# FabricInsight V100R019C00 Technical White Paper

**Issue** 01

**Date** 2019-11-11

**HUAWEI TECHNOLOGIES CO., LTD.**

### Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

### Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

## Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

# Contents

**1 Product Overview....1**
1.1 Solution Design....3
**2 Key Technical Principles....5**
2.1 Architecture....5
2.2 ERSPAN Flow Analysis....11
2.2.1 TCP Flow Collection Principle....11
2.2.2 TCP Session Traffic Calculation Principle....12
2.2.3 Packet Route Calculation Principle....13
2.2.4 Packet Transmission Latency Calculation Principle....15
2.2.5 Application Identification Principle....15
2.2.6 TCP Exception Detection Principle....16
2.3 Telemetry Performance Metric Analysis....17
2.3.1 Performance Metric Collection Principle....17
2.3.2 Dynamic Baseline Calculation Principle....23
2.3.3 Baseline Exception Detection Principle....25
2.4 Issue Analysis and Troubleshooting Analysis....27
2.4.1 Application Quality....28
2.4.1.1 Continuous Service Interruption....28
2.4.1.2 Intermittent Service Interruption....32
2.4.1.3 Host Port Not Listened On....33
2.4.2 Network Services....33
2.4.2.1 Insufficient TCAM Resources....33
2.4.2.2 Insufficiency or Sharp Change of FIB Entry Resources....34
2.4.2.3 Insufficiency or Sharp Change of ARP Entry Resources....35
2.4.2.4 Insufficiency or Sharp Change of MAC Entry Resources....35
2.4.3 Security Compliance....37
2.4.3.1 Non-compliant Traffic Interaction....37
2.4.3.2 Suspicious SYN Flood Attack....38
2.4.3.3 Suspicious Port Scanning Attack....40
2.5 RoCE Flow Analysis....42
2.5.1 Metric Data Collection Principles....42
2.5.2 Metric Data Calculation Principles....44
2.6 Edge Intelligent Analysis....46

2.6.1 Intelligent TCP Traffic Analysis....46
2.6.1.1 Data Collection Principles for Intelligent TCP Traffic Analysis....46
2.6.1.2 Calculation Principles for Intelligent TCP Traffic Analysis....47
2.6.2 Intelligent UDP Traffic Analysis....49
2.6.2.1 Data Collection Principles for Intelligent UDP Traffic Analysis....49
2.6.2.2 Data Calculation Principles for Intelligent UDP Traffic Analysis....50
**3 Function Constraints....52**
3.1 Device Types and Networking Restrictions....52
**4 Typical Application Scenarios....53**
4.1 TCP Connection Setup Failure Analysis....53
4.2 TCP RST Packet Analysis....55
4.3 Proactive Prediction of Abnormal Device Metrics and Correlation Flow Analysis....59

# 1 Product Overview

With the acceleration of digital transformation in the industry, more and more services and applications are deployed in data centers. In addition, the development of software technologies such as big data, machine learning, distribution, and servitization accelerates the pace of digital transformation in the industry. Cloudification of enterprise data centers becomes increasingly urgent, and cloud computing is becoming the basic capability of each industry. It is an urgent task for enterprises to quickly build cloud-based data centers that can support future service development. Data center networks, as the cornerstone of constructing cloud data centers, are facing great challenges. Traditional data center networks can hardly be cloudified. To handle this problem, the SDN is developed.

In the SDN era, computing resource pooling, storage resource pooling, network resource pooling, and network and service automation bring convenience to users but great challenges to network O&M. Compared with traditional network O&M, network O&M in the SDN era features in the following: proactive, real-time, and large-scale.

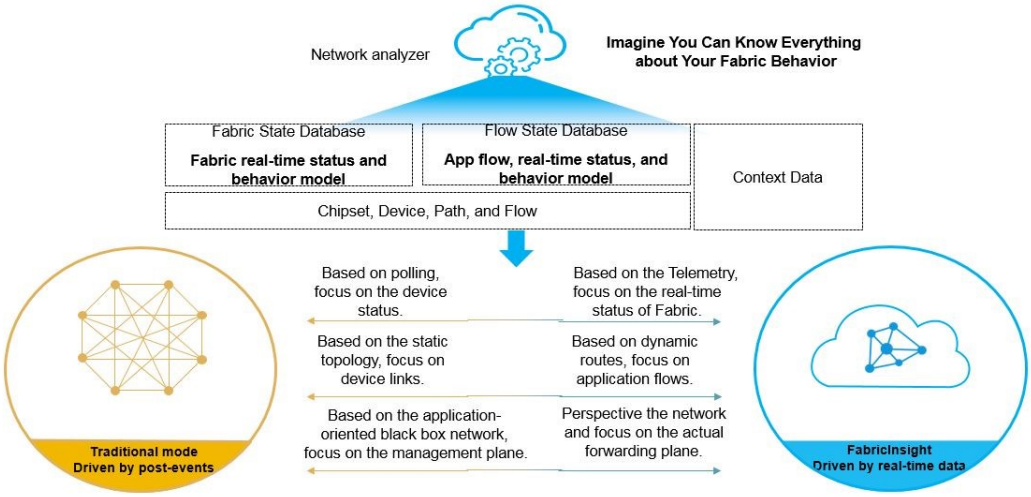
- **Proactive**: The SDN scenario requires that services can be provisioned quickly and dynamically. For example, if logical networks are created and deleted as required, network or service configuration changes frequently. Frequent configuration change increases the fault probability. The O&M system must be able to proactively and intelligently detect these faults, and use big data analysis and experience databases to help users quickly locate and rectify faults.
- **Real-time**: The O&M system can detect microburst exceptions on the network in a timely manner. For example, an enterprise customer complained that its lightweight network had the issue of transient packet loss and suspected that there were millisecond-level traffic bursts. However, these issues cannot be detected in the minute-level SNMP mechanism, let alone be optimized.
- **Large-scale**: Large-scale management has many meanings. On one hand, managed objects are extended from physical devices to virtual machines (VMs) and the NE management scale is increased by dozens of times. On the other hand, the device indicator collection granularity is improved from minutes to milliseconds to meet real-time analysis requirements, and the data volume is increased by nearly 1000 times. For active awareness and troubleshooting of issues, FabricInsight needs to collect and analyze network device indicators, and analyze the actual forwarding service flows, further increasing the data scale.

The traditional O&M management system is challenged by the preceding three features in SDN network O&M. According to a survey conducted by the EMA on over 100 enterprises,

about 70% of customers are concerned about whether the existing network O&M system is applicable to the SDN scenarios.

To deal with the O&M challenges (proactive, real-time, and large-scale) in the SDN scenario, the customer needs to change the overall O&M architecture so that the SND network can be easily used. Huawei FabricInsight, an intelligent network analysis platform, overrides the traditional monitoring focusing on resource status, detects fabric status and application behavior in real time, and breaks the boundaries between networks and applications. In addition, FabricInsight analyzes networks from the application perspective, proactively detects network or application issues, and provides automatic troubleshooting capabilities for service connectivity issues, helping users quickly demarcate and rectify faults and ensure continuous and stable running of applications.




The FabricInsight O&M architecture is constructed based on the following points:

- **Visualization: visible and clear**

  The concept of "visible" consists of two aspects: observed objects and real-time observation. Observed objects include physical objects such as devices, interfaces, and links and logical objects such as packet forwarding path, service interaction relationship, and service interaction quality. Real-time observation supports perception of millisecond-level symptoms, for example, identifying microburst traffic congestion on the network. The concept "clear" refers to the observation accuracy. On one hand, a large amount of data needs to be collected, for example, collecting all TCP flows. On the other hand, the data must be analyzed in real time to identify abnormal service flows.

- **Automation: proactive analysis and automatic troubleshooting**

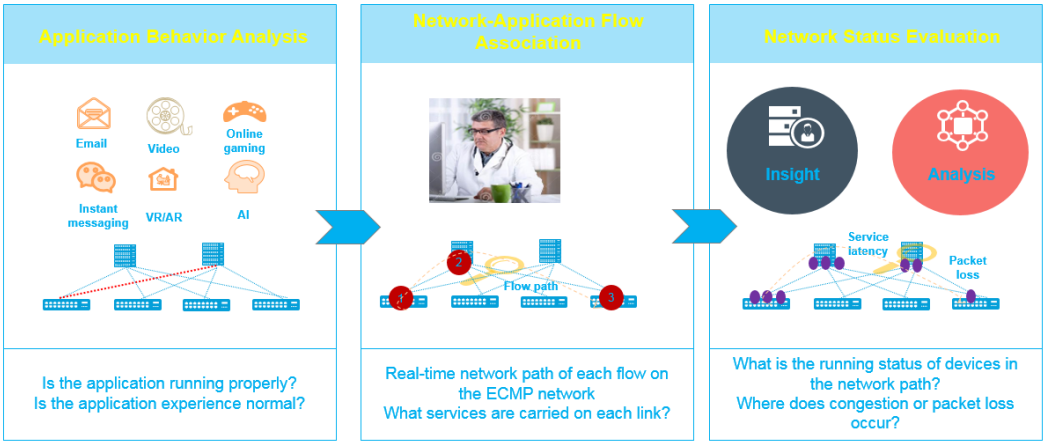
  To proactively and intelligently detect issues on the network in a timely manner, the O&M system must be able to analyze massive data and identify abnormal events on the network, for example, service connectivity issues and traffic congestion ports. In addition, the O&M system needs to determine whether to generate issue models and recommend them to users based on machine learning algorithms. For automatic troubleshooting, the O&M system must be able to analyze issue data and learn the issue case library. In addition, the O&M system must be able to orchestrate executable troubleshooting task links for different fault patterns, reducing the time required for issue demarcation and locating.

## 1.1 Solution Design

# 1.1 Solution Design

FabricInsight collects and analyzes the original TCP feature packets forwarded on the network, displays the application interaction relationship and quality, and visualizes the network traffic. In addition, FabricInsight parses packet features, and restores hop-by-hop forwarding paths of packets and forwarding traffic and latency of links to implement association between applications and networks. Then, FabricInsight collects the packet loss, traffic, and configuration of network devices through technologies such as Telemetry and proactively evaluates the network service status based on AI algorithms such as dynamic baseline and Gaussian regression. In this way, FabricInsight can build the multi-layer association analysis capability from service flow to forwarding path to network service, and display application behavior and network quality in a structured manner.

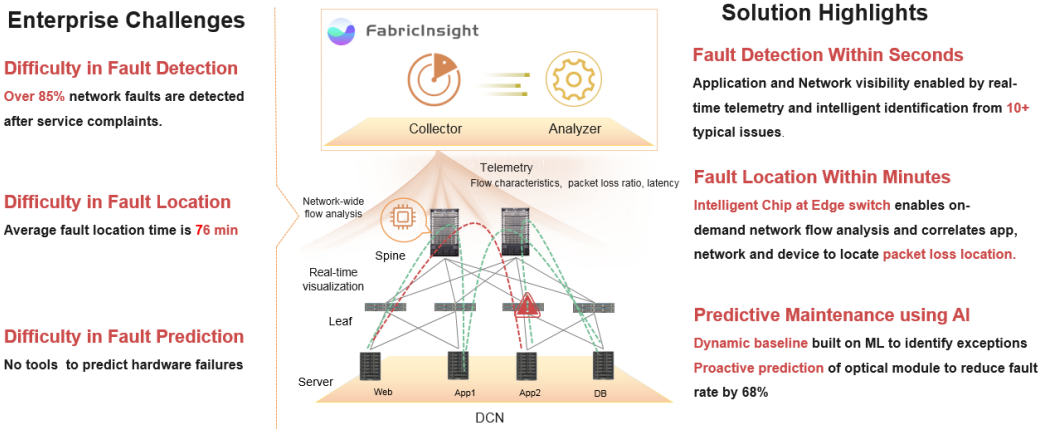
**Figure 1-1** FabricInsight solution design




FabricInsight performs big data analysis on collected ERSPAN flows and Telemetry performance metrics through distributed real-time and offline computing. In addition, FabricInsight proactively detects possible issues on the fabric based on AI algorithms such as baseline exception detection and multi-dimension clustering analysis, and intelligently analyzes and identifies whether the network or application has group issues. For service connectivity issues, FabricInsight automatically orchestrates troubleshooting procedures to support one-click automatic troubleshooting. All these help users achieve the proactive and intelligent O&M goal for proactive issue detection and minute-level issue locating and demarcation.

**Figure 1-2** Proactive and intelligent O&M



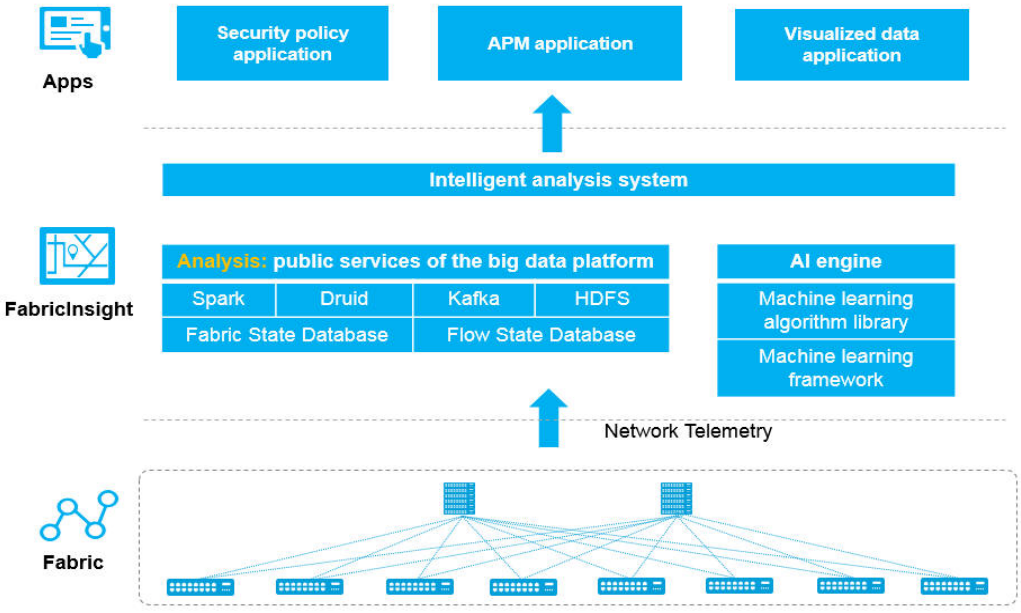
# 2 Key Technical Principles

2.1 Architecture
2.2 ERSPAN Flow Analysis
2.3 Telemetry Performance Metric Analysis
2.4 Issue Analysis and Troubleshooting Analysis
2.5 RoCE Flow Analysis
2.6 Edge Intelligent Analysis

## 2.1 Architecture

Based on Huawei Big Data platform, FabricInsight receives data from network devices in Telemetry mode and uses intelligent algorithms to analyze and display network data. The FabricInsight architecture consists of three parts: network device, FabricInsight collector, and FabricInsight analyzer.

**Figure 2-1** FabricInsight architecture



The FabricInsight analyzer uses the microservice architecture. Each service is deployed in multi-instance mode, which features high reliability and scalability. You can expand the service capacity by expanding instance nodes. Instances are independent of each other. External HTTP requests are distributed by the message bus to each node for processing. The analyzer connects to the collector in the southbound direction and uses the LVS to improve system reliability.

