance information of the SRv6 Policy. In this example, the command output shows that multiple measurement instances are automatically generated based on the SRv6 segment lists.

   ```
   [~PE1] display ifit srv6-segment-list
   ------------------------------------------------------
   -------------------
   Flow Classification                     :
   srv6-segment-list   //The flow type is tunnel-based
   measurement flow.
   Instance Id                             : 1
   Flow Id                                 : 1572866
   Flow Type                               :
   unidirectional
   Loss Measure                            : enable
   Delay Measure                           : enable
   Measure Mode                            : e2e
   Interval                                : 30(s)
   Color                                   : 10   //Color
   attribute value of the SRv6 Policy.
   Segment List Id                         : 1   //ID of
   the segment list referenced by the candidate path of
   the SRv6 Policy.
   Binding SID                             :
   2001:DB8:100::800   //Binding SID of the SRv6
   Policy.
   Reverse Binding SID                     :
   2001:DB8:300::800   //Reverse binding SID of the SRv6
   Policy.
   EndPoint                                :
   2001:DB8:3::5   //Destination address of the SRv6
   Policy.
   ------------------------------------------------------
   -------------------
   Flow Classification                     :
   srv6-segment-list
   Instance Id                             : 2
   Flow Id                                 : 1572867
   ```

```
    Flow Type                                 :
    unidirectional
    Loss Measure                              : enable
    Delay Measure                             : enable
    Measure Mode                              : e2e
    Interval                                  : 30(s)
    Color                                     : 10
    Segment List Id                           : 2
    Binding SID                               :
    2001:DB8:100::801
    Reverse Binding SID                       :
    2001:DB8:300::801
    EndPoint                                  :
    2001:DB8:3::5
```

SRv6 Policy-based IFIT-AM hop-by-hop measurement can be enabled by adding the corresponding configuration to the SRv6 Policy if an exception occurs.

```
[~PE1] segment-routing ipv6
[~PE1-segment-routing-ipv6] srv6-te policy policy1
[*PE1-segment-routing-ipv6-policy-policy1] ifit
measure-mode trace   //Enable SRv6 Policy-based
IFIT-AM hop-by-hop measurement.
[*PE1-segment-routing-ipv6-policy-policy1] commit
```

7.3 IFIT-AM DEPLOYMENT ON THE CONTROLLER

This section describes how to deploy IFIT-AM on the controller, covering deployment preparations, subscription deployment, and measurement instance deployment. Huawei's iMaster NCE-IP (NCE for short) is used as the controller in this section. According to the application scenario and measurement granularity, IFIT-AM can be classified into the following types: dynamic learning-based IFIT-AM, static IP 5-tuple-based IFIT-AM, VPN + peer-based IFIT-AM, MAC address-based IFIT-AM, and tunnel-based IFIT-AM. The different measurement types apply to different scenarios and provide a reference for deploying IFIT-AM on the controller. This section also describes how to troubleshoot VPNs when abnormalities occur — for example, threshold rules can be configured for packet loss and delay to determine faults, an automatic hop-by-hop measurement policy can be used to demarcate faults, and performance indicators of NEs, links, and interfaces can be associated to assist fault diagnosis.

7.3.1 Deployment Preparations

Before deploying IFIT-AM, check whether the environment meets deployment requirements on NCE as follows.

7.3.1.1 Collecting Live Network Information

Collect server and device information to be configured on NCE. Table 7.1 describes the information to be configured on NCE and its purpose.

TABLE 7.1 Information To Be Configured on NCE and Its Purpose ⏎

Information	Purpose
NTP server information, including the IP address, encryption mode, digest type, key value index, and key value of the NTP server.	On the NCE management plane, add the NTP server deployed on the live network as the external clock source in order to synchronize time with NEs.
Base station controller names and interface IP addresses	The information is required when core network NEs are imported into NCE.
Base station names and service IP addresses	The information is required when base stations are imported into NCE. If one base station corresponds to multiple service IP addresses, it is recommended that a template be used for configuring and importing the information. A unique name must be configured for each base station.
Basic field information of NEs	The values of an NE's basic inventory fields are synchronized from NCE Manager, but the values of predefined and user-defined extended inventory fields are left blank. Therefore, collect the values as required and export the basic inventory fields as a file. Then, fill in the collected information (e.g., NE's longitude, latitude, and area information) in the file and import the file into the system.
NEs monitored by IFIT	The NEs that are to be monitored by IFIT and subscribed to by telemetry must be selected when devices to be monitored on NCE are added.
VPN instances and NEs	The VPN instance to be monitored, flow learning mode, VPN instance's egress NE, egress NE's interface, and more must be selected when a base station flow monitoring instance is configured.

7.3.1.2 Clock Synchronization between NCE and NEs

To enable unified calculation and display the SLA information of service flows measured by IFIT in real time, NCE needs to synchronize the NTP clock with NEs. To implement this function, configure an external NTP clock source as the NTP server on NCE.

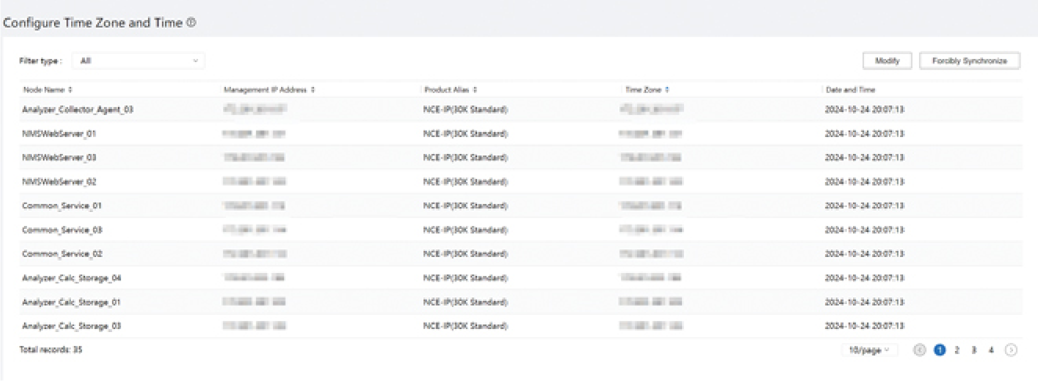
7.3.1.2.1 Precautions

The recommended Operating System (OS) for the NTP server is Linux — do not use a server or Virtual Machine (VM) running Windows as the NTP server. In addition, do not use the IP address of the NTP server for the NTP management plane node — i.e., Openness Management Platform (OMP) node — or service node, as doing so may result in the time of each node being incorrect. The following points should also be noted:

- If multiple NTP servers are configured, ensure that the time is consistent between them. Otherwise, abnormalities may occur in the NTP service.
- Do not set the clock source in a circular manner. For example, do not set A as the clock source of B, B as the clock source of C, and C as the clock source of A.
- If physical hosts have been configured to trace the time of a specified NTP server during NCE deployment, ensure that the NTP server configured on the management plane is the same as the traced NTP server.
- If the whitelist mechanism has been configured for an external NTP clock source, the IP address of the OMP node must be added to the whitelist so that the OMP node can access the NTP server.
- After the NTP server is reconfigured, management-plane applications and data, along with historical backup data of the management and product node OSs, become invalid. As a result, the backup files generated before the NTP server is configured are unavailable.

7.3.1.2.2 Procedure

1. Log in to the NCE management plane at **https://***IP address of the management plane***:31945**.
2. Choose **Maintenance** > **Time Management** > **Configure Time Zone and Time** from the main menu. The **Configure Time Zone and Time** dialog box is displayed, as shown in Figure 7.12. Check whether the time zone, date, and time of each node are consistent with those on the NE side.



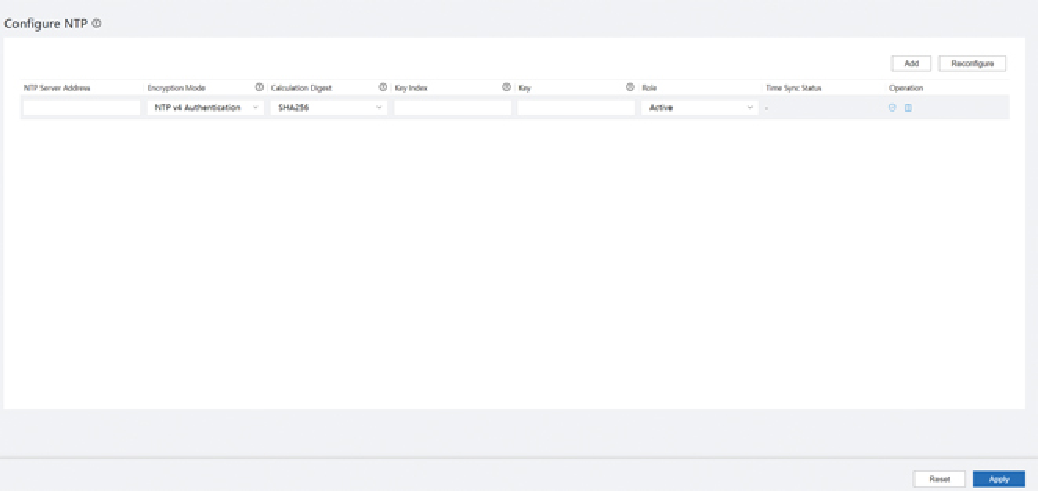
Node Name	Management IP Address	Product Alias	Time Zone	Date and Time
Analyzer_Collector_Agent_03	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13
NMSWebServer_01	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13
NMSWebServer_03	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13
NMSWebServer_02	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13
Common_Service_01	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13
Common_Service_03	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13
Common_Service_02	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13
Analyzer_Calc_Storage_04	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13
Analyzer_Calc_Storage_01	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13
Analyzer_Calc_Storage_03	[redacted]	NCE-IP(30K Standard)	[redacted]	2024-10-24 20:07:13

Total records: 35

FIGURE 7.12 Configure time zone and time dialog box. ⏎

3. If there are inconsistencies, synchronize the time zone and time of the OMP and service nodes again and rectify the NTP synchronization failure as prompted.

4. Choose **Maintenance** > **Time Management** > **Configure NTP** from the main menu. The **Configure NTP** dialog box is displayed, as shown in Figure 7.13.



NTP Server Address	Encryption Mode	Calculation Digest	Key Index	Key	Role	Time Sync Status	Operation
	NTP v4 Authentication	SHA256			Active	-	

FIGURE 7.13 Configure NTP dialog box. ⏎

CAUTION

Configuring time synchronization between NCE nodes and the NTP server will cause NCE to restart, resulting in service interruptions. Therefore, ensure that NTP time synchronization is performed at an appropriate time.

7.3.2 Adding Global Configurations

NCE supports globally enabling IFIT for devices and subscribing to all IFIT collection objects they support, simplifying operations and configurations.

The following describes how to enable IFIT globally on devices.

1. Open the Network Digital Map app and choose **Global Settings** > **Intelligent Analysis Settings** > **Congestion View** > **Monitoring Management** > **Monitoring Instance Management**, as shown in Figure 7.14.

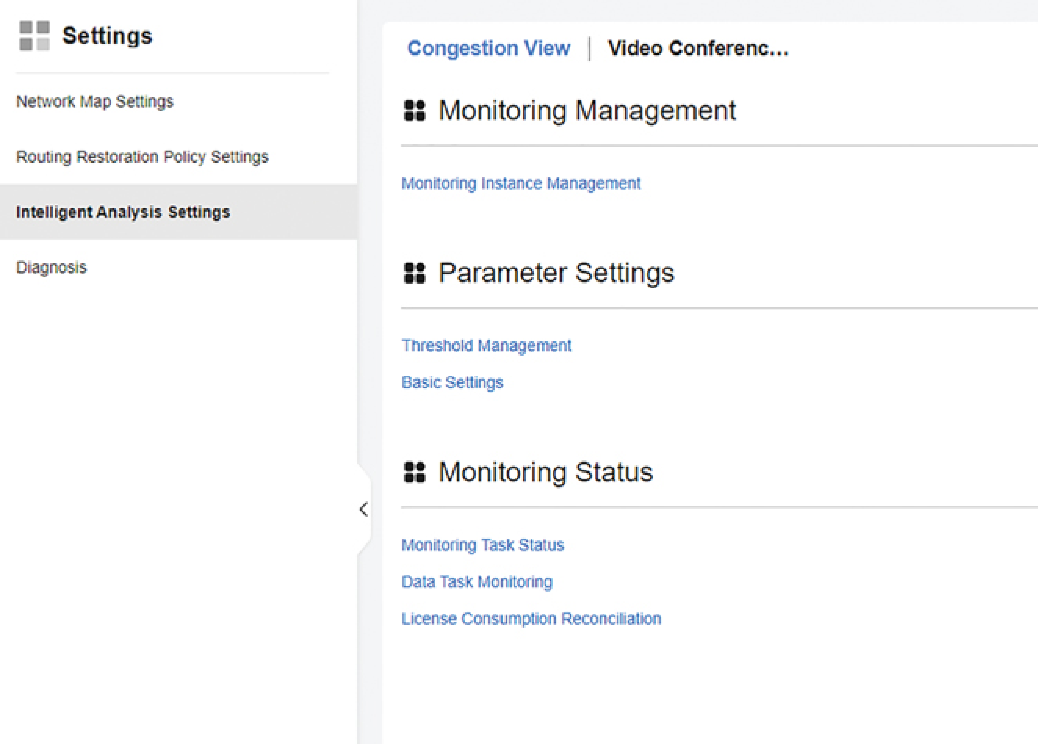
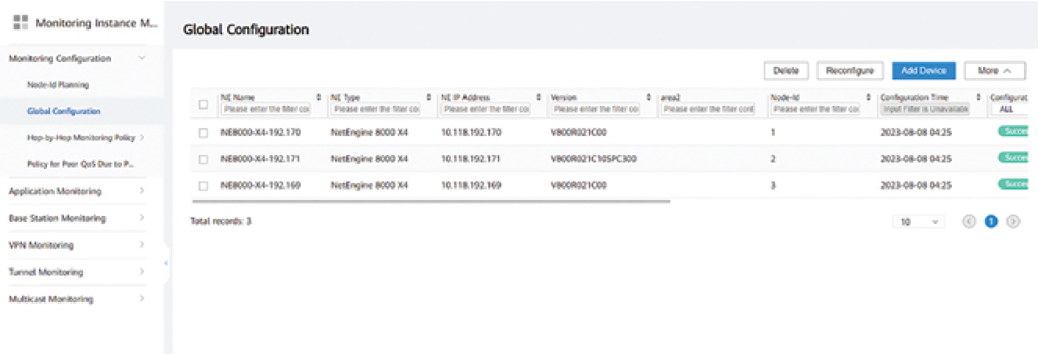


FIGURE 7.14 Selecting monitoring instance management. ⏎

2. Choose **Monitoring Configuration** > **Global Configuration**, as shown in Figure 7.15.



Global Configuration

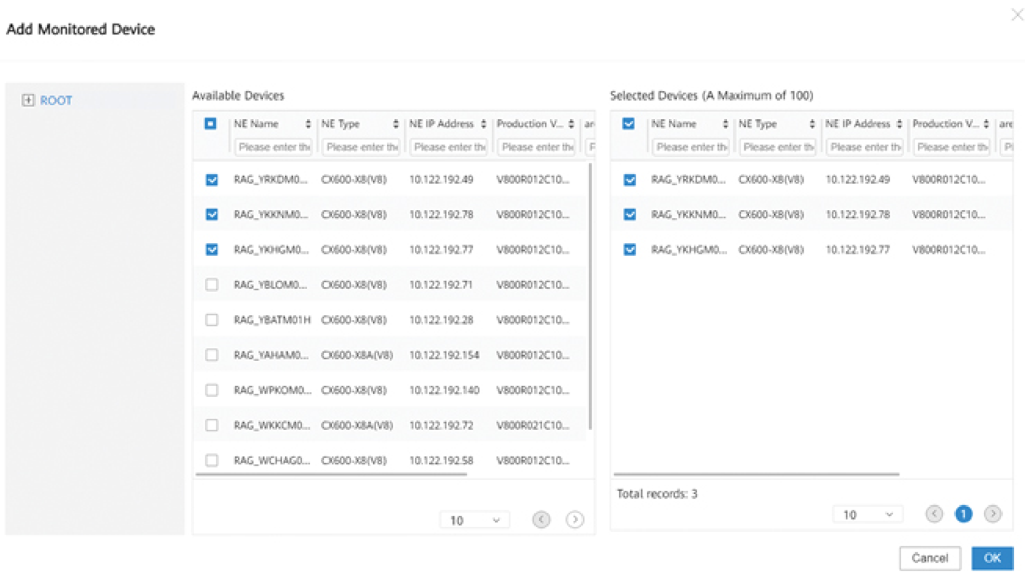
□	NE Name	NE Type	NE IP Address	Version	area2	Node-Id	Configuration Time	Configurat
	Please enter the filter co	Please enter the filter co	Please enter the filter co	Please enter the filter co	Please enter the filter cond	Please enter the filter co	Input Filter is Unavailable	ALL
□	NE8000-X4-192.170	NetEngine 8000 X4	10.118.192.170	V800R021C00		1	2023-08-08 04:25	Succe
□	NE8000-X4-192.171	NetEngine 8000 X4	10.118.192.171	V800R021C10SPC300		2	2023-08-08 04:25	Succe
□	NE8000-X4-192.169	NetEngine 8000 X4	10.118.192.169	V800R021C00		3	2023-08-08 04:25	Succe

Total records: 3

10

FIGURE 7.15 Selecting global configuration. ⏎

3. Click **Add Device**. The **Add Monitored Device** dialog box is displayed. In the **Available Devices** area, select the devices for which IFIT needs to be enabled based on the service plan, as shown in Figure 7.16.



Add Monitored Device

Available Devices

☐	NE Name	NE Type	NE IP Address	Production V...	ar
☑	RAG_YRKDM0...	CX600-X8(V8)	10.122.192.49	V800R012C10...	
☑	RAG_YKKNM0...	CX600-X8(V8)	10.122.192.78	V800R012C10...	
☑	RAG_YKHGM0...	CX600-X8(V8)	10.122.192.77	V800R012C10...	
☐	RAG_YBLOM0...	CX600-X8(V8)	10.122.192.71	V800R012C10...	
☐	RAG_YBATM01H	CX600-X8(V8)	10.122.192.28	V800R012C10...	
☐	RAG_YAHAM0...	CX600-X8A(V8)	10.122.192.154	V800R012C10...	
☐	RAG_WPKOM0...	CX600-X8(V8)	10.122.192.140	V800R012C10...	
☐	RAG_WKKCM0...	CX600-X8A(V8)	10.122.192.72	V800R021C10...	
☐	RAG_WCHAG0...	CX600-X8(V8)	10.122.192.58	V800R012C10...	

10

Selected Devices (A Maximum of 100)

☑	NE Name	NE Type	NE IP Address	Production V...	are
☑	RAG_YRKDM0...	CX600-X8(V8)	10.122.192.49	V800R012C10...	
☑	RAG_YKKNM0...	CX600-X8(V8)	10.122.192.78	V800R012C10...	
☑	RAG_YKHGM0...	CX600-X8(V8)	10.122.192.77	V800R012C10...	

Total records: 3

10

FIGURE 7.16 Added monitored device dialog box. ⏎

4. Click **OK**.

NOTE

- IFIT must be enabled globally before any type of IFIT measurement can be performed.
- For an NE managed by NCE through a user-defined management VPN, it is necessary to ensure that telemetry reports IFIT data using this VPN. To do so, open the Network Performance Analysis app and choose **Settings** > **Monitored Object Management** > **Resource Management** from the main menu. In the navigation pane, choose **Inventory Import and Export** > **IP** > **Equipment** > **NE**. Then, configure the management VPN of the NE in the corresponding **VPN Name** column.

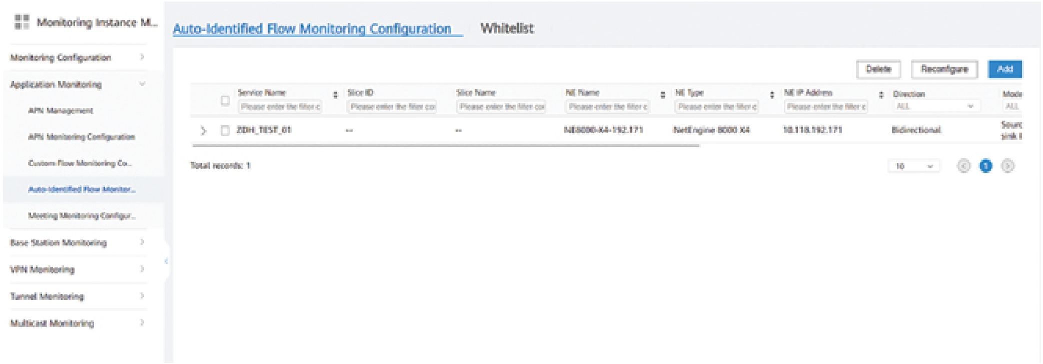
7.3.3 Dynamic Learning-Based IFIT-AM Deployment

Dynamic learning-based IFIT-AM is widely used on IP RAN mobile transport networks. After flow learning is enabled on UNIs of core-side devices, the devices automatically learn and identify flows from or to base stations.

7.3.3.1 Configuring Auto-Identified Flow Monitoring

Auto-identified flow monitoring is configured as follows:

1. Open the Network Digital Map app and choose **Global Settings** > **Intelligent Analysis Settings** > **Congestion View** > **Monitoring Management** > **Monitoring Instance Management**. For details, see the content related to monitoring instance management in Section 7.3.2.
2. In the navigation pane, choose **Application Monitoring** > **Auto-Identified Flow Monitoring Configuration**. Then, click the **Auto-Identified Flow Monitoring Configuration** tab, as shown in Figure 7.17.

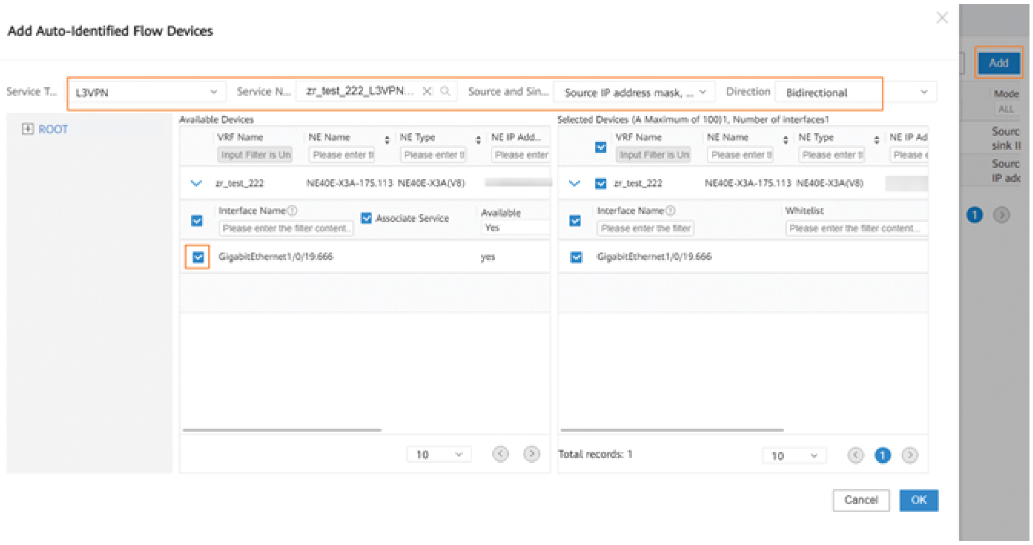


	Service Name	Slice ID	Slice Name	NE Name	NE Type	NE IP Address	Direction	Mode
	Please enter the filter c	Please enter the filter co	Please enter the filter co	Please enter the filter c	Please enter the filter c	Please enter the filter c	ALL	ALL
☐	ZDH_TEST_01	--	--	NE8000-X4-192.171	NetEngine 8000 X4	10.118.192.171	Bidirectional	Sourc sink I

Total records: 1 10

FIGURE 7.17 Auto-identified flow monitoring configuration tab page. ⏎

3. Click **Add**. The **Add Auto-Identified Flow Devices** dialog box is displayed, as shown in Figure 7.18. Select a service type from the **Service Type** drop-down list, enter a service name in the **Service Name** text box, and select **Source IP address mask, sink IP address** or **Source IP address, sink IP address** from the **Source and Sink IP Address Matching Model** drop-down list. Then, select **Unidirectional** or **Bidirectional** from the **Direction** drop-down list. In the **Available Devices** area, select a desired NE and interface to deliver the auto-identified flow monitoring configuration to the NE. Finally, click **OK**.



Service T... L3VPN Service N... zr_test_222_L3VPN... Source and Sin... Source IP address mask, ... Direction Bidirectional

Available Devices

	VRF Name	NE Name	NE Type	NE IP Add...
	Input Filter is Un	Please enter t	Please enter t	Please enter
	zr_test_222	NE40E-X3A-175.113	NE40E-X3A(V8)	

☑	Interface Name	☑ Associate Service	Available
	Please enter the filter content.		Yes
☑	GigabitEthernet1/0/19.666		yes

10

Selected Devices (A Maximum of 100)1, Number of interfaces1

☑	VRF Name	NE Name	NE Type	NE IP Ad
	Input Filter is Un	Please enter t	Please enter t	Please e
☑	zr_test_222	NE40E-X3A-175.113	NE40E-X3A(V8)	

☑	Interface Name	Whitelist
	Please enter the filter	Please enter the filter content...
☑	GigabitEthernet1/0/19.666	

Total records: 1 10

FIGURE 7.18 Add auto-identified flow devices dialog box. ⏎

4. In the **High Risk** dialog box, select **I have read the message and fully understood the operation impacts on services.** and click **OK**.
5. When the message “Operation completed” is displayed, click **OK**.

NOTE

The two matching models are as follows:

- **Source IP address mask, sink IP address**: Instances created in this model perform matching based on the source IP address mask of flows and exact matching based on the destination IP address of flows.

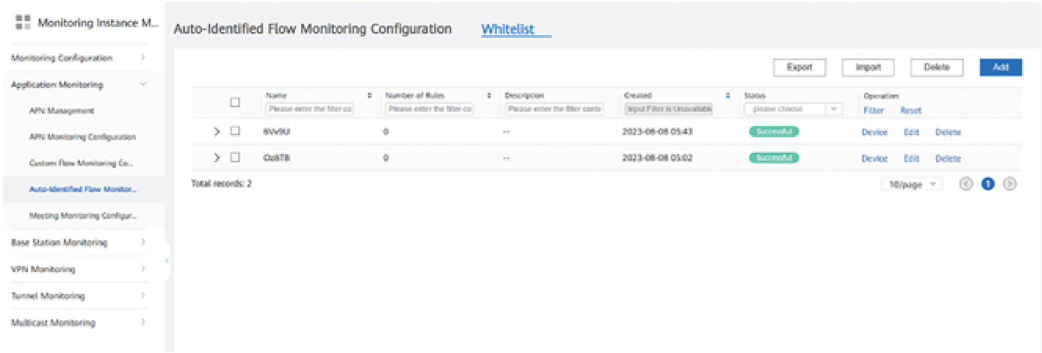
- **Source IP address, sink IP address**: Instances created in this model perform exact matching based on the source and destination IP addresses of flows.

A whitelist can also be configured so that a device reports only the flows that match the specified rules during dynamic learning. If no whitelist is configured, all flows are monitored by default.

7.3.3.2 Configuring Whitelist-Based Auto-Identified Flow Monitoring

Whitelist-based auto-identified flow monitoring is configured as follows:

1. In the navigation pane, choose **Application Monitoring** > **Auto-Identified Flow Monitoring Configuration**. Then, click the **Whitelist** tab, as shown in Figure 7.19.

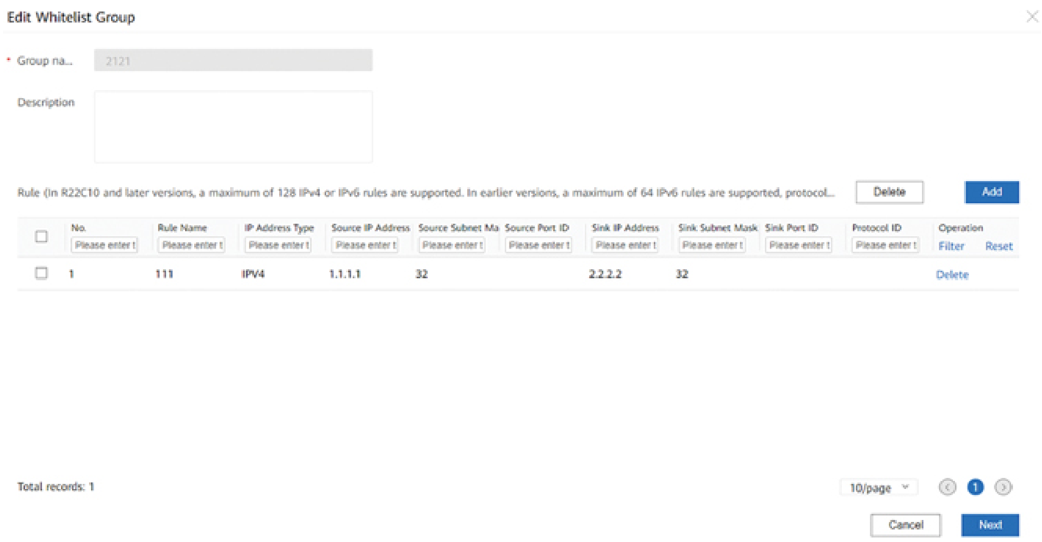


□	Name	Number of Rules	Description	Created	Status	Operation
	Please enter the filter co	Please enter the filter co	Please enter the filter conte	Input Filter is Unavailable	please choose	Filter Reset
> □	6Vv9U	0	--	2023-08-08 05:43	Successful	Device Edit Delete
> □	Oz8TB	0	--	2023-08-08 05:02	Successful	Device Edit Delete

Total records: 2

FIGURE 7.19 Whitelist tab page. ⏎

2. Click **Add** to add a whitelist group. In the **Edit Whitelist Group** dialog box (shown in Figure 7.20), set the group name and click **Add** to add a rule. Enter the source and sink IP addresses (IPv4 or IPv6), masks, and port IDs as required, and set the protocol ID.



* Group na... 2121

Description

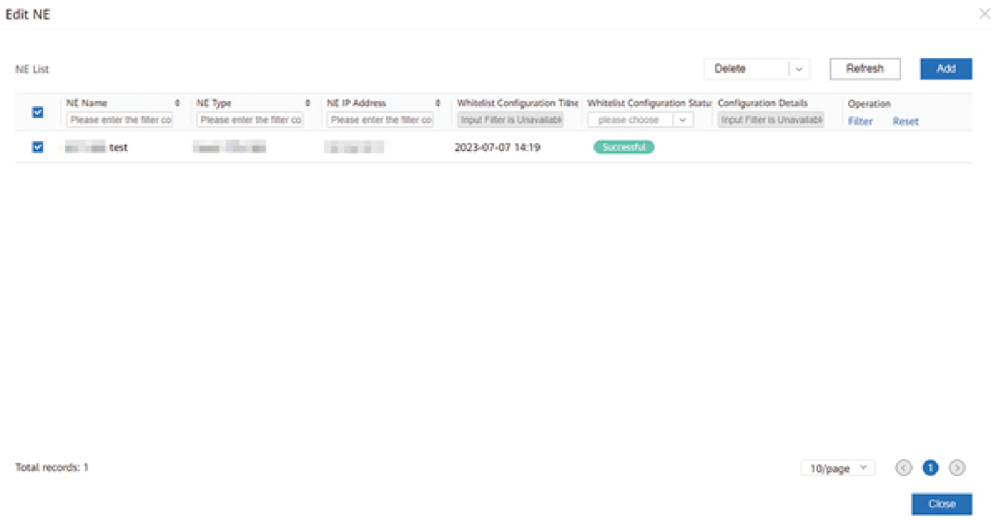
Rule (In R22C10 and later versions, a maximum of 128 IPv4 or IPv6 rules are supported. In earlier versions, a maximum of 64 IPv6 rules are supported, protocol...

□	No.	Rule Name	IP Address Type	Source IP Address	Source Subnet Ma	Source Port ID	Sink IP Address	Sink Subnet Mask	Sink Port ID	Protocol ID	Operation
	Please enter t	Please enter t	Please enter t	Please enter t	Please enter t	Please enter t	Please enter t	Please enter t	Please enter t	Please enter t	Filter Reset
□	1	111	IPV4	1.1.1.1	32		2.2.2.2	32			Delete

Total records: 1

FIGURE 7.20 Edit whitelist group dialog box. ⏎

3. After a whitelist group is created, click **Device** in the **Operation** column for the new whitelist group. On the **Edit NE** dialog box shown in Figure 7.21, click **Add** to add a desired device.



Edit NE

NE List

Delete | Refresh | Add

☑	NE Name	NE Type	NE IP Address	Whitelist Configuration Time	Whitelist Configuration Status	Configuration Details	Operation
	Please enter the filter co	Please enter the filter co	Please enter the filter co	Input Filter is Unavailable	please choose	Input Filter is Unavailable	Filter Reset
☑	[redacted] test	[redacted]	[redacted]	2023-07-07 14:19	Successful		

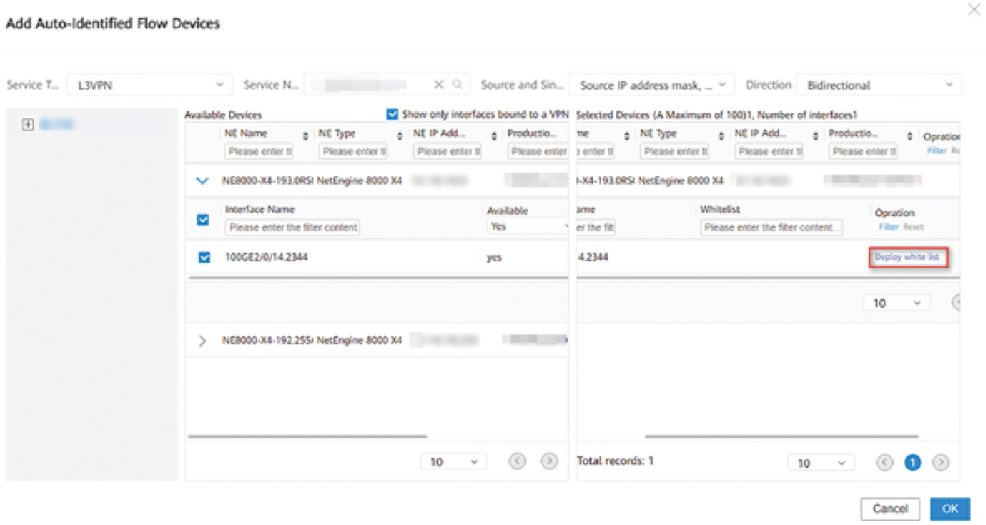
Total records: 1

10/page

Close

FIGURE 7.21 Edit NE dialog box. ⏎

4. Return to the **Auto-Identified Flow Monitoring Configuration** tab page, and click **Add**. In the **Add Auto-Identified Flow Devices** dialog box, select the desired devices and interfaces, and then click **Deploy white list** in the **Selected Devices** area to select the desired whitelist group, as shown in Figure 7.22. After the whitelist group is configured, flows passing through the corresponding interfaces of the devices are matched based on the configured rules.