**Figure 2-2** FabricInsight analyzer microservice architecture

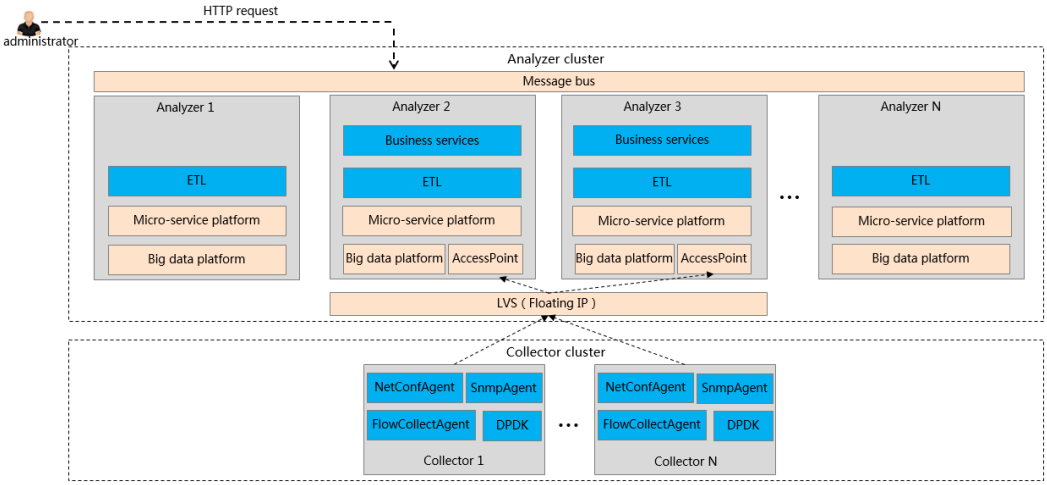
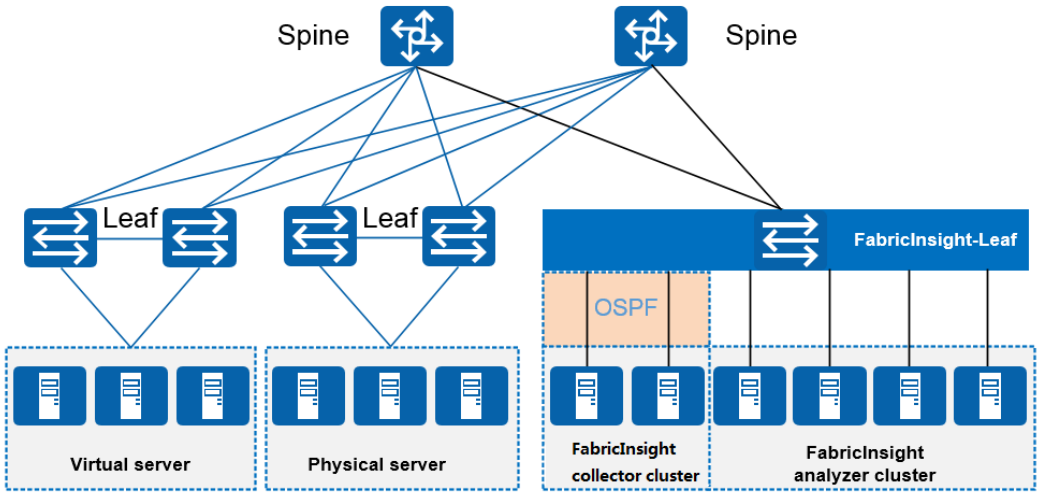



**Figure 2-3** FabricInsight networking




### Network devices

Network devices are switches on the data center network, such as the leaf and spine nodes in the figure. Currently, Huawei CE-series switches are supported. For the current FabricInsight version, devices need to report two types of data in Telemetry mode: TCP packets mirrored based on ERSPAN and performance metrics such as interface traffic reported based on (Google Remote Procedure Call Protocol) GRPC.

- ERSPAN mirrored packets: The forwarding chip on the switch identifies TCP SYN and FIN packets on the network and mirrors the packets to the FabricInsight collector

through the ERSPAN protocol. Since the forwarding chip directly identifies and mirrors the packets without using the CPU, the stability of the switch is not affected. In addition, the original packets remain unchanged and the forwarding routes of the original packets are not affected.

- GRPC performance metrics: Devices are connected as GRPC clients. Users can configure the Telemetry sampling function for a device using commands. The device then proactively establishes a GRPC connection with the target collector and sends data to the collector. The current version supports the following sampling metrics:
  - CPU and memory usage at the device and board levels
  - Number of sent and received bytes, number of discarded sent and received packets, and number of sent and received error packets at the interface level
  - Number of congested bytes at the queue level
  - Packet loss behavior data

  For details about indicator details and device models, see the product specification list.

**FabricInsight collector**

The FabricInsight collector collects data reported by switches in Telemetry mode, including TCP packets mirrored based on ERSPAN and performance metrics reported based on GRPC. For mirrored TCP packets, the collector adds timestamps to the packets, and packs and sends the packets to the analyzer for analysis. To improve the packet processing efficiency, the collector is implemented based on Intel Data Plane Development Kit (Intel DPDK). Therefore, the collector needs to support the DPDK network adapter. The Intel 82599 10GE network adapter is recommended.

**FabricInsight analyzer**

The FabricInsight analyzer cluster receives data from the collector, including TCP packets and performance metrics. The analyzer cleans different types of data using related cleaning logic, for example, calculating the forwarding path, forwarding latency, and link latency of packets. In addition, the analyzer analyzes application interaction relationships, associates applications with network paths, establishes dynamic baselines for some performance metrics based on the AI algorithms, detects exceptions, predicts the fault probability of optical modules. The analyzer can collect statistics on and analyze these data and display the analysis result.

**FabricInsight high availability**

FabricInsight uses the cluster technology to prevent service interruption upon single point of failure (SPOF). It mainly includes the collector cluster and analyzer cluster.

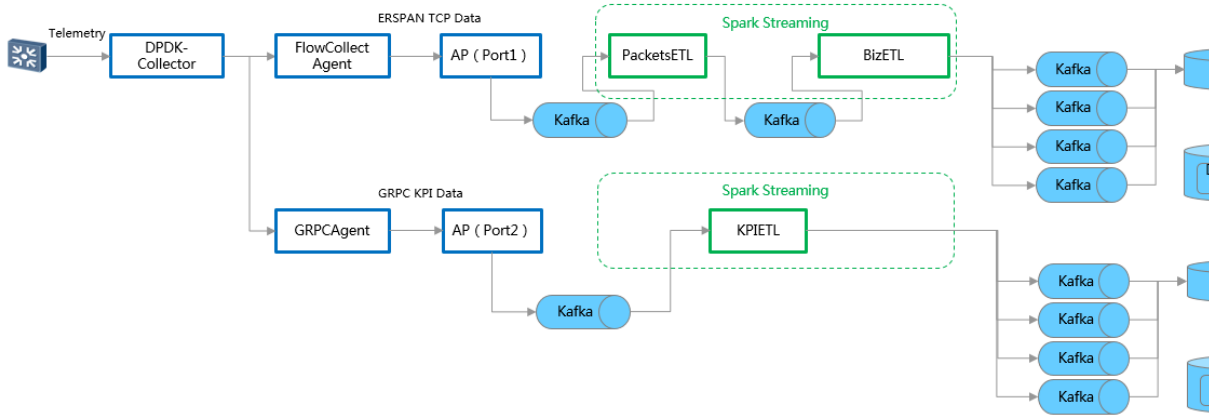
- Collector cluster: Service network ports on the collector nodes are bonded. If a service network port is faulty, functions of the collector are not affected. In the collector cluster, each collector node establishes an OSPF neighbor relationship with its leaf node, advertises a unified virtual IP (VIP) route, and uses the equal-cost multi-path (ECMP) capability of the network device to implement multi-active load balancing of the collector. When a collector node in the cluster is faulty, the collector node stops sending OSPF heartbeat packets. After the heartbeat timeout period elapses, the leaf switch triggers route recalculation and subsequent mirrored packets and performance metrics will not be sent to the collector node, implementing dynamic fault isolation.
- Analyzer cluster: Service network ports on the analyzer nodes are also bonded. If a service network port is faulty, functions of the analyzer are not affected. In addition, the analyzer uses the microservice architecture. Service functions are deployed on multiple analyzer nodes. Microservice instances are independent of each other and the same result

is returned for user requests regardless of which node the user requests to access. After services are started, they automatically register the service access routes with the message bus, which then forwards HTTP requests based on external request URLs. In addition, the message bus periodically checks whether the service port of each analyzer node is available. If the service port is unavailable, the message bus deletes the service port from the routing table. If an analyzer node is unavailable or services on an analyzer node are unavailable, the analyzer cluster can still be normally accessed by external systems.

**FabricInsight data flow**

ERSPAN mirrored packets and GRPC performance metric data are reported to the collector through switches. The collector parses the data, extracts related fields, and reports them to the AP service of the analyzer in the specified format. The AP service only receives the data and saves it to the Kafka. Three Spark Streaming cleaning tasks (PacketsETL, BizETL, and KPIETL) are executed on the analyzer cluster to obtain data from the Kafka in real time for service processing. The PacketsETL task combines TCP packets with the same quintuple into one record and writes the record to the Kafka. The BizETL task processes data cleaned by the PacketsETL task (for example, application identification and route calculation) and writes the processed data to the Kafka. Then, Druid writes the data processed by the BizETL task to the HDFS. The KPIETL task cleans performance metric data, for example, supplementing dimensions for specified metric groups based on the site requirements, processing differences of metrics such as the number of interface inbound bytes in the adjacent two periods, and writing data to the Druid through the Kafka.

**Figure 2-4** FabricInsight data flow diagram

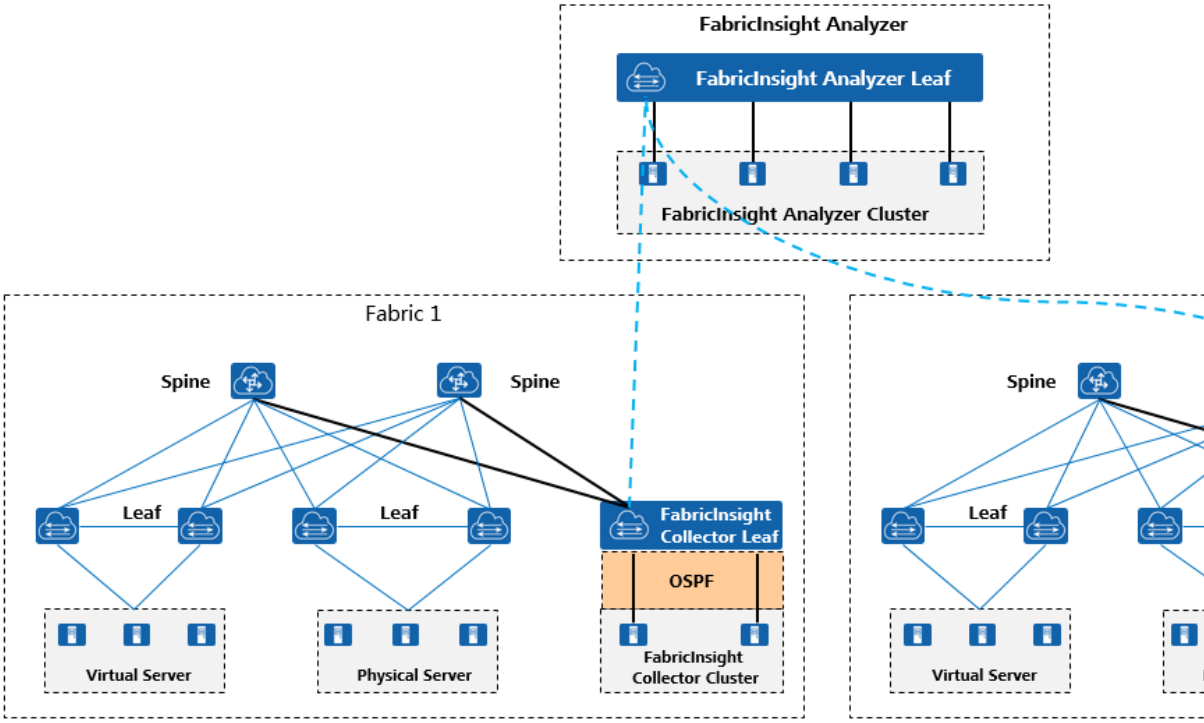



## NOTE

1. Kafka is a high-throughput and distributed message system based on the release and subscription.
2. Spark Streaming is an extension of Spark Core API and supports distributed computing and processing of elastic, high-throughput, and fault-tolerant real-time data flows.
3. Druid is a fast column-oriented distributed data storage system that supports high-speed aggregation and second-level query. It also supports million-level event access per second.
4. Hadoop Distribute File System (HDFS) is a distributed file system that provides high-throughput data access and applies to large-scale data sets.

**Multi-Fabric Management**

FabricInsight supports multi-fabric management. Each fabric can be deployed with a collector cluster and an analyzer cluster is used to manage data on all fabrics. In the multi-fabric management scenario, the collector cluster and analyzer cluster on each fabric must communicate with each other through the outband management network. The following figure shows the networking.




Similar to single-fabric management, all interaction with the devices is completed by the collector in the multi-fabric management scenario. The collector cluster on a fabric receives ERSPAN mirrored packets and Telemetry performance metric data on the fabric, and reports the data to the unified analyzer in a unified format. After receiving data reported by the fabric collectors, the analyzer cleans and imports the data into the database based on the related service logic and records the fabric label. Users can filter and view data by fabric on the GUI.

The communication bandwidth between the collector clusters and analyzer cluster must meet the requirements based on the data scale. In different data management scenarios, the required bandwidth can be estimated based on formulas in the following table.

**Table 2-1** Estimating the communication bandwidth between the collector clusters and analyzer cluster

| Data Management Scenario | Estimation Formula | Example | Remarks |
| --- | --- | --- | --- |
| ERSPAN flow analysis | Number of flows/s x 12 mirrored packets/flow x 128 bytes/packet x Data compression ratio (about 0.6) x 8 bit/s | If there are 20000 flows per second on the network, the required bandwidth is calculated as follows:<br>20000*12*128*0.6*8bps=140Mbps | ● Each flow has 12 mirrored packets. The calculation rule is as follows: Each flow contains 4 feature packets (SYN, SYNACK, FINACK, and FINACK) and each packet passes through 3 hops during network forwarding. |
| ERSPAN flow + Telemetry performance metric analysis | Number of flows/s x 12 mirrored packets/flow x 128 bytes/packet x Data compression ratio (about 0.6) x 8 bit/s +<br>(Number of devices reporting Telemetry performance metrics x 452 measurement object metric sets/per device per minute x 256 bytes x Data compression ratio (about 0.6) x 8 bit/s)/60 | If there are 10000 flows per second on the network on average and Telemetry performance metric data reporting is enabled for 100 devices, the required bandwidth is calculated as follows:<br>10000 x 12 x 128 x 0.6 x 8 bit/s + 100 x 452 x 256 x 0.6 x 8 bit/s/60 = 71 Mbit/s | ● If each device has 50 interfaces and 400 queues on average, and device and interface metrics are reported every one minute, queue congestion occurs once a minute.<br>● The device-level measurement object has two metric sets. The interface- or queue-level measurement object has one metric set.<br>● Each metric set has five collection metrics on average. On average, 256 bytes are reported for each metric of each measurement object. |

NOTE

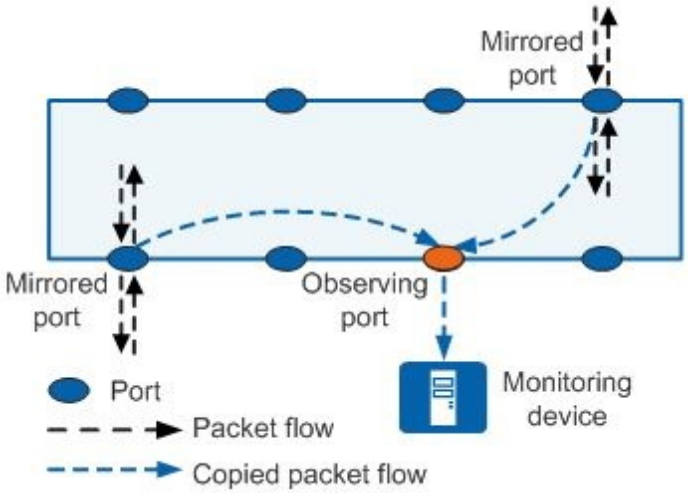
1. When the collector cluster and analyzer cluster are deployed remotely, cross-WAN communication is not supported. You are advised to directly connect the collector cluster and analyzer cluster through optical fibers.
2. The network visualization page displays data by fabric view. Each fabric view displays the overview, network topology, and abnormal packet statistics of the fabric. The page supports fabric view switchover. Information about multiple fabrics cannot be displayed in the same fabric view.
3. If all fabrics of flows interacting cross fabrics have been managed and Network Address Translation (NAT) is not performed for packets during packet forwarding, packet forwarding routes on all involved fabrics can be displayed in the packet traveling topology on the flow events page at the same time.
4. Collector clusters on different fabrics support only NTP clock synchronization and does not support 1588v2 (PTP) high-precision clock synchronization. Therefore, for packets exchanged across fabrics, the latency of inter-fabric interaction is accurate to millisecond, which has a certain precision error. However, the hop-by-hop latency on the fabric is still accurate to submicroseconds. (For details about the packet transmission latency calculation principle, see Packet Transmission Latency Calculation Principle.)