Add Auto-Identified Flow Devices

Service T... L3VPN | Service N... [redacted] | Source and Sin... Source IP address mask, ... | Direction Bidirectional

Available Devices ☑ Show only interfaces bound to a VPN

NE Name	NE Type	NE IP Add...	Productio...
Please enter t	Please enter t	Please enter t	Please enter
NE8000-X4-193.0RSI	NetEngine 8000 X4	[redacted]	[redacted]

☑	Interface Name	Available
	Please enter the filter content	Yes
☑	100GE2/0/14.2344	yes

NE Name	NE Type	NE IP Add...	Productio...
NE8000-X4-192.255I	NetEngine 8000 X4	[redacted]	[redacted]

Selected Devices (A Maximum of 100)1, Number of interfaces1

...me	NE Type	NE IP Add...	Productio...	Opration
...enter t	Please enter t	Please enter t	Please enter t	Filter Re...
...-X4-193.0RSI	NetEngine 8000 X4	[redacted]	[redacted]	

...ame	Whitelist	Opration
...er the filt	Please enter the filter content...	Filter Reset
...4.2344		Deploy white list

Total records: 1

Cancel | OK

FIGURE 7.22 Add auto-identified flow devices dialog box. ⏎

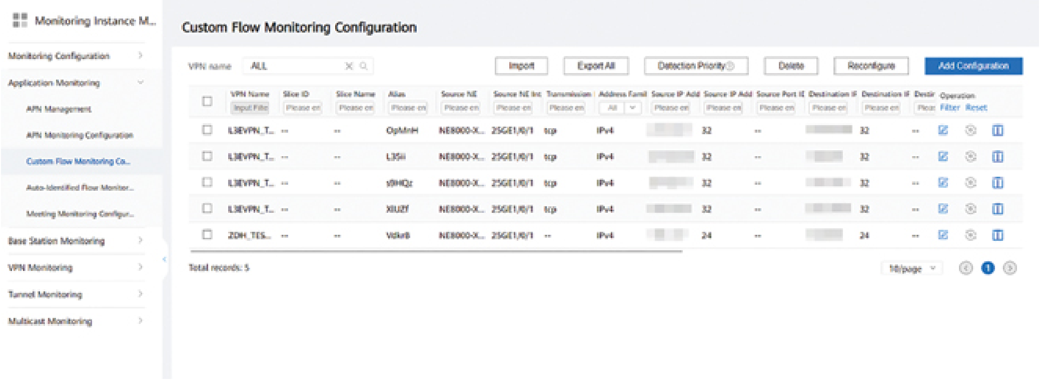
7.3.4 Static IP 5-Tuple-Based IFIT-AM Deployment

In static IP 5-tuple-based monitoring, also referred to as custom flow monitoring, the characteristics of flows to be monitored through IFIT can be specified. In addition, IFIT monitoring instances can be created on the GUI or in batches by importing a file.

7.3.4.1 Creating Instances on the GUI

Instances are created on the GUI as follows:

1. Open the Network Digital Map app and choose **Global Settings** > **Intelligent Analysis Settings** > **Congestion View** > **Monitoring Management** > **Monitoring Instance Management**. For details, see the content related to monitoring instance management in Section 7.3.2.
2. In the navigation pane, choose **Application Monitoring** > **Custom Flow Monitoring Configuration**, as shown in Figure 7.23.



Custom Flow Monitoring Configuration

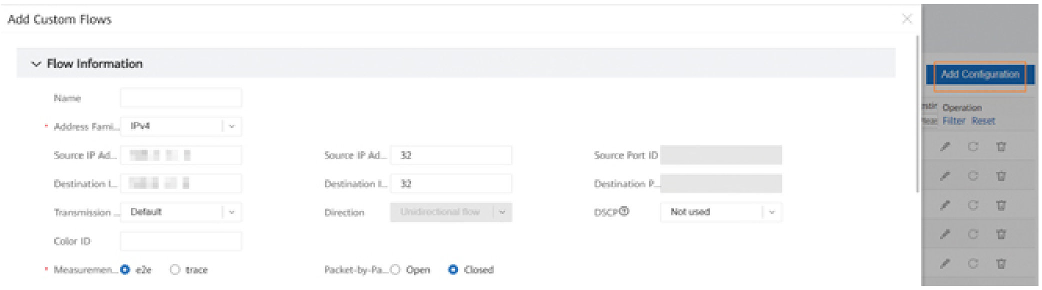
VPN name: ALL

	VPN Name	Slice ID	Slice Name	Alias	Source NE	Source NE Inc	Transmission	Address Famil	Source IP Add	Source IP Add	Source Port I[	Destination IF	Destination IF	Destir	Operation
	Input Filte	Please en	Please en	Please en	Please en	Please en	Please en	All	Please en	Please en	Please en	Please en	Please en	Pleas	Filter Reset
□	L3EVPN_T...	--	--	OpMnH	NE8000-X...	25GE1/0/1	tcp	IPv4	[redacted]	32	--	[redacted]	32	--	
□	L3EVPN_T...	--	--	L35ii	NE8000-X...	25GE1/0/1	tcp	IPv4	[redacted]	32	--	[redacted]	32	--	
□	L3EVPN_T...	--	--	s9HQz	NE8000-X...	25GE1/0/1	tcp	IPv4	[redacted]	32	--	[redacted]	32	--	
□	L3EVPN_T...	--	--	XIUZf	NE8000-X...	25GE1/0/1	tcp	IPv4	[redacted]	32	--	[redacted]	32	--	
□	2DH_TES...	--	--	Vdkr8	NE8000-X...	25GE1/0/1	--	IPv4	[redacted]	24	--	[redacted]	24	--	

Total records: 5

FIGURE 7.23 Selecting custom flow monitoring configuration. ⏎

3. Click **Add Configuration**. The **Add Custom Flows** dialog box is displayed, as shown in Figure 7.24. In the **Flow Information** area, set **Name**, **Address Family**, **Source IP Address**, **Source IP Address Mask**, **Source Port ID**, **Destination IP Address**, **Destination IP Address Mask**, **Destination Port ID**, **Transmission Protocol**, **Direction**, **DSCP**, **Measurement Mode**, and **Packet-by-Packet Delay**.



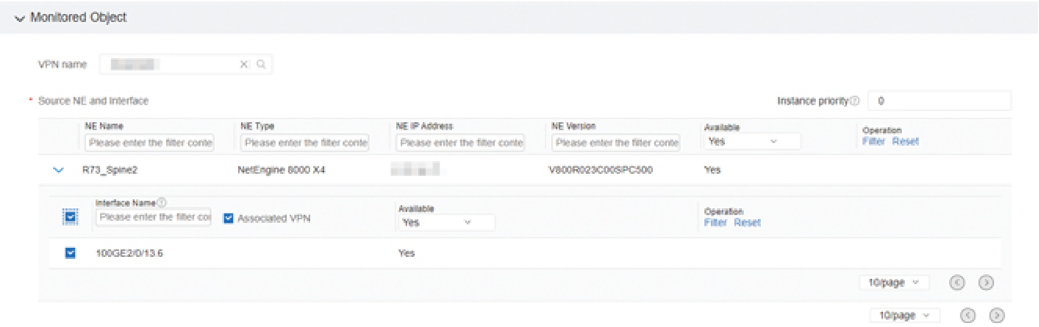
Add Custom Flows

Flow Information

Name:
* Address Fami...: IPv4
Source IP Ad...: [redacted] | Source IP Ad...: 32 | Source Port ID:
Destination I...: [redacted] | Destination I...: 32 | Destination P...:
Transmission ...: Default | Direction: Unidirectional flow | DSCP: Not used
Color ID:
* Measuremen...: ● e2e ○ trace | Packet-by-Pa...: ○ Open ● Closed

FIGURE 7.24 Add custom flows dialog box. ⏎

4. In the **Monitored Object** area, set the VPN name in the **VPN name** text box, expand an NE and select an interface from the **Source NE and Interface** list, and then set **Instance priority**, as shown in Figure 7.25.


Monitored Object

VPN name [redacted]

* Source NE and Interface — Instance priority 0

NE Name	NE Type	NE IP Address	NE Version	Available	Operation
Please enter the filter conte	Please enter the filter conte	Please enter the filter conte	Please enter the filter conte	Yes	Filter Reset
R73_Spine2	NetEngine 8000 X4	[redacted]	V800R023C00SPC500	Yes	

☑	Interface Name	☑ Associated VPN	Available	Operation
	Please enter the filter co		Yes	Filter Reset
☑	100GE2/0/13.6		Yes	

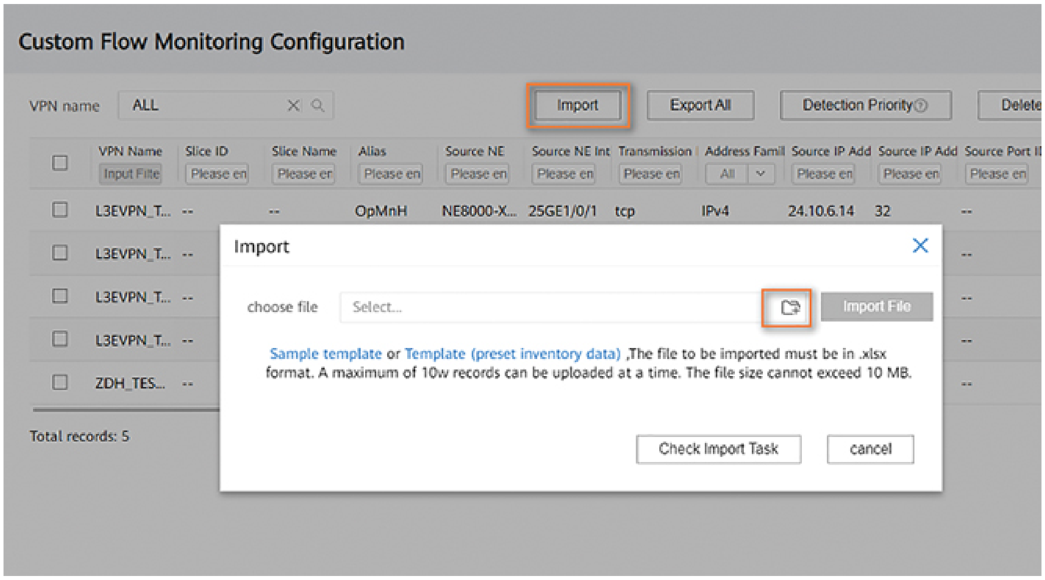
FIGURE 7.25 Configuring monitored objects. ⏎

5. Confirm the information, submit the request, and wait for the system to complete the operation.

7.3.4.2 Creating Instances in Batches

Instances are created in batches as follows:

1. Open the Network Digital Map app and choose **Global Settings** > **Intelligent Analysis Settings** > **Congestion View** > **Monitoring Management** > **Monitoring Instance Management**. For details, see the content related to monitoring instance management in Section 7.3.2.
2. In the navigation pane, choose **Application Monitoring** > **Custom Flow Monitoring Configuration**. Then, click **Import**. The **Import** dialog box is displayed, as shown in Figure 7.26.


Custom Flow Monitoring Configuration

VPN name ALL

□	VPN Name	Slice ID	Slice Name	Alias	Source NE	Source NE Int	Transmission	Address Famil	Source IP Add	Source IP Add	Source Port I
	Input Filte	Please en	Please en	Please en	Please en	Please en	Please en	All	Please en	Please en	Please en
□	L3EVPN_T...	--	--	OpMnH	NE8000-X...	25GE1/0/1	tcp	IPv4	24.10.6.14	32	--
□	L3EVPN_T...	--									--
□	L3EVPN_T...	--									--
□	L3EVPN_T...	--									--
□	ZDH_TES...	--									--

Total records: 5

Import

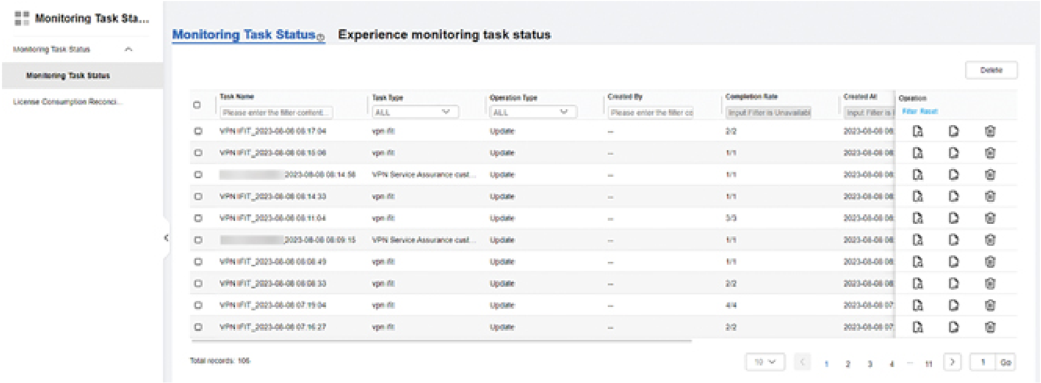
choose file Select...

Sample template or Template (preset inventory data) ,The file to be imported must be in .xlsx format. A maximum of 10w records can be uploaded at a time. The file size cannot exceed 10 MB.

FIGURE 7.26 Import dialog box. ⏎

3. Download the sample template file to be imported from the provided hyperlink, set related parameters in the file, and then save it.

4. Click **Import File** to import the file. The import is complete after the message “File uploaded successfully” is displayed.
5. Click **Check Import Task** or choose **Settings** > **Monitoring Status** > **Monitoring Task Status** from the main menu. On the **Monitoring Task Status** tab page (shown in Figure 7.27), filter tasks by **Task Name**, **Task Type**, **Operation Type**, **Created By**, etc. Then, select a desired task and click the **View Details**, **Export**, or **Delete** icon in the **Operation** column as required.



○	Task Name	Task Type	Operation Type	Created By	Completion Rate	Created At	Operation
○	Please enter the filter content...	ALL	ALL	Please enter the filter co	Input Filter is Unavailabl	Input Filter is	Filter Reset
○	VPN IFIT_2023-08-08 08:17:04	vpn ifit	Update	--	2/2	2023-08-08 08:	
○	VPN IFIT_2023-08-08 08:15:06	vpn ifit	Update	--	1/1	2023-08-08 08:	
○	[redacted] 2023-08-08 08:14:56	VPN Service Assurance cust...	Update	--	1/1	2023-08-08 08:	
○	VPN IFIT_2023-08-08 08:14:33	vpn ifit	Update	--	1/1	2023-08-08 08:	
○	VPN IFIT_2023-08-08 08:11:04	vpn ifit	Update	--	3/3	2023-08-08 08:	
○	[redacted] 2023-08-08 08:09:15	VPN Service Assurance cust...	Update	--	1/1	2023-08-08 08:	
○	VPN IFIT_2023-08-08 08:08:49	vpn ifit	Update	--	1/1	2023-08-08 08:	
○	VPN IFIT_2023-08-08 08:08:33	vpn ifit	Update	--	2/2	2023-08-08 08:	
○	VPN IFIT_2023-08-08 07:19:04	vpn ifit	Update	--	4/4	2023-08-08 07:	
○	VPN IFIT_2023-08-08 07:16:27	vpn ifit	Update	--	2/2	2023-08-08 07:	

Total records: 106

FIGURE 7.27 Monitoring task status tab page. ⏎

6. In the **Custom Flow Monitoring Configuration** dialog box, click **Export All**. After the system exports the existing instances to a file, the delivered instances can be searched for and compared.

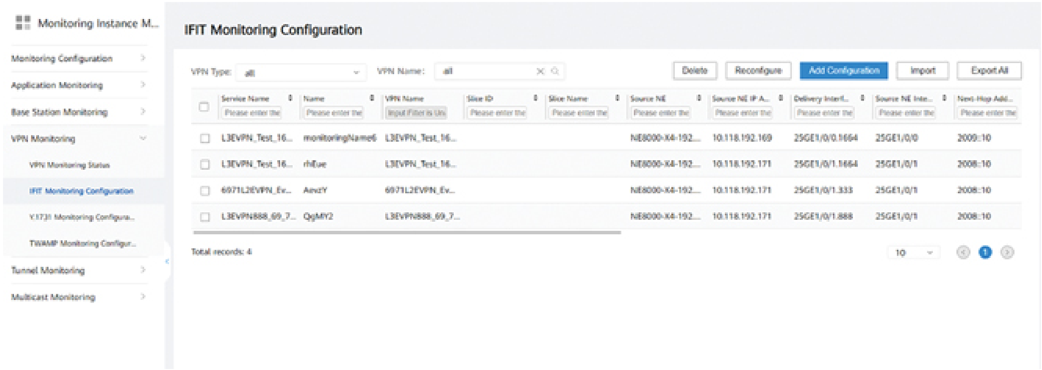
7.3.5 VPN + Peer-Based IFIT-AM Deployment

VPN + peer-based IFIT-AM is widely used in enterprise WAN scenarios (including government and finance) and carrier private line service monitoring scenarios. The measurement objects are service flows of a specified VPN between two peers. IFIT monitoring instances can be created for EVPN L3VPN, EVPN VPWS, and EVPN VPLS on the GUI or in batches by importing a file.

7.3.5.1 Creating Instances on the GUI

Instances are created on the GUI as follows:

1. Open the Network Digital Map app and choose **Global Settings** > **Intelligent Analysis Settings** > **Congestion View** > **Monitoring Management** > **Monitoring Instance Management**. For details, see the content related to monitoring instance management in Section 7.3.2.
2. In the navigation pane, choose **VPN Monitoring** > **IFIT Monitoring Configuration**, as shown in Figure 7.28.



IFIT Monitoring Configuration

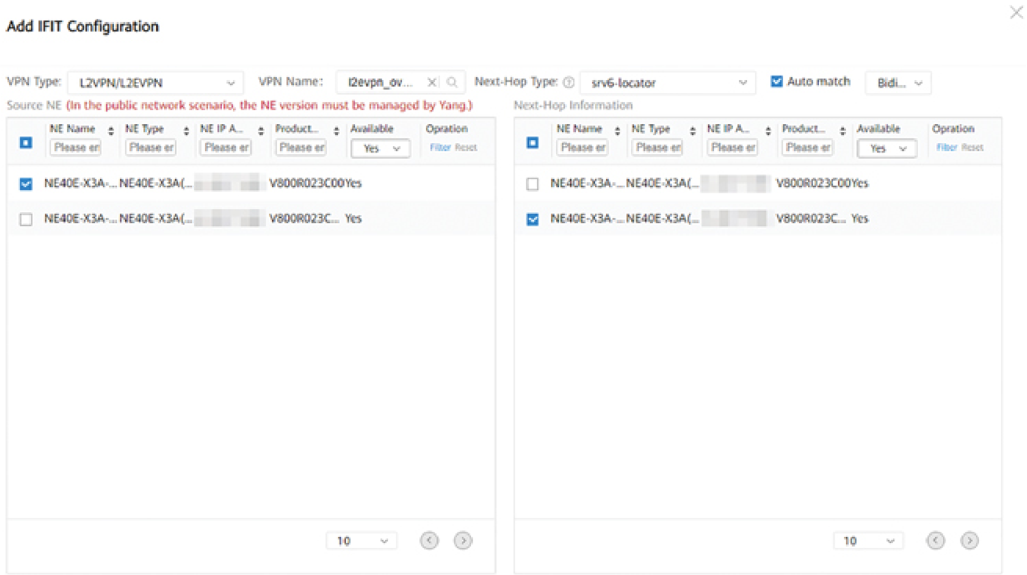
VPN Type: all   VPN Name: all

□	Service Name	Name	VPN Name	Slice ID	Slice Name	Source NE	Source NE IP A...	Delivery Interf...	Source NE Inte...	Next-Hop Add...
□	L3EVPN_Test_16...	monitoringName6	L3EVPN_Test_16...			NE8000-X4-192...	10.118.192.169	25GE1/0/0.1664	25GE1/0/0	2009::10
□	L3EVPN_Test_16...	rhEue	L3EVPN_Test_16...			NE8000-X4-192...	10.118.192.171	25GE1/0/1.1664	25GE1/0/1	2008::10
□	6971L2EVPN_Ev...	AewzY	6971L2EVPN_Ev...			NE8000-X4-192...	10.118.192.171	25GE1/0/1.333	25GE1/0/1	2008::10
□	L3EVPN888_69_7...	QgMY2	L3EVPN888_69_7...			NE8000-X4-192...	10.118.192.171	25GE1/0/1.888	25GE1/0/1	2008::10

Total records: 4

FIGURE 7.28 Selecting IFIT monitoring configuration. ⏎

3. Click **Add Configuration**. The **Add IFIT Configuration** dialog box is displayed, as shown in Figure 7.29. Set **VPN Type**, **VPN Name**, and **Next-Hop Type**. In the **Source NE** and **Next-Hop Information** areas, select NEs and interfaces.



Add IFIT Configuration

VPN Type: L2VPN/L2EVPN   VPN Name: l2evpn_ov...   Next-Hop Type: srv6-locator   ☑ Auto match   Bidi...

Source NE (In the public network scenario, the NE version must be managed by Yang.)

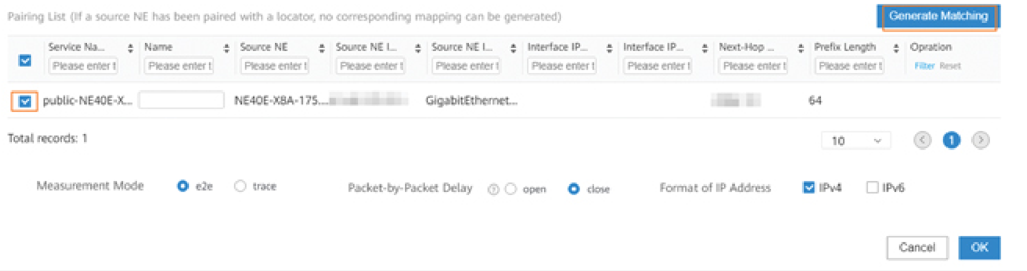
■	NE Name	NE Type	NE IP A...	Product...	Available	Opration
☑	NE40E-X3A-...	NE40E-X3A(...	[redacted]	V800R023C00	Yes	
□	NE40E-X3A-...	NE40E-X3A(...	[redacted]	V800R023C...	Yes	

Next-Hop Information

■	NE Name	NE Type	NE IP A...	Product...	Available	Opration
□	NE40E-X3A-...	NE40E-X3A(...	[redacted]	V800R023C00	Yes	
☑	NE40E-X3A-...	NE40E-X3A(...	[redacted]	V800R023C...	Yes	

FIGURE 7.29 Add IFIT configuration dialog box. ⏎

4. Click **Generate Matching**, as shown in Figure 7.30. A VPN UNI and a destination peer locator form a matching.



Pairing List (If a source NE has been paired with a locator, no corresponding mapping can be generated)

☑	Service Na...	Name	Source NE	Source NE I...	Source NE I...	Interface IP...	Interface IP...	Next-Hop ...	Prefix Length	Opration
☑	public-NE40E-X...		NE40E-X8A-175...	[redacted]	GigabitEthernet...			[redacted]	64	

Total records: 1

Measurement Mode ● e2e ○ trace   Packet-by-Packet Delay ○ open ● close   Format of IP Address ☑ IPv4 □ IPv6

FIGURE 7.30 Generating a matching. ⏎

NOTE

5. To automatically generate a matching, select **Auto match** and then select **Unidirectional** or **Bidirectional** from the drop-down list on the right.
6. Set **Measurement Mode** to **e2e** or **trace** and **Packet-by-Packet Delay** to **Open** or **Closed**.
7. Click **OK**. When the message “Operation completed” is displayed, click **OK**.

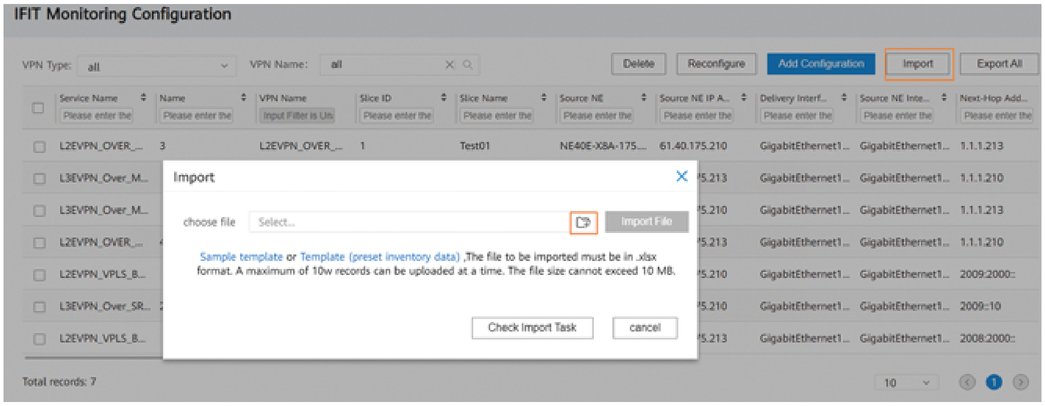
NOTE

- An IFIT instance can be bound to a device’s main interface. If the UNI of a VPN service is a sub-interface, its main interface is used when NCE delivers a VPN+peer locator-based instance. For example, if the configured interface is GigabitEthernet1/0/5.3268, the source NE interface displayed in the VPN monitoring list is GigabitEthernet1/0/5.
- If the source interface has both IPv4 and IPv6 addresses, both of them can be used to deliver an SRv6 L3VPN IFIT instance. In this case, the GUI displays two instances, which are distinguished using the service names suffixed with an address type.

7.3.5.2 Creating Instances in Batches

Instances are created in batches as follows:

1. Open the Network Digital Map app and choose **Global Settings** > **Intelligent Analysis Settings** > **Congestion View** > **Monitoring Management** > **Monitoring Instance Management**. For details, see the content related to monitoring instance management in Section 7.3.2.
2. In the navigation pane, choose **VPN Monitoring** > **IFIT Monitoring Configuration**. In the **IFIT Monitoring Configuration** dialog box, click **Import**. The **Import** dialog box is displayed, as shown in Figure 7.31.



IFIT Monitoring Configuration

VPN Type: all　VPN Name: all

□	Service Name	Name	VPN Name	Slice ID	Slice Name	Source NE	Source NE IP A...	Delivery Interf...	Source NE Inte...	Next-Hop Add...
□	L2EVPN_OVER_...	3	L2EVPN_OVER_...	1	Test01	NE40E-X8A-175...	61.40.175.210	GigabitEthernet1...	GigabitEthernet1...	1.1.1.213
□	L3EVPN_Over_M...						5.213	GigabitEthernet1...	GigabitEthernet1...	1.1.1.210
□	L3EVPN_Over_M...						5.210	GigabitEthernet1...	GigabitEthernet1...	1.1.1.213
□	L2EVPN_OVER_...	4					5.213	GigabitEthernet1...	GigabitEthernet1...	1.1.1.210
□	L2EVPN_VPLS_B...						5.210	GigabitEthernet1...	GigabitEthernet1...	2009:2000::
□	L3EVPN_Over_SR...	2					5.210	GigabitEthernet1...	GigabitEthernet1...	2009::10
□	L2EVPN_VPLS_B...						5.213	GigabitEthernet1...	GigabitEthernet1...	2008:2000::

Total records: 7

Import

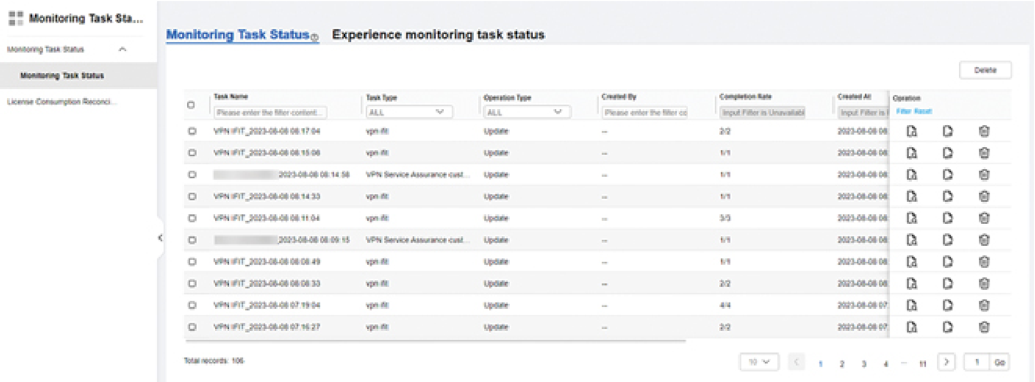
choose file: Select...

Sample template or Template (preset inventory data) ,The file to be imported must be in .xlsx format. A maximum of 10w records can be uploaded at a time. The file size cannot exceed 10 MB.

FIGURE 7.31 Import dialog box. ⏎

3. Download the sample template file to be imported from the provided hyperlink, set related parameters in the file, and then save it.

4. Select the file in which instance parameters have been set and click **Import File**. The message “File uploaded successfully” indicates that the batch task has been submitted.
5. Click **Check Import Task** or choose **Settings** > **Monitoring Status** > **Monitoring Task Status** from the main menu. On the **Monitoring Task Status** tab page, filter tasks by **Task Name**, **Task Type**, **Operation Type**, **Created By**, etc. Then select a desired task and click the **View Details**, **Export**, or **Delete** icon in the **Operation** column as required, as shown in Figure 7.32.



□	Task Name	Task Type	Operation Type	Created By	Completion Rate	Created At	Operation
	Please enter the filter content...	ALL	ALL	Please enter the filter co	Input Filter is Unavailabl	Input Filter is I	Filter Reset
□	VPN IFIT_2023-08-08 08:17:04	vpn ifit	Update	--	2/2	2023-08-08 08:	
□	VPN IFIT_2023-08-08 08:15:06	vpn ifit	Update	--	1/1	2023-08-08 08:	
□	[redacted] 2023-08-08 08:14:58	VPN Service Assurance cust...	Update	--	1/1	2023-08-08 08:	
□	VPN IFIT_2023-08-08 08:14:33	vpn ifit	Update	--	1/1	2023-08-08 08:	
□	VPN IFIT_2023-08-08 08:11:04	vpn ifit	Update	--	3/3	2023-08-08 08:	
□	[redacted] 2023-08-08 08:09:15	VPN Service Assurance cust...	Update	--	1/1	2023-08-08 08:	
□	VPN IFIT_2023-08-08 08:08:49	vpn ifit	Update	--	1/1	2023-08-08 08:	
□	VPN IFIT_2023-08-08 08:08:33	vpn ifit	Update	--	2/2	2023-08-08 08:	
□	VPN IFIT_2023-08-08 07:19:04	vpn ifit	Update	--	4/4	2023-08-08 07:	
□	VPN IFIT_2023-08-08 07:16:27	vpn ifit	Update	--	2/2	2023-08-08 07:	

Total records: 106

FIGURE 7.32 Monitoring task status tab page. ⏎

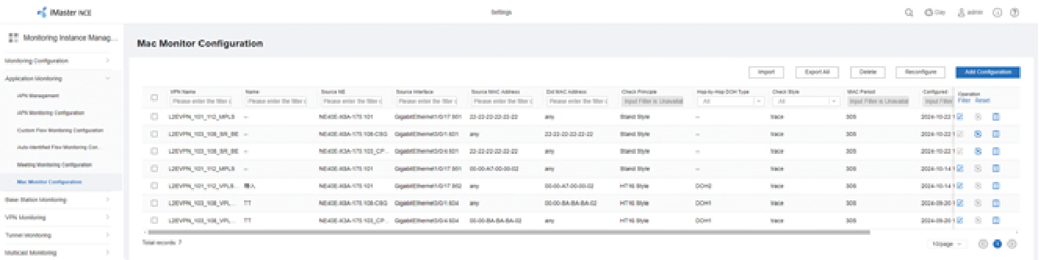
6. In the **IFIT Monitoring Configuration** dialog box, click **Export All**. After the system exports the existing instances to a file, the delivered instances can be searched for and compared.

7.3.6 MAC Address-Based IFIT-AM Deployment

MAC address-based IFIT-AM is mainly used for network troubleshooting in home broadband scenarios to meet monitoring requirements for important service assurance, service fault demarcation, and more. It performs fine-grained SLA measurement on service flows or users identified by MAC addresses.

The following describes how to deploy MAC address-based IFIT-AM in scenarios where a user goes online and reports a fault.

1. Open the Network Digital Map app and choose **Global Settings** > **Intelligent Analysis Settings** > **Congestion View** > **Monitoring Management** > **Monitoring Instance Management**. For details, see the content related to monitoring instance management in Section 7.3.2.
2. In the navigation pane, choose **Application Monitoring** > **Mac Monitor Configuration**. The **Mac Monitor Configuration** page is displayed, as shown in Figure 7.33.



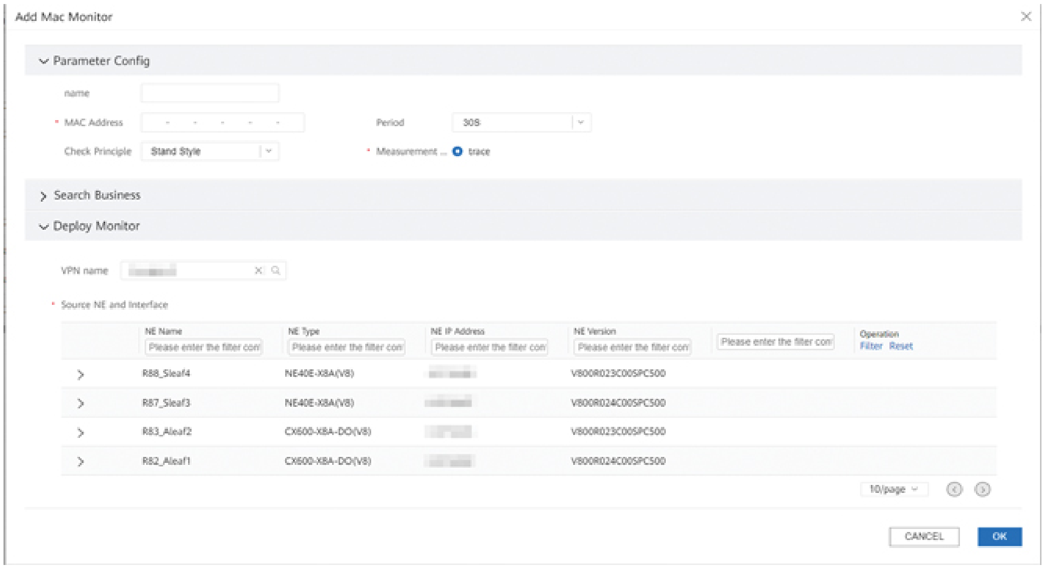
Mac Monitor Configuration

VPN Name	Name	Source NE	Source Interface	Source MAC Address	Dst MAC Address	Check Principle	Hop-by-Hop DOH Type	Check Style	MAC Period	Configured...	Operation
L2EVPN_101_112_MPLS	--	NE40E-X8A-175.101	GigabitEthernet1/0/17.501	22-22-22-22-22-22	any	Stand Style	--	trace	30S	2024-10-22 1	
L2EVPN_103_108_SR_BE	--	NE40E-X8A-175.108-CSG	GigabitEthernet3/0/1.501	any	22-22-22-22-22-22	Stand Style	--	trace	30S	2024-10-22 1	
L2EVPN_103_108_SR_BE	--	NE40E-X8A-175.103_CP...	GigabitEthernet3/0/4.501	22-22-22-22-22-22	any	Stand Style	--	trace	30S	2024-10-22 1	
L2EVPN_101_112_MPLS	--	NE40E-X8A-175.101	GigabitEthernet1/0/17.501	00-00-A7-00-00-02	any	Stand Style	--	trace	30S	2024-10-14 1	
L2EVPN_101_112_VPLS...	[illegible]	NE40E-X8A-175.101	GigabitEthernet1/0/17.502	any	00-00-A7-00-00-02	HT16 Style	DOH2	trace	30S	2024-10-14 1	
L2EVPN_103_108_VPL...	TT	NE40E-X8A-175.108-CSG	GigabitEthernet3/0/1.504	any	00-00-BA-BA-BA-02	HT16 Style	DOH1	trace	30S	2024-09-20 1	
L2EVPN_103_108_VPL...	TT	NE40E-X8A-175.103_CP...	GigabitEthernet3/0/4.504	00-00-BA-BA-BA-02	any	HT16 Style	DOH1	trace	30S	2024-09-20 1	

Total records: 7

FIGURE 7.33 Mac Monitor Configuration page. ⏎

3. Click **Add Configuration**. The **Add Mac Monitor** dialog box is displayed, as shown in Figure 7.34. Set **MAC Address**, **Period**, **Measurement Mode**, and Layer 2 Virtual Private Network (L2VPN) EVPN service name.



name

* MAC Address: - - - - -

Period: 30S

Check Principle: Stand Style

* Measurement ...: ● trace

VPN name: [redacted]

* Source NE and Interface

NE Name	NE Type	NE IP Address	NE Version		Operation
R88_Sleaf4	NE40E-X8A(V8)	[redacted]	V800R023C00SPC500		
R87_Sleaf3	NE40E-X8A(V8)	[redacted]	V800R024C00SPC500		
R83_Aleaf2	CX600-X8A-DO(V8)	[redacted]	V800R023C00SPC500		
R82_Aleaf1	CX600-X8A-DO(V8)	[redacted]	V800R024C00SPC500		

FIGURE 7.34 Add Mac Monitor dialog box. ⏎

4. Confirm the information, submit the request, and wait for the system to complete the operation.

7.3.7 Tunnel-Based IFIT-AM Deployment

Tunnel-based IFIT-AM refers to SRv6 Policy-based monitoring, which measures the traffic of each segment list in an SRv6 Policy and generates statistics for the corresponding segment list. In addition, IFIT monitoring instances can be created on the GUI or in batches by importing a file.