# 2.2 ERSPAN Flow Analysis

This section describes the key technical principles for analyzing TCP packets mirrored based on ERSPAN.

## 2.2.1 TCP Flow Collection Principle

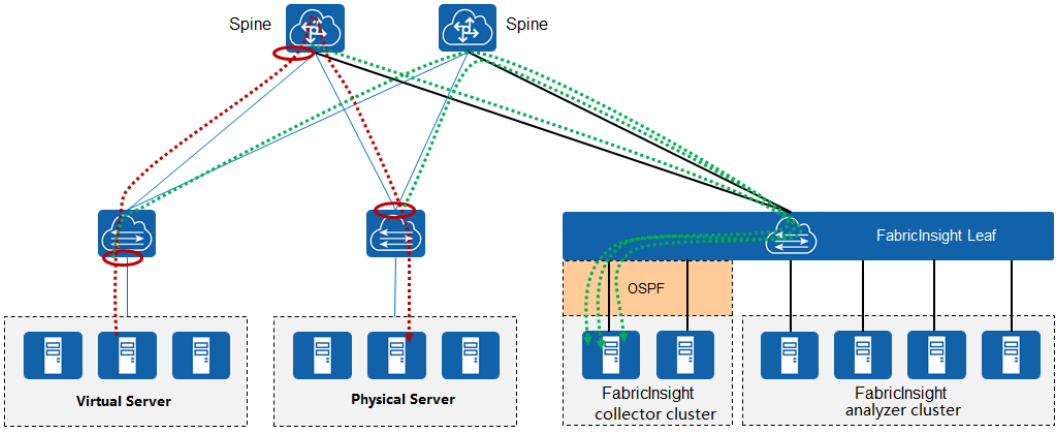
FabricInsight uses the remote flow mirroring capability of the switch to configure traffic classification on the switch to match TCP packets. Then, FabricInsight sends the packets to the monitoring device (FabricInsight collector) through the ERSPAN protocol.



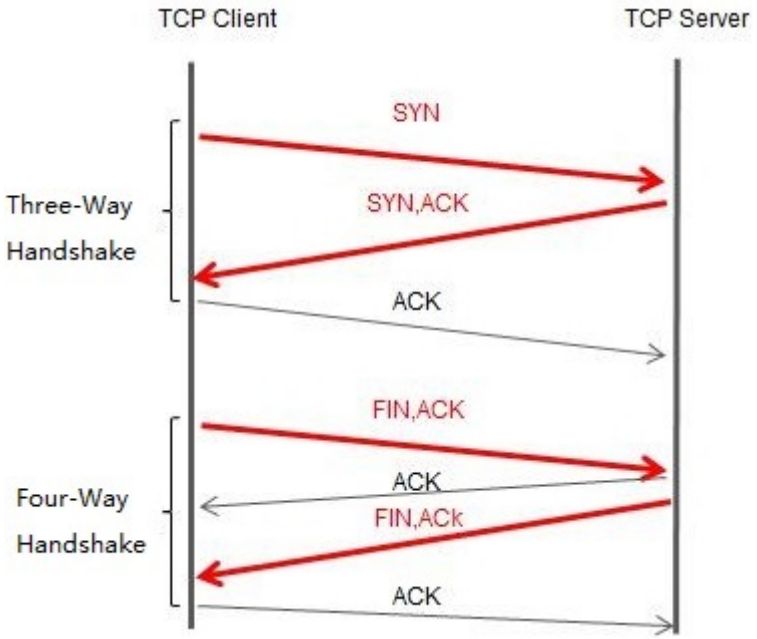

As shown in the following figure, assume that two VMs communicate with each other crossing leaf nodes. The red dotted lines indicate the packet routes. The remote mirroring in the inbound direction is enabled for each switch on the packet transmission route. If the packet passes through three hops from Leaf to Spine to Leaf, each of the three switches mirrors the packet to the FabricInsight collector once. The FabricInsight analyzer uses algorithms to restore the packet transmission route and perform related statistics and analysis.

**Figure 2-5** FabricInsight traffic mirroring




In the TCP protocol, three handshakes are required for setting up a TCP connection and four handshakes are required for tearing down the connection. To monitor TCP connection setup and teardown between applications on the network, FabricInsight needs to mirror the SYN, FIN, and RST packets in the TCP protocol to the FabricInsight collector.




To enable the ERSPAN remote flow mirroring function on the switch, you need to install the ERSPAN plug-in on the switch. For details about how to install the ERSPAN plug-in, see the related manuals of the CE-series switches. After installing the ERSPAN plug-in, you need to complete related configurations on the device and enable the flow mirroring function. For device models that support the ERSPAN enhanced feature, you need to enable the ERSPAN enhanced feature when configuring flow mirroring. The configuration commands may vary depending on the device model and version. For details, see the configuration guide of the CE-series switch.

## 2.2.2 TCP Session Traffic Calculation Principle

FabricInsight calculates the traffic of TCP sessions based on the TCP sequence number of SYN and FIN packets.

- Traffic volume in the request direction = Sequence number of the FINACK packet in the request direction - Sequence number of the SYN packet
- Traffic volume in the response direction = Sequence number of the FINACK packet in the response direction - Sequence number of the SYNACK packet

If the TCP sequence number is rotated only once in the TCP session duration, the rotated TCP sequence number can be identified and corrected. If the TCP sequence number is rotated multiple times during the TCP session duration, the rotated TCP sequence number cannot be identified through the current technology and the traffic calculation result may be incorrect.

NOTE

Each byte in the data flow transmitted over the TCP link is encoded with a sequence number. That is, the **Sequence Number** field in the TCP protocol is added to each byte. The length of **Sequence Number** is 32 bits, and the value ranges from 0 to 4294967295. When the sequence number in the TCP link life cycle is accumulated to 4294967295, the sequence number starts from 0 again, which is called sequence number rotation.

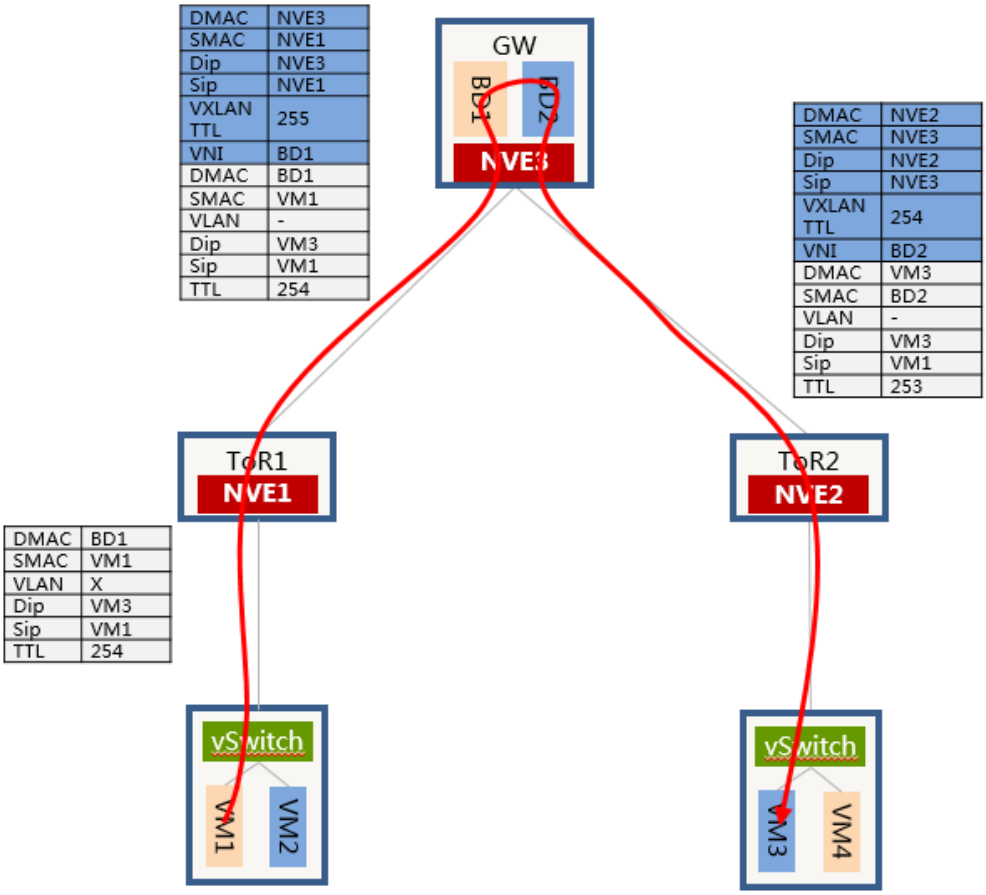
## 2.2.3 Packet Route Calculation Principle

After collecting the TCP packets mirrored by network devices, the FabricInsight analyzer calculates each TCP packet and restores each hop device of the TCP packet. The current version supports packet parsing in ERSPAN Type2 and Type3 formats.

- ERSPAN Type2 packets: The mirrored packets do not contain the forwarding port. In this case, the calculated transmission path contains the devices of each hop of the packet but the specific ports cannot be identified.
- ERSPAN Type3 packets: The mirrored packets contain the inbound forwarding port. In this case, the port of each-hop device that the packet passes through can be calculated based on the physical link data. The prerequisite is that each-hop device in the packet forwarding route must support the ERSPAN enhanced feature and have the feature enabled.

The following uses the layer-3 forwarding process of the hardware-centralized gateway as an example to describe the packet route calculation process.




As shown in the preceding figure, the overall networking mode is the hardware-centralized gateway mode. VM1 and VM3 are located on different subnets. The communication between the two subnets needs to be routed and forwarded by the gateway. During packet transmission, FabricInsight receives three copies of mirrored packets.

The first copy of mirrored packets is the mirrored packets in the inbound direction of the ToR1. The packets are IP packets sent by VM1. After being forwarded by the vSwitch, the packets are tagged with VLAN tags. After arriving at ToR1, the packets are forwarded through VxLAN. Since the packets are forwarded across subnets, the next hop is the gateway.

The second copy of mirrored packets is the mirrored packets in the inbound direction of the gateway. The packets are VxLAN packets sent by the ToR1. After receiving the packets, the ToR1 determines that the next hop is the gateway and encapsulates the VxLAN packets. The source IP address is the IP address of NVE1 and the destination IP address is the IP address of NVE3 on the gateway.

The third copy of mirrored packets is the mirrored packets in the inbound direction of the ToR2. The packets are VxLAN packets forwarded by the gateway at layer 3. When performing layer-3 forwarding, the gateway decapsulates the VxLAN packets of the BD1 and encapsulates the VxLAN packets of the BD2.

After receiving the packets, FabricInsight performs matching based on the content of the inner packets to identify the same TCP packets. After identifying the three copies of mirrored packets as the same TCP packets, FabricInsight sorts the packets based on the TTLs of inner and outer packets (because the TTLs of inner and outer packets change after packets are forwarded by the gateway at layer 3) and calculates the packet forwarding routes based on certain rules. Since the mirrored packets do not contain the device ports through which TCP

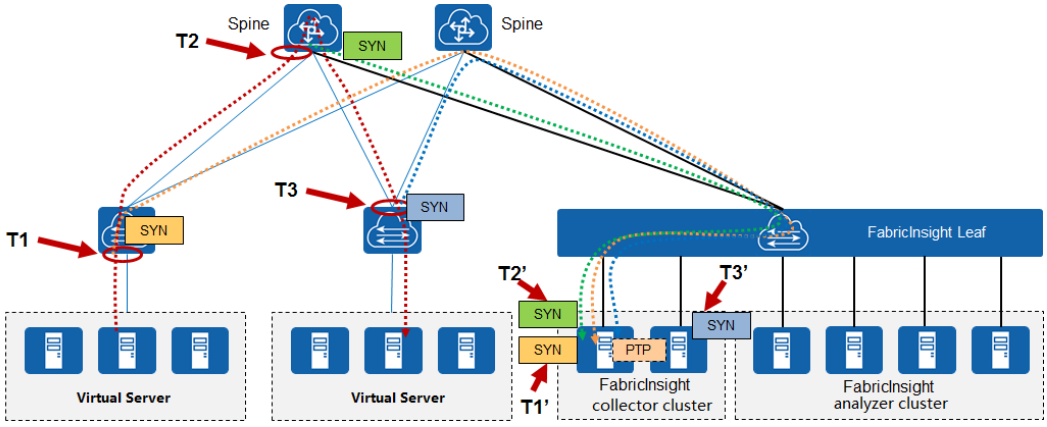
  

packets are transmitted, the calculated routes can only be accurate to device but cannot be accurate to port on the device.

## 2.2.4 Packet Transmission Latency Calculation Principle

FabricInsight mirrors TCP SYN, FIN, and RST packets in the inbound direction of the switch to the FabricInsight collector through remote flow mirroring of the switch. After adding the timestamp to the packets, the FabricInsight collector calculates the packet transmission routes and transmission latency of each hop.

**Figure 2-6** FabricInsight latency calculation principle



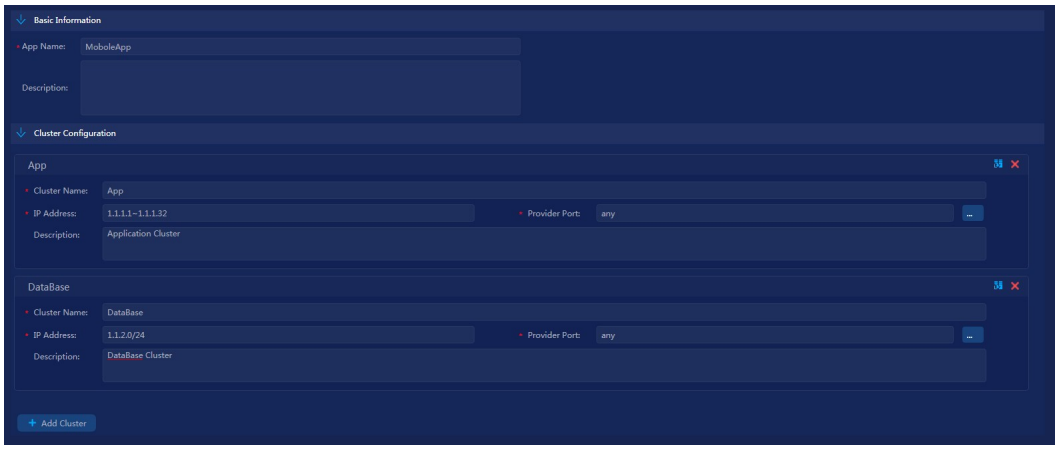

As shown in the preceding figure, after receiving SYN packets, the switch immediately sends them to the FabricInsight collector. Two FabricInsight collectors form a cluster and the collectors use the OSPF protocol to implement load balancing. The leaf switch where FabricInsight is located performs load balancing based on the IP address of mirrored packets and sends the packets to any collector in the collector cluster. The 1588v2 clock synchronization is used to ensure clock synchronization between the collector servers. A SYN packet is transmitted through three switches at time T1, T2, and T3, and three mirrored packets are generated. The three mirrored packets arrive at the collector at time T1', T2', and T3'. FabricInsight calculates the route latency as follows: T2'-T1' and T3'-T2'. However, the actual route latency is as follows: T2-T1 and T3-T2. Because the packet transmission routes and actual packet processing collectors are different, the sequence of the timestamps in the three mirrored packets may be different from the transmission sequence on the original route. As a result, the calculated route latency may be different from the actual route latency.

## 2.2.5 Application Identification Principle

When processing a reported TCP flow, FabricInsight can identify the application to which the TCP flow belongs. The application identification function is implemented based on the mapping between the application and IP address entered on the GUI. During processing, FabricInsight finds the matching application information based on the source IP address, destination IP address, and destination port. Information on the application configuration page is entered based on the following hierarchy: application > cluster > network segment. An application can be configured with multiple clusters and a cluster can also be configured with multiple network segments.

**Figure 2-7** Application information configuration page




Assume that the source IP address and destination IP address of a TCP flow is 1.1.1.11 and 1.1.2.22, respectively. According to the application information in the preceding figure, the source IP address belongs to the APP cluster and the destination IP address belongs to the database cluster. Therefore, this TCP flow is an interaction in the MobileApp application.

## 2.2.6 TCP Exception Detection Principle

FabricInsight can detect exceptions in the following table.

**Table 2-2**

| Exception type | Exception Type Description |
| --- | --- |
| Retransmission of TCP signaling packets | If the peer end of the TCP signaling packet (SYN, SYNACK, or FINACK) does not respond within a specified period, the TCP retransmission mechanism is triggered to resend the signaling packet. |
| TCP connection setup failure | SYN and SYNACK TCP retransmission times out or the server directly replies an RST packet after the client sends a SYN packet.<br>After detecting that the SYNACK packet is retransmitted, FabricInsight waits for two minutes. If FIN and RST packets are reported within two minutes, connection setup is successful. If no other packets are received within the two minutes, the SYNACK connection fails to be set up. |
| TCP RST packet | The RST is set. |
| TTL exception | The TTL value of the inner packet is smaller than 3. |
| TCP flag exception | SYN and FIN are both set.<br>SYN and RST are both set.<br>FIN, PSH, and URG are set at the same time.<br>SYN and PSH are both set.<br>FIN is set but ACK is not set. |

# 2.3 Telemetry Performance Metric Analysis

This section describes the key technical principles for analyzing performance metrics based on the GRPC protocol.

## 2.3.1 Performance Metric Collection Principle

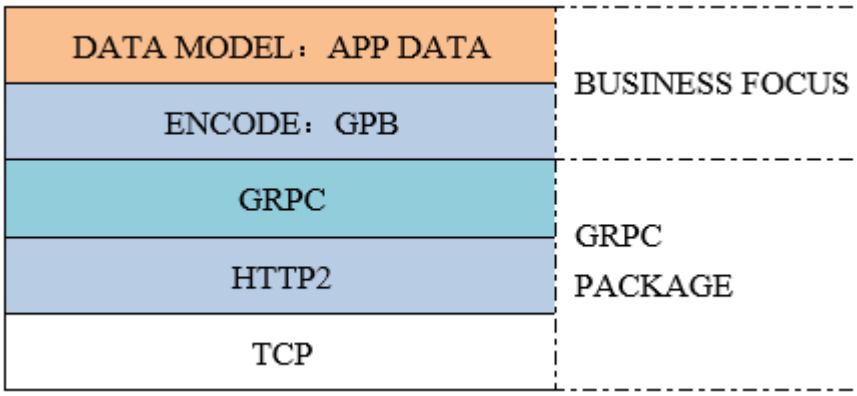
FabricInsight uses the Telemetry feature of CE switches to collect performance metrics of devices, interfaces, and queues, enabling users to actively monitor and predict network faults. The Telemetry feature uses the GRPC protocol to push data from devices to the FabricInsight collector. Before using this feature, you need to import Telemetry license on the device.

**GRPC Protocol Overview**

The protocol is a high-performance general RPC open-source software framework running over HTTP/2 protocols at the transport layer. Both communication parties perform secondary development based on the framework, so that both communication parties focus on services and do not need to pay attention to bottom-layer communication implemented by the GRPC software framework.

The following figure shows the GRPC protocol stack layers.

**Figure 2-8** GRPC protocol stack layers




The layers are described as follows:

- TCP layer: This is a bottom-layer communication protocol, which is based on the TCP connection.
- HTTP2 layer: The HTTP2 protocol carries GRPC, using HTTP2 features including bidirectional streams, flow control, header compression, and multiplexing request of a single connection.
- GRPC layer: This layer defines the protocol interaction format for remote procedure calls.
- GPB encoding layer: Data transmitted through the GRPC protocol is encoded in Google Protocol Buffers (GPB) format.
- Data model layer: Communication parties need to understand data models of each other so that they can correctly interact with each other.

Users can configure the Telemetry sampling function for a device using commands. The device then functions as a GPRC client to proactively establish a GRPC connection with the target collector and push data to the collector.

**GPB Encoding Introduction**

The GRPC protocol uses the GPB encoding format to carry data. GPB provides a mechanism for serializing structured data flexibly, efficiently, and automatically. Similar to XML and JSON, GPB is also an encoding mode. However, GPB is a binary encoding mode with good performance and high efficiency. GPB has v2 and v3 versions. Currently, devices support the v3 version.

GRPC connections are established according to the GRPC definition and messages carried by GRPC described in the .proto file. GPB uses the .proto file to describe a dictionary for encoding, that is, describing the data structure. During encoding, FabricInsight automatically generates code based on the .proto file, conducts secondary development based on the generated code, and encodes and decodes the GPB, implementing device connection and parsing of message formats defined in the .proto file.

**Service Data .proto Files**

The following table describes the service .proto files and metric sampling paths supported by FabricInsight of the current version.

**Table 2-3**

| Monitored Object | Measurement Metric | Metric Sampling Path | Earliest Device Version | Supported Device Type | Minimum Sampling Precision (FabricInsight Specifications) |
|---|---|---|---|---|---|
| Device/ Board | CPU usage | huawei-devm:devm/ cpuInfos/cpuInfo | V200R005C00 | CE6810EI/ CE6810LI/ CE6850EI/ CE6850HI/ CE6850U-HI/ CE6851HI/ CE6855HI/ CE6860EI/ CE6870EI/ CE6875EI/ CE6880EI<br>CE7850EI/ CE7855EI<br>CE8850EI/ CE8860EI<br>CE12804/ CE12808/ CE12812/ CE12816/ | 10000 ms |
| | Memory usage | huawei-devm:devm/ memoryInfos/memoryInfo | V200R005C00 | | 10000 ms |
| | FIB entry (forwarding plane) | huawei-fibstatus:fibstatus/ ipv6RouteEntryCounts/ ipv6RouteEntryCount<br>huawei-fibstatus:fibstatus/ ipv6RouteResources/ ipv6RouteResource<br>huawei-fibstatus:fibstatus/ ipRouteEntryCounts/ ipRouteEntryCount<br>huawei-fibstatus:fibstatus/ ipRouteResources/ ipRouteResource | V200R005C10 | | 10000 ms |

| Monitored Object | Measurement Metric | Metric Sampling Path | Earliest Device Version | Supported Device Type | Minimum Sampling Precision (FabricInsight Specifications) |
|---|---|---|---|---|---|
| | MAC entry (forwarding plane) | huawei-mac:mac/bdMacTotalNumbers/bdMacTotalNumber<br>huawei-mac:mac/macAddrSummarys/macAddrSummary | V200R005C10 | CE12804S/CE12808S/CE12804E/CE12808E/CE12816E | 10000 ms |
| Chip | TCAM | huawei-qos:qos/qosAclResourceUsageStats/qosAclResourceUsageStat<br>huawei-qos:qos/qosBankResourceUsageStats/qosBankResourceUsageStat | V200R005C10 | | 10000 ms |
| Interface | Number of received/sent packets, number of received/sent broadcast packets, number of received/sent multicast packets, number of received/sent unicast packets, number of received/sent bytes | huawei-ifm:ifm/interfaces/interface/ifStatistics | V200R005C00 | | 1000 ms |

| Monitored Object | Measurement Metric | Metric Sampling Path | Earliest Device Version | Supported Device Type | Minimum Sampling Precision (FabricInsight Specifications) |
|---|---|---|---|---|---|
| | Number of discarded received/ sent packets, number of received/ sent error packets | huawei-ifm:ifm/interfaces/ interface/ifStatistics | V200R005C00 | | 1 min |
| Queue | Number of Buffer bytes | huawei-qos:qos/ qosPortBufUsageStats/ qosPortBufUsageStat | V200R005C00 | | CE5880EI and CE6880EI: 2 ms<br>Others: 2100 ms |
| Packet loss behavior | Forwarding packet loss and congested packet loss | huawei-qos:qos/ qosGlobalCfgs/ qosCaptureDropstats/ qosCaptureDropstat | V200R005C00 | CE6865-48S8CQ-EI/ CE8850-64CQ-EI/ CE12800E-X | 10000 ms |

| Monitored Object | Measurement Metric | Metric Sampling Path | Earliest Device Version | Supported Device Type | Minimum Sampling Precision (FabricInsight Specifications) |
|---|---|---|---|---|---|
| Optical link | Current, voltage, temperature, and receive/transmit power | huawei-devm:devm/ports/port/opticalInfo | V200R005C00 | CE6810EI/ CE6810LI/ CE6850EI/ CE6850HI/ CE6850U-HI/ CE6851HI/ CE6855HI/ CE6860EI/ CE6870EI/ CE6875EI/ CE6880EI CE7850EI/ CE7855EI/ CE8850EI/ CE8860EI/ CE12804/ CE12808/ CE12812/ CE12816/ CE12804S/ CE12808S | 30 minutes |

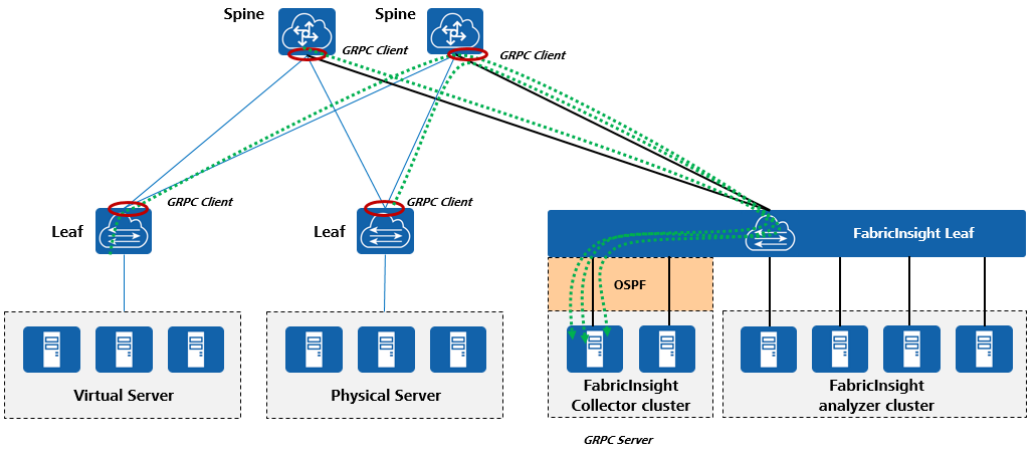
NOTE

1. For details about the models and versions of devices supporting the Telemetry feature in FabricInsight of this version, see the specifications list.
2. In this version, FabricInsight does not provide the Telemetry configuration delivery capability. That is, Telemetry sampling commands cannot be configured and delivered on FabricInsight GUI. Users need to manually configure the Telemetry sampling command on the device.

**Networking**

After the Telemetry performance metric subscription rule is configured on CE switches, the switches collect metric data based on the specified period and send the data to FabricInsight for analysis. The following figure shows the networking.

**Figure 2-9** Networking for collecting Telemetry performance metric data




The collector cluster advertises OSPF VIP routes externally. Devices report Telemetry performance metric data and ERSPAN mirrored packets by using the VIP routes as the destination address. The collector cluster receives data packets through the DPDKCollector process. The DPDKCollector process parses the packet header and distributes packets to the backend agent for parsing based on the packet type.

**Configuration Using Commands**

In this version, FabricInsight does not provide the Telemetry configuration delivery capability. Users need to manually configure the Telemetry sampling command on the device. The following describes how to collect performance data of the Ethernet3/0/0 interface at an interval of 1 minute and report the data to the collector.

**Step 1** Enable the Telemetry function.

<HUAWEI> system-view

[~HUAWEI] telemetry

[*HUAWEI-telemetry] sample enable

**Step 2** Configure a sampling task group and configure data paths to be sampled in the task group.

Add a sampling path in the sensor-group view. The filter criterion in the square brackets ([]) indicates that only Ethernet3/0/0 is subscribed.

[*HUAWEI-telemetry] sensor-group test

[*HUAWEI-telemetry-sensor-group-test] sensor-path huawei-ifm:ifm/interfaces/interface[ifName=Ethernet3/0/0]/ifStatistics

[*HUAWEI-telemetry-sensor-group-test] commit

[*HUAWEI-telemetry-sensor-group-test] quit

**Step 3** Configure a data sending destination group and configure a destination address to where the data is sent in the destination group. If the OSPF VIP advertised by the collector cluster is 1.1.1.1, the default listening port number of the collector GRPC packet receiving process is 30001.

[*HUAWEI-telemetry] destination-group test

[*HUAWEI-telemetry-destination-group-test] ipv4-address 1.1.1.1 port 30001 protocol grpc no-tls

[*HUAWEI-telemetry-destination-group-test] commit

[*HUAWEI-telemetry-destination-group-test] quit

**Step 4** Configure a subscription, that is, associate the sampling task group, one-minute sampling interval, and data sending target to trigger sampling.

[*HUAWEI-telemetry] subscription test

[*HUAWEI-telemetry-subscription-test] sensor-group test sample-interval 60000

[*HUAWEI-telemetry-subscription-test] destination-group test

[*HUAWEI-telemetry-subscription-test] commit

[~HUAWEI-telemetry-subscription-test] quit

**----End**

## 2.3.2 Dynamic Baseline Calculation Principle

FabricInsight predicts baselines for metrics including the device CPU/memory usage and number of interface received/sent packets through AI algorithms such as time sequence data feature decomposition and aperiodic sequence Gaussian fitting algorithms. Compared with the static threshold in the traditional NMS domain, the dynamic baseline is based on the historical data of a period of time and works with the anomaly detection algorithm based on the dynamic baseline to precisely detect metric deterioration on the network in advance. In the current version, FabricInsight establishes baselines for the CPU/memory usage of all connected CE devices, baselines for routing entries such as the ARP, FIB, and MAC entries, and baselines for metrics such as number of received/sent packets of interfaces with physical links.

The details are as follows.

**Table 2-4**

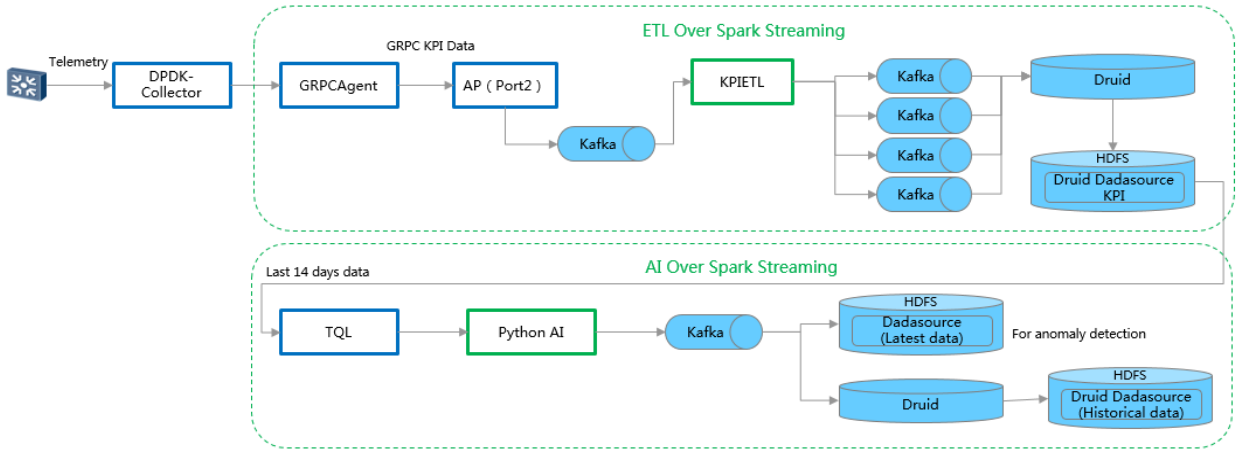
| Monitored Object | Metric with Default Baseline | Monitored Object Scope with Default Baseline | Maximum Period of Historical Training Data | Baseline Calculation Period | Baseline Retention Period |
|---|---|---|---|---|---|
| Device/ Board | CPU usage | All connected devices with Telemetry performance metric reporting enabled | Last 14 days | 1 day | One month |
| | Memory usage | | | | |

| Monitored Object | Metric with Default Baseline | Monitored Object Scope with Default Baseline | Maximum Period of Historical Training Data | Baseline Calculation Period | Baseline Retention Period |
|---|---|---|---|---|---|
| Chip | ARP entries, FIBv4/ FIBv6 routing entries, and MAC routing entries | All connected devices with Telemetry performance metric reporting enabled | Last 14 days | 1 day | One month |
| Interface | Number of received/ sent packets. | All connected devices with Telemetry performance metric reporting enabled and all interfaces with physical links on the devices | Last 14 days | 1 day | One month |
| | Number of received/ sent error packets. | | | | |
| | Number of discarded received/ sent packets. | | | | |
| | Number of received/ sent broadcast packets | | | | |

According to the table, the dynamic baseline is calculated every other day in offline mode. The predicted baseline of the next day is calculated at a time. The granularity of the generated dynamic baseline data is the same as that of the raw data. For devices, boards, and interfaces, the minimum data granularity of dynamic baseline data is one minute. The dynamic baseline is calculated based on the Spark Streaming framework. The following figure shows the complete data flow diagram.