7.3.7.1 Creating Instances on the GUI

Instances are created on the GUI as follows:

1. Open the Network Digital Map app and choose **Global Settings** > **Intelligent Analysis Settings** > **Congestion View** > **Monitoring Management** > **Monitoring**

Instance Management. For details, see the content related to monitoring instance management in Section 7.3.2.

2. In the navigation pane, choose **Tunnel Monitoring** > **Monitoring Configuration**, as shown in Figure 7.35.

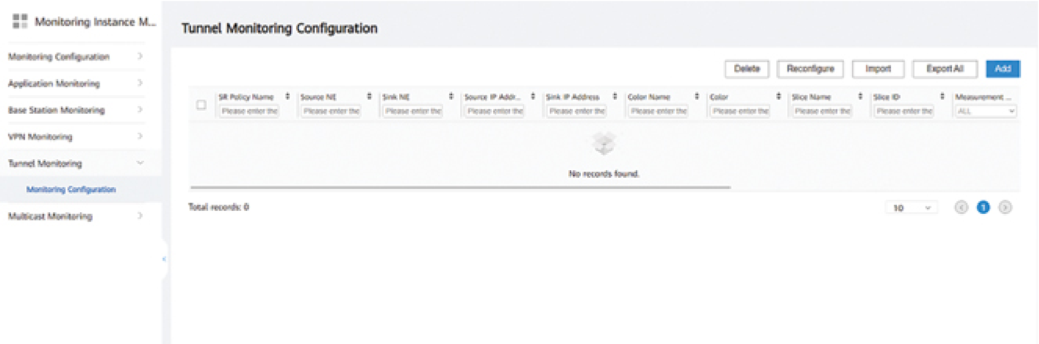


FIGURE 7.35 Selecting monitoring configuration. ⏎

3. Click **Add**. The **Add Tunnel to Be Monitored** dialog box is displayed, as shown in Figure 7.36. Click the text box next to **Subnet**. In the **Select Subnet** dialog box, select a desired subnet and click **OK**. In the **Available NEs** area, filter NEs by **NE Name**, **NE Type**, **NE IP Address**, **Production Version**, and **Available**.

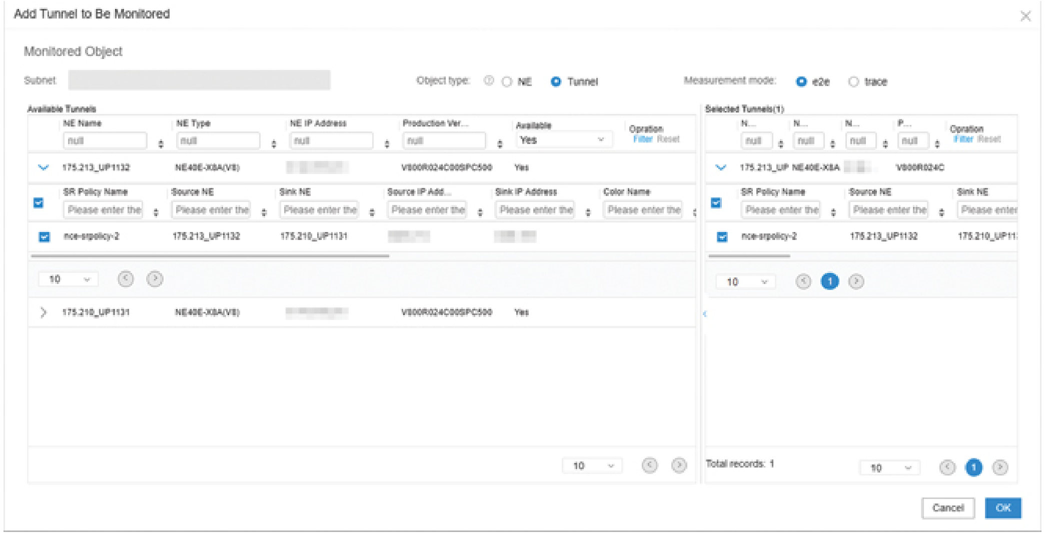


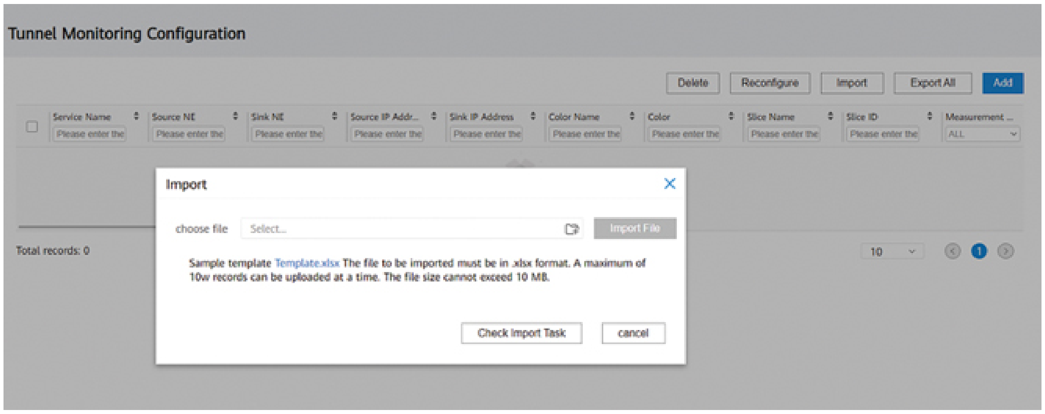
FIGURE 7.36 Add tunnel to be monitored dialog box. ⏎

4. Set **Object type** to **NE** or **Tunnel**. If **Object type** is set to **NE**, tunnel instances with the selected NE as the source are created in batches. In cases where a new tunnel is created for the selected NE, a monitoring instance needs to be created again. If **Object type** is set to **Tunnel**, the selected SRv6 Policy is monitored.
5. Set **Measurement mode** to **e2e** or **trace**.
6. Click **OK**.

7.3.7.2 Creating Instances in Batches

Instances are created in batches as follows:

1. Open the Network Digital Map app and choose **Global Settings** > **Intelligent Analysis Settings** > **Congestion View** > **Monitoring Management** > **Monitoring Instance Management**. For details, see the content related to monitoring instance management in Section 7.3.2.
2. In the navigation pane, choose **Tunnel Monitoring** > **Monitoring Configuration**. In the **Tunnel Monitoring Configuration** dialog box, click **Import**. The **Import** dialog box is displayed, as shown in Figure 7.37.



Sample template Template.xlsx The file to be imported must be in .xlsx format. A maximum of 10w records can be uploaded at a time. The file size cannot exceed 10 MB.

FIGURE 7.37 Import dialog box. ⏎

3. Download the sample template file to be imported from the provided hyperlink, set related parameters in the file, and then save it.
4. Select the file in which instance parameters have been set and click **Import File**. The message “File uploaded successfully” indicates that the batch task has been submitted.
5. Click **Check Import Task** or choose **Settings** > **Monitoring Status** > **Monitoring Task Status** from the main menu. On the **Monitoring Task Status** tab page (shown in Figure 7.38), filter tasks by **Task Name**, **Task Type**, **Operation Type**, **Created By**, etc. Then, select a desired task and click the **View Details**, **Export**, or **Delete** icon in the **Operation** column as required.

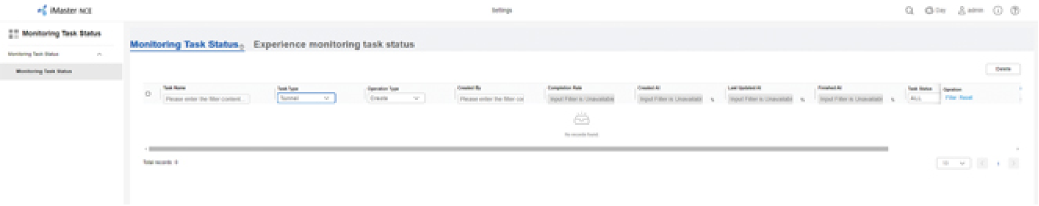


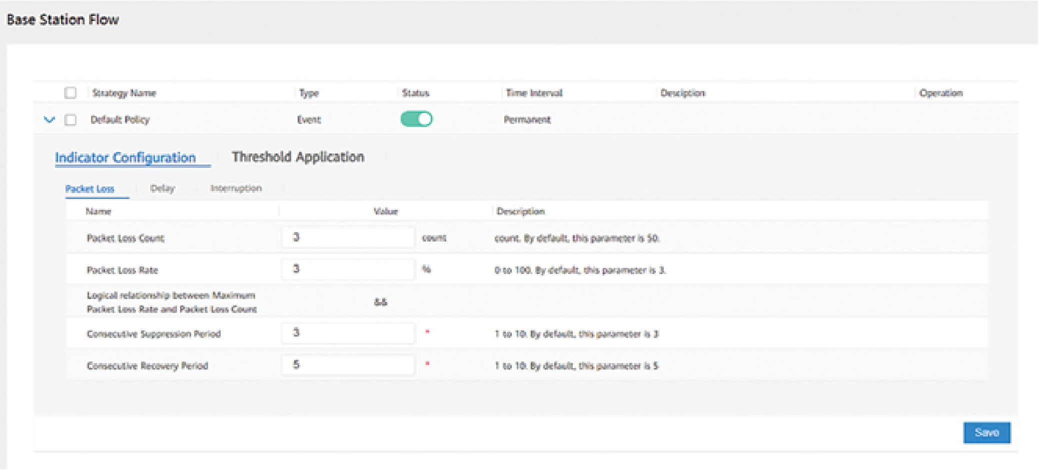
FIGURE 7.38 Monitoring task status tab page. ⏎

6. In the **Tunnel Monitoring Configuration** dialog box, click **Export All**. After the system exports the existing instances to a file, the delivered instances can be searched for and compared.

7.3.8 VPN Troubleshooting

If a VPN fault occurs, indicator fields in the performance report can be checked for abnormalities, and then the fault can be located through E2E or hop-by-hop data association analysis. The following uses IFIT-AM measurement for the mobile transport service VPN as an example, in which dynamic learning-based IFIT-AM is deployed for the mobile transport service. The dynamically learned monitored object is referred to as a “base station flow” in this section. To improve device forwarding performance, the E2E mode is deployed for each base station flow by default. If an exception occurs, the automatic hop-by-hop monitoring policy is used to switch the measurement mode to trace. The operations involved in this process are as follows:

1. Define poor-QoS determination rules. Configure packet loss, delay, and interruption thresholds for E2E flows, as shown in Figure 7.39. If the indicators of a base station flow exceed the configured thresholds, this flow is considered poor QoS, indicating that an exception occurs.



Base Station Flow

□	Strategy Name	Type	Status	Time Interval	Desciption	Operation
□	Default Policy	Event	●	Permanent		

Indicator Configuration | Threshold Application

Packet Loss | Delay | Interruption

Name	Value		Description
Packet Loss Count	3	count	count. By default, this parameter is 50.
Packet Loss Rate	3	%	0 to 100. By default, this parameter is 3.
Logical relationship between Maximum Packet Loss Rate and Packet Loss Count	&&		
Consecutive Suppression Period	3	*	1 to 10. By default, this parameter is 3
Consecutive Recovery Period	5	*	1 to 10. By default, this parameter is 5

FIGURE 7.39 Defining poor-QoS determination rules. ⏎

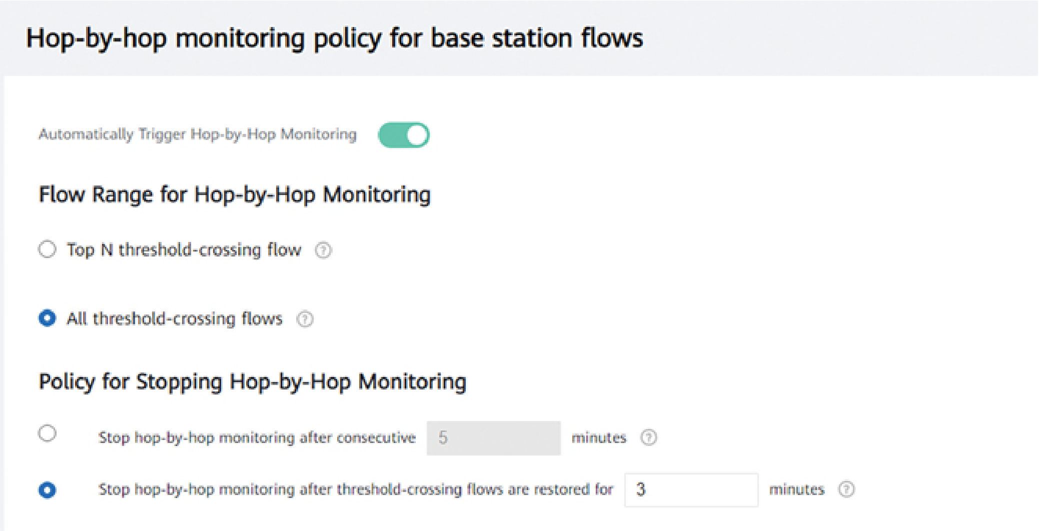
Table 7.2 describes thresholds for packet loss causing poor QoS.

TABLE 7.2 Thresholds for Packet Loss Causing Poor QoS ⏎

Threshold Name	Description
Packet Loss Count	Total number of packets lost in a 1-minute period. It is calculated by subtracting the number of packets reported at the egress of an IFIT-AM measurement instance from that reported at the ingress.
Packet Loss Rate	Maximum packet loss rate of an IFIT-AM measurement instance in a 1-minute period.
Logical relationship between Maximum Packet Loss Rate and Packet Loss Count	Threshold-crossing for packet loss, causing poor QoS, is determined only when both **Packet Loss Rate** and **Packet Loss Count** are exceeded.

Threshold Name	Description
Consecutive Suppression Period	Poor QoS caused by packet loss is determined only when the thresholds are exceeded for several consecutive periods.
Consecutive Recovery Period	The fault is considered rectified only when the thresholds are not exceeded for several consecutive periods.

2. Configure an automatic hop-by-hop monitoring policy. As shown in Figure 7.40, toggle on **Auto Trigger Hop-by-Hop Monitoring**. The hop-by-hop mode is then automatically triggered if base station flows have poor QoS. Data of each hop can be displayed, and the hop where an exception occurs can be located, helping to identify and locate faults.



Hop-by-hop monitoring policy for base station flows

Automatically Trigger Hop-by-Hop Monitoring ●

Flow Range for Hop-by-Hop Monitoring

○ Top N threshold-crossing flow

● All threshold-crossing flows

Policy for Stopping Hop-by-Hop Monitoring

○ Stop hop-by-hop monitoring after consecutive 5 minutes

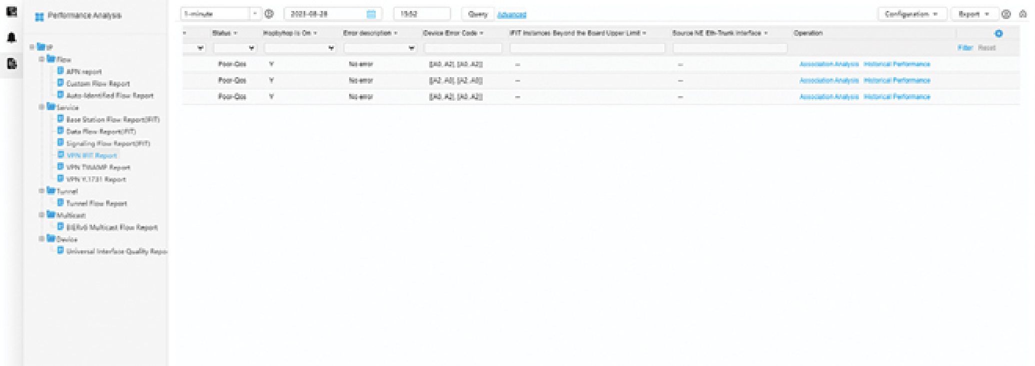
● Stop hop-by-hop monitoring after threshold-crossing flows are restored for 3 minutes

FIGURE 7.40 Configuring an automatic hop-by-hop monitoring policy. ⏎

Set **Flow Range for Hop-by-Hop Monitoring** to **Top N threshold-crossing flow** or **All threshold-crossing flows**. **Top N threshold-crossing flow** indicates that multiple flows whose source or destination is a base station may cross the threshold in a given period. All threshold-crossing flows are sorted in descending order of packet loss rate or delay. The top 1 flow is selected for each of N base stations, and the automatic hop-by-hop monitoring policy is applied to these flows. **All threshold-crossing flows** indicates that the automatic hop-by-hop monitoring policy is applied to all monitored and threshold-crossing flows.

Set **Policy for Stopping Hop-by-Hop Monitoring** to **Stop hop-by-hop monitoring after consecutive *XX* minutes** or **Stop hop-by-hop monitoring after threshold-crossing flows are restored for *XX* minutes**. **Stop hop-by-hop monitoring after consecutive *XX* minutes** indicates that the measurement mode is switched to E2E, regardless of whether the fault is rectified. **Stop hop-by-hop monitoring after threshold-crossing flows are restored for *XX* minutes** indicates that the measurement mode is switched to E2E *XX* minutes after the fault is rectified.

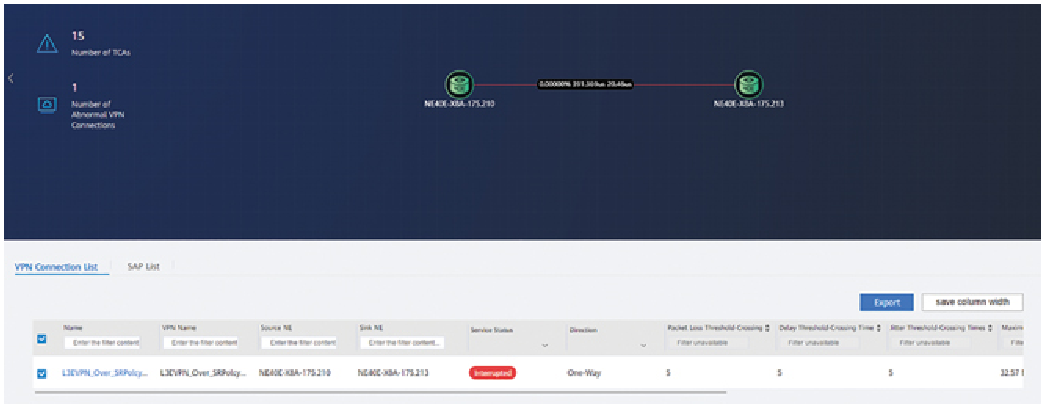
3. View flow performance statistics. Click <Image 7-1 Here> to access the **Performance Analysis** page, as shown in Figure 7.41. In the corresponding VPN service report, statistics about poor-QoS flows can be viewed by filtering rows whose **Status** is **Poor-Qos**. Check whether hop-by-hop measurement is enabled for the flows. If **Hopbyhop Is On** is **Y**, click **Association Analysis** in the **Operation** column for each flow to view details about the flow.



Status	Hopbyhop Is On	Error description	Device Error Code	IFIT Instances Beyond the Board Upper Limit	Source NE Eth-Trunk Interface	Operation
Poor-Qos	Y	No-error	[[A0, A2], [A0, A2]]	--	--	Association Analysis Historical Performance
Poor-Qos	Y	No-error	[[A2, A0], [A2, A0]]	--	--	Association Analysis Historical Performance
Poor-Qos	Y	No-error	[[A0, A2], [A0, A2]]	--	--	Association Analysis Historical Performance

FIGURE 7.41 Performance analysis page. ⏎

4. The VPN details page (shown in Figure 7.42) shows the number of threshold-crossing alarms (number of times that the packet loss rate, delay, and jitter exceed the thresholds) and the number of abnormal VPN connections on the left and the overall topology of the VPN on the right. The topology can be viewed by all, normal, or abnormal VPN connections. Abnormal VPN service links are marked.

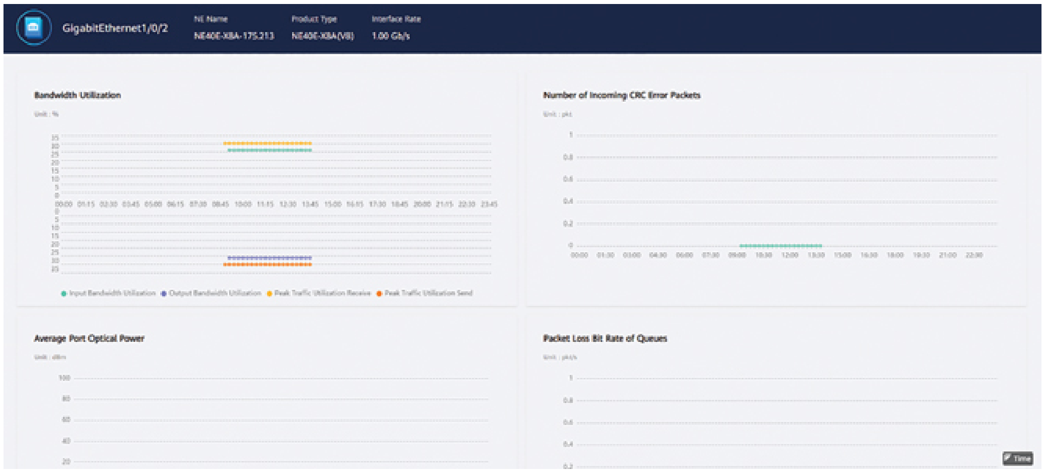


Name	VPN Name	Source NE	Sink NE	Service Status	Direction	Packet Loss Threshold-Crossing	Delay Threshold-Crossing Time	Jitter Threshold-Crossing Times	Maxim
Enter the filter content	Enter the filter content	Enter the filter content	Enter the filter content...			Filter unavailable	Filter unavailable	Filter unavailable	Filte
L3EVPN_Over_SRPolicy...	L3EVPN_Over_SRPolicy...	NE40E-X8A-175.210	NE40E-X8A-175.213	Interrupted	One-Way	5	5	5	32.57 I

FIGURE 7.42 VPN details page. ⏎

a. On the VPN details page, the **VPN Connection List** tab page displays detailed information about all service access points connected to the VPN. The information includes the packet loss, delay, jitter threshold-crossing times, maximum rate, and upstream and downstream rates. Abnormal VPNs can be filtered, and multiple technologies (e.g., IFIT, Y.1731, and TWAMP) can be used to monitor VPNs — the monitoring results can be displayed on the VPN details page. Note that different measurement technologies have different performance indicators for indicating the packet loss rate, jitter, and delay.

        i. IFIT performance indicators: Maximum packet loss rate for packet loss measurement, maximum delay for delay measurement, and maximum jitter for jitter measurement
        ii. Y.1731 performance indicators: Maximum near-end packet loss rate for packet loss measurement, maximum two-way delay for delay measurement, and maximum two-way jitter for jitter measurement
        iii. TWAMP performance indicators: Maximum two-way packet loss rate for packet loss measurement, maximum two-way delay for delay measurement, and maximum two-way jitter for jitter measurement
    b. If multiple service flows exist in a VPN connection using IFIT, the upstream rate is the total upstream rate of all service flows, and the downstream rate is the total downstream rate of all service flows. A VPN connection using TWAMP or Y.1731 does not involve the upstream or downstream rate.
    c. The **SAP List** tab page displays detailed information about the service interfaces connected to the VPN. The information includes the Tx/Rx bandwidth utilization, Tx/Rx peak bandwidth utilization, and Tx peak rate. As shown in Figure 7.43, 24-hour performance indicator trend charts of service interfaces can be viewed by clicking the service interface name of a service access point, providing useful insights for troubleshooting.



GigabitEthernet1/0/2

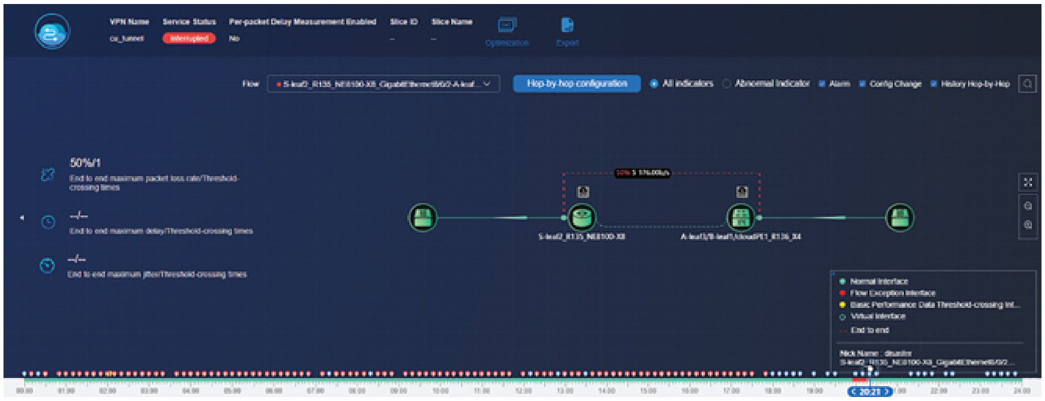
NE Name: NE40E-X8A-175.213

Product Type: NE40E-X8A(V8)

Interface Rate: 1.00 Gb/s

FIGURE 7.43 Performance indicator trend charts of the interface. ⏎

5. On the **VPN Connection List** tab page, click the link for a VPN connection in the **Name** column. The association analysis page for the VPN connection is displayed, as shown in Figure 7.44. This page displays the E2E maximum packet loss rate/threshold-crossing times, E2E maximum delay/threshold-crossing times, and E2E maximum jitter/threshold-crossing times on the left, and the service path of the VPN connection on the right. The topology can display all indicators or only abnormal ones based on the flow direction. Abnormal VPN connections, indicators, and links are marked. And the last 7 days of SLA data can be played back to accurately restore historical faulty service topology paths, fault diagnosis results, and flow analysis information, making it easier to trace and locate historical issues.



VPN Name: cs_tunnel
Service Status: Interrupted
Per-packet Delay Measurement Enabled: No
Slice ID: --
Slice Name: --

FIGURE 7.44 Association analysis page for the VPN connection. ⏎

On this page, it is also possible to display NE, link, and interface information and view flow specifications. Specifically:

a. Double-click an NE or link that is marked abnormal in the VPN service topology to view the trend charts of its maximum delay, number of lost packets, maximum packet loss rate, maximum jitter, and average rate over 24 hours, as shown in Figure 7.45. The charts facilitate troubleshooting.

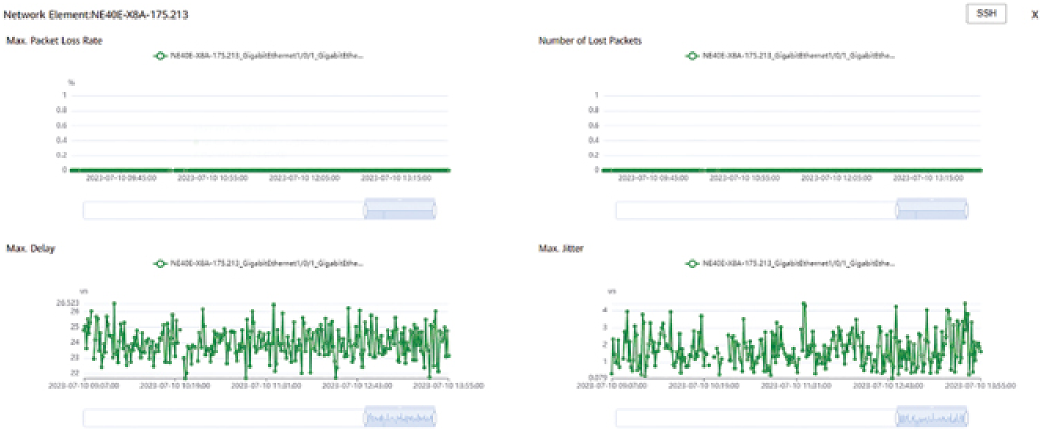
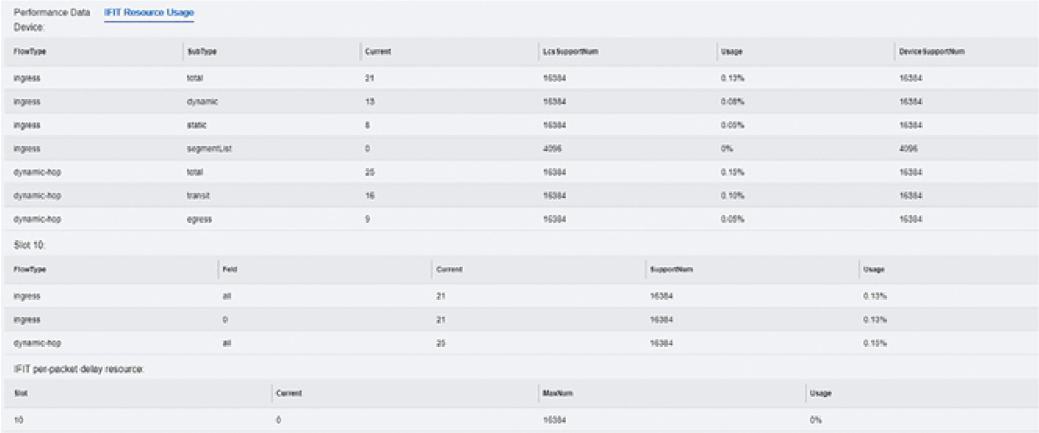


FIGURE 7.45 Trend charts of the abnormal NE. ⏎

b. Double-click an NE in the VPN service topology. As shown in Figure 7.46, the **Performance Data** and **IFIT Resource Usage** tab pages are displayed below the topology. The **IFIT Resource Usage** tab page provides details about the IFIT resource usage of the NE.



Device:

FlowType	SubType	Current	Lcs SupportNum	Usage	Device SupportNum
ingress	total	21	16384	0.13%	16384
ingress	dynamic	13	16384	0.08%	16384
ingress	static	8	16384	0.05%	16384
ingress	segmentList	0	4096	0%	4096
dynamic-hop	total	25	16384	0.15%	16384
dynamic-hop	transit	16	16384	0.10%	16384
dynamic-hop	egress	9	16384	0.05%	16384

Slot 10:

FlowType	Field	Current	SupportNum	Usage
ingress	all	21	16384	0.13%
ingress	0	21	16384	0.13%
dynamic-hop	all	25	16384	0.15%

IFIT per-packet delay resource:

Slot	Current	MaxNum	Usage
10	0	16384	0%

FIGURE 7.46 IFIT resource usage. ⏎

Table 7.3 describes the fields for device-level IFIT resource usage.

TABLE 7.3 Fields for Device-Level IFIT Resource Usage ⏎

Field	Description
FlowType	Type of the statistical item of the IFIT-AM measurement instance generated by a flow on the device: • ingress: ingress node instance • dynamic-hop: transit and egress node instances
SubType	Subtype of the statistical item of the IFIT-AM measurement instance generated by a flow on the device: • total: total statistics • dynamic: dynamic flow • static: static flow • segmentList: tunnel-level • transit: transit node • egress: egress node
Current	Current number of IFIT-AM measurement instances
LcsSupportNum	Number of IFIT-AM measurement instances supported by the license
Usage	Resource usage of IFIT-AM measurement instances
DeviceSupportNum	Number of IFIT-AM measurement instances supported by the device

Table 7.4 describes the fields for board-level IFIT resource usage.

TABLE 7.4 Fields for Board-Level IFIT Resource Usage ⏎

Field	**Description**
FlowType	Type of the statistical item of the IFIT-AM measurement instance generated by a flow on the device: • ingress: ingress node instance • dynamic-hop: transit and egress node instances
FeId	Chip ID
Current	Current number of IFIT-AM measurement instances
SupportNum	Number of supported IFIT-AM measurement instances
Usage	Resource usage of IFIT-AM measurement instances

c. Click an interface that is marked abnormal in the VPN flow topology to view the interface's KPI trend charts over 24 hours. The KPIs include the input bandwidth utilization, output bandwidth utilization, rate of discarded queue packets, and incoming Cyclic Redundancy Check (CRC) error packets, as shown in Figure 7.47. The default display rule is as follows: If threshold-crossing occurs, the threshold-crossing indicators and the corresponding threshold lines are displayed; if no threshold-crossing occurs, all indicators are displayed, but the threshold lines are not.

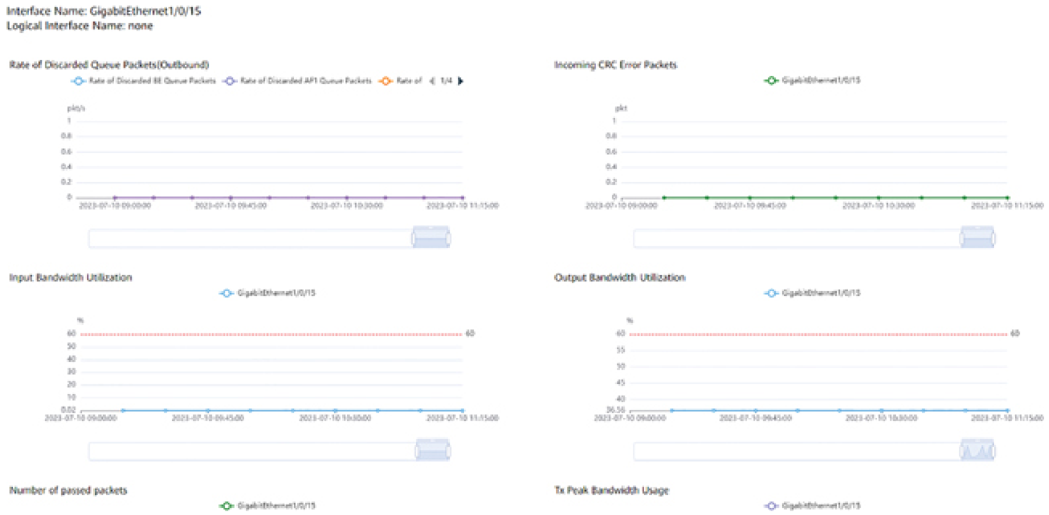
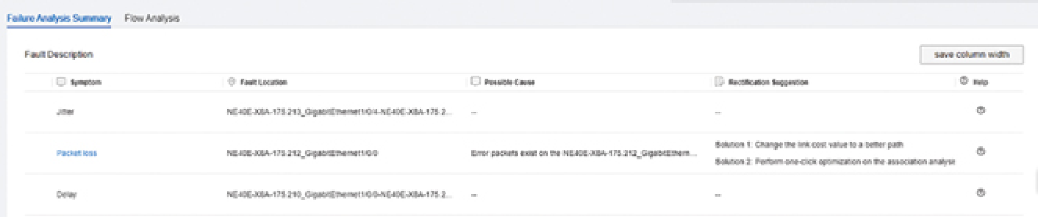

Interface Name: GigabitEthernet1/0/15
Logical Interface Name: none

FIGURE 7.47 Trend charts of the abnormal interface. ⏎

6. On the **Failure Analysis Summary** tab page below the topology, view the fault analysis summary of the VPN connection, as shown in Figure 7.48. The summary provides suggestions based on the fault location, possible cause, and indicator trend, facilitating fault rectification.



Symptom	Fault Location	Possible Cause	Rectification Suggestion	Help
Jitter	NE40E-X8A-175.213_GigabitEthernet1/0/4-NE40E-X8A-175.2...	--	--	
Packet loss	NE40E-X8A-175.212_GigabitEthernet1/0/0	Error packets exist on the NE40E-X8A-175.212_GigabitEthern...	Solution 1: Change the link cost value to a better path Solution 2: Perform one-click optimization on the association analyse	
Delay	NE40E-X8A-175.210_GigabitEthernet1/0/0-NE40E-X8A-175.2...	--	--	

FIGURE 7.48 Fault analysis summary. ⏎

For a jitter fault, only the fault location is displayed. For packet loss and delay faults, see Table 7.5.

TABLE 7.5 Description of Packet Loss and Delay Faults ⏎

Fault Type	Fault Symptom	Fault Location	Possible Cause	Rectification Suggestion	Data Sour
Packet loss threshold-crossing	Interface CRC error packet	NE+interface	Error packets exist on an interface of an NE.	Check whether the optical module is faulty and whether the optical fiber is loose.	Abn indic
	Queue packet loss on an interface on the flow path	NE+interface	Queue packet loss occurs on an interface of an NE.	Expand the capacity.	Abn indic
	No interface exception and no queue packet loss	NE (link)+interfaces at both ends of the NE	-- is displayed.	-- is displayed.	Non
Delay threshold-crossing	Queue packet loss on an interface on the flow path	NE+interface	Queue packet loss occurs on an interface of an NE.	Expand the capacity.	Abn indic

Fault Type	Fault Symptom	Fault Location	Possible Cause	Rectification Suggestion	Data Sour
	No queue packet loss on the flow path	NE (link)+interfaces at both ends of the NE	-- is displayed.	-- is displayed.	None

7. On the **Flow Analysis** tab page below the topology, view the trend charts of the E2E maximum packet loss rate, E2E maximum delay, E2E maximum jitter, and maximum rate, as shown in Figure 7.49. The charts facilitate troubleshooting.

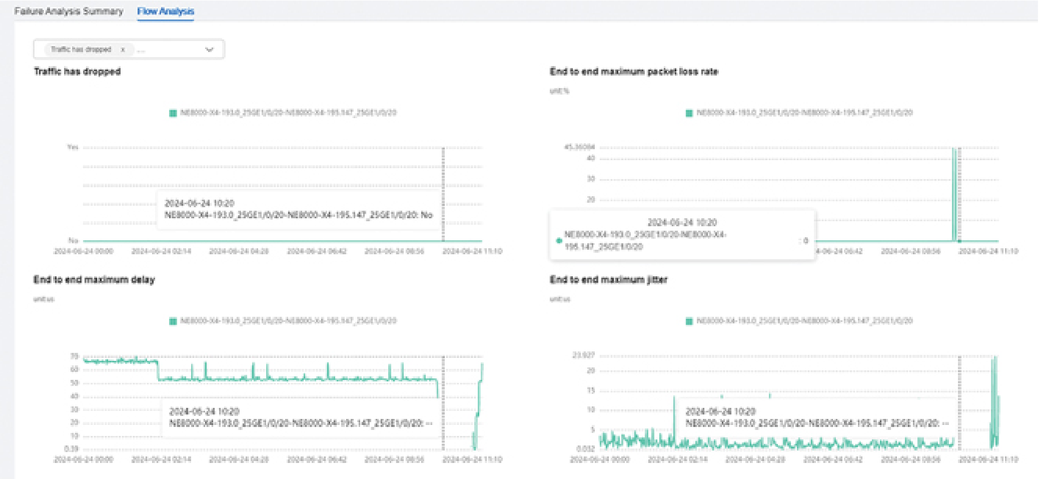

Failure Analysis Summary
Flow Analysis
Traffic has dropped ×
Traffic has dropped
NE8000-X4-193.0_25GE1/0/20-NE8000-X4-195.147_25GE1/0/20
Yes
No
2024-06-24 10:20
NE8000-X4-193.0_25GE1/0/20-NE8000-X4-195.147_25GE1/0/20: No
2024-06-24 00:00
2024-06-24 02:14
2024-06-24 04:28
2024-06-24 06:42
2024-06-24 08:56
2024-06-24 11:10
End to end maximum packet loss rate
unit:%
NE8000-X4-193.0_25GE1/0/20-NE8000-X4-195.147_25GE1/0/20
45.36094
40
30
20
2024-06-24 10:20
NE8000-X4-193.0_25GE1/0/20-NE8000-X4-195.147_25GE1/0/20
: 0
End to end maximum delay
unit:us
NE8000-X4-193.0_25GE1/0/20-NE8000-X4-195.147_25GE1/0/20
70
60
50
40
30
20
10
0.39
2024-06-24 10:20
NE8000-X4-193.0_25GE1/0/20-NE8000-X4-195.147_25GE1/0/20: --
End to end maximum jitter
unit:us
NE8000-X4-193.0_25GE1/0/20-NE8000-X4-195.147_25GE1/0/20
23.927
20
15
10
5
0.032
2024-06-24 10:20
NE8000-X4-193.0_25GE1/0/20-NE8000-X4-195.147_25GE1/0/20: --

FIGURE 7.49 Flow analysis. ⏎

7.4 STORIES BEHIND IPV6 ON-PATH TELEMETRY

7.4.1 Network and Controller

The importance of the network “brain” (controller) has been overemphasized in the SDN era, with many believing that the controller can be used to solve all network problems. This belief stems from the forwarding and control planes being separated based on OpenFlow, and network devices (forwarders) being fully controlled by the controller. In practice, however, we found that this ideal architecture is troublesome. Later when I met Mr. Guoli Yin, he inspired me a lot with his summary of SDN: Network functions can make up for insufficiencies in device functions, and controller functions can make up for insufficiencies in network functions. For example, device function insufficiency, such as low device reliability, can be addressed using network reliability technologies like E2E path Hot Standby (HSB) and FRR, and network function insufficiency, such as the inability to implement global optimization through distributed network path calculation, can be addressed using the controller to perform global path optimization. This summary reflects the importance of the system. It is essential to solve problems by fully leveraging the functions of devices, networks, and controllers.

During the process of IPv6 Enhanced innovation, I gained a deeper understanding of the system: A network is like a “body,” and if it is not powerful, the “brain” (controller) cannot be smart enough. It is difficult to develop a good controller on traditional MPLS networks due to the complexities in MPLS technology. And while VXLAN simplifies service deployment and enables the controller on data center networks, its functions are relatively simple. With the emergence of IPv6 Enhanced technologies, SRv6 enables the controller to have flexible and scalable path programming capabilities, and IPv6 on-path telemetry enables the controller to have stronger path visualization capabilities. SRv6 is like “hands” and IPv6 on-path telemetry is like “eyes.” It is only possible to make the “brain” smart and flexible enough through the cooperation of the “body,” “hands,” and “eyes.”

III

CHAPTER 8

Industry Development and Technology Prospects of IPv6 On-Path Telemetry

DOI: 10.1201/9781003677901-11

As 5G deployment continues to grow and cloud-network services gain steam, IPv6 on-path telemetry is making significant progress in standardization and industrialization. This chapter describes its standardization progress, related industry activities, and commercial deployment, and explores its future technology prospects.

8.1 INDUSTRY DEVELOPMENT OF IPV6 ON-PATH TELEMETRY

8.1.1 Progress of IPv6 On-Path Telemetry Standardization

The IETF, ETSI, and CCSA are currently working on standardizing IPv6 on-path telemetry. Figure 8.1 shows the standard layout of IPv6 on-path telemetry, which encompasses the IFIT framework, management plane, control plane, and data plane.

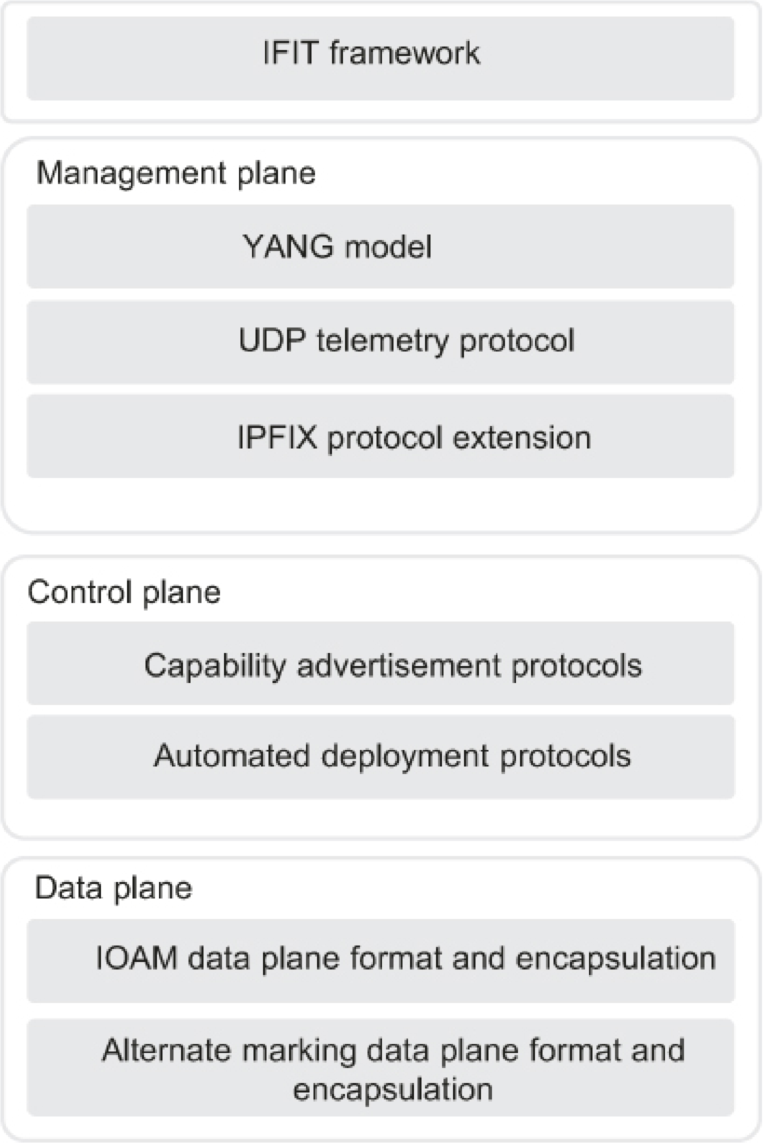

IFIT framework
Management plane
YANG model
UDP telemetry protocol
IPFIX protocol extension
Control plane
Capability advertisement protocols
Automated deployment protocols
Data plane
IOAM data plane format and encapsulation
Alternate marking data plane format and encapsulation

FIGURE 8.1 Standard layout of IPv6 on-path telemetry. ⏎

8.1.1.1 Standards and Drafts Related to the IPv6 On-Path Telemetry Framework

IFIT serves as a reference framework for IPv6 on-path telemetry, outlining an automated telemetry architecture that is easy to implement. This framework is defined by the ETSI Experiential Networked Intelligence Industry Specification Group (ENI ISG) in Group Report (GR) ENI 012. The architecture utilizes technologies such as intelligent flow selection, data reporting suppression, and DNP to address many challenges in large-scale network deployment.

A framework for alternate marking deployment is defined in *draft-ietf-ippm-alt-mark-deployment*[1], which covers methods for setting a measurement domain, configuring the ingress and egress nodes of this domain, and ensuring correct restoration of flows with alternate marking applied when the flows leave this domain. Similarly, RFC 9378[2] provides a framework for IOAM deployment.

The following three tables provide further details about the standards and drafts related to the data, control, and management planes of IPv6 on-path telemetry.

8.1.1.2 Standards and Drafts Related to the Data Plane

The data plane protocol extensions for IPv6 on-path telemetry primarily focus on defining the relevant data formats and the process of encapsulation within various transport protocols. Table 8.1 provides an overview of the involved standards and drafts.

TABLE 8.1 Standards and Drafts Related to the Data Plane for IPv6 On-Path Telemetry ⏎

Category	**Standard/Draft**	**Overview**
Alternate marking	RFC 9341[3]	Defines the alternate marking method and how to use it to measure packet loss, delay, and jitter.
	RFC 9342[4]	Extends the alternate marking method to measure any kind of unicast flow whose packets can traverse several different paths in the network. This extension can be regarded as a multipoint-to-multipoint measurement technique.
	RFC 9343[5]	Defines IPv6 data plane encapsulation for alternate marking.
	draft-ietf-mpls-inband-pm-encapsulation[6]	Defines MPLS data plane encapsulation for alternate marking.
	draft-zhou-ippm-enhanced-alternate-marking[7]	Extends the basic alternate marking data fields based on RFC 9341 and RFC 9343, enhancing measurement capabilities.
IOAM	RFC 9197[8]	Defines the IOAM data plane format in passport mode.

Category	Standard/Draft	Overview
	RFC 9326[9]	Defines the IOAM data plane format in postcard mode.
	RFC 9322[10]	Defines flags in IOAM options.
	RFC 9486[11]	Defines IPv6 data plane encapsulation for IOAM.
	RFC 9630[12]	Describes the optimization and extension of IOAM in multicast scenarios.

8.1.1.3 Standards and Drafts Related to the Control Plane

The control plane protocol extensions of IPv6 on-path telemetry are mainly used to advertise capability information and enable functions. Table 8.2 provides an overview of the involved standards and drafts.

TABLE 8.2 Standards and Drafts Related to the Control Plane for IPv6 On-Path Telemetry ⏎

Category	Standard/Draft	Overview
Capability advertisement	draft-wang-lsr-igp- extensions-ifit[13]	Defines IGP extensions for on-path telemetry capability advertisement.
	draft-wang-idr-bgpls-extensions-ifit[14]	Defines BGP-LS extensions for on-path telemetry capability advertisement.
	draft-ietf-idr-bgp-ifit-capabilities[15]	Defines BGP extensions for on-path telemetry capability advertisement based on the route next hop.
	RFC 9259[16]	Defines the formats of Echo Request and Echo Reply messages in order to query the IOAM capability of a node.
Function enabling	draft-ietf-idr-sr-policy-ifit[17]	Defines BGP SR Policy extensions for the automatic deployment of on-path telemetry along with SR Policies.

Category	Standard/Draft	Overview
	draft-ietf-pce-pcep-ifit[18]	Defines PCEP extensions for the automatic deployment of on-path telemetry along with paths.

8.1.1.4 Standards and Drafts Related to the Management Plane

The management plane protocol extensions for IPv6 on-path telemetry include three key aspects: configuring the on-path telemetry function on devices, reporting device data efficiently, and defining a controller's northbound interface. Table 8.3 provides an overview of the involved standards and drafts.

TABLE 8.3 Standards and Drafts Related to the Management Plane for IPv6 On-Path Telemetry ⏎

Category	Standard/Draft	Overview
Function configuration	RFC 9617[19]	Defines the YANG model for IOAM. This model provides IOAM configuration interfaces to allow the use of IOAM on specified flows through NETCONF.
	draft-ydt-ippm-alt- mark-yang[20]	Defines the YANG model for alternate marking. This model provides alternate marking configuration interfaces to allow the use of alternate marking on specified flows through NETCONF.
Data report	draft-ietf-netconf- udp-notif[21]	Defines a UDP-based protocol for reporting the on-path telemetry data collected from network devices to the controller at high speed.
	draft-ietf-netconf-distributed-notif[22]	Defines a distributed data reporting method, which directly reports the on-path telemetry data collected from the line cards of network devices to the controller.

Category	Standard/Draft	Overview
	draft-fz-ippm-on-path-telemetry-yang[23]	Defines the YANG model for reporting on-path telemetry information.
	draft-ietf-opsawg-ipfix-alt-mark[24]	Defines the IPFIX extension for reporting alternate marking-based on-path telemetry information.
	draft-spiegel-ippm-ioam-rawexport[25]	Defines the IPFIX extension for reporting IOAM-based on-path telemetry information.
Northbound interface	RFC 9375[26]	Defines the YANG model for reporting network measurement information, including on-path telemetry information, through the controller's northbound interface.

8.1.2 Industry Activities for IPv6 On-Path Telemetry

To further consolidate industry consensus and promote applications of IPv6 on-path telemetry, the industry has carried out several activities. These include getting major device vendors on board, conducting IPv6 on-path telemetry interoperability tests, and holding WG discussions and industry forums.

The industry's first complete IPv6 on-path telemetry solution was showcased by Huawei at Interop Tokyo 2019, winning the Best of Show Award thanks to its exceptional performance. Leveraging millisecond-level in-band flow measurement, this solution can detect network service quality in real time, locate silent faults in seconds, and support fast service recovery, bringing intelligent network O&M one step closer.

In November 2019, China's Expert Committee for Promoting Large-Scale IPv6 Deployment approved establishing the IPv6+ Technology Innovation WG. The group aims to strengthen system innovation with next-generation IPv6 Internet technologies based on the achievements of China's large-scale IPv6 deployment. It also seeks to integrate the advantages of IPv6 technology stakeholders (including academia, device vendors, carriers, and enterprises) to verify and demonstrate new IPv6 Enhanced network technologies (including SRv6, network slicing, DetNet, BIERv6, and IFIT) and new applications in network routing protocols, management automation, intelligence, and security. (DetNet is short for Deterministic Networking, and BIERv6 is short for Bit Index Explicit Replication over IPv6.) This will help to continuously improve IPv6 technology standards. IPv6 on-path telemetry is an important technique of IPv6 Enhanced innovation and has already gained widespread attention.

A year later at the end of 2020, the ETSI established a new Industry Specification Group (ISG) called IPv6 Enhanced Innovation (IPE) to promote IPv6 innovation and development. IPv6 on-path telemetry was again showcased at the IPE Webinar conferences held by ETSI in September 2020 and October 2021. Later, IPE's work was transferred to the IPv6 Global Forum for further promotion.

Starting in 2021, a dedicated session on IPv6 on-path telemetry has been held at the MPLS SDN&NFV&AI World Congress, where the architecture and key technologies involved in on-path telemetry were introduced. (NFV is short for Network Functions Virtualization.)

In its efforts to promote the maturity of IPv6 Enhanced technologies and the related industry, the Technology and Standards Research Institute from China Academy of Information and Communications Technology (CAICT) launched the "IPv6 Enhanced Ready" test project. This project tests the IPv6 Enhanced readiness of devices and solutions for both device vendors and network service providers. CAICT also launched the "IPv6 Enhanced Ready 2.0" assessment project in June 2023, with IPv6 on-path telemetry being a pivotal feature to be evaluated. To date, several vendors, including Huawei, have successfully passed the "IPv6 Enhanced Ready 2.0 & SRv6 Ready" certification developed by CAICT's China Telecommunication Technology Labs (CTTL).

The preceding activities have played a key role in promoting the application of IPv6 on-path telemetry. As this innovative technology becomes more widely used on carrier and enterprise networks, the IPv6 on-path telemetry industry will mature and gain a stronger foothold.

8.1.3 Commercial Deployment of IPv6 On-Path Telemetry

As 5G continues to advance, carrier users are placing higher requirements on network quality. It has therefore become increasingly important to effectively monitor the performance of mobile transport networks. Furthermore, with the continued growth of cloud computing, service cloudification has become a top priority for enterprise users. Efficient O&M is vital in cloud network scenarios as solutions such as high-quality IP leased line services and financial WAN emerge. IPv6 on-path telemetry is poised to play a significant role in these scenarios and is being widely adopted by carriers and enterprises worldwide.

IFIT alternate marking is a commercially available solution for IPv6 on-path telemetry. In China, carriers such as China Telecom, China Mobile, and China Unicom have piloted and deployed the IFIT alternate marking solution on a large scale in various provinces. Similarly, carriers around the world, including in Europe, Asia Pacific, and Southern Africa, are actively deploying this innovative solution to improve the intelligent service level of their networks and services. Examples include POST Luxembourg, MTN South Africa, and Singtel. On top of this, multiple industries and enterprises are putting the solution to work. Notable deployment examples include the provincial government networks of Guangdong and Guangxi in China, and the financial backbone networks of Agricultural Bank of China, Industrial and Commercial Bank of China, and Bank of Communications. Preliminary estimates reveal that the IFIT alternate marking solution has already been deployed at over 200 sites.

8.1.3.1 Intelligent O&M of 5G Transport Networks

Leveraging IP RAN, China Telecom has introduced new large-capacity devices and cutting-edge technologies such as SRv6, Flexible Ethernet (FlexE), and IFIT to build a 5G Smart Transport Network (STN). This network carries mobile backhaul services (e.g., 3G, 4G, and 5G services) and 5G+cloud-network services, including government/enterprise Ethernet private lines, cloud private lines, and cloud private networks[27], as depicted in Figure 8.2. The STN covers local networks across all provinces in China, making it one of the largest IPv6 Enhanced-enabled networks in the world. It drastically simplifies network protocols and achieves network intelligence, agility, and efficiency.

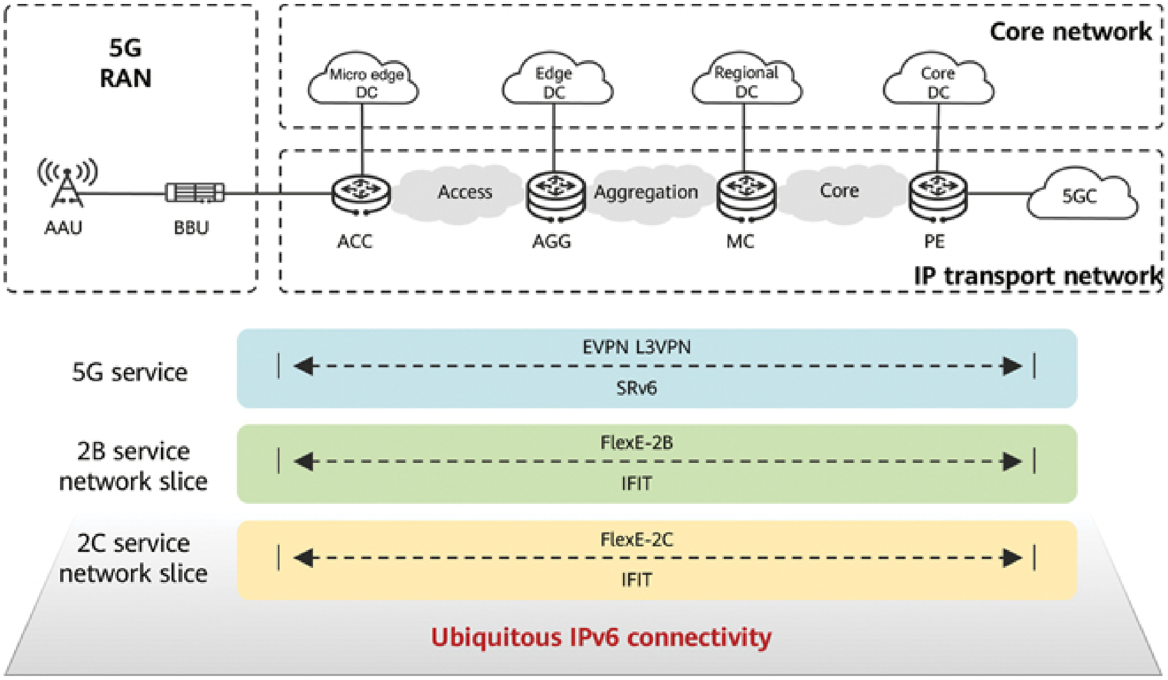


FIGURE 8.2 China telecom's E2E IPv6 Enhanced 5G STN. ⏎

China Telecom has introduced a comprehensive IFIT system into its 5G STN. This system has transformed the previous user complaint–driven manual troubleshooting process that took hours to complete. Now, it only takes one minute to detect poor-QoE services and another minute to demarcate and automatically fix faults, with no customer complaints. With SRv6 deployed across domains, IFIT can be used together with the intelligent management and control system to quickly detect and analyze poor-QoE services in cross-domain scenarios, and quickly demarcate and automatically rectify faults.

8.1.3.2 Quality Assurance of the Data Private Network for the Beijing Winter Olympics

The Beijing Winter Olympics was a landmark event that required multiple network communication services across 87 venues and service facilities spread over two cities and three competition areas, and across multiple transportation lines between Beijing and Zhangjiakou. Such services included shared Internet, Internet private lines, and media+. It imposed strict requirements on the terminal access efficiency, network packet loss rate, round-trip delay, access- and core-layer network availability, and network troubleshooting time. Furthermore, different levels of venues had different indicator requirements. To meet these demanding requirements, China Unicom built the IPv6 Enhanced data private network[28], whose architecture is shown in Figure 8.3.

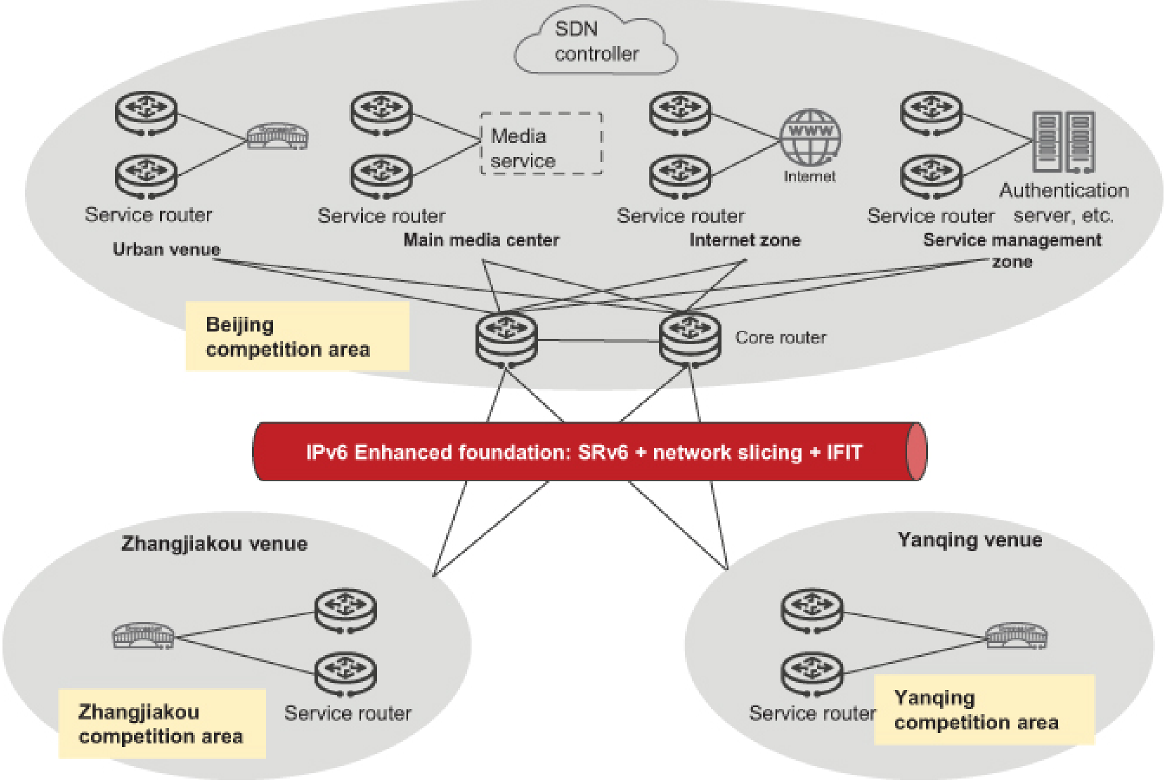


FIGURE 8.3 IPv6 Enhanced data private network architecture for the Winter Olympics. ⏎

IFIT was deployed across the entire network to monitor end-to-end SLAs and hop-by-hop forwarding information of services in real time. It allowed poor-QoE services to be actively measured within minutes, transforming passive O&M into proactive preventive O&M and empowering O&M personnel to accurately diagnose root causes with the help of AI.

8.1.3.3 Visualized O&M of an E-government Cloud Network

In line with its commitment to advancing digital governance, Guangdong has built a next-generation e-government cloud extranet. Figure 8.4 shows the networking architecture of Guangdong's IPv6 Enhanced-enabled e-government cloud network, which includes both the extranet and intra-cloud network.

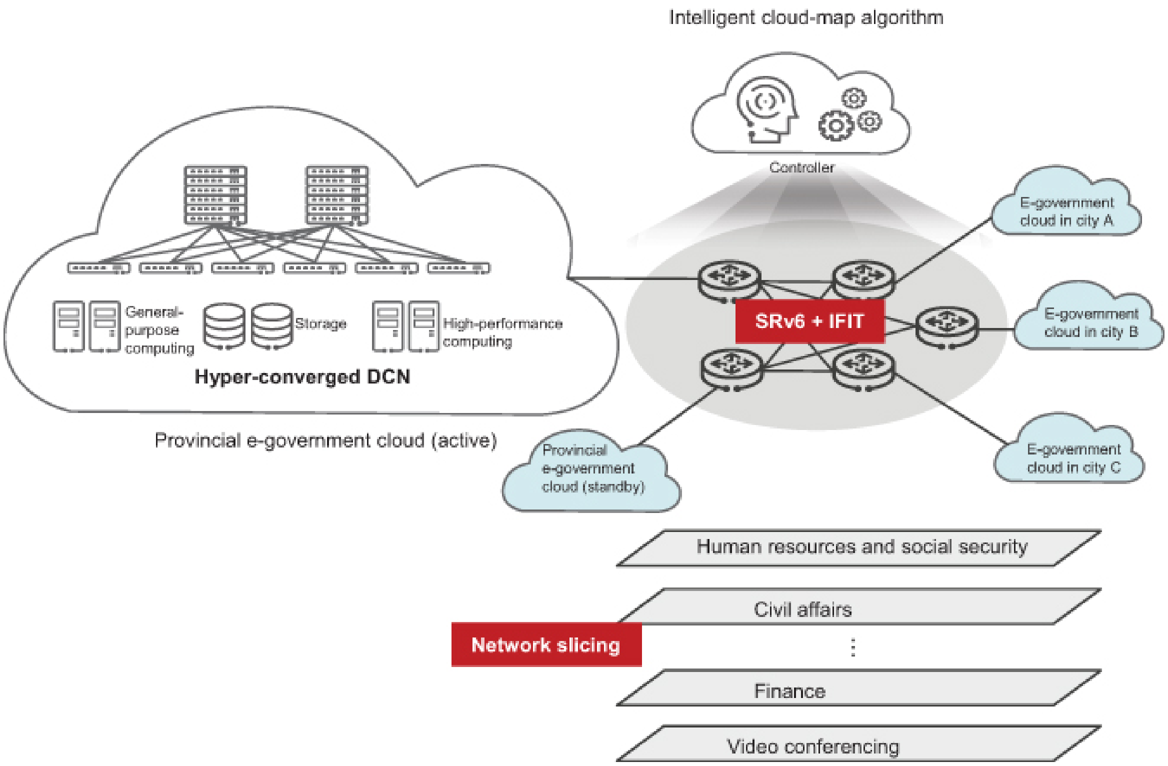

Intelligent cloud-map algorithm
Controller
E-government cloud in city A
E-government cloud in city B
E-government cloud in city C
SRv6 + IFIT
General-purpose computing
Storage
High-performance computing
Hyper-converged DCN
Provincial e-government cloud (active)
Provincial e-government cloud (standby)
Human resources and social security
Civil affairs
Network slicing
⋮
Finance
Video conferencing

FIGURE 8.4 Architecture of the IPv6 Enhanced-enabled e-government cloud network in Guangdong province. ⏎

IFIT enables real-time monitoring of service quality and forms the basis of a visualized network operation platform that can display information about devices, assets, lines, traffic, and more. This allows the network health status to be viewed online in real time, providing valuable insights into the status of the e-government cloud network. An intelligent algorithm is used to predict the probability of traffic congestion occurring and the risks in device/line operations, enabling proactive adjustment of traffic paths to avoid faults. It can also automatically fix common faults such as line congestion and bit errors and perform optimization.

Take the video conference private network as an example. The provincial network platform of the Guangdong e-government extranet provides uninterrupted support for video conferences, achieving zero service faults and zero packet loss. It supported 1500 concurrent users during peak hours, with the total bandwidth exceeding 1.5 Gbps. This network ensured that smooth video conferencing services were available 24/7.

8.2 TECHNOLOGY PROSPECTS OF IPV6 ON-PATH TELEMETRY

The 5G and cloud era is driving the development of IP networks toward IPv6 Enhanced. In the future, IP networks must have three features: intelligent ultra-broadband, intelligent connection, and intelligent O&M. The latter feature is vital for ensuring the SLA of future network services and in implementing automated and intelligent IP networks. It analyzes real-time performance measurement data of the entire network and takes actions (i.e., intervenes, adjusts, and optimizes) to avoid possible network risks. This transforms the traditional fault-driven network O&M mode into a proactive and predictive one.

IPv6 Enhanced is the optimal choice for intelligent IP networks. And an important part of IPv6 Enhanced is IPv6 on-path telemetry, one of the core technologies for intelligent O&M. IPv6 on-path telemetry is designed to build a complete on-path telemetry system that implements fast fault detection and self-healing, meeting intelligent O&M requirements in the 5G and cloud era. Looking ahead, new intelligent O&M requirements and challenges will emerge, brought by the cloudification of various industries — the IoT, Internet of Vehicles (IoV), and industrial Internet — and intelligent connectivity of everything. This on-path telemetry system therefore needs to be further improved. For example, more measurement types and parameters need to be added to improve the measurement precision, greater automation is needed to simplify deployment, and the volume of sent data needs to be reduced to improve performance.

8.2.1 Continuous Optimization of IOAM

IOAM is the ideal option for on-path telemetry, but it poses significant engineering challenges due to its data plane information measurement and massive data reporting. These two aspects will therefore continue to be optimized.

8.2.1.1 Path Tracing

Path Tracing (PT)[29] optimizes IOAM tracing options by tracing network paths with a standardized, optimized information recording format.

PT records a data packet's path as a sequence of interface IDs and provides a record of E2E delay, per-hop delay, and payload on each outbound interface along the packet delivery path. It supports tracing of up to 14 hops with a 40-byte IPv6 HBH extension header, minimizing bandwidth overhead.

8.2.1.1.1 Midpoint Compressed Data Record

Each PT midpoint along a data delivery path records the Midpoint Compressed Data (MCD) into the Hop-by-Hop Option for Path Tracing (HBH-PT) header.