**Figure 2-10** Data flow diagram for calculating the dynamic baseline



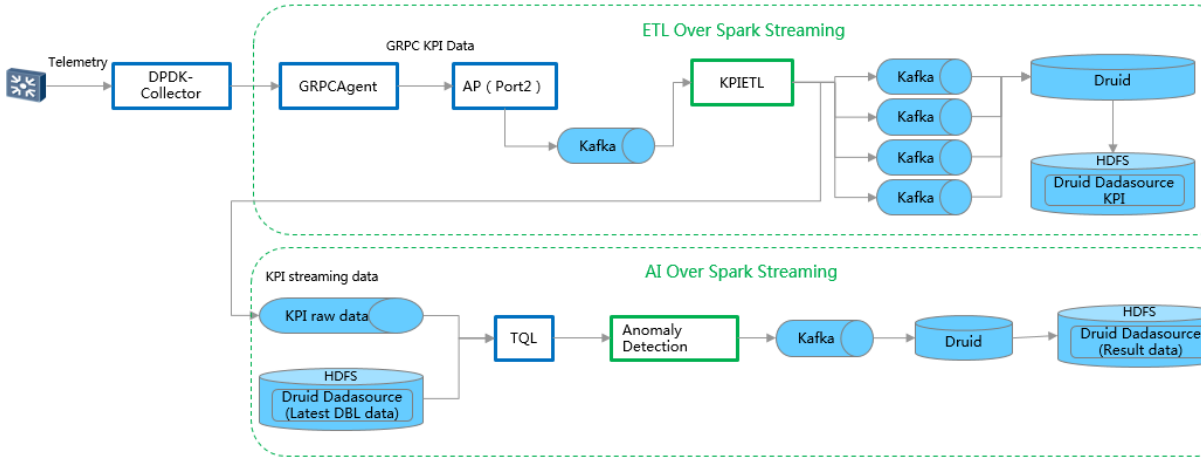

The dynamic baseline is based on the Spark Streaming offline computing framework. The framework periodically obtains the training data of a certain period from the Druid table of storage performance metrics through periodic tasks to predict the baseline. The AI operator is implemented based on Python and is responsible for data set preprocessing and dynamic baseline prediction. The operator depends on the Spark Streaming framework for distributed computing. After completing the calculation, the AI operator exports the baseline data of the next day to the Kafka queue of the specified topic based on the predefined data format. To improve the efficiency and quasi-real-time performance of baseline exception detection, baseline data in Kafka queues must be sliced by hour and written into HDFS as the data source for baseline exception detection in addition to be persisted in Druid.

## 2.3.3 Baseline Exception Detection Principle

Abnormal data needs to be displayed in quasi-real time. Therefore, different from offline calculation of dynamic baseline, baseline exception detection uses the real-time calculation framework based on the Spark Streaming. Real-time computing refers to the process of directly consuming metric data cleaned from the KPIETL and detecting exceptions based on the dynamic baseline of the last day. Therefore, the granularity of exception detection data is the same as that of original performance metric data. For devices, boards, and interfaces, the minimum granularity of baseline exception data is one minute. By default, FabricInsight performs exception detection calculation on metrics with dynamic baselines. The following figure shows the complete data flow diagram.

**Figure 2-11** Data flow diagram for calculating baseline exception detection




As shown in the preceding figure, dynamic baseline exception detection depends on the input of two data sources.

- Performance metrics of the original granularity: Metrics requiring dynamic baselines in output data of KPIETL cleaning need to be imported to the Druid and another topic. This topic is the data input of the baseline exception calculation framework.
- Predicted dynamic baseline on the current day: The data is generated by the dynamic baseline offline calculation framework and is sliced by hour. For details, see the previous section. Before submitting data to the Spark Streaming computing framework, the exception detection task obtains the corresponding dynamic baseline time slice data based on the timestamp of the original performance data, and submits the data to the computing framework after TQL Join.

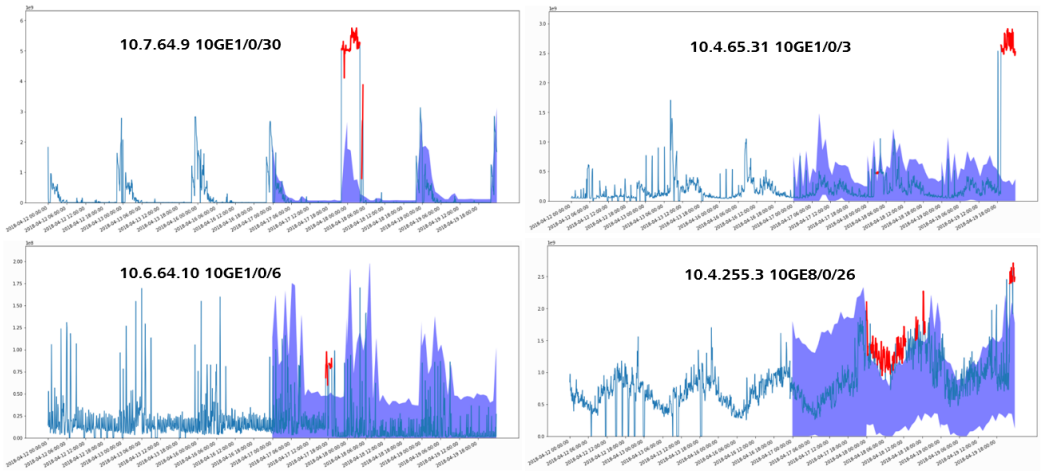
The core logic of exception detection is also implemented by the Python operator. The Spark Streaming framework is used for distributed calculation. The operator executes the following logic:

- Point-by-point data comparison: Check whether the original data exceeds the baseline by the granularity of period.
- Identification and counting of consecutive out-of-range data: Check whether the out-of-range data is in consecutive periods and record the number of consecutive periods when out-of-range data is generated.
- Alarm suppression and combination: Suppress alarms based on specified rules to prevent excessive redundant baseline data from being generated. By default, a baseline exception is recorded only when the baseline is exceeded for three consecutive periods. In addition, the system automatically combines these out-of-baseline records into one record and the baseline exception record imported into the database contains the start time and end time of the exception.
- Output of the final baseline exception data: Write the calculation result to the storage exception Druid table.

The following figure shows the simulation result of the periodic sequence exception detection algorithm.

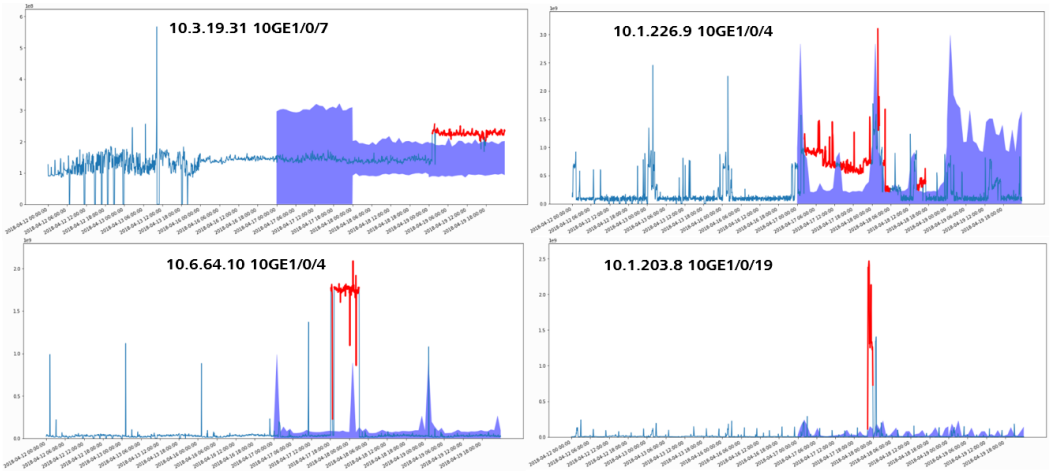
**Figure 2-12** Simulation result of the periodic sequence anomaly detection algorithm for interface traffic




As shown in the figure, the blue line indicates the raw performance metric data, the blue shadow area indicates the dynamic baseline prediction data of the interface traffic of the measured object, and the red line indicates the baseline exception data detected based on the specified rules.

The following figure shows the simulation result of the non-periodic sequence exception detection algorithm.

**Figure 2-13** Simulation result of the non-periodic sequence anomaly detection algorithm for interface traffic




As shown in the figure, the blue line indicates the raw performance metric data, the blue shadow area indicates the dynamic baseline prediction data of the interface traffic of the measured object, and the red line indicates the baseline exception data detected based on the specified rules.

# 2.4 Issue Analysis and Troubleshooting Analysis

FabricInsight performs big data analytics on collected ERSPAN flows and Telemetry performance metrics through real-time and offline computing. In addition, FabricInsight

proactively detects possible issues on a fabric based on AI algorithms such as baseline anomaly detection and multi-dimension cluster analysis, and intelligently analyzes and identifies whether a network or an application has issues that occur on a large scale. For service connectivity issues, FabricInsight orchestrates troubleshooting procedures to support one-click automatic troubleshooting. All these functions help users achieve proactive and intelligent O&M including proactive issue detection and minute-level issue locating and demarcation.

Based on the actual O&M scenarios of customers, FabricInsight collects issues to and analyzes issues in the issue case library on live networks of the customers, and summarizes more than 10 typical issue scenarios in terms of the application quality, network service, and security compliance. In addition, FabricInsight proactively analyzes and identifies issues in different scenarios. If an issue is detected, FabricInsight automatically generates an alarm. Users can configure remote alarm notification rules to detect issues in real time.

# 2.4.1 Application Quality

Application quality issues are mainly used to proactively identify applications with abnormal interaction behaviors, for example, sessions that fail to set up TCP connections continuously and sessions that are intermittently disconnected repeatedly during connection setup. For these issues, FabricInsight orchestrates related troubleshooting procedures based on different issue patterns and provides the automatic troubleshooting capability. Users can perform one-click troubleshooting on the GUI. FabricInsight analyzes the result of each troubleshooting step and provides the final troubleshooting conclusion. Operations on the GUI are simple, and the troubleshooting result is clear, which greatly reduces the time required for issue demarcation and locating. The following sections describe the application scenarios, issue identification principles, and constraints for different issue patterns.

## 2.4.1.1 Continuous Service Interruption

**Application Scenario**

Users need to identify sessions (triplet of the source IP address, destination IP address, and destination port, which is hereinafter referred to as triplet) with continuous TCP connection setup failures on a network, and use the AI clustering algorithm to analyze whether an issue occurs on a large scale on the network or an application.

Sessions with continuous TCP connection setup failures include:

1. Sessions for which TCP connection setup never succeeds
2. Sessions for which TCP connection setup succeeds but then always fails

**Issue Identification Principle**

FabricInsight calculates sessions with continuous TCP connection setup failures on the network in real time based on the Spark Streaming framework. Then, FabricInsight uses dynamic baselines and real-time anomaly detection technologies to identify the time when the number of failures increases sharply and identify sessions with burst continuous TCP connection setup failures. Finally, FabricInsight analyzes whether an issue occurs on a large scale on the network or applications based on the data.

**Step 1** FabricInsight calculates sessions with continuous TCP connection setup failures on the network in offline mode. The statistical time range is from 00:00 on the natural day of the current Spark computing task to the current task execution time. The measured objects include sessions for which TCP connection setup never succeeds and sessions for which TCP

connection setup succeeds but then always fails. The offline task calculation period is 5 minutes.

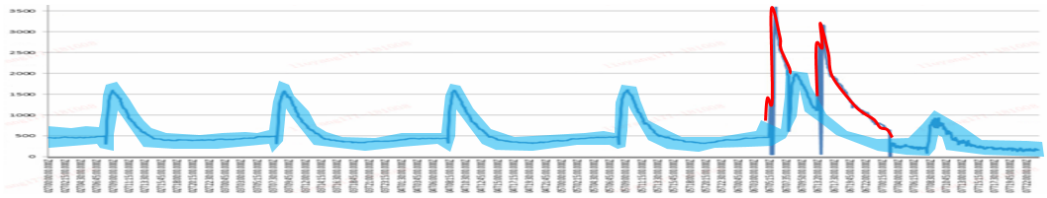
**Step 2** FabricInsight creates dynamic baselines every 5 minutes based on the number of sessions calculated in step 1, and detects the time when the number of sessions exceeds the baseline. For details about dynamic baseline and anomaly detection, see **2.3 Telemetry Performance Metric Analysis**.

**Step 3** FabricInsight analyzes new sessions (triplet) with continuous TCP connection setup failures at the exception time points, and calculates the information entropy in terms of the source IP address, destination IP address, source IP+destination IP address, and destination IP +destination port. Then, FabricInsight recommends analysis dimensions for users based on the entropy calculation result.

**Step 4** FabricInsight performs multi-dimension cluster analysis on the analysis result in step 3 to determine whether the issue occurs individually or on a large scale. If the issue is a new issue that occurs on a large scale, FabricInsight generates an issue alarm, prompting the user to solve the issue in time.

**----End**

**Figure 2-14** Dynamic baseline and anomaly detection simulation for the number of sessions with continuous TCP connection setup failures on the live network of a customer



The figure shows the trend of the number of sessions with continuous TCP connection setup failures on the live network of a customer for six consecutive days. The simulation system creates a dynamic baseline based on the data and performs anomaly detection. The blue line indicates the dynamic baseline and the red line indicates the exception curve. As shown in the figure, the number of sessions with continuous TCP connection setup failures is nearly doubled at the exception time point. Based on the preceding issue identification steps, FabricInsight performs cluster analysis on new abnormal sessions at the exception time point to check whether an issue occurs on a large scale.

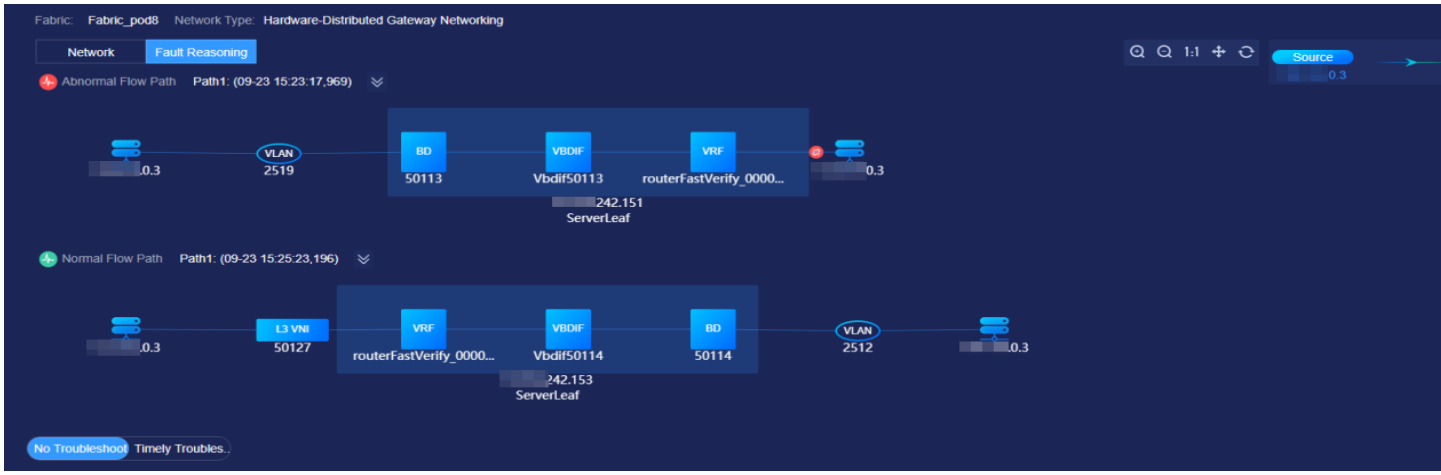
**Automatic Troubleshooting Principle**

There are many possible causes for continuous service interruption issues. These issues may be caused by the network or applications. Based on the expert experience library and troubleshooting process, FabricInsight summarizes a unified troubleshooting model and provides an automatic troubleshooting framework that can be orchestrated and requires no manual intervention. Troubleshooting actions involve check on the network. Users can perform one-click troubleshooting, improving the troubleshooting efficiency.

| Troubleshooting Object | Possible Cause | Action | Automatic/ Manual | Principle |
| --- | --- | --- | --- | --- |
| Abnormal flow path | Faulty edge devices | Check the forwarding types (Layer 3 or Layer 2 forwarding) and the next hop type (VXLAN tunnel, VAS, or VM) of the breakpoint edge device. | Automatic | Query historical data of normal and abnormal ERSPAN flows on the destination host, create a logical topology at the overlay layer, and compare the logical topology of normal and abnormal ERSPAN flows to identify the faulty edge device and breakpoint scenario. If no normal flow path exists, the leaf node at the last hop of the abnormal flow is regarded as the faulty edge device. |
| Breakpoint device | Configuration change, ARP/ route loss, route flapping, interface going Down, outbound interface link congestion, M-LAG route flapping, firewall policy blocking, IP address conflict, ARP entry overrun, ARP attack, MAC address flapping, MAC entry overrun, loop, and soft error | Compare configuration files, entries, logs, and VXLAN status in the normal and abnormal periods. | Automatic | Analyze information including configuration files, entries, logs, and VXLAN status about the faulty edge device collected when faults occur and collected in normal situations by FabricInsight to identify the cause of the failure to associate the source and destination IP addresses. |

**Figure 2-15** Details and automatic troubleshooting GUI of an issue




NOTE

1. To improve the accuracy of automatic troubleshooting, users need to set the fabric networking type on the fabric resource management page.
2. Troubleshooting can be performed only when abnormal flow paths exist.
3. Automatic troubleshooting depends on configuration file changes, device logs, entry information (such as ARP and FIB entries), VXLAN status, and firewall policy information. The following requirements must be met to improve the accuracy of automatic troubleshooting results:

   a. For the troubleshooting items for which configuration file changes need to be verified, configure the devices to report configuration file change alarms to FabricInsight.

   b. For the troubleshooting items for which device logs need to be verified, configure the devices to send Syslogs to FabricInsight.

   c. For the troubleshooting items for which entry information and VXLAN status need to be verified, FabricInsight needs to connect to devices through Telnet or STelnet to obtain the information. Therefore, users need to enable Telnet or STelnet on the devices and set Telnet or STelnet connection parameters on FabricInsight in advance.

   d. For the troubleshooting items for which firewall policies need to be verified, FabricInsight needs to connect to devices through the NETCONF protocol to obtain the firewall security policy information. Therefore, users need to assign the permission to connect to firewalls through NETCONF for FabricInsight and set NETCONF connection parameters on FabricInsight in advance.

   e. In NAT mapping-related troubleshooting scenarios, import the NAT mapping relationship on the **Inventory** > **Network Resource** > **NAT** page.