The MCD contains the following information:

- MCD.OIF (Outgoing Interface ID): An 8- or 12-bit interface ID associated with the outbound physical interface of a router.
    - Allocated by a network service provider, an interface ID does not need to be globally unique across the entire network. As long as an E2E path can be inferred from the interface ID chain, the same interface ID can be repeated multiple times on the network.
    - Interface IDs can be programmed on a device using CLI/NETCONF or other means (this is outside the scope of this book).
    - A network service provider can choose to use either an 8- or 12-bit interface ID, but this must be consistent across the entire network.
    - In a Link Aggregation Group (LAG), each member interface is configured with a unique interface ID.
- MCD.OIL (Outgoing Interface Load): A 4-bit representation of the outbound interface load (i.e., current throughput relative to the interface bandwidth).
    - MCD.TTS (Truncated Timestamp): An 8-bit timestamp that encodes the time at which a data packet leaves a router.
    - PT uses the 64-bit timestamp format. RFC 8877 recommends two 64-bit timestamp formats[30]: 64-bit truncated PTP timestamp and 64-bit NTP timestamp.
    - A TTS template is configured for each outbound interface on the device.
    - A TTS template defines the position of 8 bits to be selected from the egress timestamp.
    - A PT midpoint implementation may support one or more TTS templates, with each TTS providing a different time precision.
    - An operator configures an outbound interface with a single TTS template. Deciding which TTS template to use for a given interface is based on the type of the link connected to that interface.
    - All routers on the network must run a time synchronization protocol (e.g., PTP or NTP).

8.2.1.1.2 Data Plane Behavior of PT Nodes

1. Data plane behavior of a PT source

    A PT source is a source node that starts a PT probing instance and generates PT probes. It generates a PT probe packet for each configured PT probing instance at the specified probe rate as follows:

i. Generates a new IPv6 data packet.
ii. Sets the IPv6 Source Address (SA) as per the PT probing instance configuration.
iii. Sets the IPv6 Destination Address (DA) to the first Segment ID (SID) from the SRv6 segment list.
iv. Sets the IPv6 Next Header field to zero (HBH).
v. Sets the DSCP and Flow Label values as per the PT probing instance configuration.
vi. Appends an IPv6 HBH header with the HBH-PT.
vii. Sets all bits of the HBH-PT MCD stack to zero.
viii. Appends an SRH if the segment list has more than one SID.
ix. Sets the Next Header field to 60 (IPv6 Destinations Options header).
x. Writes the remaining SIDs of the SID list in the SRH.
xi. Appends an IPv6 Destinations Option header with the IPv6 Destination Option for Path Tracing (DOH-PT).
xii. Sets the Next Header field of the IPv6 Destinations Options header to 59 (IPv6 No Next Header).
xiii. Adds padding bytes after the IPv6 Destinations Option header to reach the desired packet size according to the MTU sweeping range configuration.
xiv. Sets the 16-bit Session ID field of the DOH-PT according to the PT probing instance configuration.
xv. Performs an IPv6 FIB lookup to determine the outbound interface (IFACE-OUT) on which the packet will be forwarded.
xvi. Records Transmit 64-bit timestamp (SRC.T64) in the T64 field of the DOH-PT.
xvii. Records IFACE-OUT ID (SRC.OIF) in the IF_ID field of the DOH-PT.
xviii. Records IFACE-OUT Load (SRC.OIL) in the IF_LD field of the DOH-PT.
xix. Forwards the packet through the outbound interface.

2. Data plane behavior of a PT midpoint

   A PT midpoint is a transit node that delivers PT packets. When a PT midpoint receives an IPv6 packet containing the IPv6 HBH-PT option, it computes and records its own MCD information.

3. Data plane behavior of a PT sink

   A PT sink is a node that receives PT probes from the source. These probes contain information recorded by every PT midpoint along the path. The sink then forwards the probes to a Regional Collector (RC) after recording its own PT information. Once received, the RC parses and stores the information in a database. Using the PT information, the RC constructs the packet delivery path and determines the timestamp at each node.

8.2.1.1.3 PT Header Formats

PT defines a new IPv6 option to be carried in the IPv6 HBH Options Header — HBH-PT. Figure 8.5 depicts its format.

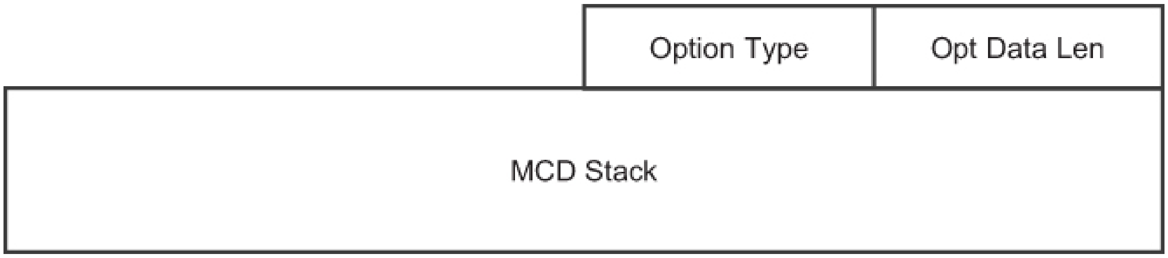


FIGURE 8.5 IPv6 HBH-PT format. ⏎

Table 8.4 describes the involved fields.

TABLE 8.4 Fields in the IPv6 HBH-PT ⏎

Field	Length	Description
Option Type	8 bits	The specific value is to be assigned. The 3 high-order bits of the option must be set to 001. 00 indicates that nodes that do not support the HBH-PT skip HBH. 1 indicates that nodes that support the HBH-PT update the option.
Opt Data Len	8 bits	Length (in bytes) of the MCD stack.
MCD Stack	Variable	Metadata scratchpad where PT midpoints record their MCD.

PT also defines a new IPv6 option to be carried in the IPv6 DOH — DOH-PT. Figure 8.6 depicts its format.

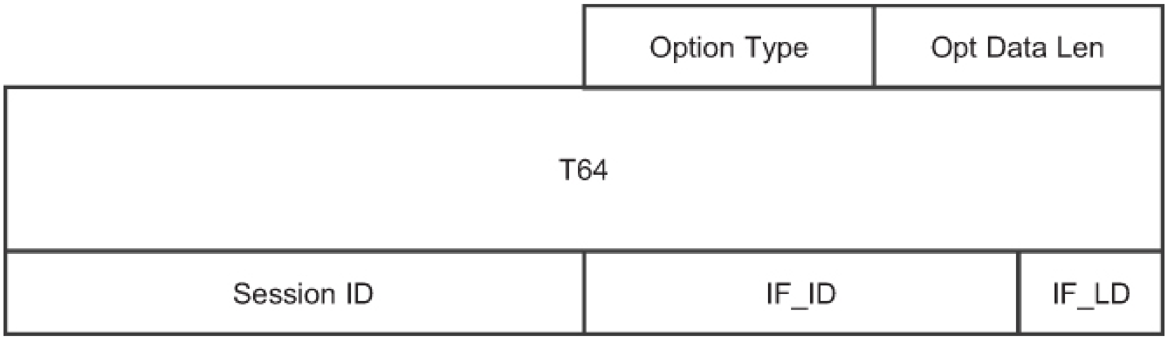


FIGURE 8.6 IPv6 DOH-PT format. ⏎

Table 8.5 describes the involved fields.

TABLE 8.5 Fields in the IPv6 DOH-PT ⏎

Field	**Length**	**Description**
Option Type	8 bits	The specific value is to be assigned. The 3 high-order bits of the option must be set to 000. The first two zeros indicate that nodes that do not support the DOH-PT skip the IPv6 DOH. The last zero indicates that nodes cannot change the DOH-PT in routes.
Opt Data Len	8 bits	Length (in bytes) of the DOH-PT (fixed at 12).
T64	64 bits	Timestamp.
Session ID	16 bits	Session identifier set by the source node generating probes. It is used to associate probes of the same session. Value 0 indicates that this field is not set.
IF_ID	12 bits	Interface ID.
IF_LD	4 bits	Interface load.

8.2.1.1.4 Summary of the PT Solution

Earlier descriptions have taken PT nodes using active measurement, which requires probe packets, as an example to explain data plane behaviors. In practice, however, the PT solution can also be used for on-path measurement. The advantages of the PT solution are as follows.

- Low overhead: A 40-byte HBH header allows for 14-hop path measurement — 1 at the PT source, 12 at PT midpoints, and 1 at the PT sink.
- Line rate and hardware friendliness:
    - Achieves the line rate in current hardware using the regular forwarding pipeline.
    - Leverages mature hardware capabilities (basic shift operation) without the need to adjust the packet size at every node along the path.
- Scalable fine-grained timestamp: The PT source and sink support 64-bit timestamps, and the PT midpoints support 8-bit timestamps.
- Scalable load measurement.

The PT solution uses the passport measurement method. Compared with IOAM tracing options, PT drastically reduces the overhead of on-path measurement. However, it supports only limited types of measurement data (mainly the outbound interface, delay, and load). This introduces a scalability problem if PT is adapted to support more types of data.

8.2.1.2 Lightweight IOAM for SRv6 Network

Lightweight IOAM for SRv6 network[31] reuses the segment lists in the SRv6 SRH to record the forwarding information of involved nodes. This ensures that the packet length does not increase during on-path measurement.

SRv6 SIDs are typically not used after they are processed according to RFC 8754[32]. Because processed SIDs are still retained in the SRH and transmitted to the destination of the packet, their space can be reused for other purposes, such as carrying performance measurement or IOAM information.

Given that IOAM data needs to meet the accuracy requirement defined in RFC 9197[8], this solution assumes that the rewriteable length of a SID is at least 64 bits. In addition, to determine which node the IOAM data is related to, it is necessary to retain the locator (LOC) part that identifies the node in the SID.

In order to indicate the type of IOAM data and additional operations (e.g., IOAM operations), a new SID field called FLAG is defined in this solution. With the FLAG field, the complete SID format is LOC:FLAG:FUNCT. The IOAM data stored in the SID is structured in the following format: <FLAG><IOA Mdata>.

The offset and length of the FLAG field can be configured by a network administrator. For example, this field can be an 8-bit value between the LOC and FUNCT fields, and the offset can be the 48th bit. In other words, bits 0–47 form the LOC, bits 48–55 form the FLAG, and bits 56–127 form the FUNCT fields. The following values may be used in the FLAG field:

- 0: non-IOAM.
- 1: timestamp.
- 2: data packet counter.
- 3: queue depth.
- 4: ingress_if_id and egress_if_id (short format).
- 5: Hop_Lim and node_id.
- 6: namespace-specific data.
- 7: buffer occupancy.
- 8: checksum complement.
- 9–255: reserved.

While lightweight IOAM for SRv6 network ensures that the packet length does not increase during on-path measurement, it has limitations on application scenarios and needs to be selected based on actual requirements given that IOAM tracing options are expected to trace more types of information.

8.2.2 Finer-Grained Path Visualization

IOAM defines several IOAM Trace-Types that identify information about nodes through which packets pass[8]. As IP forwarding evolves, more types of forwarding information may be needed for finer-grained path visualization. In particular, visualizing network forwarding policies is crucial.

IP networks often use ACLs, Policy-Based Routing (PBR), or FlowSpec routes as forwarding policies. Unlike IP routing, which forwards packets over the shortest path, these policies forward packets based on classification information such as 5-tuple to meet service or security requirements. However, the following issues may arise because there is no effective way to monitor the validity of these policies.

- Accumulation of forwarding policies makes maintenance more and more difficult.
- Whether service packets use the correct forwarding policy cannot be effectively verified.

IOAM can be used to record the forwarding policies applied to packets, offering a more effective way to visualize forwarding policies. By collecting statistics on actual service packet forwarding policies, it is possible to determine which forwarding policies are applied and re-evaluate or remove those that have not been used for a long time. It is also possible to determine whether a packet’s forwarding path meets expectations or needs to be adjusted based on the associated forwarding policy. These characteristics facilitate effective maintenance of network forwarding policies.

8.2.3 On-Path Telemetry for IP Multicast

IPv6 on-path telemetry is widely used for unicast services. Given that multicast is another important type of IP network service, equal attention needs to be paid to multicast on-path telemetry.

IP multicast is extensively used in scenarios such as real-time interactive online conferences, IPTV, and real-time data transmission in online financial markets. Multicast packet loss and delay adversely affect application performance and user experience.

A multicast channel is generally identified by a source and a group (S, G). IP multicast packets of a specific (S, G) are replicated and sent to multiple receivers, creating multiple copies of the original packets on the network. Consequently, the trace data obtained when IOAM tracing options are used for multicast packets is replicated into the packet copy for each branch of the multicast tree. Except the data from the final leaf branch, most of the other data is redundant. This redundancy increases packet header overhead, wastes bandwidth, and complicates data processing.

To solve these problems, RFC 9630 defines the following two solutions[12].

8.2.3.1 IOAM DEX Option Enhancement Solution

Postcard-based on-path telemetry solutions (e.g., IOAM DEX) can eliminate data redundancy because each node on the multicast tree sends a postcard with only local data. However, these solutions cannot accurately correlate the measurement data with multicast paths due to the lack of branching information.

For instance, in the multicast tree shown in Figure 8.7, node B has two branches, one to node C and the other to node D. Node C leads to node E, and node D leads to node F. With postcard-based solutions, the received postcards are insufficient for determining whether node E is the next hop of node C or node D. It is necessary to collect information about the multicast path tree by other means (e.g., through multicast trace, or MTrace) and correlate it with measurement information. However, such correlation is undesirable because it introduces extra work and complexity.

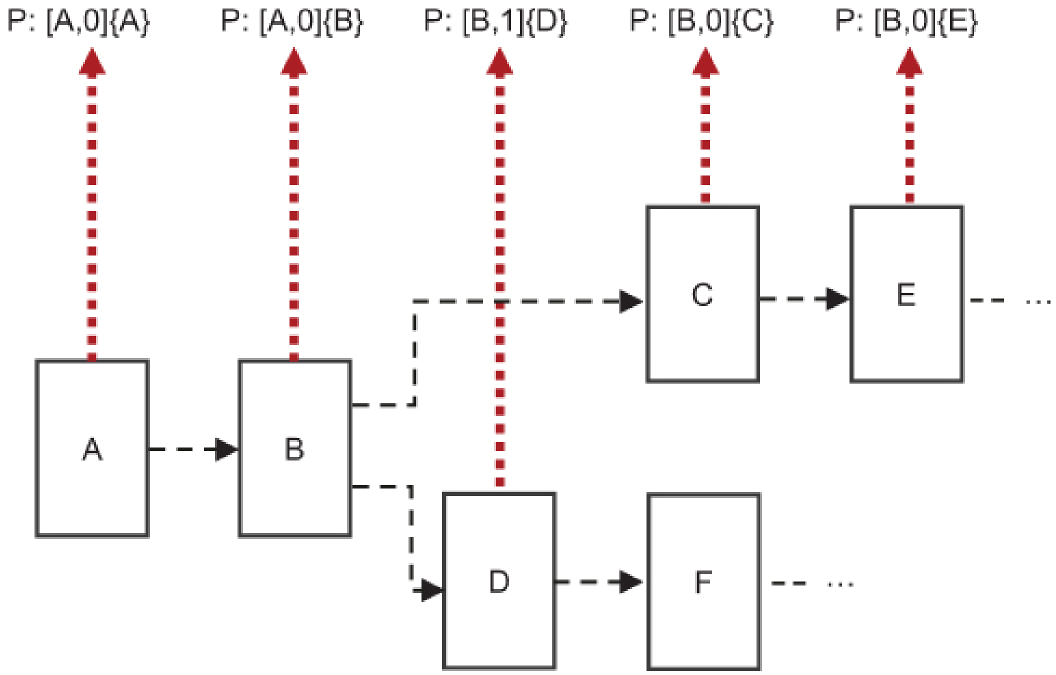


FIGURE 8.7 Example of per-hop postcard-based telemetry. ⏎

This problem exists because there is no identifier (either implicit or explicit) to correlate the telemetry data on each branch. One way to correlate the measurement data with the multicast path tree is to transfer the branch identifier by using the IOAM DEX option. The branching node ID plus an index is used to make the branch identifier globally unique.

In Figure 8.7, P represents a postcard packet, the square brackets contain branch identifiers, and the braces contain telemetry data about specific nodes. Here, [B, 0] is used as the branch identifier for the branch from node B to node C, and [B, 1] is used for the branch to D. The branch identifier is carried in the multicast packet until the next branch fork node. In the postcards each node sends, the node must export the branch identifier in the received IOAM DEX Option-Type header. The branch identifier, along with other fields such as Flow ID and Sequence Number, is sufficient for the data collector to reconstruct the topology of the multicast tree.

Each branch fork node needs to generate a unique branch identifier (i.e., multicast branch ID) for each branch in its multicast tree instance and include it in the IOAM DEX Option-Type header. The multicast branch ID remains unchanged until the next branch fork node. Such a branch ID contains two parts: Branching Node ID and Interface Index. Figure 8.8 shows the format of a multicast branch ID.

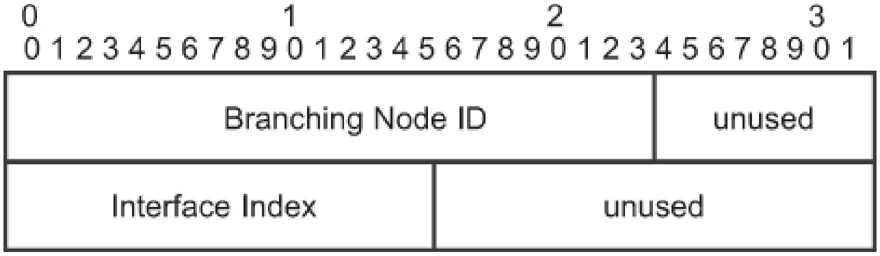


FIGURE 8.8 Multicast branch ID format. ⏎

Table 8.6 describes the involved fields.

TABLE 8.6 Fields in a Multicast Branch ID ⏎

Field	Length	Description
Branching Node ID	3 bytes	Node ID. Complies with the node ID specification in IOAM[8].
Interface Index	2 bytes	Interface index.
unused	1 byte or 2 bytes	Unused bits, set to 0.

In the IOAM DEX Option-Type header, Multicast Branch ID is carried as an optional field following the optional fields Flow ID and Packet Sequence Number. Figure 8.9 shows the format of such a header.

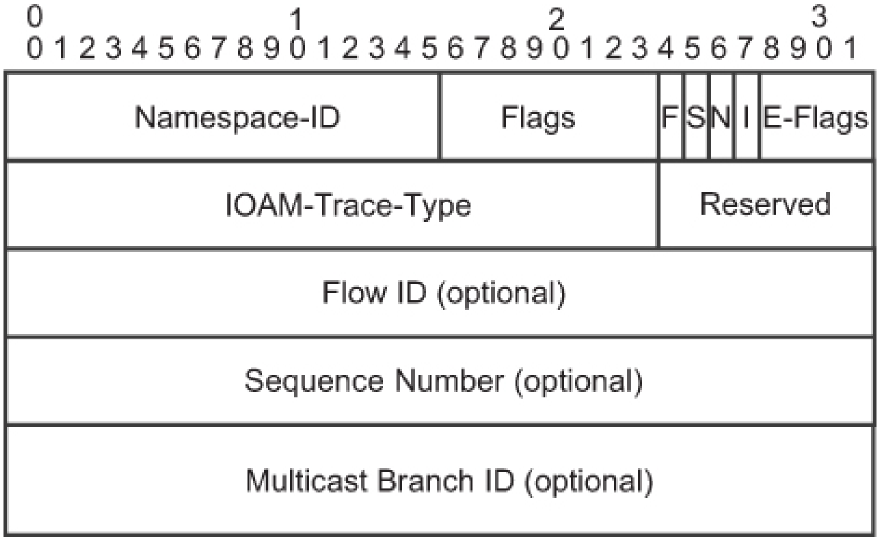


FIGURE 8.9 Format of the IOAM DEX Option-Type header carrying multicast branch ID. ⏎

Table 8.7 describes the involved fields.

TABLE 8.7 Fields in the IOAM DEX Option-Type Header Carrying Multicast Branch ID ⏎

Field	Length	Description
Namespace-ID	16 bits	IOAM namespace identifier. IOAM processing is performed only when the locally configured namespace ID is the same as the one in the packet. 0x0000 is the default namespace ID, which must be supported by all IOAM nodes.
Flags	8 bits	Flag bits, which are not yet defined.
F	1 bit	Flow identifier flag. Value 1 indicates that the optional Flow ID field is valid.
S	1 bit	Sequence number flag. Value 1 indicates that the optional Sequence Number field is valid.
N	1 bit	Node ID flag. When both this field and field I are set to 1, the optional Multicast Branch ID field is present. The two fields must be both set or cleared. Otherwise, the packet header format is considered malformed, and the packet must be discarded. Two extension flag bits are used. This is because Multicast Branch ID needs more than 4 bytes to encode, but RFC 9326[9] specifies that each extension flag only indicates the presence of a 4-byte optional data field.
I	1 bit	Interface index flag. When both this field and field N are set to 1, the optional Multicast Branch ID field is present.
E-Flags	4 bits	Extension flags (similar to the F, S, N, and I fields) that carry extension information indications.
IOAM-Trace-Type	24 bits	Type of IOAM trace data to be output.
Reserved	8 bits	Reserved field.

Field	Length	Description
Flow ID	32 bits	Optional flow ID. When multiple IOAM measurement nodes report data, the involved controller calculates measurement data based on the flow ID.
Sequence Number	32 bits	Optional packet sequence number. The value of this field starts from 0 and increases by 1 each time a packet is counted.
Multicast Branch ID	32 bits	Optional multicast branch ID.

After receiving the multicast branch ID information from its upstream node, a node must carry this information in its exported telemetry data. This is necessary to allow the original multicast tree to be correctly reconstructed based on the telemetry data.

8.2.3.2 Hybrid Solution of IOAM Tracing Options + Postcard-Based Telemetry

This solution involves both IOAM tracing options and postcard-based telemetry. To avoid data redundancy caused by using IOAM tracing options in multicast scenarios, the trace information of each branch needs to carry data (including the multicast branch ID) related to each branch fork node.

Figure 8.10 shows an example of per-section postcard-based telemetry, where trace data is exported on a per-section basis in a multicast tree. P represents a postcard packet, and the braces contain telemetry data about specific nodes. Nodes B and D are two branch fork nodes, each of which exports a postcard covering the trace data for the previous section. The end node of each path also exports the data of the last section as a postcard.

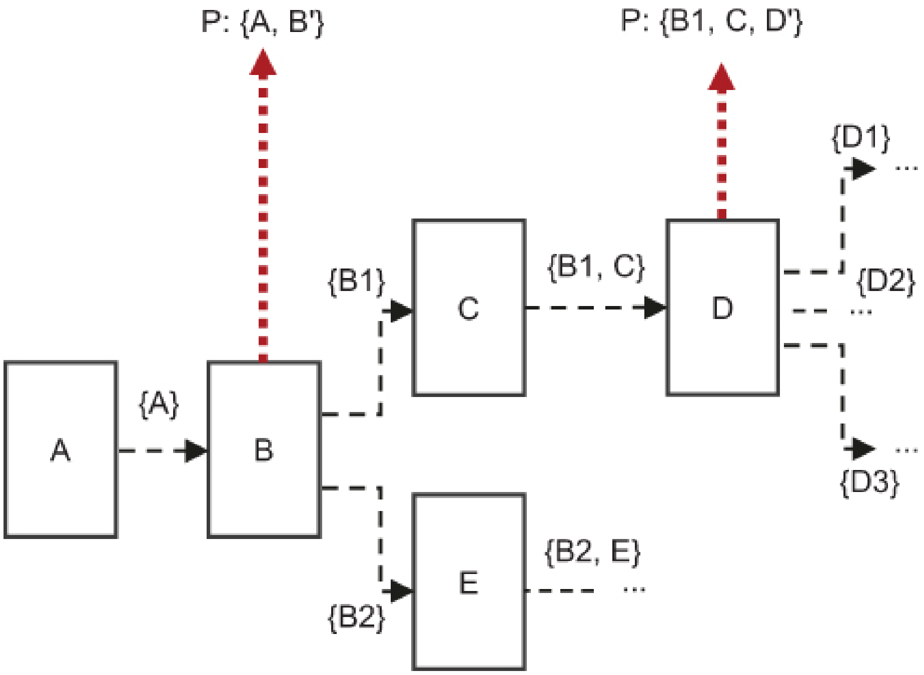


FIGURE 8.10 Example of per-section postcard-based telemetry. ⏎

This solution eliminates the need to modify the IOAM tracing options header format as specified in RFC 9197. It only requires configuring branch fork nodes and leaf nodes, exporting postcards that contain the trace data collected on the nodes so far, and refreshing the IOAM headers and data in the packet (e.g., clearing the node data list to all zeros and resetting the RemainingLen field to the initial value).

8.2.4 E2E On-Path Telemetry for Application-Network Convergence

IPv6 on-path telemetry mainly targets applications and terminals, but networks lack the ability to clearly perceive them. Current methods, such as IP 5-tuple-based identification, are not intuitive, accurate, or fine-grained. To improve application-level network measurement, IPv6 on-path telemetry can be combined with the emerging Application-Aware IPv6 Networking (APN6) technology. APN6 uses the extension header in IPv6 packets to convey application and user information into a network so that the network can provide finer-grained services oriented to applications and users[33]. When IPv6 on-path telemetry is applied to service flows, this information facilitates telemetry for specific applications and users.

Another problem facing IPv6 on-path telemetry is that it typically only measures performance of service flows within a network. In practice, a service flow needs to pass through terminals, networks, and clouds. Performance along each path segment — from a terminal to network and from a network to cloud — may also significantly affect the service quality. To facilitate E2E fault demarcation and location on the entire path, IPv6 on-path telemetry needs to be implemented not only within the network, but also across the entire terminal-network-cloud path. In addition, the applications on the terminal or cloud side can further adjust services and paths according to the E2E measurement result. For example, a video application can adjust the bit rate based on a network measurement status or perform congestion control at the transport layer.

Implementing on-path telemetry across the entire terminal-network-cloud path faces the following challenges:

- The postcard mechanism is difficult to apply. This mechanism requires a unified, centralized controller to collect and analyze data. But it is difficult to implement such a controller because the terminal, network, and cloud are in different administrative domains. In addition, on-path telemetry requires clock synchronization between measurement points, posing engineering difficulties.
- The passport mechanism increases the packet length, although it does not require a centralized controller to collect information reported by nodes along the path. This increase brings the observer effect in on-path telemetry, leading to inaccurate measurement results.

To address the preceding challenges, the congestion measurement mechanism[34] extends Explicit Congestion Notification (ECN) by recording more information without increasing the packet header length, avoiding the problems caused by passport-based on-path telemetry.

Congestion measurement extends the conventional single-bit ECN to multiple bits, allowing network devices to update congestion information at each hop with finer granularity. As a result, the congestion information field in a packet that reaches the receiver accurately indicates not only the presence of congestion but also the congestion degree across the path. This nuanced approach provides a richer set of data for decision-making, enhancing the precision of congestion control and facilitating load balancing and network debugging.

Figure 8.11 shows an overview of the congestion measurement procedure. First, the sender must mark the packet with data fields for congestion measurement. These fields specify what kind of congestion information the sender intends to collect from transit nodes. As the packet traverses the network, each transit node should inspect the data fields and update the congestion information accordingly. Upon receipt of the packet, the receiver extracts the updated congestion information in the packet and then sends it back to the sender.

After acquiring the congestion information that reflects the packet's journey, the sender uses this information to make informed adjustments to its sending rate or load balancing decisions.

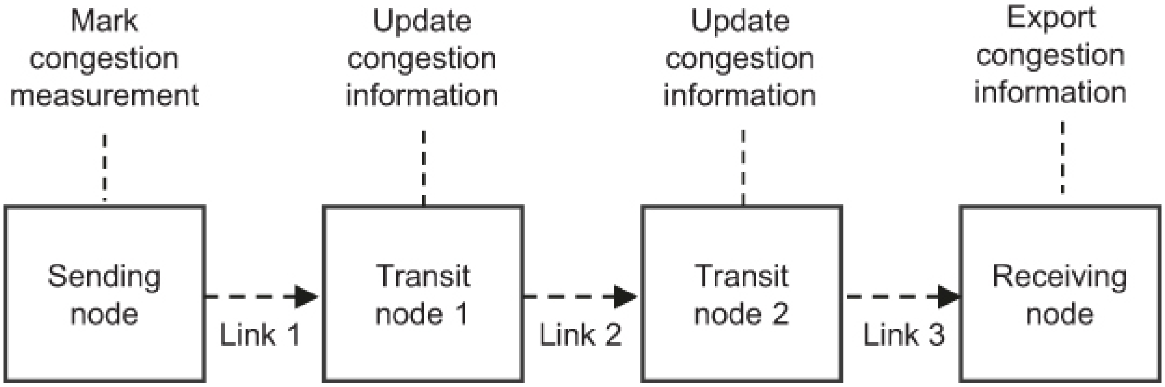

Mark congestion measurement
Update congestion information
Update congestion information
Export congestion information
Sending node
Link 1
Transit node 1
Link 2
Transit node 2
Link 3
Receiving node

FIGURE 8.11 Overview of the congestion measurement procedure. ⏎

Figure 8.12 shows the format of data fields for congestion measurement.

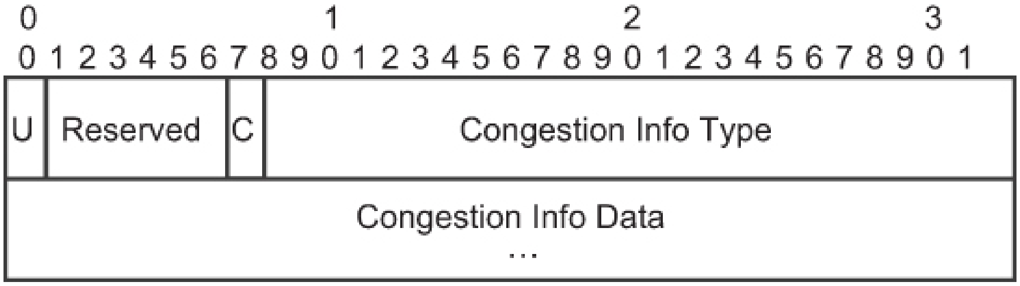


FIGURE 8.12 Data fields for congestion measurement. ⏎

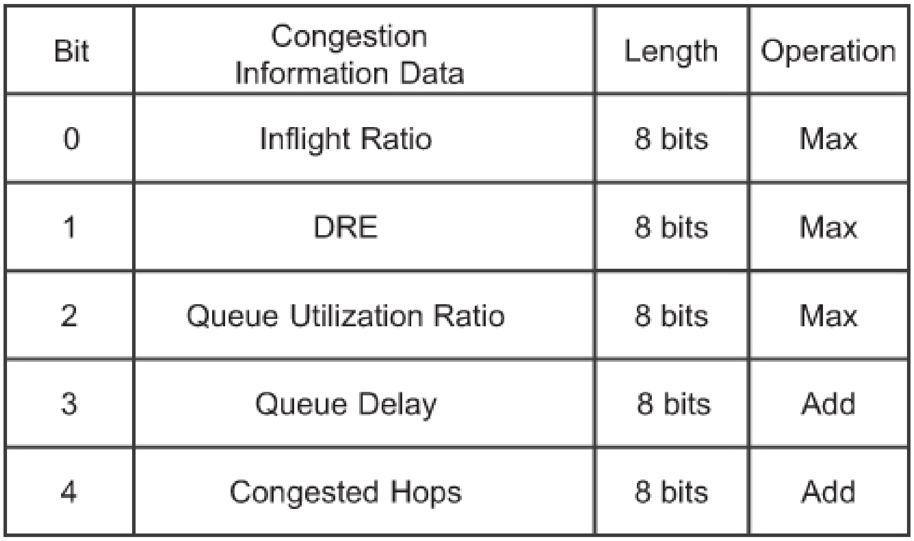
Table 8.8 describes the involved fields.

TABLE 8.8 Data Fields for Congestion Measurement ⏎

Field	**Length**	**Description**
U	1 bit	Indicates whether the Congestion Info Data field needs to be updated by transit nodes. If this bit is set, the transit nodes will update the field. Otherwise, the transit nodes do not update it.
Reserved	6 bits	Reserved bit.
C	1 bit	If the C bit is 1, Congestion Info Data is customized and used only in limited domains such as a data center network. If the C bit is 0, Congestion Info Type is a bitmap.

Field	Length	Description
Congestion Info Type	24 bits	Congestion information type. It is a 24-bit bitmap that specifies the types of the present congestion information data. Note that multiple types of congestion information data can coexist in each packet for the endpoint to collect detailed raw congestion information.
Congestion Info Data	Variable	Congestion information data. Transit nodes must update this field based on the local load status.

Figure 8.13 lists the congestion information data.



Bit	Congestion Information Data	Length	Operation
0	Inflight Ratio	8 bits	Max
1	DRE	8 bits	Max
2	Queue Utilization Ratio	8 bits	Max
3	Queue Delay	8 bits	Add
4	Congested Hops	8 bits	Add

FIGURE 8.13 Congestion information data. ⏎

Table 8.9 describes the involved fields.

TABLE 8.9 Fields Involved in Congestion Information Data ⏎

Field	Length	Description
Inflight Ratio	8 bits	Path utilization rate, which is the maximum link utilization rate obtained through hop-by-hop update along the path.
DRE	8 bits	Discounting Rate Estimator[35], which is used to estimate the path utilization rate based on historical data. The path utilization rate is the maximum link utilization rate obtained through hop-by-hop update.

Field	Length	Description
Queue Utilization Ratio	8 bits	Maximum queue utilization rate obtained through hop-by-hop update along the path.
Queue Delay	8 bits	Queuing delay (in ms), which is the total queuing delay of all devices along the path.
Congested Hops	8 bits	Number of transit nodes with congestion. The value is the total number of times that the ECN threshold is exceeded along the path. (Exceeding the ECN threshold means that the node is congested.)

8.3 STORIES BEHIND IPV6 ON-PATH TELEMETRY

8.3.1 IPv6 Enhanced Innovation and Driving Experience

During the promotion of IPv6 Enhanced innovation, many people often wonder about its value. Initially, we focused on explaining the value from the perspective of its technical principles and advantages. But now we use a more intuitive approach to help people understand its value: likening IPv6 Enhanced innovation to driving a car, something that people are familiar with in their daily lives.

Decades ago, before the advent of Google Maps or navigation systems, driving somewhere often involved planning the shortest route using a physical map and relying on roadside signs for guidance along the way. Given the lack of knowledge about traffic conditions, we often encountered frequent congestion while driving. There was also the possibility of roadworks forcing us to make spontaneous changes to the route. Similarly, on traditional IP networks, packets are forwarded hop by hop based on the destination IP address, with forwarding performed using the best-effort approach along the shortest path. From this analogy, we can see that the experience of IP services often has shortcomings.

Driving experience today has improved significantly thanks to the emergence of tools such as navigation systems and dashboard cameras and the implementation of traffic mechanisms such as dedicated lanes. Tools such as Google Maps and navigation systems enable drivers to plan efficient routes based on traffic conditions and avoid congested or construction areas. Similarly, SRv6 and Network Digital Map serve as navigation systems for service packets on IP networks, enabling the packets to be forwarded along planned paths and improving service experience.

Dashboard cameras are another tool that did not exist decades ago. At that time, it was challenging to prove liability in the event of a traffic accident, bringing inconvenience for all parties involved. Even with the introduction of monitoring cameras, there were still areas that were not covered, and the issue of determining liability remained. It was not until the emergence of dashboard cameras that drivers could easily track their entire journey, minimizing the hassle of disputes over liability if traffic accidents occurred. Similarly, IPv6 on-path telemetry is like equipping service packets on IP networks with a dashboard camera, enabling efficient fault demarcation and location during network failures.

Another aspect in our analogy of IPv6 Enhanced innovation to driving a car is dedicated lanes. Roads used to be relatively narrow and did not require lane division, but as they have become wider and wider, lanes are divided and some are dedicated for certain traffic (e.g., buses). This allows buses to travel smoothly even during peak hours. IPv6 network slicing technology is similar to providing dedicated lanes. It uses dedicated resources to ensure the forwarding of IP service packets, guaranteeing the experience of high-priority services.