### 2.4.1.2 Intermittent Service Interruption

**Application Scenario**

Users need to identify sessions (triplet) with intermittent TCP connection setup failures on a network, and use the AI clustering algorithm to analyze whether an issue occurs on a large scale on the network or applications. Intermittent TCP connection setup failure refers to that TCP connection setup succeeds occasionally for sessions with a specific triplet but sometimes repeatedly fails.

**Issue Identification Principle**

FabricInsight calculates the sessions with intermittent TCP connection setup failures on the network in real time based on the Spark Streaming framework. Then, FabricInsight uses dynamic baselines and real-time anomaly detection technologies to identify time points when the failures increase sharply and sessions with intermittent connection setup failures. This issue identification process is similar to that of continuous service interruption issues. Finally, FabricInsight analyzes whether an issue occurs on a large scale on the network or applications based on the data.

**Step 1** FabricInsight collects statistics on sessions with intermittent TCP connection setup failures on the network in real time. The statistical time range is from 00:00 on the natural day of the current Spark computing task to the current task execution time. The measured objects include sessions with intermittent TCP connection setup failures. The real-time task calculation period is 1 minute.

**Step 2** FabricInsight creates dynamic baselines every minute for the number of sessions calculated in step 1, and detects the time points when the number of sessions exceeds the baseline. For details about the dynamic baseline and anomaly detection principles, see **2.3 Telemetry Performance Metric Analysis**.

**Step 3** FabricInsight analyze new sessions (triplet) with intermittent TCP connection setup failures at the exception time points, and calculates the information entropy in terms of the source IP

address, destination IP address, source IP+destination IP address, and destination IP +destination port. Then, FabricInsight recommends analysis dimensions for users based on the entropy calculation result.

**Step 4** FabricInsight performs multi-dimension cluster analysis on the analysis result in step 3 to determine whether the issue occurs individually or on a large scale. If the issue is a new issue that occurs on a large scale, FabricInsight automatically generates an issue alarm, prompting the user to solve the issue in time.

**----End**

**Troubleshooting Principle**

For details, see **2.4.1.1 Continuous Service Interruption**.

### 2.4.1.3 Host Port Not Listened On

**Application Scenario**

Users need to identify hosts that fail to respond to some services. That is, application ports on hosts send RST packets to respond to TCP connection setup requests. This issue occurs because TCP listening is not enabled on ports.

**Issue Identification Principle**

**Step 1** FabricInsight collects statistics on hosts with the following features on the network in real time:

1. A host port sends RST packets to respond to TCP connection setup requests.
2. The last hop of a flow is a server leaf node.

**Step 2** FabricInsight collects further statistics on unreachable application ports based on the calculation result in step 1, and generates issue data.

**----End**

## 2.4.2 Network Services

Network service issues are used to proactively identify whether entry usage of the network device forwarding plane on the fabric is abnormal. For example, FIB route forwarding entries are insufficient or change sharply. For such issues, FabricInsight trains the dynamic baseline based on the static threshold or entry usage historical data to proactively identify exceptions in real time. In addition, FabricInsight can display the forwarding entry usage snapshot at the exception time point. For example, if the FIB entry usage is abnormal, FabricInsight allows users to view the resource usage of each VRF instance at the exception time point, enabling users to quickly analyze whether VRF instances with abnormal behavior exist.

### 2.4.2.1 Insufficient TCAM Resources

**Application Scenario**

Users need to identify devices with insufficient TCAM resources on the network, locate the specific board, chip, stage, or resource type (slice, rules, meter, and counter), and view the TCAM resource usage of each service at the exception time point.

**Issue Identification Principle**

FabricInsight collects the TCAM resource usage of devices through Telnet (STelnet) or gRPC, and compares the collected data with the static threshold to check whether the TCAM resources are sufficient. In addition, FabricInsight collects details about TCAM resources used by each service at the exception time point.

**Troubleshooting Suggestions**

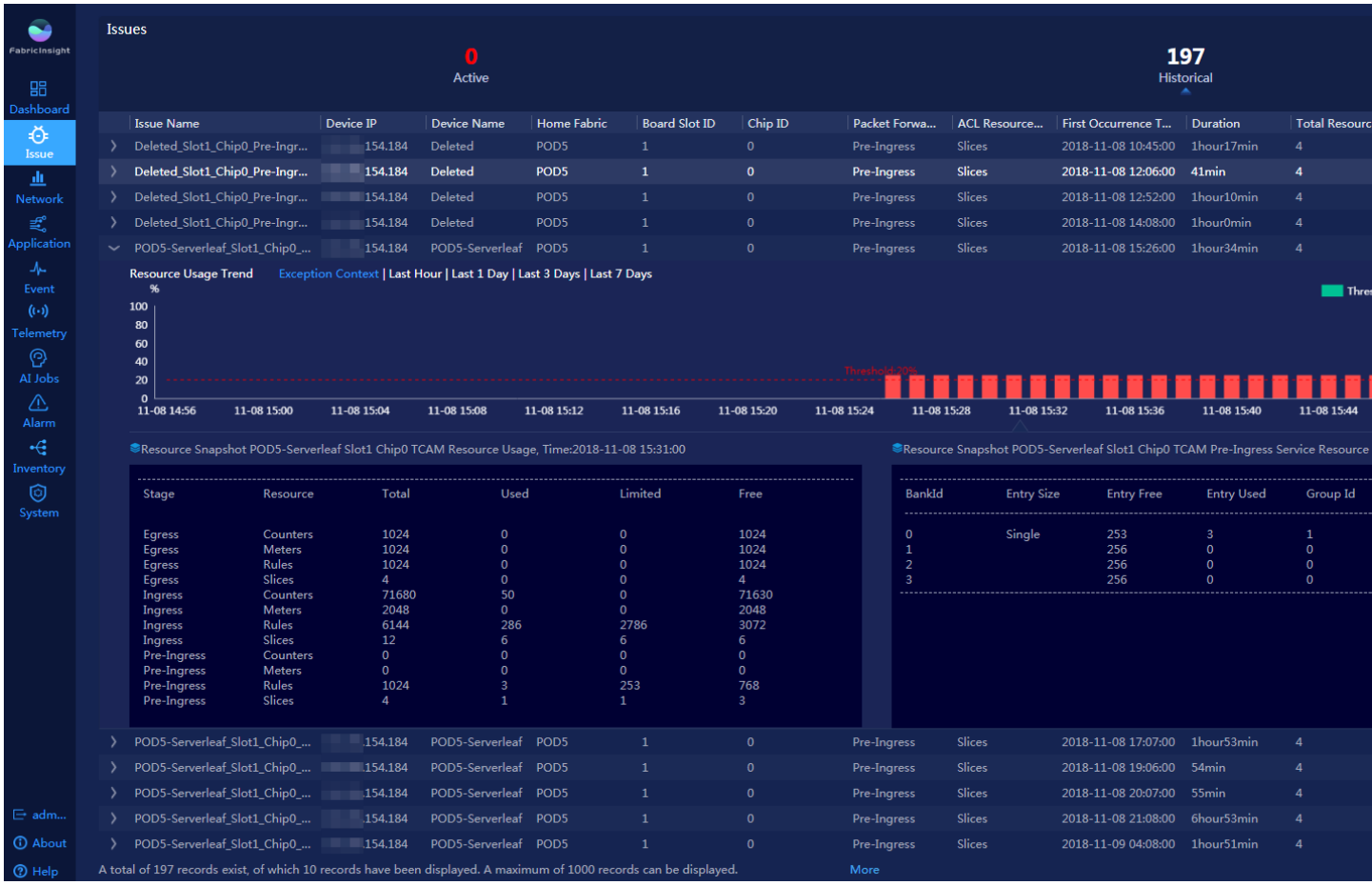
**Step 1** On the **Insufficient TCAM Resources** tab page (**Figure 2-16**), view active Insufficient TCAM Resources issues and check the chip, board, device, and resource type (slice, rule, meter, and counter).

**Step 2** View the snapshot of the TCAM resource usage at the exception time point and check the TCAM resource usage of each service.

**Step 3** Solve the issue based on the TCAM resource usage of each service.

**----End**

**Figure 2-16** Insufficient TCAM Resources tab page




## 2.4.2.2 Insufficiency or Sharp Change of FIB Entry Resources

**Application Scenario**

Users need to identify devices where FIB entry resources are insufficient or the usage of FIB entries changes sharply on the network, which is accurate to the specific board, chip, and resource type (V4, V6, V6 64, V6 [65, 128), and V6 128).

**Issue Identification Principle**

FabricInsight collects the FIB entry resource usage and total number of FIB entry resources of devices through Telnet (STelnet) or GRPC. By comparing the collected data with the static threshold or training the dynamic baseline based on historical entry usage data, FabricInsight proactively identifies issues that resources are insufficient or resource usage changes sharply in real time. In addition, FabricInsight collects the detailed FIB entry resource usage of each VRF instance at the exception time point.

**Troubleshooting Suggestions**

**Step 1** On the issue page of insufficiency or sharp change of FIB entry resources, view the active issues and check the chip, board, device, and resource type (V4, V6, V6 64, V6 [65, 128), and V6 128) of the issues.

**Step 2** View the snapshot of the FIB entry resource usage at the exception time point, and check the resource usage details of each VRF instance.

**Step 3** Solve the issue based on the resource usage of each VRF instance.

**----End**

### 2.4.2.3 Insufficiency or Sharp Change of ARP Entry Resources

**Application Scenario**

Users need to identify devices where ARP entry resources are insufficient or the resource usage changes sharply on the network, which is accurate to the specific board and chip.

**Issue Identification Principle**

FabricInsight collects the ARP entry resource usage and total number of ARP entry resources of devices through Telnet (STelnet). By comparing the collected data with the static threshold or training the dynamic baseline based on historical entry usage data, FabricInsight proactively identifies issues that resources are insufficient or resource usage changes sharply in real time. In addition, FabricInsight collects the detailed ARP entry resource usage at the exception time point.

**Troubleshooting Suggestions**

**Step 1** On the issue page of insufficiency or sharp change of ARP entry resources, view active issues and check the chip, board, and device of the issues.

**Step 2** View the snapshot of the ARP entry resource usage at the exception time point, and check the resource usage details of each VPN instance.

**Step 3** Solve the issue based on the resource usage of each VPN instance.

**----End**

### 2.4.2.4 Insufficiency or Sharp Change of MAC Entry Resources

**Application Scenario**

Users need to identify devices where MAC entry resources are insufficient or the resource usage changes sharply on the network, which is accurate to the specific board.

**Issue Identification Principle**

FabricInsight collects the MAC entry resource usage and total number of MAC entry resources of devices through Telnet (STelnet) or GRPC. By comparing the collected data with the static threshold or training the dynamic baseline based on historical entry usage data, FabricInsight proactively identifies issues that resources are insufficient or resource usage changes sharply in real time. In addition, FabricInsight collects the MAC entry details at the exception time point.

**Troubleshooting Suggestions**

**Step 1** On the issue page of insufficiency or sharp change of MAC entry resources, view active issues and check the board and device of the issues.

**Step 2** View the snapshot of the MAC entry resource usage at the exception time point, and check the MAC entry resource usage details.

**Step 3** Provide a repair solution based on the MAC entry resource usage details.

**----End**

**Table 2-5** Summary and comparison of network service entry resource issues

| Issue Type | Data Collection Protocol | Data Collection Period | Issue Identification Mode | Support View of Resource Usage Snapshot at Exception Time Point |
|---|---|---|---|---|
| Insufficient TCAM Resources | Telnet/STelnet/ GRPC (GRPC is used by default.) | Telnet/STelnet: 5 minutes, 15 minutes, 30 minutes, or 1 hour (default)<br>GRPC: 1 minute | Static threshold | Supported |
| Insufficiency or Sharp Change of FIB Entry Resources | Telnet/STelnet/ GRPC (GRPC is used by default.) | Telnet/STelnet: 5 minutes, 15 minutes, 30 minutes, or 1 hour (default)<br>GRPC: 1 minute | Static threshold + dynamic baseline | Supported |
| Insufficiency or Sharp Change of ARP Entry Resources | Telnet/STelnet | Telnet/STelnet: 5 minutes, 15 minutes, 30 minutes, or 1 hour (default) | Static threshold + dynamic baseline | Supported |
| Insufficiency or Sharp Change of MAC Entry Resources | Telnet/STelnet/ GRPC (GRPC is used by default.) | Telnet/STelnet: 5 minutes, 15 minutes, 30 minutes, or 1 hour (default)<br>GRPC: 1 minute | Static threshold + dynamic baseline | Supported |

# 2.4.3 Security Compliance

Security compliance issues are used to proactively identify potential SYN flood attacks, port scanning attacks, and non-compliant TCP sessions on the fabric. In attack scenarios, FabricInsight comprehensively analyzes related data and identifies the location of the suspected attack source, for example, the first device that the attack source SYN packet passes through or the real host where the attack source is located. This helps users check whether the attack is initiated from the external network or internal network. For non-compliant TCP sessions, FabricInsight identifies abnormal sessions based on rules configured by users, helping users audit non-complaint traffic.

## 2.4.3.1 Non-compliant Traffic Interaction

FabricInsight uses the ERSPAN technology to collect all interacting TCP sessions on the network. The captured TCP sessions are not repudiated. There are various service isolation scenarios. For example, the network administrator can restrict that two service departments cannot communicate with each other. With modern network technologies, security isolation of services can be implemented through multiple methods, for example, a security blocking policy can be configured on the firewall. If the configuration is missing or tampered with by mistake, service isolation fails and non-compliant service interaction occurs. Traditional O&M methods can hardly identify the non-compliant traffic. On contrast, FabricInsight can analyze ERSPAN packets to identify and collect statistics on the non-compliant traffic.

**Application Scenario**

Users need to identify TCP sessions with non-compliant interaction on the network.

**Issue Identification Principle**

FabricInsight calculates non-compliant sessions meeting the configured rules in real time based on the Spark Streaming framework. Then, FabricInsight aggregates and displays issues by the rule.

**Step 1** Users manually create non-compliant session matching rules on the rule setting page of this type of issue. FabricInsight allows users to create multiple rules at the same time. In each rule, users need to configure source objects and destination objects that are not supposed to have TCP interaction. Source and destination objects can be flexibly set based on IP address segments. Users can also select the entered application models.

**Step 2** FabricInsight calculates whether the ERSPAN packets meet the non-compliant session matching rules configured in step 1 in real time. The calculation period of the Spark task is 10 seconds.

**Step 3** FabricInsight aggregates the calculation results generated in step 2 by the rule granularity and generates issues.

**----End**

**Troubleshooting Suggestions**

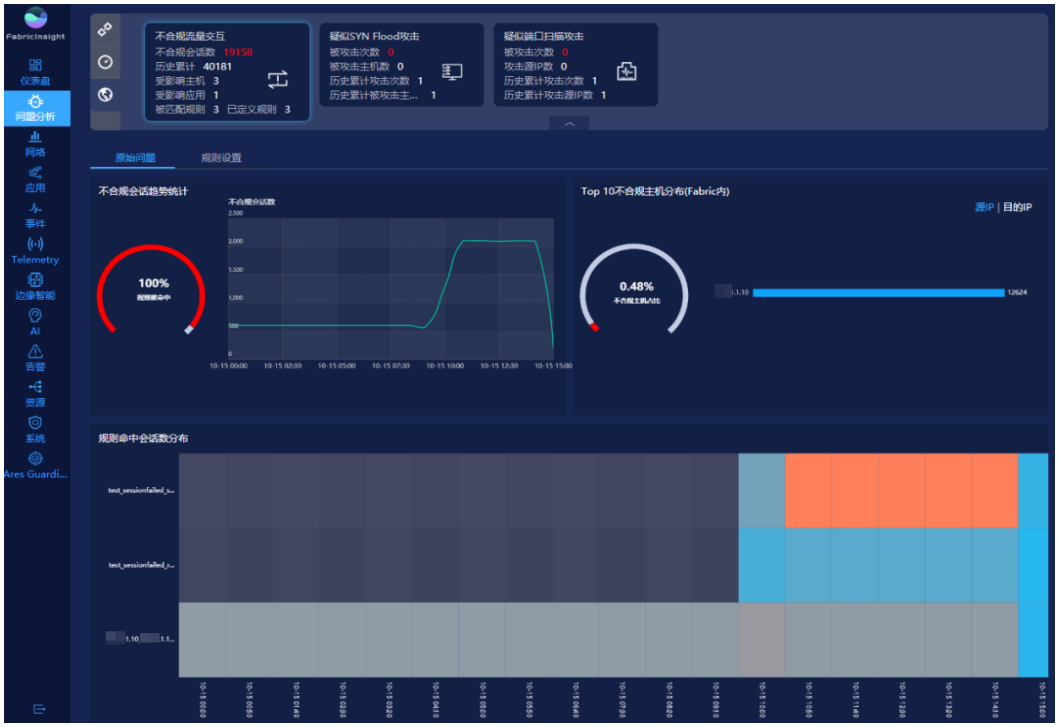
**Step 1** On the issue page (**Figure 2-17**), check the specific rules, sessions with non-compliant interaction, and non-compliant session trend.

**Step 2** Click a rule to view details, for example, non-compliant session trend and top non-compliant session (triplet/2-tuple of the source IP address and destination IP address) distribution of the rule to locate the specific host.

**----End**

**Figure 2-17** Non-compliant Traffic Interaction tab page




NOTE

1. By default, FabricInsight does not preset any non-compliant session matching rules. Users are advised to create rules based on actual O&M scenarios.
2. To ensure real-time issue calculation efficiency, FabricInsight does not allow users to configure both the source IP address and destination IP address using wildcards (*).
3. A maximum of 20 abnormal session matching rules can be created.
4. Users are not allowed to configure a rule repeatedly.

## 2.4.3.2 Suspicious SYN Flood Attack

### Application Scenario

Users need to identify possible TCP SYN flood attacks on the network, analyze the impacts that the attack behavior has on the target host, and locate the attack source.

### Issue Identification Principle

FabricInsight calculates whether TCP SYN flood attacks exist on the network in real time based on the Spark Streaming framework, and calculates the attack source location based on the actual packet forwarding path.

**Step 1** FabricInsight calculates the ERSPAN packets in real time and checks whether the destination host meets the SYN flood attack rate threshold. Users can adjust the default threshold on the issue setting page. The threshold conditions are as follows:

1. The TCP half-connection request rate of the destination host reaches a threshold. The TCP half-connection refers to that the destination host responds with a SYN ACK packet after receiving a SYN packet from the source IP address but receives no ACK packet from the source IP address. As a result, the TCP connection cannot be set up successfully. If the destination host has a large number of TCP half-connections, the half-

connection queue resources of the TCP protocol stack in the operating system will be used up, and the host cannot respond to other normal session requests.

2. The TCP connection request rate of the destination host reaches a threshold. Normally, the TCP SYN packets received by the host on the fabric are relatively stable. If the number of TCP connection requests received by a host reaches a high threshold at a certain time, the host may suffer from SYN flood attacks.

If either of the preceding conditions is met, a suspected SYN flood attack is identified. The Spark task calculation period is 10 seconds.

**Step 2** FabricInsight check whether a destination host meets the SYN flood attack threshold. Once the destination host meets the SYN flood attack rate threshold, FabricInsight identifies a suspected SYN flood attack issue and records information such as the attacked host, attack time, and attack duration.

**----End**

**Troubleshooting Suggestions**

The SYN flood attack source usually uses a large number of forged IP addresses to launch attacks. Once an attack occurs, network O&M personnel can hardly trace the attack source based on the forged IP addresses. FabricInsight analyzes the original packets, extracts original attack packets from a large number of packets, and restores the paths of these attack packets. By collecting statistics on the first-hop device of attack packets, FabricInsight can identify the network access location of the attack source, which greatly improves the efficiency for locating the attack source host.

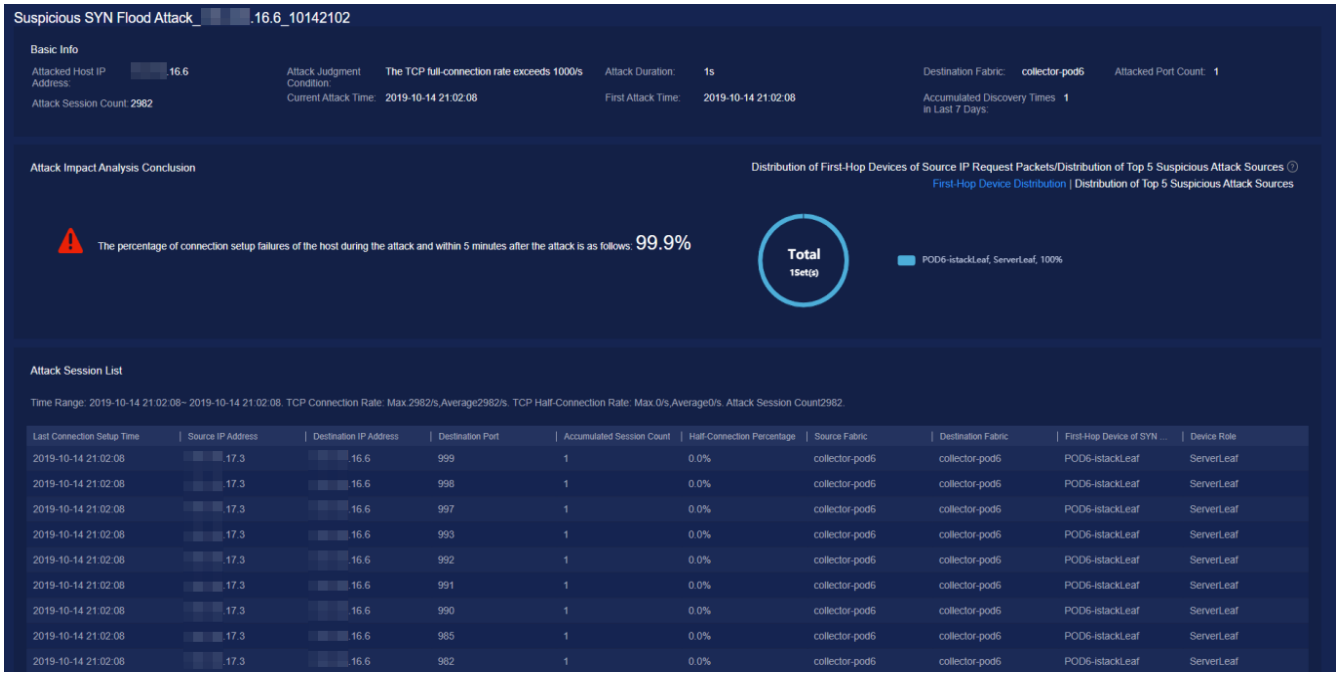
**Step 1** On the issue page, view the issue list and check hosts suffering from SYN flood attacks.

**Step 2** Click an issue in the issue list. On the issue details page that is displayed, check the distribution of first-hop devices of the TCP SYN packets from the attack source.

1. If the first-hop devices are mainly BorderLeaf devices, the attack source is a host out of the fabric.
2. If the first-hop devices are mainly ServerLeaf devices, the attack source is a host on the fabric. In this case, users only need to further check hosts connected to these ServerLeaf devices to locate the attack source.

**----End**

**Figure 2-18** SYN flood attack issue details page




NOTE

1. To accurately locate the attack source, users need to configure ERSPAN mirroring for devices such as ServerLeaf and BorderLeaf devices on the fabric.
2. If ServerLeaf and BorderLeaf devices support the ERSPAN enhancement feature, it is recommended that ERSPAN enhancement be enabled when configuring ERSPAN. In this case, users can use FabricInsight to check the ingress port on the first-hop device of the attack packets, further narrowing down the attack source host scope.

## 2.4.3.3 Suspicious Port Scanning Attack

### Application Scenario

Users need to identify possible TCP port scanning attacks on the network and analyze and locate the attack source.

### Issue Identification Principle

FabricInsight calculates whether TCP port scanning attacks exist on the network in real time based on the Spark Streaming framework, and calculates the attack source location based on the actual packet forwarding path.

**Step 1** FabricInsight calculates the ERSPAN packets in real time and analyzes whether the TCP packets sent by a source IP address meets the port scanning attack rate threshold. Users can adjust the default threshold on the issue setting page. The threshold conditions are as follows:

1. Among the TCP SYN packets sent by the source IP address at a time point, the number of packets with different destination ports reaches a threshold. This corresponds to the scenario where the attack source scans application ports enabled on the attacked host.
2. Among the TCP SYN packets sent by the source IP address at a time point, the number of packets with the same destination port but different IP addresses reaches a threshold. This corresponds to the scenario where the attack source scans hosts having the specific application port enabled.

If either of the preceding conditions is met, a suspected port scanning attack is identified. The Spark task calculation period is 10 seconds.

**Step 2** If a source IP address meets the port scanning attack rate threshold, FabricInsight identifies a suspected port scanning attack issue and records information such as the source IP address, attack time, and attack duration.

**----End**

**Troubleshooting Suggestions**

Similar to SYN flood attacks, port scanning attack sources usually use forged IP addresses to launch attacks. Once an attack occurs, network O&M personnel can hardly trace the attack source based on the forged IP addresses. By analyzing original packets, FabricInsight proactively identifies the network access location of the attack source, which greatly improves the efficiency of locating the attack source host.

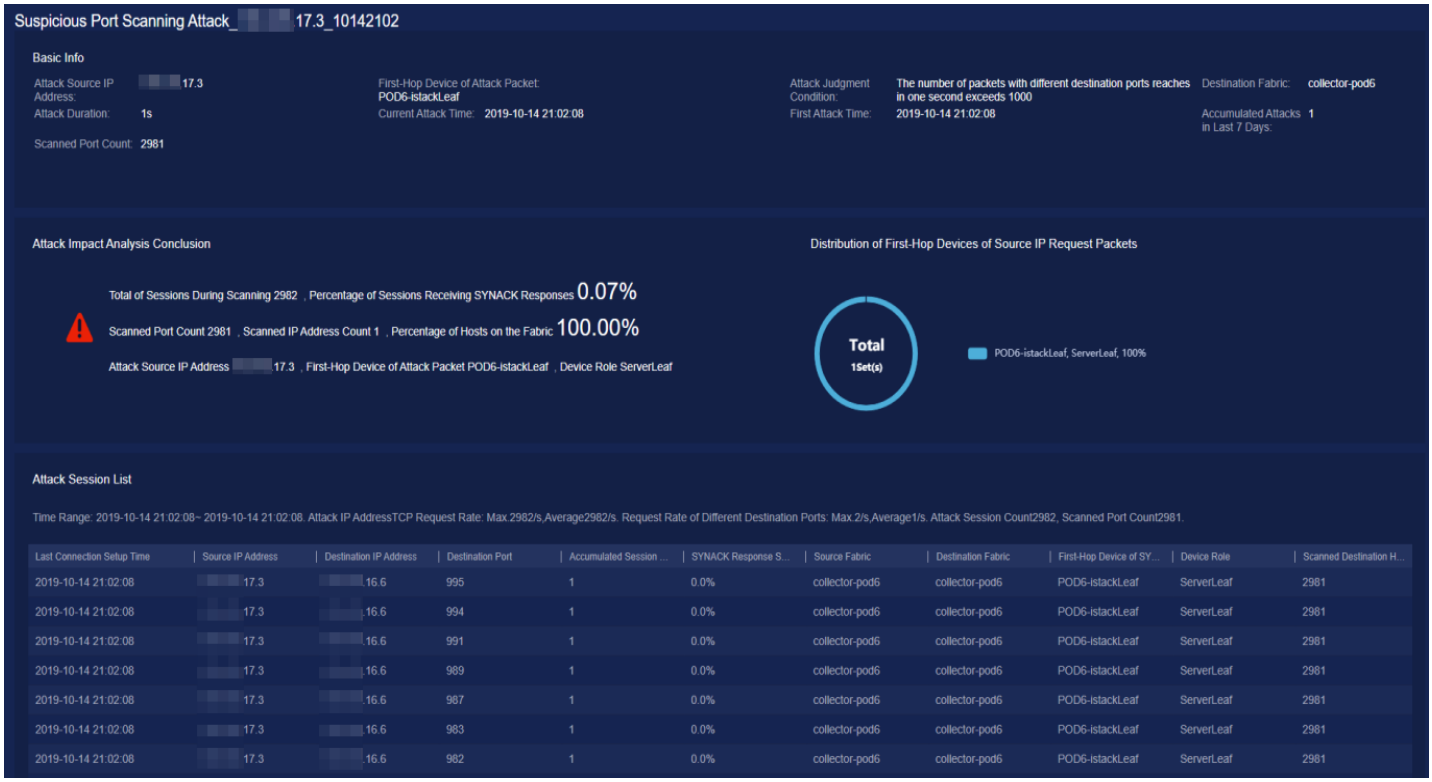
**Step 1** On the issue page, view the issue list and check source IP addresses initiating port scanning attacks.

**Step 2** Click an issue in the issue list. On the issue details page that is displayed, check the distribution of first-hop devices of the TCP SYN packets from the attack source.

1. If the first-hop devices are mainly BorderLeaf devices, the attack source is a host out of the fabric.
2. If the first-hop devices are mainly ServerLeaf devices, the attack source is a host on the fabric. In this case, users only need to further check hosts connected to these ServerLeaf devices to locate the attack source.

**----End**

**Figure 2-19** Port scanning attack issue details page



NOTE

1. To accurately locate the attack source, users need to configure ERSPAN mirroring for devices such as ServerLeaf and BorderLeaf devices on the fabric.
2. If ServerLeaf and BorderLeaf devices support the ERSPAN enhancement feature, it is recommended that ERSPAN enhancement be enabled when configuring ERSPAN. In this case, users can use FabricInsight to check the ingress port on the first-hop device of the attack packets, further narrowing down the attack source host scope.

# 2.5 RoCE Flow Analysis

This chapter describes the key technical principles involved in RoCE flow analysis. The principles include metric data collection and calculation.

## 2.5.1 Metric Data Collection Principles

Collection commands must be configured on CE switches to collect metric data of RoCE flows. After collection commands are configured on CE switches, the switches report packets to the NP chip based on ACL rules for metric calculation. Devices report metric data to FabricInsight every 10 seconds. FabricInsight receives metric data reported by devices through the NetStream protocol, summarizes the data, and displays the data on the GUI. Devices can collect all CM packets and some data packets.