There are many similar analogies. One example is likening IPv6-based DetNet technology to high-speed trains where arrival and departure times are strictly controlled. Such strict control is necessary to meet the requirement for deterministic delay in IP services. In summary, innovative IPv6 Enhanced technologies are akin to equipping service packets on IP networks with various tools that enhance the driving experience. By understanding the need for tools to enhance our driving experience, we can also understand the significance and value of innovative IPv6 Enhanced technologies for IP network transport. Today, AI technologies are thriving, and autonomous driving has become the next step for further enhancing the driving experience. Similarly, autonomous driving networks have become an important direction for the development of IP networks. Needless to say, IPv6 Enhanced innovation will continue to develop and play an important role in the intelligent network era.

8.3.2 Information Compression for IPv6 Enhanced Innovation

The development of innovative IPv6 Enhanced technologies has been influenced by the evolving needs. The new requirements of 5G, cloud, and other new applications, along with breakthroughs in network software and hardware capabilities, have driven the support for new network functions via flexible extensions of IPv6 extension headers. Such functions include SRv6 network programming, IPv6 network slicing, and IPv6 on-path telemetry. But while breakthroughs in network hardware make IPv6 Enhanced innovation possible, it does not mean that IP headers can be extended infinitely. Moreover, interface

MTU on networks has limitations, whereby if the IP header becomes too long, the effective payload is reduced, severely impacting transmission efficiency. To control the length of the IPv6 packet header, one of the key technical challenges for IPv6 Enhanced innovation is how to carry information efficiently using limited space. Various techniques have been developed to address this challenge. For example, the SRv6 header compression technology[36] eliminates redundant locator block information from the segment list in the SRH. Another example is the postcard mode in IPv6 on-path measurement. Unlike the passport mode, this mode ensures that the length of the on-path measurement information remains unchanged in IPv6 headers. Information control is important in not only the data plane but also the management and control planes for IPv6 Enhanced innovation. For example, in IPv6 network slicing, the topology attribute is separated from the resource attribute to effectively reduce the amount of IGP flooding information. And in IPv6 on-path telemetry, the alternate marking method reports compressed information based on the measurement period (differing from IOAM per-packet reporting).

While these innovative IPv6 Enhanced technologies can be likened to automotive driving tools, such tools have far more CPU and memory resources than network devices. In IPv6 Enhanced innovation, breakthroughs in network hardware and software capabilities bring significant benefits. At the same time, it is necessary to strictly control the increase of information in the data, management, and control planes. These two aspects must be closely integrated, and this also provides valuable insight for the future development of IPv6 Enhanced innovation.

REFERENCES

1. Fioccola G, Zhu K, Graf T et al. Alternate marking deployment framework [EB/OL]. (2024-07-03) [2024-09-30]. Draft-ietf-ippm-alt-mark-deployment-01. ⏎
2. Brockners F, Bhandari S, Bernier D et al. In situ operations, administration, and maintenance (IOAM) deployment [EB/OL]. (2023-04) [2024-09-30]. RFC 9378. ⏎
3. Fioccola G, Cociglio M, Mirsky G et al. Alternate-marking method [EB/OL]. (2022-12) [2024-09-30]. RFC 9341. ⏎
4. Fioccola G, Cociglio M, Sapio A et al. Clustered alternate-marking method [EB/OL]. (2022-12) [2024-09-30]. RFC 9342. ⏎
5. Fioccola G, Zhou T, Cociglio M et al. IPv6 application of the alternate-marking method [EB/OL]. (2022-12) [2024-09-30]. RFC 9343. ⏎

6. Cheng W, Min X, Zhou T et al. Encapsulation for MPLS performance measurement with alternate-marking method [EB/OL]. (2024-09-12) [2024-09-30]. Draft-ietf-mpls-inband-pm-encapsulation-18. ⏎
7. Zhou T, Fioccola G, Liu Y et al. Enhanced alternate marking method [EB/OL]. (2024-05-27) [2024-09-30]. Draft-zhou-ippm-enhanced- alternate-marking-15. ⏎
8. Brockners F, Bhandari S, Mizrahi T. Data fields for in situ operations, administration, and maintenance (IOAM) [EB/OL]. (2022-05) [2024-09-30]. RFC 9197. ⏎
9. Song H, Gafni B, Brockners F et al. In situ operations, administration, and maintenance (IOAM) direct exporting [EB/OL]. (2022-11-15) [2024-09-30]. RFC 9326. ⏎
10. Mizrahi T, Brockners F, Bhandari S et al. In situ operations, administration, and maintenance (IOAM) loopback and active flags [EB/OL]. (2022-11-15) [2024-09-30]. RFC 9322. ⏎
11. Bhandari S, Brockners F. IPv6 options for in situ operations, administration, and maintenance (IOAM) [EB/OL]. (2023-09) [2024-09-30]. RFC 9486. ⏎
12. Song H, Mcbride M, Mirsky G et al. Multicast on-path telemetry using in situ operations, administration, and maintenance (IOAM) [EB/OL]. (2024-08) [2024-09-30]. RFC 9630. ⏎
13. Wang Y, Zhou T, Qin F et al. IGP extensions for in-situ flow information telemetry (IFIT) capability advertisement [EB/OL]. (2021-01-29) [2024-09-30]. Draft-wang-lsr-igp-extensions-ifit-01. ⏎
14. Wang Y, Zhou T, Liu M et al. BGP-LS extensions for in-situ flow information telemetry (IFIT) capability advertisement [EB/OL]. (2021-01-14) [2024-09-30]. Draft-wang-idr-bgpls-extensions-ifit-00. ⏎
15. Fioccola G, Pang R, Wang S et al. Advertising in-situ flow information telemetry (IFIT) capabilities in BGP [EB/OL]. (2024-07-05) [2024-09-30]. Draft-ietf-idr-bgp-ifit-capabilities-05. ⏎
16. Ali Z, Filsfils C, Matsushima S et al. Operations, administration, and maintenance (OAM) in segment routing over IPv6 (SRv6) [EB/OL]. (2022-06-23) [2024-09-30]. RFC 9259. ⏎
17. Qin F, Yuan H, Yang S et al. BGP SR policy extensions to enable IFIT [EB/OL]. (2024-04-19) [2024-09-30]. Draft-ietf-idr-sr-policy-ifit-08. ⏎
18. Yuan H, Wang X, Yang P et al. Path computation element communication protocol (PCEP) extensions to enable IFIT [EB/OL]. (2024-07-05) [2024-09-30]. Draft-ietf-pce-pcep-ifit-05. ⏎
19. Zhou T, Guichard J, Brockners F et al. A Yang data model for in situ operations, administration, and maintenance (IOAM) [EB/OL]. (2024-08)

[2024-09-30]. RFC 9617. ⏎

20. Graf T, Wang M, Fioccola G et al. A Yang data model for the alternate marking method [EB/OL]. (2024-09-02) [2024-09-30]. Draft-ydt-ippm-alt-mark-yang-03. ⏎
21. Zheng G, Zhou T, Graf T et al. UDP-based transport for configured subscriptions [EB/OL]. (2024-07-04) [2024-09-30]. Draft-ietf-netconf-udp-notif-14. ⏎
22. Zhou T, Zheng G, Voit E et al. Subscription to distributed notifications [EB/OL]. (2024-04-28) [2024-09-30]. Draft-ietf-netconf-distributed-notif-09. ⏎
23. Fioccola G, Zhou T. On-path telemetry YANG data model [EB/OL]. (2024-06-19) [2024-09-30]. Draft-fz-ippm-on-path-telemetry-yang-00. ⏎
24. Graf T, Fioccola G, Zhou T et al. IPFIX alternate-marking information [EB/OL]. (2024-07-08) [2024-09-30]. Draft-ietf-opsawg-ipfix-alt-mark-00. ⏎
25. Spiegel M, Brockners F, Bhandari S et al. In-situ OAM raw data export with IPFIX [EB/OL]. (2024-02-12) [2024-09-30]. Draft-spiegel-ippm- ioam-rawexport-07. ⏎
26. Wu B, Wu Q, Boucadair M et al. A Yang data model for network and VPN service performance monitoring [EB/OL]. (2023-04) [2024-09-30]. RFC 9375. ⏎
27. Shui Y. China Telecom Wu Youming: Leading IP network transformation and innovation around four technologies [N/OL] C114. (2020-09-01) [2024-09-30]. ⏎
28. Tu L, Cao C, Tong T. Innovative private network solution for the beijing winter olympics based on IPv6+[J]. *Posts and Telecommunications Design Technology*. (2022-04) [2024-09-30]. ⏎
29. Filsfils C, Abdelsalam A, Camarillo P et al. Path tracing in SRv6 networks [EB/OL]. (2023-10-23) [2024-09-30]. Draft-filsfils-spring-path-tracing-05. ⏎
30. Mizrahi T, Fabini J, Morton A. Guidelines for defining packet timestamps [EB/OL]. (2020-09) [2024-09-30]. RFC 8877. ⏎
31. Li C, Cheng W, Chen M et al. A light weight IOAM for SRv6 network programming [EB/OL]. (2021-02-04) [2024-09-30]. Draft-li-spring-light-weight- srv6-ioam-02. ⏎
32. Filsfils C, Dukes D, Previdi S et al. IPv6 segment routing header (SRH) [EB/OL]. (2020-03) [2024-09-30]. RFC 8754. ⏎
33. Li Z, Peng S, Li C et al. Application-aware IPv6 Networking (APN6) encapsulation [EB/OL]. (2021-08-26) [2024-09-30]. Draft-li-6man-app-

aware- ipv6-network-03. ⏎

34. Shi H, Zhou T, Li Z. Data fields for congestion measurement [EB/OL]. (2024-03-04) [2024-09-30]. Draft-shi-ippm-congestion-measurement-data-00. ⏎
35. Alizadeh M, Edsall T, Dharmapurikar S et al. Conga: Distributed congestion-aware load balancing for datacenters [EB/OL]. (2014-08-17) [2024-09-30]. ⏎
36. Cheng W, Filsfils C, Li Z et al. Compressed SRv6 segment list encoding [EB/OL]. (2024-07-22) [2024-09-30]. Draft-ietf-spring-srv6-srh-compression-18. ⏎

CHAPTER 9

Journey along IPv6 On-Path Telemetry

DOI: 10.1201/9781003677901-12

9.1 HARD PRACTICE

Huawei's research into on-path measurement techniques began long ago. In 2013, the Network Solution Department proposed the IP FPM technique, which utilized alternate marking for on-path measurement. The name "IP FPM" was coined by Susan Hares from Futurewei. Susan, a highly respected figure in the IP field and a co-chair of the IETF Inter-Domain Routing (IDR) WG, was then leading IP technology research and standards promotion at Futurewei. To promote the related technical standards in the IETF, she proposed the IP FPM name, which was recognized after multiple rounds of email discussions.

Recognizing the value of IP FPM innovation, Huawei's Datacom Product Line quickly decided to implement it in router products. The core mechanism of IP FPM is alternate marking in the data plane. At that time, the primary application scenario was on-path measurement on IPv4 service packets carried over MPLS tunnels. However, neither IPv4 nor MPLS had suitable extension space to support alternate marking. We discussed with colleagues in Huawei's solution department whether a simplified technical scheme could be used. One idea was to utilize the three bits in the MPLS flow label's[1] Traffic Class (TC)[2] field for alternate marking. In such a scheme, the IPv4 packets carried over MPLS tunnels would not need to be alternately marked — on-path measurement would be implemented through alternate marking at the MPLS tunnel level, replacing the alternate marking at the IPv4 service flow level. However, our colleagues believed that the application scenarios of this scheme were limited. For example:

- If packets were alternately marked based on MPLS tunnels, it would prevent the measurement of packet loss and delay from an upstream board to the downstream board on an ingress node or egress node. For a router with a distributed architecture, an MPLS label would be pushed for an IPv4 service packet on a downstream board of an ingress node, and an MPLS label would be popped on an upstream board of an egress node to restore the original packet. This made it impossible to implement MPLS-based marking from the upstream board to the downstream board on the ingress or egress node.
- An MPLS tunnel on the network may experience path changes due to node or link faults. Consequently, packets might not reach the original path's egress node or the upstream board of the egress node, affecting the ability to measure packet loss and delay during the fault period.

Due to these reasons, the solution team insisted on implementing alternate marking based on the IPv4 header. But given there are few fields available for this purpose in the IPv4 header, it was proposed — after several rounds of comparison and verification — to use an unused bit from each of the ToS and Flags fields to implement alternate marking when the network plan is clear. However, this data plane extension still could not solve the issue in on-path measurement scenarios where paths change. To identify packets of the corresponding service flow for on-path measurement when paths change, all possible nodes through which an IPv4 service flow might pass would need to be configured with policies based on IP 5-tuple. Due to the uncertainty of path changes on a network, this meant that nearly every node would need to be configured with the corresponding IP 5-tuple policies, and the information would need to be reported to a central point for summarization and data analysis. While today we would instinctively consider using network controllers for processing, they were a relatively new concept back in 2014. At that time, only traditional NMSs existed, and they were not even used on some networks. To support multi-point collection and synchronization of on-path measurement data, IP FPM also required a complex control plane protocol mechanism.

Concerned about the complexity of IP FPM, we spent a long time communicating with colleagues from the solution department, given that we were responsible for implementing it. We recommended avoiding complex scenarios like path changes and instead adopting a simpler, more practical approach to implementing IP FPM. However, such an approach had limited application and could not meet the ambition for widely used innovation. We therefore had to initiate the design and implementation of IP FPM based on the ideal scenarios. To support industry promotion, we also continued to drive standard formulation in the IETF and released the corresponding RFC[3]. However, IP FPM proved somewhat impractical after its launch. The complex solutions and configurations discouraged users from using it, and the O&M simplifications brought by on-path measurement were impossible. Given the failure to gain market acceptance despite the significant research and development efforts invested in IP FPM, the IP FPM innovation has not achieved the expected success. Many of the colleagues involved in IP FPM shifted toward other technical directions, and the once fervent IP FPM innovation fell silent.

9.2 RELENTLESS EXPLORATION

Since 2014, SDN has gained immense popularity, with intelligent O&M emerging as a crucial use case during its development. And AlphaGo's victory in 2016 over the then Go world champion Lee Sedol sparked another wave of interest in AI, with its popularity rivaling that of ChatGPT and the current Large Language Models (LLMs). These events had a profound influence on us. Aiming to leverage AI to simplify IP network O&M, we subsequently submitted a draft of Network Artificial Intelligence (NAI) to the IETF[4] and registered an NAI project on the Open Network Operating System (ONOS) website.

In the days that followed, I was preoccupied with the development of NAI. From a utilitarian perspective, the IETF defines protocol standards for interworking. But there is generally no interworking involved in utilizing AI for data processing within a controller, meaning that the opportunity for standardization is limited. More importantly, from a practical perspective, a network runs according to rules. If all running data can be reported, the controller can clearly locate exceptions and root causes of network faults. There is no need to use the fuzzy and probability-based AI method to guess what these exceptions and root causes are. This conundrum is analogous to the choice between Western medicine and traditional Chinese medicine. If Western medicine can use tools to detect various body indicators for directly and conveniently determining the cause of a disease, there is no need for traditional Chinese medicine, which relies on long-accumulated and insightful experience to determine the cause. These two medical approaches are not mutually exclusive; practitioners of traditional Chinese medicine can accumulate treatment experience faster and more accurately with the support of health check data from medical devices. Similarly, for network O&M, we should not exclusively pursue the AI trend. More importantly, we should instead invent effective tools to clearly visualize network and service status and reduce O&M difficulties. At the time, the concept of telemetry gradually gained attention, sparking our interest. Consequently, we launched related research work on telemetry.

The research work on telemetry was also not smooth. In 2017, we established a telemetry standards project under the guidance of the IP Standards Strategy Committee (IPSSC). Zhou Tianran and I conducted most of the insight, analysis, and research work from the perspective of the standard, with no formal technical research project in place. Zhou knew Dr. Song Haoyu from Futurewei after having worked together on SDN research. Dr. Song was also conducting research on telemetry and on-path measurement, and communicated with us from time to time. All three of us have produced documents such as scenario analysis and technical solution analysis centered on telemetry and on-path measurement techniques. Because this was not a formal research project, few people paid attention to our telemetry research, aside from those involved in the IPSSC project report. Zhou, who sat opposite me in the office, occasionally expressed frustration about the perceived value of our research work. Although I was uncertain about the next steps in telemetry research, I firmly believed that this area was crucial for addressing fundamental IP network O&M issues. I encouraged him to have faith in our technological innovations and devote himself to research.

9.3 RESTARTING INNOVATIONS

At the beginning of 2018 — a critical year for our telemetry research — the telemetry research project was successfully initiated. Miao Fuyou and I selected several people to join our research team, including Dai Longfei, Xu Ling, and Gu Yunan, in addition to Zhou Tianran. Liu Min, with expertise in forwarding adaptation development, was later transferred to our department and joined the team. Although the team was established, its members had limited experience in product design and development. To address this, I frequently organized discussions to guide them in identifying critical design issues and exploring specific solutions during IP on-path measurement research. This interaction paid dividends, as it helped enhance the team's scenario analysis and system design capabilities. During the initial discussion on the telemetry technique, two issues in particular were highlighted.

One issue relates to the telemetry technical framework. Telemetry involves extremely complex techniques, which can easily cause confusion. After systematically analyzing telemetry-related techniques, we defined the NTF and submitted a draft[5] to the IETF.

The second issue relates to the naming of the IP on-path telemetry solution. At that time, IOAM had only the tracing options, meaning it supported only the passport mode. After analyzing the approach and finding numerous limitations in terms of scalability, we, therefore, introduced the postcard mode for on-path measurement. We also proposed the UDP-based telemetry technique, along with methods like intelligent flow selection and data reporting suppression, to prevent the technical solution from becoming unfeasible due to excessive information reporting. Discussions soon became very complex because of these solutions intertwining with IOAM techniques. In some discussions, the two parties who debated fiercely eventually realized that the solutions they were debating were entirely different. We all strongly felt the need to clearly define terms for our technical solution. I therefore invited everyone to share their ideas and make decisions during our meetings. At that time, I was trying to diet and lose weight. We also often missed lunch due to numerous heated discussions, sparked by the team's enthusiasm for telemetry research. Weight loss was therefore another hot topic for the team. Gu Yunan and Liu Min were both well-versed in dieting, and Wang Zhongzhen (a colleague from the forwarding department who guided our microcode design) also had successful dieting experience. The result of this was that our team's communication group on social media became called the "Dieters." When we voted together on all the possible names of our on-path telemetry solution, the name IFIT was an obvious choice. We then submitted the IFIT framework draft[6] to the IETF. Since then, the term IFIT, conveying a good meaning, has gradually become more widespread.

During the study of IFIT, SR-MPLS remained dominant in SR technology, but we clearly saw the severe limitations in MPLS extensibility, especially the inability to support the IOAM tracing options through the extensions of the MPLS label stack. Following extensive discussions and the exploration of various extension methods, we concluded that the most practical approach was to add an IPv6-like extension header at the end of the MPLS label stack to encapsulate information. We subsequently applied for the MPLS extension header patent and submitted the related draft[7] to the IETF. And, recognizing the limited development potential for MPLS, we further solidified our decision to utilize the IPv6 extension header mechanism for extending network functions. In summarizing the innovation of IPv6 Enhanced, Kevin Hu — then president of Huawei's Data Communication Product Line — said that people who can take the lead in IP innovation are often those who first identify the network problems and requirements. The IFIT research is undoubtedly a good interpretation of this. In 2018, we conducted research on both SRv6 and IFIT. SRv6 and IPv6-based IOAM tracing options posed high requirements on network processor chips, and these requirements were considered carefully during the planning of next-generation chips. Thanks to sound technical preparations, the subsequent restrictions on Huawei's chip supply did not have a significant impact on the IPv6 Enhanced innovation. In retrospect, I was struck with fear, yet I felt incredibly lucky.

IFIT research has not been without its challenges. Initially, we devoted significant effort to designing and prototyping postcard-based on-path measurement. Even so, there was a lack of quantifiable comparisons between the postcard and passport modes in terms of the on-path measurement data volume. Something always felt off to me. In October 2018, Dr. Dai Longfei was assigned to perform a special analysis of the reported data. Soon after, he provided a thorough comparative analysis, and the results were shocking.

- The data volume reported by on-path measurement was staggering. Based on the scenario assumptions provided by Dr. Dai, on-path measurement generated up to 1.5 million records per second and about 5.5 TB of data per day. The consumption of resources by such a vast amount of data and the pressure it exerts on the overall system performance would be overwhelming. It is difficult to envision carriers being willing to incur such a substantial cost to support on-path telemetry.
- Another unexpected but understandable discovery was that the total amount of data reported in postcard mode is much higher than that reported in passport mode. Although the valid information reported is the same, in postcard mode, all nodes along the path need to report data, which includes a packet header on each node. In passport mode, only the egress node reports data, which therefore includes only one packet header. The additional packet headers account for the higher consumption in postcard mode compared to passport mode. Assume that *n* additional such headers are generated when on-path measurement is performed on one service packet. Now consider how many service packets a 400G interface could forward per second. The cumulative volume of excess reported data is astonishing.

These two results had a significant impact on me. They revealed that the cost of on-path telemetry is too high, limiting its scope of use, and that the postcard mode offers no special advantage over the passport mode due to the obvious difference in the amount of reported data. After these results were released, I felt disheartened, lacked motivation, and was unsure of what steps to take next.

Fortunately, things soon took a turn for the better. After learning about our research on on-path telemetry, Huang Jinming from the Packet Transport Network (PTN) product line visited us to discuss solutions for enhancing OAM capabilities. We explored the IP FPM and alternate marking methods for which IPv6 extension headers could be used to carry related instructions, and postcard-based on-path measurement could be adopted. Because the packet loss rate and delay were measured based on the measurement period, the volume of data reported to the controller would be significantly smaller compared to IOAM. In addition, IPv6 extensions carrying the flow ID of service packets eliminated the complex configurations where network nodes identify flows using IP 5-tuple policies. Although the IFIT solution based on the alternate marking method did not entirely align with our initial intent of researching per-flow and per-packet measurement, its deployment cost was significantly lower, and it could be really deployed for commercial use. We therefore proposed defining IPv6 on-path telemetry based on the alternate marking mechanism as IFIT 1.0. When conditions permit in the future, IFIT supporting IOAM tracing options or postcard-based optimized IOAM could be implemented and defined as IFIT 2.0.

9.4 NEW BEGINNINGS

After determining the IFIT design, we began prototype development. Liu Min and the microcode expert Lu Bo (arranged by Liu Shaowei, then director of Huawei Data Communication R&D Management Department, to join our team) contributed greatly to the development of the IFIT data plane, while Xu Ling focused on developing IFIT management and control plane functions. Like ants moving a mountain, we finished developing the IFIT prototype system step by step. After implementing the IFIT function, it was then necessary to intuitively display the on-path telemetry results via Graphical User Interfaces (GUIs). Developing IFIT GUIs based on NCE required a significant investment, which was unrealistic at that time. Zhou Tianran, who had previously conducted research and development based on SDN open source, recommended using open source tools for IFIT GUI presentation. I believed this was a feasible option, so I assigned Dr. Dai Feilong to handle the GUI development.

Dr. Dai, a mathematician by training, was inquisitive and had strong hands-on abilities. He quickly learned and mastered a variety of IT open source software and tools for developing functions related to reporting and displaying IFIT information. However, the GUI product was consistently unsatisfactory. This made me a little anxious, because I was heavily involved in promoting SRv6 and IPv6 Enhanced innovation and could not always provide guidance. Around the beginning of 2019, Feng Su and Hao Jianwu from Huawei's network sales team in Japan contacted us to discuss showcasing advanced IP innovative technologies at Interop in Japan, aiming to establish the brand of our datacom products. Without hesitation, I recommended IFIT and introduced Dr. Dai to them. Feng and Hao, both experienced in project development and management, held weekly meetings with Dr. Dai to discuss IFIT design and GUI presentation. Over two months, they consistently raised requirements to Dr. Dai, tracked his progress to ensure the completion of development, and urged him to continuously optimize the implementation. This had the profound effect of transforming a mathematician into a qualified network development engineer.

A week before Interop, Dr. Dai traveled to Japan to prepare the IFIT exhibition environment. Despite thorough preparations at the headquarters, unexpected issues frequently arose on site. A dozen open source software packages needed to be installed, creating a variety of issues. Huawei frontline personnel were anxious and pushed Dr. Dai to work through two consecutive nights. He finally completed the task just before the demonstration equipment was packaged for transport to the event. Dr. Dai was responsible for the IFIT demonstration at Interop and approached it with confidence and enthusiasm. Ultimately, IFIT won the Special Award at the Interop exhibition. Through the struggles of IFIT innovation, Dr. Dai finally gained success and confidence — I was sincerely happy for him.

While the IFIT prototype was being successfully demonstrated, we finally achieved a breakthrough in standardization. At first, we had aimed to collaborate with the other vendor on IOAM standards, holding multiple rounds of discussions with its expert within the IETF. Unfortunately, we made little progress. Because of difficulties in the cooperation, we initially submitted multiple drafts in the IP Performance Metrics (IPPM) WG based on IFIT research results[8, 9, 10, 11, 12] and collaborated with other industry experts. The expert from the other vendor was responsible for IOAM standards in the IETF, but the complexity of the technologies involved slowed down the standardization process. Moreover, many new drafts for on-path measurement were introduced within the IPPM WG and were closely associated with IOAM. However, it remains unclear how these drafts should be developed. To address this issue, the WG chairs and transport Area Director (AD) designated the other vendor's expert to convene side meetings during the 104th IETF meeting to discuss promoting the IOAM standards.

In March 2019, the 104th IETF meeting was held in Prague, Czech Republic. Zhou Tianran, Song Haoyu, Gong Jun, and I attended the IOAM side meetings, which were hosted by the other vendor's expert, and once witnessed a very heated discussion. Faced with pressure from multiple parties, the expert had to reach a compromise. After three rounds of side meeting discussions, it was finally decided which technical solutions would be standardized within IOAM — one of them was our proposed postcard-based on-path measurement mechanism. We also made concessions and agreed to include the postcard mode as part of the IOAM technology, defining it as the direct export option. The 104th IETF meeting was one of the most important IETF meetings I attended. In the meeting, I formally assumed my duties as the freshly elected member of the IETF Internet Architecture Board, the most crucial draft (SRH) we promoted for SRv6 passed the last call in the IPv6 Maintenance (6MAN) WG, and significant advancements were also achieved in the standardization of IFIT innovation.

Later, Zhou Tianran, Giuseppe (standards expert from the Huawei European Research Center), and other industry experts initiated the standardization of the IPv6 alternate marking mechanism in the 6MAN WG. During this process, experts recommended updating the original IP FPM-related RFCs. The outcome of this was that three RFCs[13, 14, 15] were published. Through these efforts, we completed key standardization work on the IOAM and alternate marking mechanisms, laying a solid foundation for further industry development.

9.5 CONCLUSION

From the failure of IP FPM to the success of IFIT, it is difficult even for many Huawei employees to understand why similar technologies have such different outcomes.

In the process of innovation, critical problems can be extremely valuable. From IP FPM to IFIT, the value and significance of on-path measurement always remained. Despite the setbacks we experienced in IP FPM innovation, we did not throw the baby out with the bathwater when faced with doubts about innovation. Instead, we patiently waited for the right moment.

During the IFIT innovation, we also faced various doubts. However, I always believed that solving the basic problems of IP network O&M was essential. When we realized the possibility of combining IPv6 extensions and on-path measurement mechanisms to resolve these issues, we dedicated ourselves to making it happen. Under the guidance of the IPSSC, the research, standardization, development, and solution teams collaborated closely to make the IFIT alternate marking mechanism a commercially viable IPv6 on-path telemetry solution.

I have always believed that technology should be developed with warmth, rather than being seen as a cold entity. Due to the absence of efficient technical tools, when a network issue arises, engineers have to invest considerable time and effort into replicating the issue, obtaining packet headers of affected services, and extracting useful information from vast logs and debug data. Consequently, they often work overtime and sometimes stay up all night. Additionally, IP is often regarded as a low-level technology that relies solely on best-effort forwarding. Even after the development of IPv6 Enhanced innovations, many people still believe that IP should remain simple. The significance of IPv6 Enhanced innovations, represented by IPv6 on-path telemetry, extends beyond generating revenue and profits for products. More profoundly, these innovations liberate network engineers from laborious work and enable IP networks to provide better services. History will show that the simplicity of IP is merely a characteristic of a particular era, not an inherent quality of IP. Network operators should not sacrifice network O&M during service development, nor should they wait until a disaster occurs before making improvements.

Our journey with IPv6 on-path telemetry is far from over — we still need to achieve fine-grained measurement and visual presentation, and determine how to handle vast amounts of data. All these goals must be addressed through ongoing innovation. The development of IP on-path telemetry mechanisms can be likened to the advancement of medical equipment. Starting with basic chest X-rays, progressing to Computed Tomography (CT) scanners, and eventually reaching Magnetic Resonance Imaging (MRI), technologies continue to evolve, accompanied by increasing demands for resources and escalating costs. IPv6 on-path telemetry innovation must be developed continuously at a reasonable cost to meet network O&M needs at various levels and enable healthier network growth.

REFERENCES

1. Bryant S, Filsfils C, Drafz U et al. Flow-aware transport of pseudowires over an MPLS packet switched network [EB/OL]. (2011-11) [2024-09-30]. RFC 6391. ↩
2. Andersson L, Asati R. Multiprotocol label switching (MPLS) label stack entry: “EXP” field renamed to “traffic class” field [EB/OL]. (2009-02) [2024-09-30]. RFC 5462. ↩
3. Fioccola G, Capello A, COCIGLIO M et al. Alternate-marking method for passive and hybrid performance monitoring [EB/OL]. (2018-01-29) [2024-09-30]. RFC 8321. ↩
4. Li Z, Zhang J. An architecture of network artificial intelligence (NAI) [EB/OL]. (2017-05-04) [2024-09-30]. Draft-li-rtgwg-network-ai-arch-00. ↩
5. Song H, Li ZQ, Martinez-Julia P et al. Network telemetry framework [EB/OL]. (2019-09-07) [2024-09-30]. Draft-song-opsawg-ntf-03. ↩
6. Song H, Qin F, Chen H. Framework for in-situ flow information telemetry [EB/OL]. (2024-04-25) [2024-09-30]. Draft-song-opsawg-ifit-framework-21. ↩
7. Song H, Zhou T, Andersson L et al. MPLS network actions using post-stack extension headers [EB/OL]. (2024-04-13) [2024-09-30]. Draft-song-mpls-extension-header-13. ↩
8. Song H, Li Z, Zhou T et al. In-situ OAM processing in tunnels [EB/OL]. (2018-12-29) [2024-09-30]. draft-song-ippm-ioam-tunnel-mode-00. ↩
9. Song H, Zhou T. In-situ OAM data type extension [EB/OL]. (2018-10-18) [2024-09-30]. Draft-song-ippm-ioam-data-extension-01. ↩
10. Song H, Zhou T. Control in-situ OAM overhead with segment IOAM [EB/OL]. (2018-10-19) [2024-09-30]. Draft-song-ippm-segment-ioam-01. ↩
11. Song H, ZHOU T. In-situ OAM data validation option [EB/OL]. (2018-10-18) [2024-09-30]. Draft-song-ippm-ioam-data-validation-option-02. ↩
12. Song H, MIRSKY G, Zhou T et al. On-path telemetry using packet marking to trigger dedicated OAM packets [EB/OL]. (2023-12-04) [2024-09-30]. Draft-song-ippm-postcard-based-telemetry-16. ↩
13. Fioccola G, Cociglio M, Mirsky G et al. Alternate-Marking method [EB/OL]. (2022-12) [2024-09-30]. RFC 9341. ↩

14. Fioccola G, Cociglio M, Sapio A et al. Clustered alternate-marking method [EB/OL]. (2022-12) [2024-09-30]. RFC 9342. ⏎
15. Fioccola G, Zhou T, Cociglio M et al. IPv6 application of the Alternate-Marking method [EB/OL]. (2022-12) [2024-09-30]. RFC 9343. ⏎

Appendix A

IPv6 Active Measurement Techniques

IPv6 on-path telemetry measures the real service quality of the network that services traverse by analyzing real service packets. However, in some scenarios, such as measuring the service quality of a backup path or evaluating the service quality of a newly planned path, it is desirable to measure such service quality when there is no service traffic. In such scenarios, active measurement techniques can be employed.

Active measurement techniques send dedicated test packets to measure network service quality, such as the packet loss rate, delay, and jitter. There are multiple active measurement techniques that can be used for IPv6, among which Two-Way Active Measurement Protocol (TWAMP) and Simple Two-way Active Measurement Protocol (STAMP) will be the focus of this appendix.

A.1 TWAMP

TWAMP is an active measurement technique that generates test flows based on five-tuple information (source IP address, destination IP address, source port number, destination port number, and protocol type). The test packets in each test flow are sent through the measured path, and the response packets are sent back through the same path. TWAMP then analyzes the received response packets to measure the performance and status of IP links.

A.1.1 TWAMP Architecture

TWAMP consists of four logical entities: Session-Sender, Session-Reflector, Control-Client, and Server. The Control-Client and Server are part of the control plane and handle the negotiation (initialization), startup, and stopping of measurement tasks. The Session-Sender and Session-Reflector are part of the data plane and perform measurement actions. Specifically, the Session-Sender sends test packets, and the Session-Reflector responds to these packets[1].

In real-world applications, TWAMP can be implemented using either the Full or Light architecture, depending on where the four logical entities are deployed.

- Full architecture: The Session-Sender and Control-Client are integrated as one entity called the Controller, and the Session-Reflector and Server are integrated as another entity called the Responder. The Controller communicates with the Responder through TCP-based TWAMP control packets to establish a test session. After the session is established, the Controller sends UDP-based TWAMP test packets to the Responder, and the Session-Reflector on the Responder responds to these packets. Figure A.1 shows TWAMP in the Full architecture.

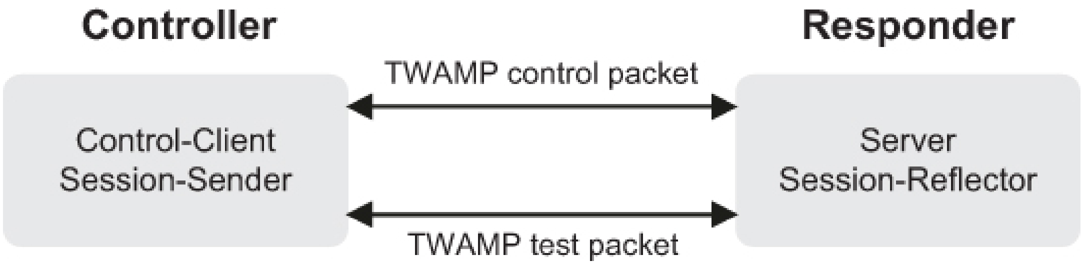


FIGURE A.1 TWAMP in the Full architecture. ⏎

- Light architecture: The Session-Sender, Control-Client, and Server are integrated as one entity called the Controller, and the Session-Reflector itself functions as the Responder. Unlike the Full architecture, the Light architecture does not require control plane session negotiation. Instead, session information is delivered to the Controller to control the sending of test request packets. The Responder only needs to receive test request packets, copy the sequence number in them to the same field in response packets, and then send the response packets back. The Light architecture, shown in Figure A.2, makes TWAMP easier to implement and is more widely deployed. The following sub-section describes this architecture in detail.

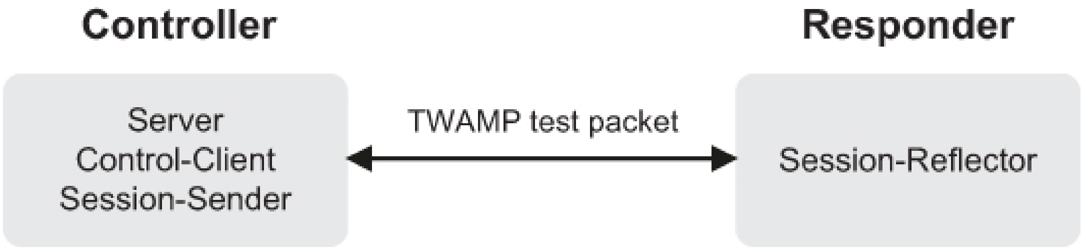


FIGURE A.2 TWAMP in the Light architecture. ⏎

A.1.2 TWAMP Light

The packet exchange process used in TWAMP in the Light architecture — also called TWAMP Light — is shown in Figure A.3. The operations involved in this process are as follows:

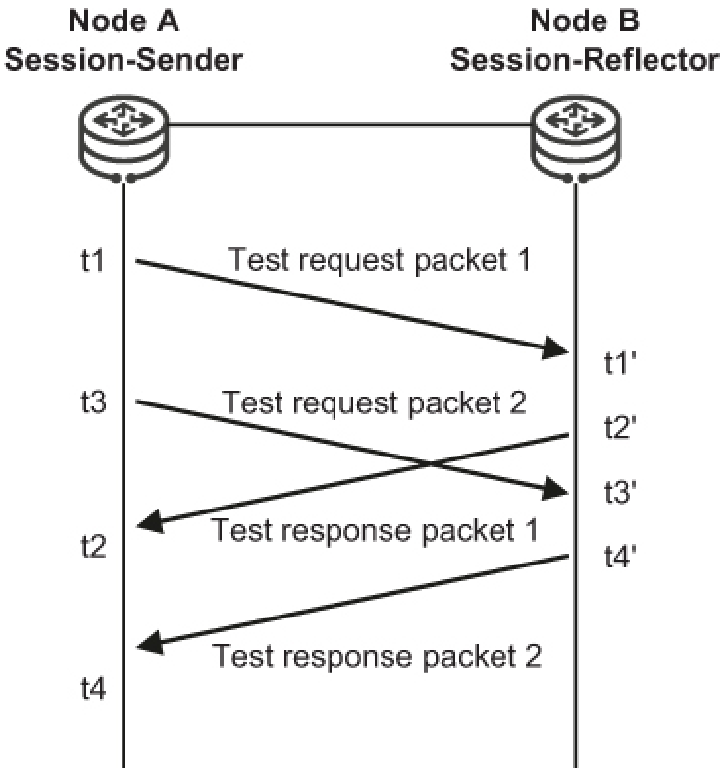


FIGURE A.3 TWAMP Light packet exchange process. ⏎

1. The Session-Sender (node A) sends a test request packet carrying the sending timestamp *t*1.

2. After receiving the packet, the Session-Reflector (node B) records the receiving timestamp $t1'$, adds the sending timestamp $t2'$ to the response packet, and sends the packet to the Session-Sender.
3. After receiving the packet, the Session-Sender records the receiving timestamp $t2$.

Based on the four timestamps, the delays in a single measurement period can be calculated as follows:

- Forward delay = $t1' - t1$
- Reverse delay = $t2 - t2'$
- Round-trip delay = $(t1' - t1) + (t2 - t2')$

A TWAMP Light test request packet sent by the Session-Sender carries information such as the packet sequence number and timestamp. Figure A.4 shows the format of such a packet.

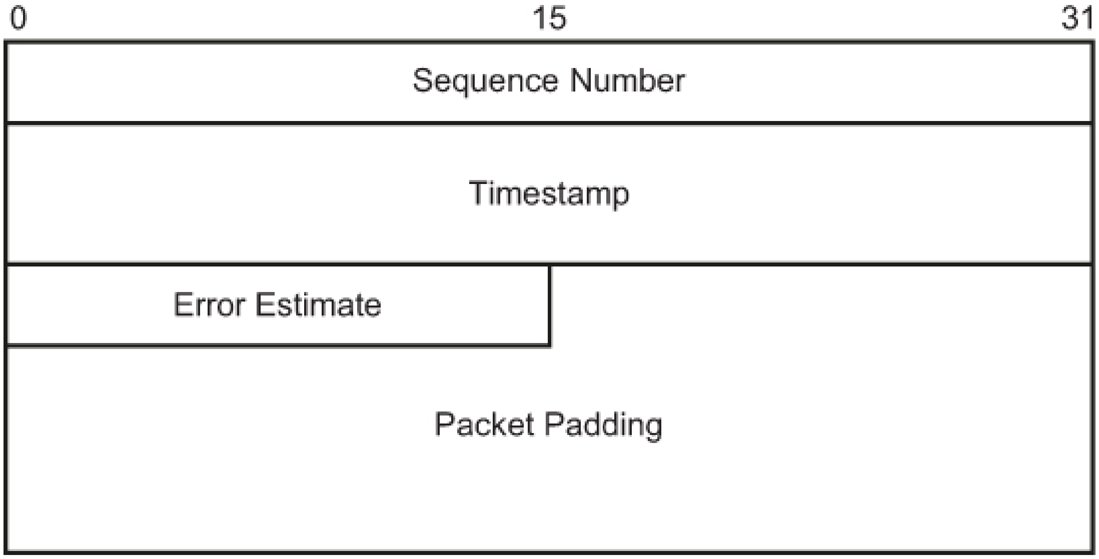
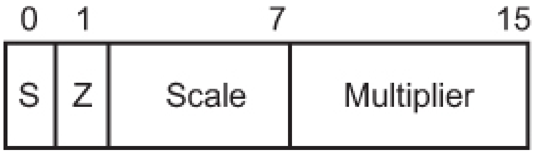


FIGURE A.4 Format of a TWAMP Light test request packet. ⏎

Table A.1 describes the fields in a TWAMP Light test request packet (Figure A.4).

TABLE A.1 Fields in a TWAMP Light Test Request Packet ⏎

Field	Length	Description

Field	**Length**	**Description**
Sequence Number	4 bytes	Sequence number of a test request packet.
Timestamp	8 bytes	Time when a test request packet is sent.
Error Estimate	2 bytes	Estimate of the clock error. Figure A.5 shows the format of this field.  FIGURE A.5 Error estimate format. • *S*: If this bit is set to 1, an external clock source is used. • *Z*: reserved field whose value is fixed at 0. • Scale and Multiplier: used to calculate the clock error. They are unsigned integers and occupy 6 bits and 8 bits, respectively. Note that Multiplier must not be set to 0. Error estimate = Multiplier × 2^{-32} × 2^{Scale} (in seconds).
Packet Padding	Variable	Padding for alignment. This field ensures that the length of the test request packet is the same as that of the test response packet, simplifying the measurement processing logic.

A TWAMP Light test response packet sent by the Session-Reflector carries the sending timestamp copied from the received test request packet, the receiving timestamp of the request packet, the sending timestamp of the response packet, and other information. Figure A.6 shows the format of such a packet.

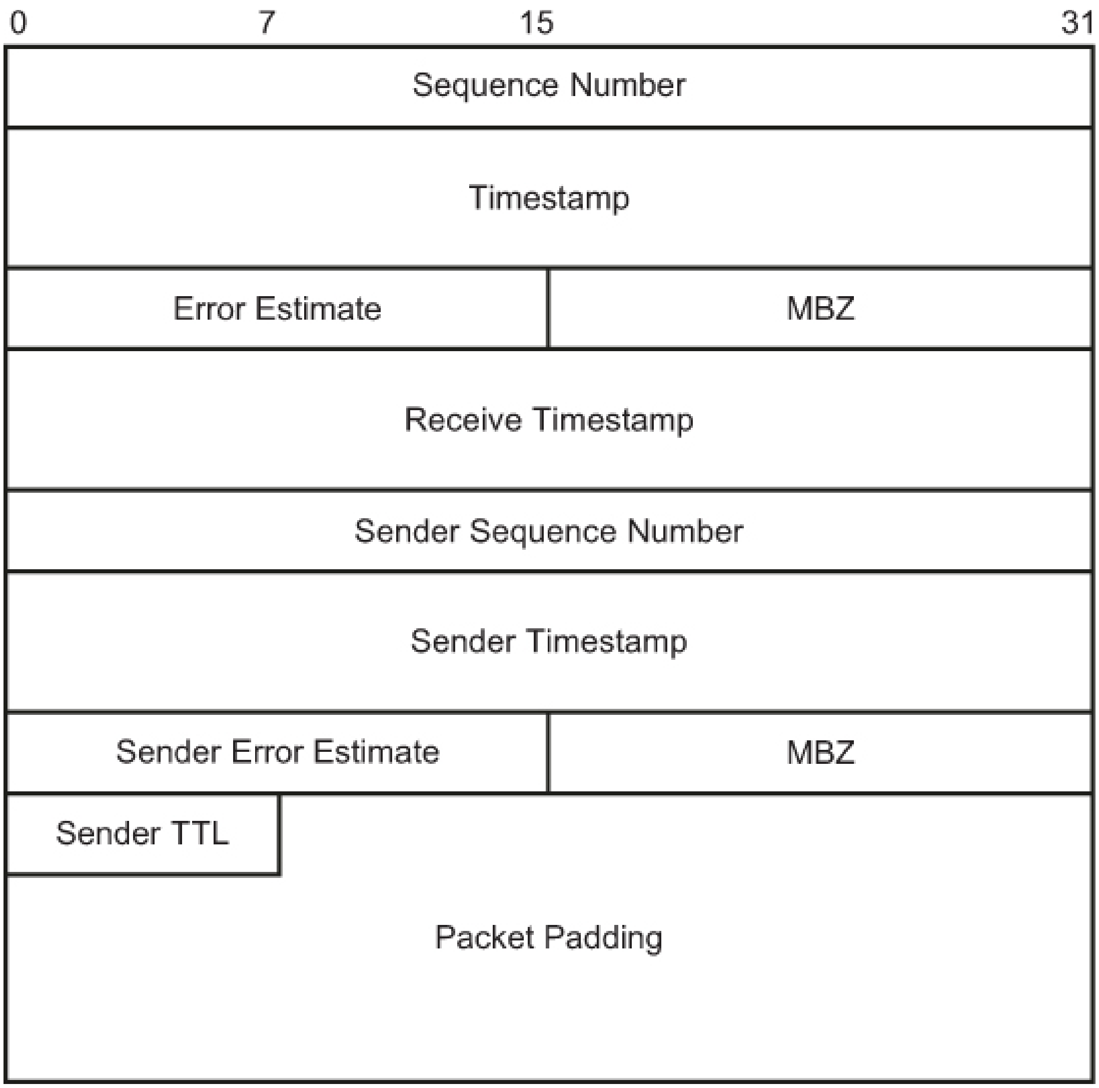


FIGURE A.6 Format of a TWAMP Light test response packet. ⏎

Table A.2 describes the fields in a TWAMP Light test response packet.

TABLE A.2 Fields in a TWAMP Light Test Response Packet ⏎

Field	**Length**	**Description**
Sequence Number	4 bytes	Sequence number of a test response packet.
Timestamp	8 bytes	Time when a test response packet is sent.
Error Estimate	2 bytes	Estimate of the clock error. The interpretation of this field is the same as that in a request packet.

Field	Length	Description
MBZ	2 bytes	Must-be-zero. It is a reserved field whose value is fixed at 0.
Receive Timestamp	8 bytes	Time when a test request packet is received.
Sender Sequence Number	4 bytes	Sequence number copied from the corresponding field in the request packet sent by the Session-Sender.
Sender Timestamp	8 bytes	Timestamp copied from the corresponding field in the request packet sent by the Session-Sender.
Sender Error Estimate	2 bytes	Error Estimate copied from the corresponding field in the request packet sent by the Session-Sender.
MBZ	2 bytes	Reserved field whose value is fixed at 0.
Sender TTL	1 byte	Time To Live (TTL) copied from the IP header of the request packet sent by the Session-Sender.
Packet Padding	Variable	Padding for alignment. This field can reuse the Packet Padding field in the request packet to ensure that the payload length in the response packet is the same as that in the request packet. The Session-Reflector can reduce the padding as required.

Based on the three timestamps carried in a response packet and the timestamp when it receives such a packet, the Session-Sender calculates the forward delay, reverse delay, and round-trip delay. In addition, the two-way packet loss can be calculated by comparing the number of sent request packets to the number of received response packets within a period of time.

A.2 STAMP

TWAMP in the Full architecture requires multiple processes, such as control plane negotiation and data plane measurement, making it complex to deploy. And although TWAMP Light is easier to implement, it lacks sufficient cross-vendor interoperability and extensibility. STAMP was designed to address these issues. This standardized active measurement technique simplifies the architecture, specifies the test packet format and operation process, and supports cross-vendor interoperability[2]. Figure A.7 shows the STAMP architecture.

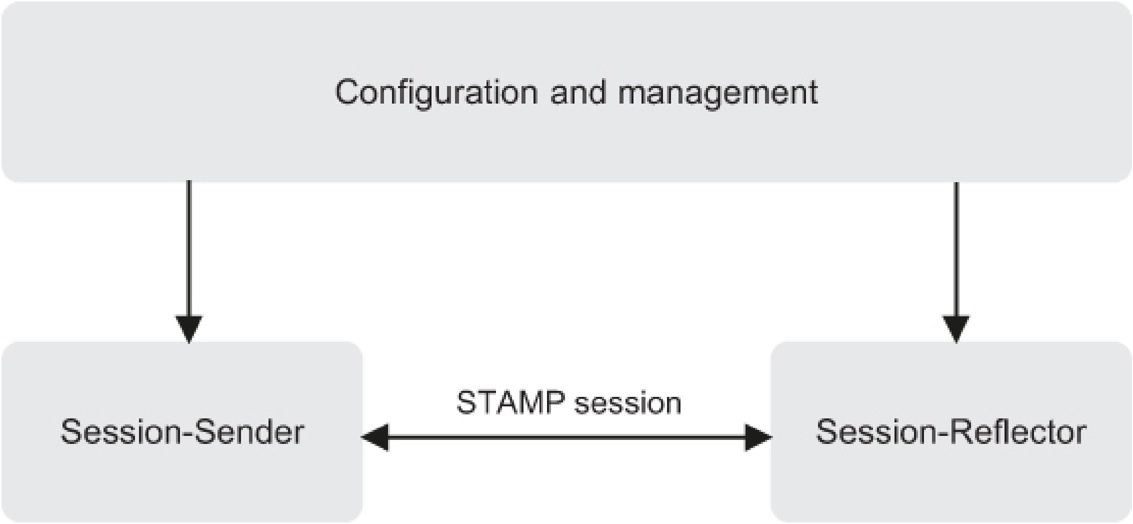


FIGURE A.7 STAMP architecture. ⏎

In the STAMP architecture, the configuration and management module is responsible for configuring STAMP sessions. Each such session is a bidirectional detection packet flow between the Session-Sender and Session-Reflector within a certain period of time.

The process involved in two-way packet loss measurement starts with the Session-Sender sending a test request packet to the Session-Reflector through UDP. Then, after receiving the packet, the Session-Reflector performs operations (e.g., directly copying the sequence number in the request packet to the response packet) according to local configurations.

STAMP can work in either of the following modes:

- Stateless mode: The Session-Reflector directly reflects the received test request packets without maintaining the state of the test session. In this mode, only two-way packet loss can be calculated.

- Stateful mode: The Session-Reflector maintains the session state and adds a sequence number to each test response packet. This sequence number increases independently based on each session. In this mode, the Session-Sender calculates the one-way packet loss by comparing the sequence numbers in the request and response packets.

Figure A.8 shows the format of a STAMP test request packet sent from the Session-Sender to the Session-Reflector.

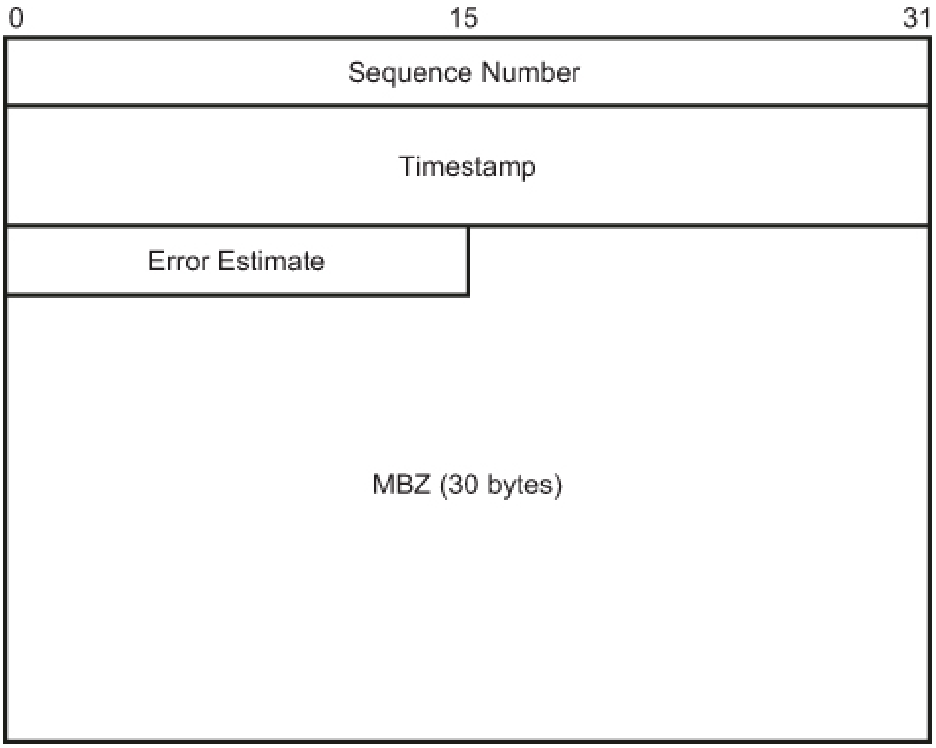


FIGURE A.8 Format of a STAMP test request packet. ⏎

Table A.3 describes the fields in a STAMP test request packet.

TABLE A.3 Fields in a STAMP Test Request Packet ⏎

Field	**Length**	**Description**
Sequence Number	4 bytes	Sequence number of a test request packet. For each session, the value starts from 0 and increments by 1 with each transmitted packet.
Timestamp	8 bytes	Time when a test request packet is sent.

Field	Length	Description
Error Estimate	2 bytes	Estimate of the clock error. The interpretation of this field is the same as that in a TWAMP Light test request packet.
MBZ	30 bytes	The value of this field must be 0 on transmission and be ignored on receipt. It provides space for adding information such as the timestamp and sequence number to the response packet sent by the Session-Reflector. The Session-Sender uses this field to maintain the symmetry between the size of a request packet and that of a response packet, simplifying data plane processing.

Figure A.9 shows the format of a STAMP test response packet reflected by the Session-Reflector to the Session-Sender.

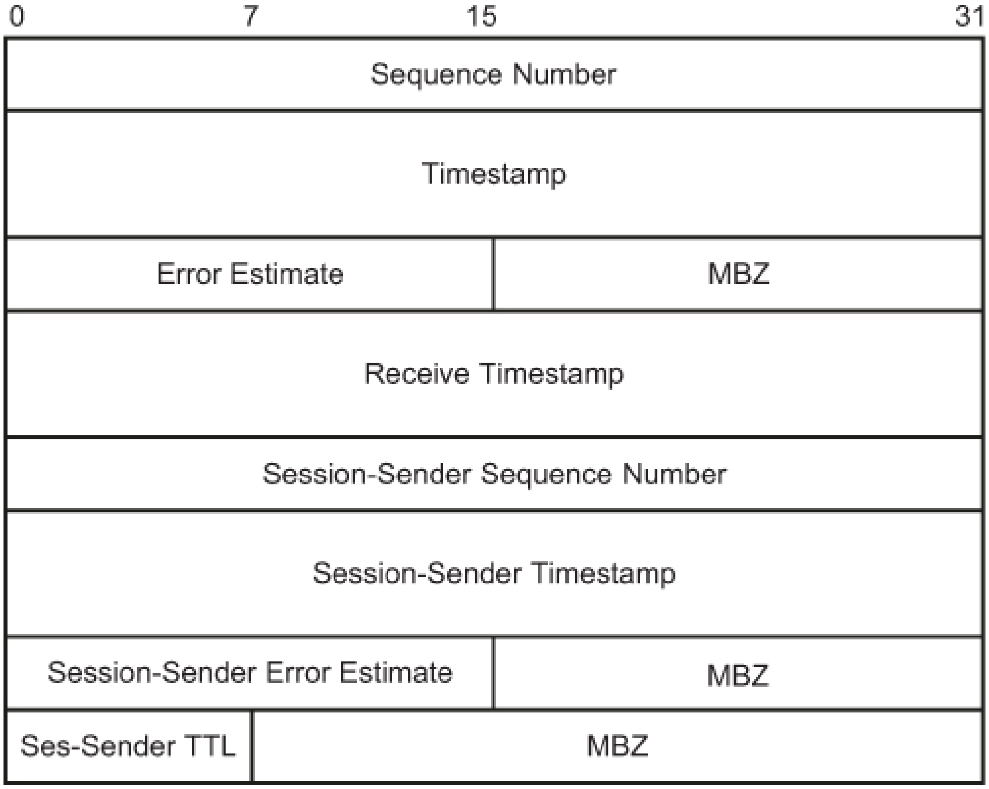


FIGURE A.9 Format of a STAMP test response packet. ⏎

Table A.4 describes the fields in a STAMP test response packet.

TABLE A.4 Fields in a STAMP Test Response Packet ⏎

Field	Length	Description
Sequence Number	4 bytes	Sequence number of a test response packet. In stateless mode, the value is copied from the Sequence Number field of the test request packet. In stateful mode, the value starts from 0 and increments by 1 with each transmitted packet.
Timestamp	8 bytes	Time when a test response packet is sent.
Error Estimate	2 bytes	Estimate of the clock error. The interpretation of this field is the same as that in a TWAMP Light test request packet.
MBZ	2 bytes	Reserved field whose value is fixed at 0.
Receive Timestamp	8 bytes	Time when a test request packet is received.
Session-Sender Sequence Number	4 bytes	Sequence number copied from the corresponding field in the request packet sent by the Session-Sender.
Session-Sender Timestamp	8 bytes	Timestamp copied from the corresponding field in the request packet sent by the Session-Sender.
Session-Sender Error Estimate	2 bytes	Error Estimate copied from the corresponding field in the request packet sent by the Session-Sender.
MBZ	2 bytes	Reserved field whose value is fixed at 0.
Ses-Sender TTL	1 byte	TTL copied from the IP header of the request packet sent by the Session-Sender.
MBZ	3 bytes	Reserved field whose value is fixed at 0.

STAMP and TWAMP Light share similar methods for calculating basic performance indicators in scenarios such as two-way packet loss and delay measurement. To accommodate future evolution requirements in new scenarios, STAMP is being developed to offer new measurement methods, including absolute packet loss measurement and high-precision one-way delay measurement. For details, see related drafts such as *draft-gandhi-ippm-simple-direct-loss*[3] and *draft-ietf-spring-stamp-srpm*[4].

REFERENCES

1. Hedayat K, Krzanowski R, Morton A et al. A two-way active measurement protocol (TWAMP) [EB/OL]. (2008-10) [2024-09-30]. RFC 5357. ⏎
2. Mirsky G, Jun G, Nydell H et al. Simple two-way active measurement protocol [EB/OL]. (2020-03) [2024-09-30]. RFC 8762. ⏎
3. Gandhi R, Filsfils C, Voyer D et al. Simple two-way direct loss measurement procedure [EB/OL]. (2024-08-07) [2024-09-30]. Draft-gandhi- ippm-simple-direct-loss-08. ⏎
4. Gandhi R, Filsfils C, Voyer D et al. Performance measurement using simple two-way active measurement protocol (STAMP) for segment routing networks [EB/OL]. (2024-08-02) [2024-09-30]. Draft-ietf-spring-stamp-srpm-15. ⏎

Appendix B

Time Synchronization Technologies

IPv6 on-path telemetry requires time synchronization between devices on the measurement path. The following sections describe two major time synchronization technologies: NTP and PTP.

B.1 NTP

NTP[1] is widely used by network devices and computers to implement clock synchronization. Its purpose is to ensure that all participating network devices and computers are synchronized to basically the same time, based on which they provide various applications.

B.1.1 NTP Network Structure

An NTP network consists of the reference clock, time servers, clients, and interconnected transmission paths. The primary time server synchronizes its time with the reference clock, which is usually a radio clock or Global Positioning System (GPS). Layer-2 time servers synchronize time from the primary time server or other layer-2 time servers on the network, and then transmit the time information to one or more downstream time servers or clients. This process ensures that the system clocks of all devices on the network are basically the same.

The time servers are connected in a hierarchical structure, in which each layer is called a stratum (numbered from 0). Figure B.1 shows the typical NTP networking. The reference clock at the top layer runs at stratum 0, and a server that synchronizes time through a stratum n server runs at stratum n + 1. The number represents the distance from the reference clock and is used to prevent cyclical dependencies in the hierarchy. In the figure, the primary time servers run at stratum 1, and the layer-2 time servers and layer-3 time servers run at stratum 2 and stratum 3, respectively, according to their locations.

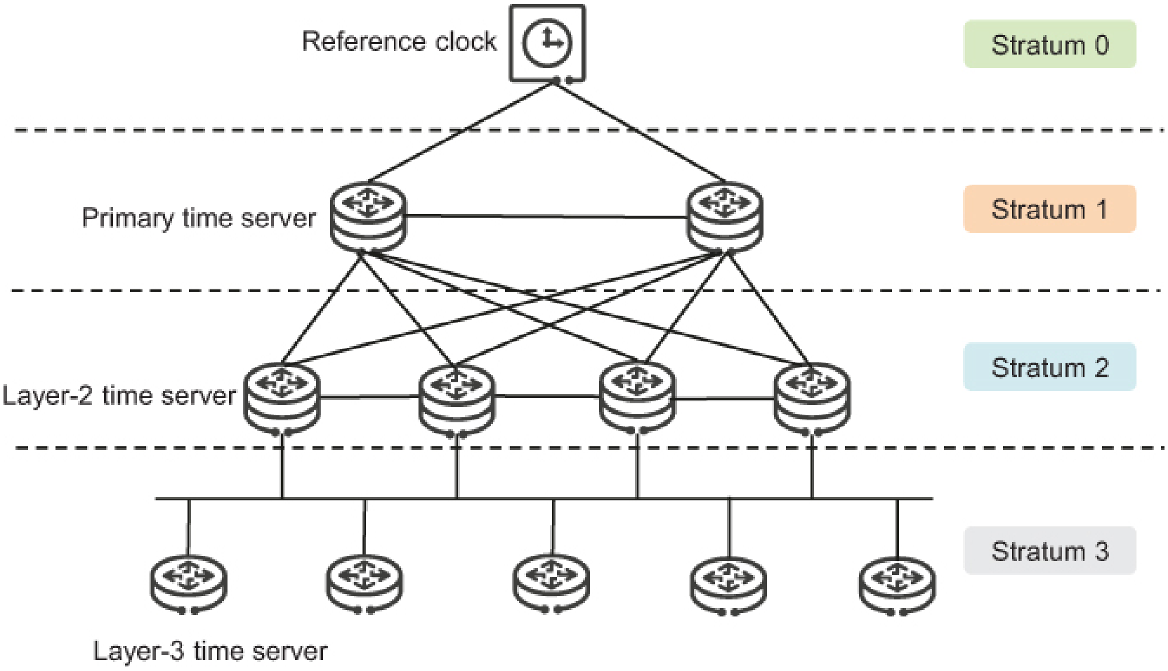


FIGURE B.1 Typical NTP networking. ⏎

B.1.2 Operating Modes of NTP

Table B.1 lists the operating modes of NTP.

TABLE B.1 Operating Modes of NTP ⏎

Operating Mode	Description	Usage Scenario
Client/Server mode	The client synchronizes its time with that of the server.	This mode is used when a device synchronizes its time with that of the upper-layer time server at a lower stratum.

Operating Mode	Description	Usage Scenario
Peer mode	Peers synchronize time with each other.	This mode is usually used for synchronization between devices at the same stratum. This helps implement backup between these devices. Even if the communication between a device and all upper-layer time servers fails, the device can still synchronize time from the time servers at the same stratum.
Broadcast mode	The server periodically sends time synchronization packets to a broadcast address. Broadcast clients listen for the broadcast packets from the server and synchronize their time with that of the server according to the received broadcast packets.	In broadcast mode, a large number of devices can synchronize their time with the time of one time server on the same network, simplifying network configuration.
Multicast mode	The server periodically sends time synchronization packets to a multicast address. Clients listen for the multicast packets from the server and synchronize their time with that of the server according to the received multicast packets.	This mode applies to scenarios where numerous clients are distributed on the network. In multicast mode, the server multicasts packets to the clients of a multicast group. This reduces the number of transmitted NTP packets, thereby alleviating pressure on the network.

B.1.2.1 Client/Server Mode

Figure B.2 shows the exchange process in client/server mode. The device operating in client mode synchronizes its time with the time of the server. The time server provides synchronization information for the client but does not modify its own clock. In client/server mode, the server is also called a unicast server, differentiating it from a broadcast server in broadcast mode.

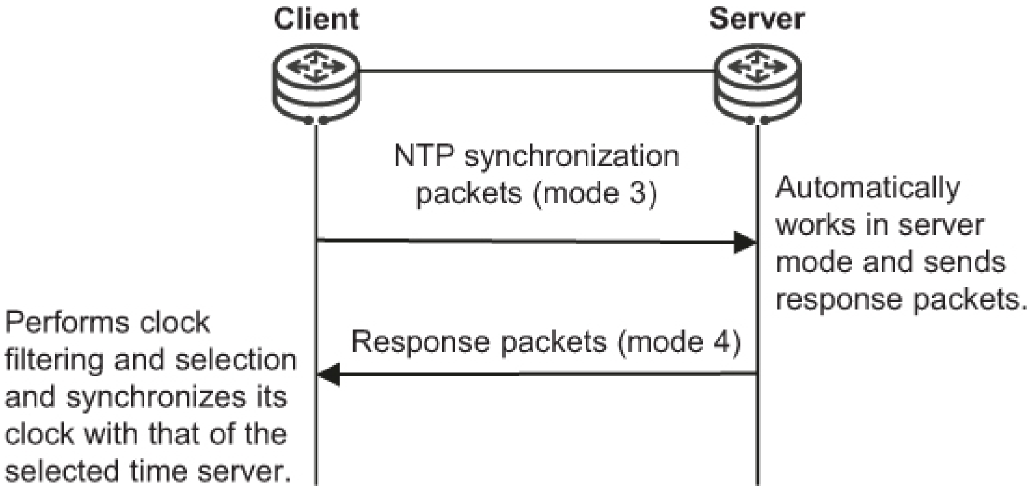

Client
Server
NTP synchronization packets (mode 3)
Automatically works in server mode and sends response packets.
Performs clock filtering and selection and synchronizes its clock with that of the selected time server.
Response packets (mode 4)

FIGURE B.2 Exchange process in client/server mode. ⏎

The exchange process in client/server mode is as follows:

1. The client periodically sends packets to the server. The Mode field in the packets is set to 3, indicating the client mode.
2. After receiving the packets from the client, the server replies with response packets containing the required information. The Mode field in the response packets is set to 4, indicating the server mode.
3. After receiving the response packets, the client performs clock filtering and selection, and then synchronizes the local clock with the clock of the selected time server.

B.1.2.2 Peer Mode

Figure B.3 shows the exchange process in peer mode.

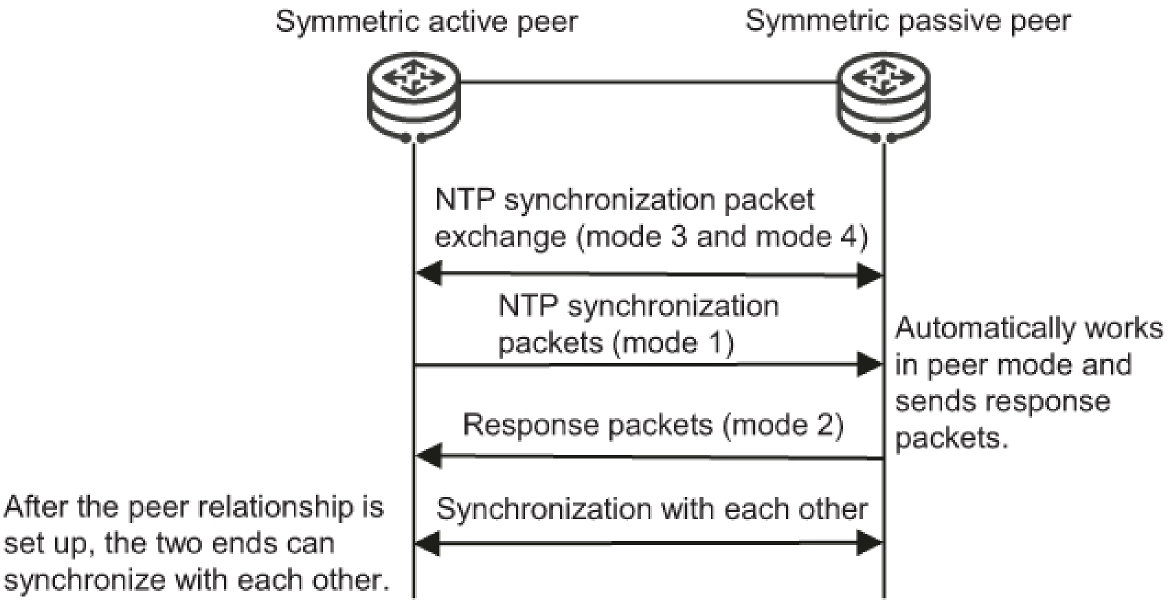

Symmetric active peer
Symmetric passive peer
NTP synchronization packet exchange (mode 3 and mode 4)
NTP synchronization packets (mode 1)
Automatically works in peer mode and sends response packets.
Response packets (mode 2)
After the peer relationship is set up, the two ends can synchronize with each other.
Synchronization with each other

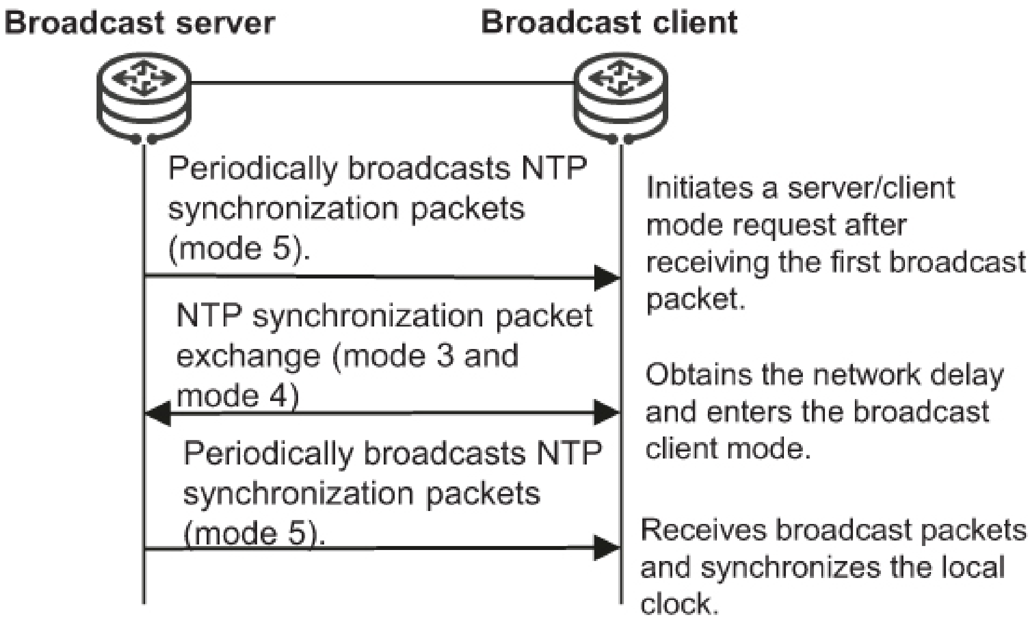
FIGURE B.3 Exchange process in peer mode. ⏎

The exchange process in peer mode is as follows:

1. The symmetric active peer sends an NTP packet with the Mode field being 3 (client mode) to the symmetric passive peer, which replies with an NTP response packet with the Mode field being 4 (server mode).
2. The symmetric active peer periodically sends packets with the Mode field being 1 (symmetric active peer) to the symmetric passive peer.
3. After receiving the packets, the symmetric passive peer replies with response packets with the Mode field being 2 (symmetric passive peer). The symmetric passive peer does not need to be configured. It does not establish a connection or set relevant state variables until it receives an NTP packet.
4. After the peer relationship is set up, the symmetric active and passive peers can synchronize with each other.

B.1.2.3 Broadcast Mode

Figure B.4 shows the exchange process in broadcast mode. In this mode, the time server (broadcast server) provides synchronization information for all clients but does not modify its own clock.



Broadcast server
Broadcast client
Periodically broadcasts NTP synchronization packets (mode 5).
Initiates a server/client mode request after receiving the first broadcast packet.
NTP synchronization packet exchange (mode 3 and mode 4)
Obtains the network delay and enters the broadcast client mode.
Periodically broadcasts NTP synchronization packets (mode 5).
Receives broadcast packets and synchronizes the local clock.

FIGURE B.4 Exchange process in broadcast mode. ⏎

The exchange process in broadcast mode is as follows:

1. The broadcast server periodically sends NTP packets with the Mode field being 5 (broadcast or multicast mode) to a broadcast address.
2. Broadcast clients listen for the broadcast packets from the broadcast server. After receiving the first NTP packet from the broadcast server, each broadcast client interacts with the broadcast server in client/server mode for a short time to obtain the packet roundtrip delay, which is necessary for time synchronization.
3. The broadcast clients continue to listen for subsequent broadcast packets and synchronize their time with that of the broadcast server according to the received broadcast packets.

B.1.2.4 Multicast Mode

Figure B.5 shows the exchange process in multicast mode. In this mode, the time server (multicast server) provides synchronization information for all clients in a multicast group but does not modify its own clock.

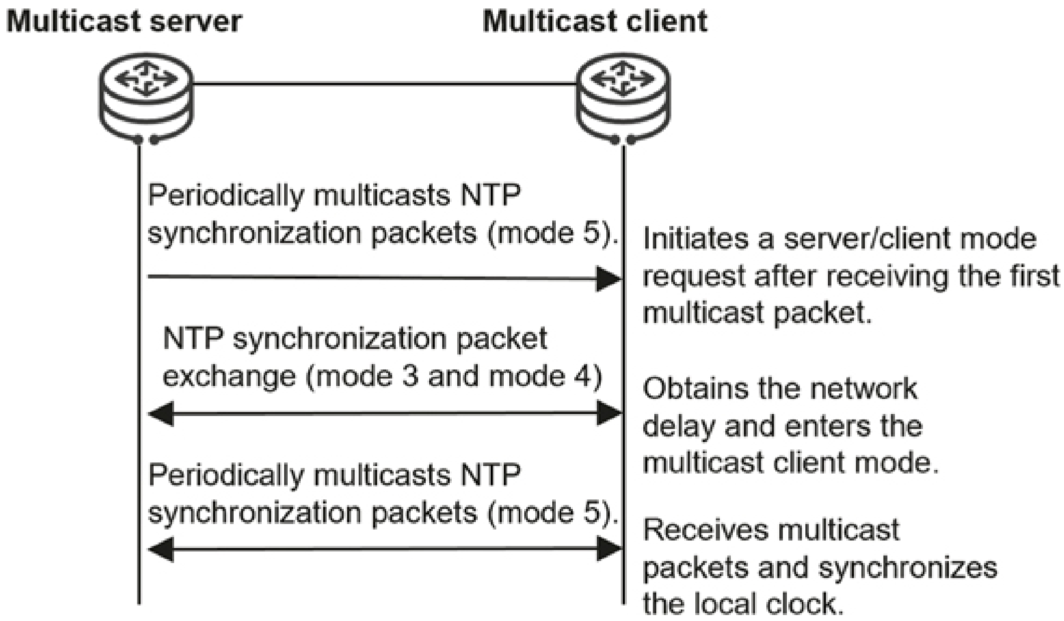

Multicast server
Multicast client
Periodically multicasts NTP synchronization packets (mode 5).
Initiates a server/client mode request after receiving the first multicast packet.
NTP synchronization packet exchange (mode 3 and mode 4)
Obtains the network delay and enters the multicast client mode.
Periodically multicasts NTP synchronization packets (mode 5).
Receives multicast packets and synchronizes the local clock.

FIGURE B.5 Exchange process in multicast mode. ⏎

The exchange process in multicast mode is as follows:

1. The multicast server periodically sends NTP packets with the Mode field being 5 (broadcast or multicast mode) to a multicast address.
2. Multicast clients listen for the multicast packets from the multicast server. After receiving the first NTP packet from the multicast server, each multicast client interacts with the multicast server in client/server mode for a short time to obtain the packet roundtrip delay, which is necessary for time synchronization.
3. The multicast clients continue to listen for subsequent multicast packets and synchronize their time with that of the multicast server according to the received multicast packets.

B.2 PTP

B.2.1 PTP Standards

IEEE 1588, also known as PTP, refers to *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. It defines the basic functions and processing specifications for implementing high-precision time synchronization on networks. Released in 2002, IEEE 1588v1 (also referred to as IEEE 1588-2002) is mainly used in industrial automation and test measurement. In 2008, with IP-based networks advancing and 3G networks becoming more widespread, IEEE made revisions to the standard, officially releasing IEEE 1588v2[2] (also referred to as IEEE 1588-2008) to address the growing demand for time synchronization on telecom networks. In the telecom field, IEEE 1588v2 offers specific improvements over IEEE 1588v1. For example, IEEE 1588v2 defines encapsulation for Layer 2 and Layer 3 networks, introduces the transparent clock (TC) model, and uses extended options to enhance protocol features and functions. In 2019, IEEE released IEEE 1588v2.1[3] (also referred to as IEEE 1588-2019). This standard includes more extended protocol features, such as IEEE 1588 multi-domain and IEEE 1588 security. IEEE 1588v2.1 and IEEE 1588v2 are compatible with each other but not with IEEE 1588v1.

Although defined as a time synchronization protocol, IEEE 1588 can be used for both high-precision time synchronization and high-precision clock synchronization (high-precision frequency synchronization) between devices. In the following sections, IEEE 1588 refers to IEEE 1588v1, IEEE 1588v2, and IEEE 1588v2.1. If a specific version is required, it is specified in full.

IEEE 1588 has been widely used in various fields, including communications, industry, and electric power. These fields have their interworking protocols redefined according to their different application scenarios, implementation functions, and time synchronization precision requirements. For example, ITU-T G.8275.1, defined by ITU-T based on IEEE 1588v2 and IEEE 1588v2.1, is a carrier-class networkwide precise time synchronization protocol mainly used in hop-by-hop time synchronization scenarios in the telecom field, whereas Society of Motion Picture and Television Engineers (SMPTE) ST 2059-2, also based on IEEE 1588v2 and IEEE 1588v2.1, is a high-precision time synchronization protocol applied to video networks in the media asset field.

B.2.2 Basic Concepts in PTP

B.2.2.1 PTP Clock Domain

A PTP clock domain is a logical group of devices that use PTP for clock synchronization. A physical network can be logically divided into multiple PTP clock domains. Each clock domain has its own independent synchronization time, with which devices in the same domain synchronize.

A device can transmit time information from multiple PTP clock domains over a transport network to provide different synchronization times for multiple carrier networks. A device also supports multiple PTP clock domains (PTP logical groups), each performing time synchronization independently.

B.2.2.2 PTP Clock Types

IEEE 1588v2 and IEEE 1588v2.1 define three PTP clock types: Ordinary Clock (OC), Boundary Clock (BC), and TC.

- OC: An OC has only one PTP clock port. When an OC synchronizes its time with an upstream node through the PTP clock port, this OC is also called an OC timeReceiver clock (hereinafter referred to as OC timeReceiver). When an OC advertises time signals to a downstream node through the PTP port, the clock is called a Grand timeTransmitter clock (hereinafter referred to as Grand timeTransmitter).
- BC: A BC has multiple PTP clock ports. It functions as a timeReceiver when it synchronizes its time with an upstream device through one PTP port and functions as a timeTransmitter when it advertises time signals to downstream devices through other PTP ports. A BC can also be used to obtain the standard time from the BITS through a non-PTP port, such as a 1 Pulse Per Second (PPS) + Time of Day (ToD) port, and advertise time signals to downstream devices through other PTP ports.
- TC: A TC has multiple PTP ports, but, unlike OC and BC, it does not need to synchronize time with other devices. Instead, a TC only forwards PTP messages between its PTP ports and corrects the forwarding delay and link delay for PTP event messages without synchronizing its time through any of its PTP ports. TC is classified as E2E TC or P2P TC.

TC + OC: A special type of TC, TC + OC has the same functions as a TC in terms of time transmission (i.e., it forwards PTP messages and corrects the forwarding delay and link delay for PTP event messages) and performs clock or time synchronization on an OC port. TC + OC can be classified as E2E TC + OC or P2P TC + OC.

Figure B.6 shows the hierarchy of the three types of clocks on a time synchronization network.

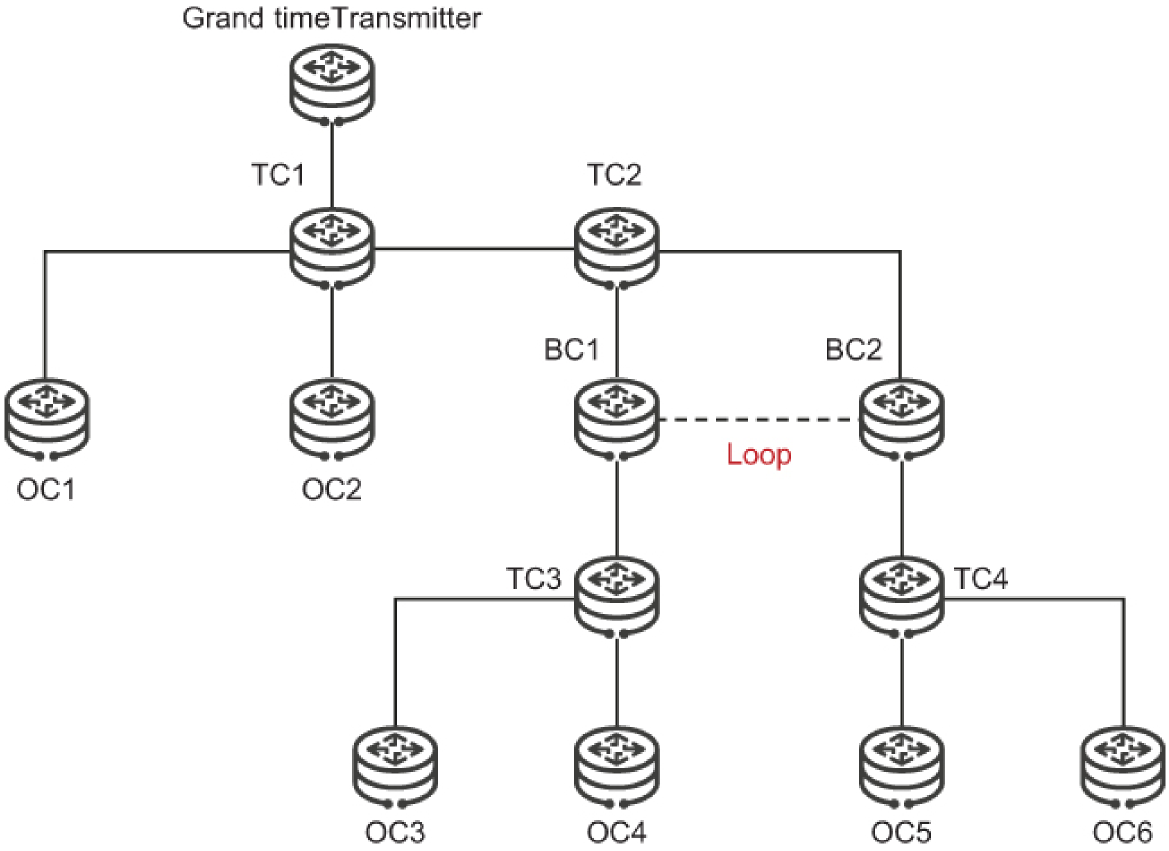


FIGURE B.6 Hierarchy of the three types of clocks on a network. ⏎

In Figure B.6, the Grand timeTransmitter is generally used as a clock source device of the system, and the loop indicates a physical topology loop. IEEE 1588 can use the PTP source selection algorithm to ensure that no loops occur during PTP time transmission. The looped link in this figure is only used as a backup link of BC1 and BC2, not by BC1 and BC2 to synchronize time.

B.2.2.3 PTP Clock Source Selection

On an IEEE 1588v2 time synchronization network, all clocks are organized into a timeTransmitter–timeReceiver hierarchy, where timeTransmitter and timeReceiver are upstream and downstream nodes that respectively advertise and receive the synchronization time. The Grand timeTransmitter serves as the reference time for the system. The topology can be established through static configuration or automatically generated through the Best timeTransmitter Clock Algorithm (BTCA).

In IEEE 1588, the Announce message is used to exchange time source information between PTP nodes. Such information includes the priority of the Grand timeTransmitter, stratum, time precision, and number of hops to reach the

Grand timeTransmitter. With the preceding information, PTP nodes determine the Grand timeTransmitter, select which port to use for clock synchronization with the Grand timeTransmitter, and determine the timeTransmitter–timeReceiver relationship between two nodes. The aim of source selection through BTCA is to establish a loop-free, fully connected spanning tree, with the Grand timeTransmitter as the root.

After a timeTransmitter–timeReceiver relationship is set up between two nodes, the timeTransmitter periodically sends Announce messages to the timeReceiver. If the timeReceiver does not receive any Announce message from the timeTransmitter within a specified period, it considers the timeTransmitter–timeReceiver relationship invalid. The timeReceiver then selects another PTP port or PTP node to establish a new timeTransmitter–timeReceiver relationship and perform time synchronization.

B.2.2.4 External Time Synchronization

While IEEE 1588 enables time synchronization between clock nodes, it cannot synchronize the time of these nodes with UTC. To achieve this, it is necessary to connect the Grand timeTransmitter to an external time source (e.g., a BITS device) through an external time port. This allows the Grand timeTransmitter to obtain the synchronization time in non-PTP mode. Figure B.7 shows the networking involved in external time synchronization. The external time source obtains reference time signals from the Global Navigation Satellite System (GNSS), such as the US's GPS, Europe's Galileo navigation satellite system, Russia's Global Navigation Satellite System (GLONASS), and China's BeiDou Navigation Satellite System (BDS).

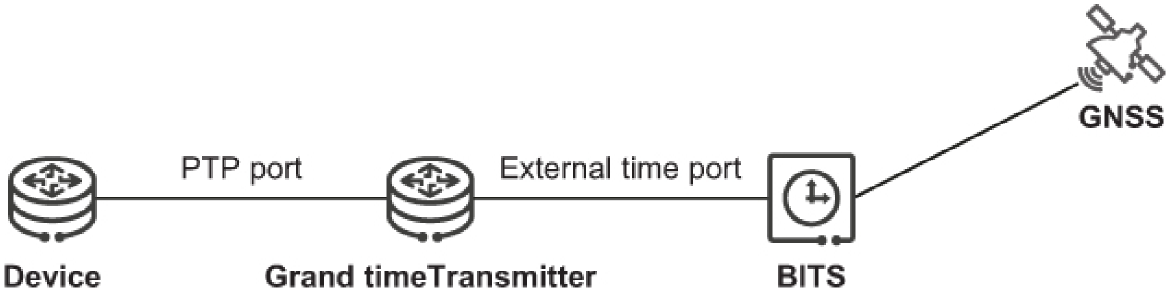


FIGURE B.7 External time synchronization. ⏎

B.2.3 PTP Time Synchronization Fundamentals

The fundamentals of IEEE 1588v2 time synchronization are similar to those of NTP time synchronization. Specifically, the timeTransmitter and timeReceiver exchange Sync (synchronization) messages carrying timestamps with each other and calculate the roundtrip delay between the two devices based on the transmit and receive timestamps in these messages. If the forward and return delays are the same, the one-way delay is half of the roundtrip delay. Based on the messages received from the timeTransmitter, the timeReceiver obtains the time offset between the two devices and then adjusts its time accordingly to implement time synchronization with the timeTransmitter. However, due to delay variation and differences between the forward and return delays, this method does not achieve high synchronization precision. In some high-precision synchronization scenarios, it is necessary to compensate for the asymmetry in the transmit and receive delays between the timeTransmitter and timeReceiver. For details about the technologies used to achieve this, see IEEE 1588v2.1[3].

PTP implements time synchronization between the timeTransmitter and timeReceiver by continuously measuring the delay variation between them and adjusting the time accordingly. PTP defines two delay measurement modes: Delay and Peer Delay.

B.2.3.1 Delay Mode

The Delay mode is applicable to E2E delay measurement. Figure B.8 shows the process of time synchronization in Delay mode.

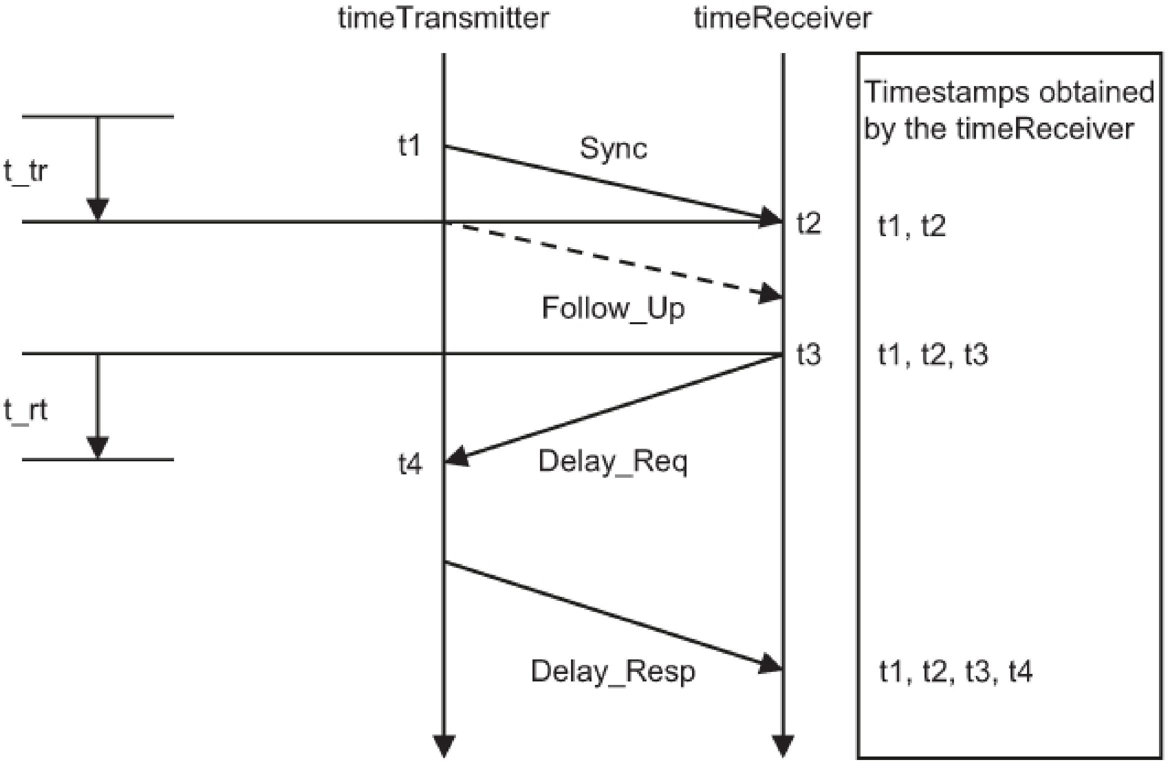


FIGURE B.8 Time synchronization in Delay mode. ⏎

Table B.2 describes the types of messages involved in Delay-based time synchronization.

TABLE B.2 Description of the Types of Messages Involved in Delay-Based Time Synchronization ⏎

Message Type	**Description**
Sync	A Sync message is transmitted by a timeTransmitter to its timeReceiver. It either contains the transmit timestamp or is followed by a Follow_Up message containing this timestamp. It can be used to measure the packet transmission delay from the timeTransmitter to the timeReceiver.
Follow_Up	In scenarios where PTP time synchronization is performed in two-step mode, a Follow_Up message is used as a special Sync message to advertise the transmit timestamp.
Delay_Req	A Delay_Req message is used to request the timeReceiver to return the receive timestamp of the Delay_Req message through a Delay_Resp message.
Delay_Resp	A Delay_Resp message is used to return the receive timestamp of a Delay_Req message.

The time synchronization process in Delay mode is as follows:

1. The timeTransmitter periodically sends a Sync message carrying the transmit timestamp $t1$ to the timeReceiver. Upon receipt of the message, the timeReceiver records the receive timestamp $t2$ for it. If a Sync message cannot carry timestamp $t1$ in scenarios that require high precision, timestamp $t1$ can be carried in a Follow_Up message.
2. The timeReceiver periodically sends a Delay_Req message to the timeTransmitter and records the transmit timestamp $t3$ of the Delay_Req message. Upon receipt of the message, the timeTransmitter records the receive timestamp $t4$ for it and returns a Delay_Resp message carrying timestamp $t4$ to the timeReceiver.
3. The timeReceiver calculates the following parameters based on the obtained timestamps ($t1$, $t2$, $t3$, and $t4$):
   - RoundTripDelay (roundtrip delay on the link between the timeTransmitter and timeReceiver) = $(t4 - t1) - (t3 - t2)$

- MeanPathDelay (one-way delay on the link from the timeTransmitter to timeReceiver, assuming that the forward and return delays are symmetric) = $[(t4 - t1) - (t3 - t2)]/2$
- Offset (time offset of the timeReceiver relative to the timeTransmitter) = $t2 - t1$ − MeanPathDelay

4. The timeReceiver adjusts the local time based on the preceding calculation results to synchronize the time with the timeTransmitter.

This process is performed repeatedly to maintain time synchronization between the timeReceiver and timeTransmitter.

B.2.3.2 Peer Delay Mode

The Peer Delay mode is applicable to point-to-point delay measurement. In this mode, all nodes, including the timeReceiver and timeTransmitter, must know the upstream link delay. To achieve this, they calculate the adjacent link delay by sending messages to adjacent nodes. Figure B.9 shows the process of delay measurement in Peer Delay mode.

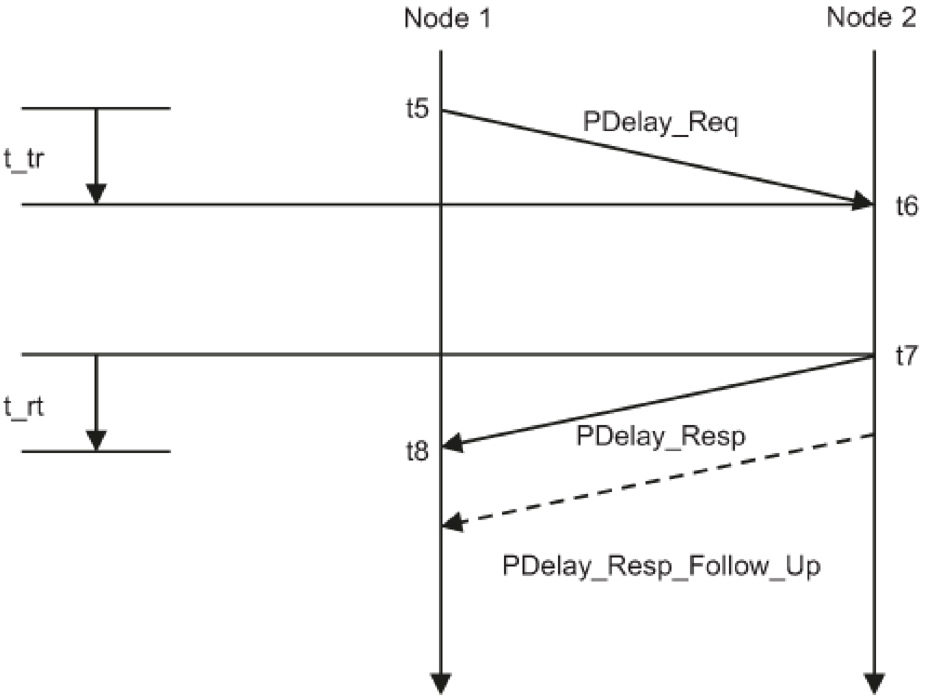


FIGURE B.9 Delay measurement in Peer Delay mode. ⏎

Table B.3 describes the types of messages involved in Peer Delay-based delay measurement.

TABLE B.3 Description of the Types of Messages Involved in Peer Delay-Based

Delay Measurement ⏎

Message Type	Description
PDelay_Req	A PDelay_Req message is transmitted from one PTP port to another PTP port and is used to calculate the delay on the PTP link between the two ports.
PDelay_Resp	A PDelay_Resp message is used to respond to a received PDelay_Req message.
PDelay_Resp_Follow_Up	In scenarios where PTP time synchronization is performed in two-step mode, a PDelay_Resp_Follow_Up message is used to carry the transmit timestamp of a PDelay_Resp message or the time difference between the time when a PDelay_Resp message is sent and the time when a PDelay_Req message is received.

The process of delay measurement in Peer Delay mode is described as follows:

1. Node 1 periodically sends a PDelay_Req message to node 2 and locally saves the transmit timestamp $t5$ of the message. After receiving the message, node 2 records the timestamp $t6$ for it.
2. Node 2 sends a PDelay_Resp message carrying the value of $t7 - t6$ to node 1 and records the transmit timestamp $t7$ for the message. After receiving the message, node 1 records the timestamp $t8$ for it. In scenarios where PTP time synchronization is performed in two-step mode, the corresponding timestamp can also be carried through a PDelay_Resp_Follow_Up message.
3. Node 1 calculates the following parameters based on the obtained timestamps ($t5$, $t7 - t6$, and $t8$):
   - RoundTripDelay (roundtrip delay on the link between node 1 and node 2) = $(t8 - t5) - (t7 - t6)$.
   - MeanPathDelay (one-way delay on the link from node 1 to node 2, assuming that the forward and return delays are symmetric) = $[(t8 - t5) - (t7 - t6)]/2$.

In this process, the link delay is calculated and updated in real time, but time synchronization is not performed. To implement time synchronization, the following steps are also required:

1. The timeTransmitter periodically sends a Sync message carrying the transmit timestamp $t1$ to the timeReceiver. After receiving the message, the timeReceiver records the receive timestamp $t2$ for it.
2. The timeReceiver calculates its time offset relative to the timeTransmitter as follows: Offset = $t2 - t1$ − MeanPathDelay.
3. The timeReceiver adjusts the local time based on the preceding calculation results to synchronize the time with the timeTransmitter.

B.3 TIME SYNCHRONIZATION PROTOCOL SELECTION

NTP is generally implemented through software and is easy to deploy, but it achieves time synchronization precision of only 10 ms to 100 ms. In contrast, PTP is implemented through hardware and measures the link delay based on timestamps added at points closest to the packet receiving or sending end (at the hardware layer). PTP provides time synchronization precision at the nanosecond level, but it depends on the device hardware support.

Network requirements will determine which time synchronization technology to select: PTP is recommended for networks that require high synchronization precision and have sufficient hardware capabilities, whereas NTP is recommended for other networks.

REFERENCES

1. Mills D, Martin J, Burbank J et al. Network time protocol version 4: Protocol and algorithms specification [EB/OL]. (2010-06) [2024-09-30]. RFC 5905. ⏎
2. IEEE Std 1588-2008. IEEE standard for a precision clock synchronization protocol for networked measurement and control systems. (2008-07-24) [2024-09-30]. IEEE 1588v2. ⏎
3. IEEE Std 1588-2019. IEEE standard for a precision clock synchronization protocol for networked measurement and control systems. (2020-06-16) [2024-09-30]. IEEE 1588v2. ⏎

Acronyms and Abbreviations

3G	Third generation of mobile communication technology
3GPP	3rd generation partnership project
5G	5th generation of mobile communication technology
5GC	5G core network
6MAN	IPv6 maintenance
AAU	Active antenna unit
ACC	Access node
ACE	Access control entry
ACL	Access control list
ACM SIGCOMM	Association for computing machine special interest group on data communication
AD	Area director
ADN	Autonomous driving network
AGG	Aggregation node
AI	Artificial intelligence
AM	Alternate marking
AMF	Access and mobility management function
API	Application program interface
APN6	Application-aware IPv6 networking
AS	Autonomous system

ASBR	Autonomous system boundary router
B2B	Business to business
BBU	Baseband unit
BC	Boundary clock
BFD	Bidirectional forwarding detection
BGP	Border gateway protocol
BGP-LS	Border gateway protocol-link state
BITS	Building-integrated timing supply
BMP	BGP monitoring protocol
BRAS	Broadband remote access server
BTCA	Best time transmitter clock algorithm
CBOR	Concise binary object representation
CC	Continuity check
CCSA	China Communications Standards Association
CE	Customer edge
CLI	Command line interface
CPE	Customer premises equipment
CPU	Central processing unit
CPV	Control plane verification
CRC	Cyclic redundancy check
CV	Connectivity verification
DetNet	Deterministic networking
DEX	Direct export
DM	Delay measurement
DNP	Dynamic network probe
DOH	Destination options header
DPV	Data plane verification
DRE	Discounting rate estimator
DSCP	Differentiated services code point
E2E	End-to-end
E2E TC	End-to-end transparent clock
EAM	Enhanced alternate marking
ECA	Event-condition-action

ECMAScript	European Computer Manufacturers Association Script
ECMP	Equal-cost multiple path
ECN	Explicit congestion notification
ENI ISG	Experiential Networked Intelligence Industry Specification Group
ESN	Equipment serial number
ETSI	European Telecommunications Standards Institute
EVPL	Ethernet virtual private line
EVPN	Ethernet virtual private network
FCAPS	Fault, configuration, accounting, performance, security
FIB	Forward information base
FlowSpec	Flow specification
FM	Fault management
FRR	Fast reroute
FSM	Finite state machine
FTP	File transfer protocol
GENEVE	Generic network virtualization encapsulation
GIS	Geographic information system
GLONASS	Global navigation satellite system
GNSS	Global navigation satellite system
GPB	Google protocol buffers
GPS	Global positioning system
GR	Group report
GRE	Generic routing encapsulation
gRPC	Google remote procedure call
GSMA	Global System for Mobile Communications Association
HBH	Hop-by-hop options header
HMAC	Hash-based message authentication code
HSB	Hot standby

HTTP	Hypertext transfer protocol
HTTPS	Hypertext transfer protocol secure
HVPN	Hierarchy VPN
IANA	Internet assigned numbers authority
ICMPv6	Internet control message protocol version 6
ID	Identifier
IDR	Inter-domain routing
IETF	Internet engineering task force
IFIT	In-situ flow information telemetry
IGP	Interior gateway protocol
INT	In-band network telemetry
IOAM	In-situ operation, administration, and maintenance
IoT	Internet of things
IP	Internet protocol
IPE	IPv6 Enhanced innovation
IPFIX	IP flow information export
IP FPM	IP flow performance measurement
IPinIP	IP in IP encapsulation
IPPM	IP performance metrics
IP RAN	IP radio access network
IPSSC	IP Standards Strategy Committee
IPTV	Internet protocol television
IPv4	Internet protocol version 4
IPv6	Internet protocol version 6
IS-IS	Intermediate system to intermediate system
IT	Information technology
ITU-T	International telecommunication union-telecommunication standardization sector
JSON	Java script object notation
KPI	Key performance index
L2NM	L2VPN network model

L2VPN	Layer 2 virtual private network
L3NM	L3VPN network model
L3VPN	Layer 3 virtual private network
LAG	Link aggregation group
LDP	Label distribution protocol
LLM	Large language model
LM	Loss measurement
LSP	Label switched path
LSPA	Label switched path attributes
MAC	Media access control
MC	Metro core
MCD	Midpoint compressed data
MDT	Model driven telemetry
MIB	Management information base
MP2MP	Multipoint-to-multipoint
MP2P	Multipoint-to-point
MPLS	Multi-protocol label switching
MPLS TE	MPLS traffic engineering
MPLS-TP	MPLS transport profile
M-SDO	Multiple Standards Developing Organization
MTrace	Multicast trace
MTU	Maximum transmission unit
NAI	Network artificial intelligence
NCE	Network cloud engine
NE	Network element
NETCONF	Network configuration protocol
NFV	Network functions virtualization
NH	Next header
NHC	Next hop dependent characteristics
NMS	Network management system
NNI	Network-to-network interface
NSH	Network service header
NTF	Network telemetry framework

NTP	Network time protocol
OAM	Operations, administration, and maintenance
OC	Ordinary clock
OIF	Outgoing interface ID
OIL	Outgoing interface load
OLT	Optical line terminal
OMP	Openness management platform
ONOS	Open network operating system
ONT	Optical network terminal
OPSAWG	Operations and Management Area Working Group
OSPFv3	Open shortest path first version 3
OSS	Operational support system
P2P	Point-to-point
P2P TC	Peer-to-peer transparent clock
P4	Programming protocol-independent packet processors
PCC	Path computation client
PCE	Path computation element
PCEP	Path computation element protocol
PE	Provider edge
PIF	Protocol independent forwarding
PM	Performance measurement
POF	Protocol oblivious forwarding
POT	Proof of transit
PSAMP	Packet sampling protocol
PT	Path tracing
PTN	Packet transport network
PTP	Precision time protocol
PTSF	Packet timing signal fail
QoS	Quality of service
RAN	Radio access network
RC	Regional collector

REST	Representational state transfer
RFC	Request for comments
RH	Routing header
RIB	Routing information base
RI LSA	Router information opaque link state advertisement
RPC	Remote procedure call
RR	Route reflector
RTT	Round trip time
SAFI	Subsequent address family identifier
SDN	Software defined network
SFC	Service function chain
sFlow	Sampled flow
SFTP	Secure file transfer protocol
SID	Segment ID
SLA	Service level agreement
SND	Specific NE driver
SNMP	Simple network management protocol
SPE	Superstratum provider edge
SRH	Segment routing header
SR-TE	Segment routing-traffic engineering
SRv6	Segment routing over IPv6
SSP	Specific service plugin
STAMP	Simple two-way active measurement protocol
STN	Smart transport network
TC	Transparent clock
TC	Traffic class
TCP	Transmission control protocol
TE	Traffic engineering
TIH	Telemetry information header
TLS	Transport layer security
TLV	Type-length-value

TM Forum	Telemanagement forum
T-MPLS	Transport MPLS
ToD	Time of day
ToS	Type of service
TTL	Time to live
TTS	Truncated timestamp
TWAMP	Two-way active measurement protocol
UDP	User datagram protocol
UI	User interface
UNI	User-to-network interface
UPE	User-end provider edge
UPF	User plane function
UTC	Coordinated universal time
VIP	Very important person
VPLS	Virtual private LAN service
VPN	Virtual private network
VPWS	Virtual private wire service
VR	Virtual reality
VRP	Versatile routing platform
VXLAN-GPE	Generic protocol extension for VXLAN
WAN	Wide area network
XML	Extensible markup language
YANG	Yet another next generation