**Figure 2-20** Data packets that can be collected

| Code[4-0] | Description |
|---|---|
| 00000 | SEND First |
| | |
| 00010 | SEND Last |
| 00011 | SEND Last with Immediate |
| 00100 | SEND Only |
| 00101 | SEND Only with Immediate |
| 00110 | RDMA WRITE First |
| | |
| 01000 | RDMA WRITE Last |
| 01001 | RDMA WRITE Last with Immediate |
| 01010 | RDMA WRITE Only |
| 01011 | RDMA WRITE Only with Immediate |
| 01100 | RDMA READ Request |
| 01101 | RDMA READ response First |
| | |
| 01111 | RDMA READ response Last |
| 10000 | RDMA READ response Only |
| 10001 | Acknowledge |
| 10010 | ATOMIC Acknowledge |
| 10011 | CmpSwap |
| 10100 | FetchAdd |

**Collection Configurations**

Run the following commands to enable a CE switch to report metric data of RoCE flows:

// Enable metric (throughput and RTT) data collection globally or based on ACL rules.

traffic-analysis rocev2 [global | acl acl-number] inbound

// Enable metric (packet loss) data collection globally.

traffic-analysis rocev2 drop global

NOTE

- In distributed storage scenarios, you are advised to collect the following metrics globally: throughput and RTT.
- In the HPC and AI Fabric scenarios, you are advised to collect the following metric globally: packet loss. The throughput and RTT metrics are not collected globally. You can configure ACL rules based on the site requirements to collect the throughput and RTT metrics.

**Data Reporting Template**

The following table describes the NetStream template for reporting RoCE flow metrics.

**Table 2-6** Template for reporting RoCE flow metrics

| Field | Description |
| --- | --- |
| SIP | Source IP address. |
| DIP | Destination IP address. |
| SQP | Source queue pair. |
| DQP | Destination queue pair. |
| Flow Start Time | Start time of a flow. |
| Flow End Time | Last update time of a flow. |
| Client To Sever Interface | Inbound interface in the request direction. |
| Client To Sever RTT | RTT in the request direction. |
| Client To Sever RTT Timestamp | Timestamp of the last update of the RTT in the request direction. |
| Client To Sever Write Throughput | Write throughput in the request direction. |
| Client To Sever Write Throughput Timestamp | Timestamp of the last update of the write throughput in the request direction. |
| Client To Sever Read Throughput | Read throughput in the request direction. |

| Field | Description |
| --- | --- |
| Client To Sever Read Throughput Timestamp | Timestamp of the last update of the read throughput in the request direction. |
| Client To Sever RTX | Number of lost packets in the request direction. |
| Sever To Client Interface | Inbound interface in the response direction. |
| Sever To Client RTT | RTT in the response direction. |
| Sever To Client RTT Timestamp | Timestamp of the last update of the RTT in the response direction. |
| Sever To Client Write Throughput | Write throughput in the response direction. |
| Sever To Client Write Throughput Timestamp | Timestamp of the last update of the write throughput in the response direction. |
| Sever To Client Read Throughput | Read throughput in the response direction. |
| Sever To Client Read Throughput Timestamp | Timestamp of the last update of the read throughput in the response direction. |
| Sever To Client RTX | Number of lost packets in the response direction. |

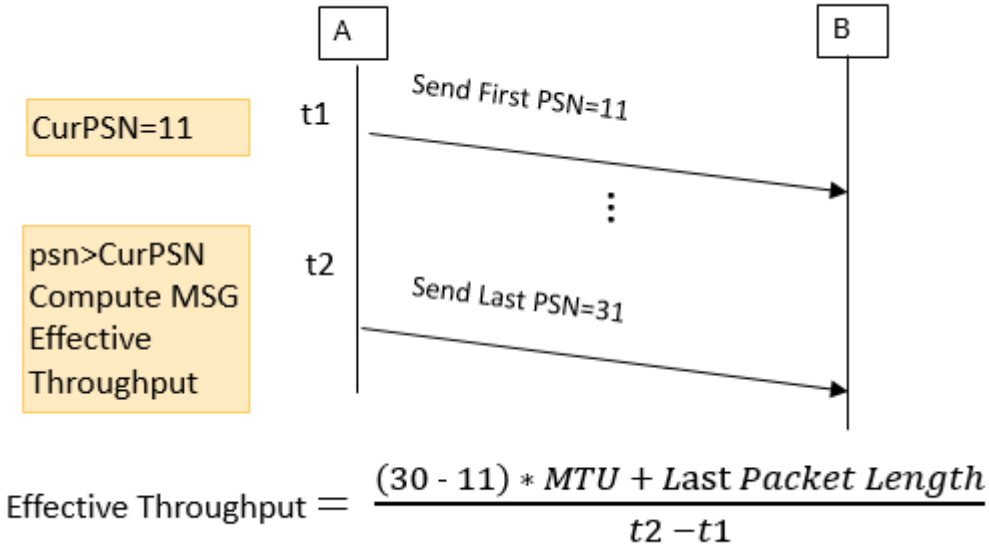
## 2.5.2 Metric Data Calculation Principles

**Throughput**

Only the throughput in a single direction is calculated. The system calculates the valid transmission volume of the MSG (excluding retransmitted volume) based on the PSN difference between the last and first packets. The calculation formula is as follows: Valid throughput = Valid transmission volume of the MSG/MSG time.

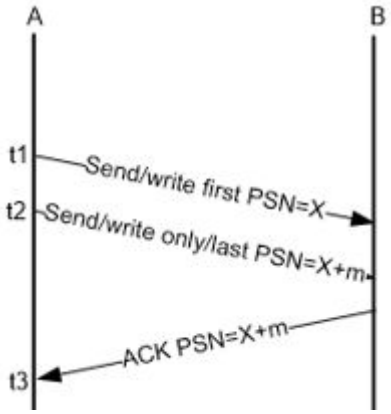
**Figure 2-21** Throughput calculation




**RTT**

To calculate the RTT, you need to record only the time of two packets. The time difference of the two packets is the RTT. The RTT in the following figure equals to t3-t2.

**Figure 2-22** RTT calculation

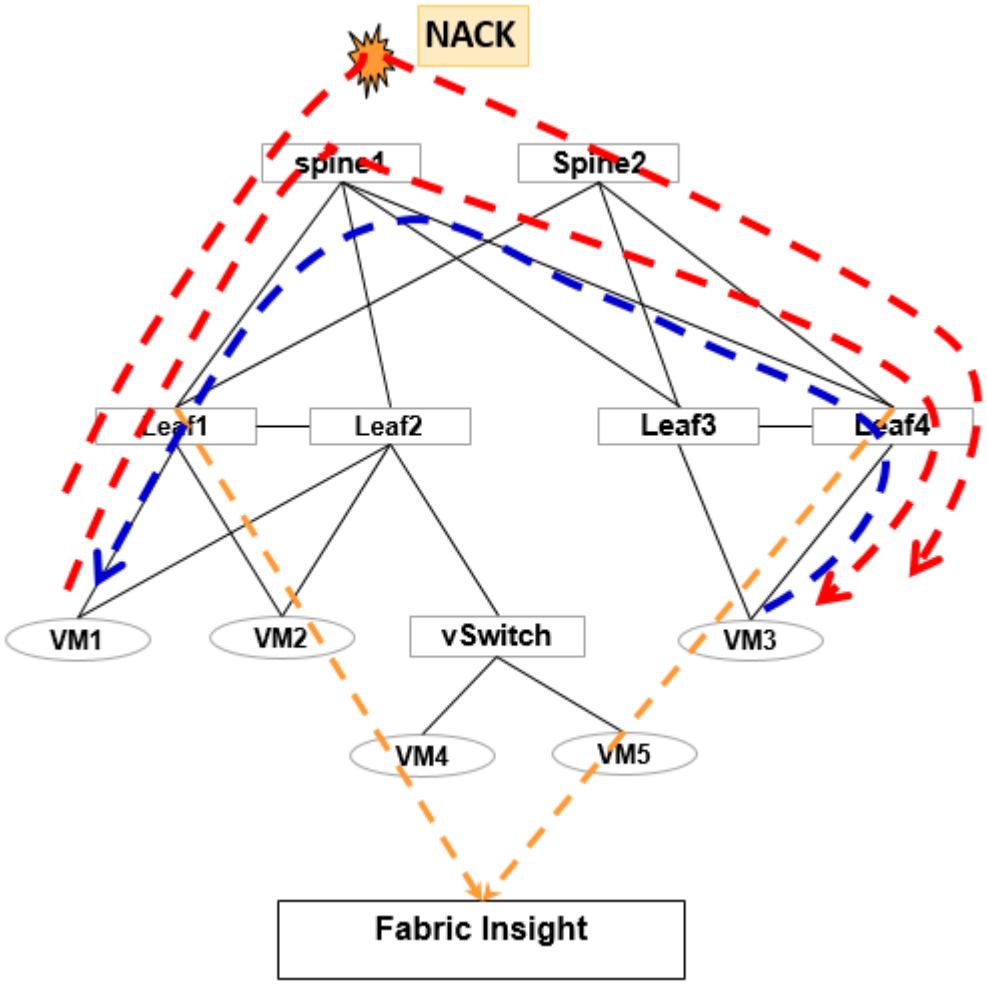



**Packet loss**

In the RoCEv2 protocol, when some packets in a message are discarded, the sequence number received by the destination network adapter is different from that sent by the source network adapter. The destination network adapter sends a NACK RoCEv2 packet to the source network adapter, indicating that the current message needs to be retransmitted.

**Figure 2-23** Lost packet quantity calculation




NOTE

The mainstream mellanox NICs do not support selective retransmission. If a packet is lost, all packets of the entire message will be retransmitted.

# 2.6 Edge Intelligent Analysis

This chapter describes the key technical principles involved in intelligent TCP and UDP traffic analysis.

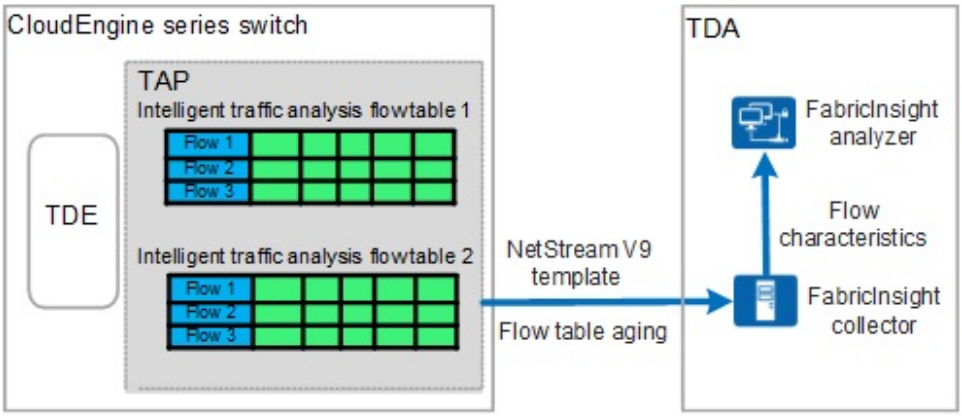
## 2.6.1 Intelligent TCP Traffic Analysis

### 2.6.1.1 Data Collection Principles for Intelligent TCP Traffic Analysis

To collect data for intelligent TCP traffic analysis, you need to configure a collection command on the CE switch. After the collection command is configured, the CE switch reports packets to the NP chip for TCP flow analysis and calculation based on ACL matching rules. FabricInsight uses the NetStream protocol to receive data in the bidirectional and unidirectional TCP flow tables reported by devices, analyzes the data, and displays the data in a visualized manner.

Figure 1-1 Flow table collection for intelligent TCP traffic analysis




As shown in the preceding figure, the client with IP address 10.1.1.1 interacts with the server with IP address 10.1.2.1. The intelligent traffic analysis function is deployed on the CE switch to analyze the flow and report the obtained flow analysis result to the FabricInsight collector. The operations are as follows:

Step 1: Configure an advanced ACL rule to match the TCP flow between the client with IP address 10.1.1.1 and the server with IP address 10.1.2.1.

acl number 3055

rule 4 permit tcp source 10.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255

Step 2: Enable intelligent TCP traffic analysis.

traffic-analysis tcp acl 3055 inbound

Step 3: Enable unidirectional intelligent TCP traffic analysis.

traffic-analysis tcp one-way sequence 0x0000000A 0x000000FF

Step 4: Configure flow aging for intelligent TCP traffic analysis.

traffic-analysis tcp timeout inactive 100

traffic-analysis tcp timeout ip tcp-session

Step 5: Configure the output of the intelligent TCP traffic analysis. The destination IP address is the floating IP address of the collector.

traffic-analysis tcp export source ip 10.1.3.1

traffic-analysis tcp export host ip 10.1.3.2 6000

### 2.6.1.2 Calculation Principles for Intelligent TCP Traffic Analysis

After device configurations are successfully delivered, the device analyzes the matched TCP packets and reports the quintuple flow table information to the FabricInsight collector. The following figure shows the fields in the flow table.

**Figure 2-24** Keys in quintuple information for creating a TCP flow table

| Key | Description |
| --- | --- |
| SPORT | Specifies the source port number of a service flow. |
| SIP | Specifies the source IP address of a service flow. Currently, only IPv4 addresses are supported.<br>For a flow table created based on SYN packets, the SIP is the source IP address of the SYN packets.<br>For a flow table created based on TCP intermediate data packets, the SIP is the source IP address of the first packet received by the traffic-analysis processor (TAP). |
| DPORT | Specifies the destination port number of a service flow. |
| DIP | Specifies the destination IP address of a service flow. Currently, only IPv4 addresses are supported. |
| Protocol | Indicates the protocol type. Only TCP is supported. |

**Figure 2-25** TCP traffic characteristics that can be analyzed by the TAP

| Characteristic | Description |
| --- | --- |
| Number of discarded packets | The TAP can collect the following statistics on packets transmitted in both directions:<br>Total number of discarded packets<br>Number of packets discarded on the upstream device |
| RTT | The TAP can calculate the round trip time (RTT) for packets transmitted in both directions. The RTT is the average sliding latency calculated based on packets transmitted in both directions. It is precise to the nearest microsecond. |
| Number of packets | The TAP can count the numbers of packets transmitted in both directions. |
| Flow status | The TAP can analyze the TCP flow status in the current flow table. The flow status can be one of the following:<br>SYN status<br>SYN + ACK status<br>ACK status<br>TCP connection establishment status<br>TCP connection termination status |
| Flow table creation time | The TAP can collect statistics on the time when a TCP flow table is created. |

| Characteristic | Description |
| --- | --- |
| Inbound interface of packets | The TAP can identify the inbound interfaces of packets transmitted in both directions. |
| VNI | The TAP can identify the VNIs of packets transmitted in both directions. |

Devices periodically report TCP flow characteristics (including bidirectional packet loss, RTT, and number of packets) based on the configured active aging time. FabricInsight calculates the traffic characteristics of each device in real time based on the Spark Streaming framework. The total number of lost packets, number of upstream lost packets, and number of packets reported by the device in the request/response direction are accumulated and not cleared each time after being reported. During data cleaning, FabricInsight calculates the difference based on the data in the previous and next periods to obtain the actual data in each period.

# 2.6.2 Intelligent UDP Traffic Analysis

## 2.6.2.1 Data Collection Principles for Intelligent UDP Traffic Analysis

To collect data for intelligent UDP traffic analysis, you need to configure a collection command on the CE switch. After the collection command is configured, the CE switch reports packets to the NP chip for UDP flow analysis and calculation based on ACL matching rules. FabricInsight uses the NetStream protocol to receive data in the UDP flow tables reported by devices, analyzes the data, and displays the data in a visualized manner.

As shown in the preceding figure, the client with IP address 10.1.1.1 interacts with the server with IP address 10.1.2.1. The intelligent traffic analysis function is deployed on the CE switch to analyze the flow and report the obtained flow analysis result to the FabricInsight collector. The operations are as follows:

Step 1: Configure an advanced ACL rule to match the UDP flow between the client with IP address 10.1.1.1 and the server with IP address 10.1.2.1.

acl number 3055

rule 5 permit udp source 10.1.1.0 0.0.0.255 destination 10.1.2.0 0.0.0.255

Step 2: Enable intelligent UDP traffic analysis.

traffic-analysis udp acl 3055 inbound

Step 3: Configure flow aging for intelligent UDP traffic analysis.

traffic-analysis udp timeout inactive 30

Step 4: Configure the output of the intelligent UDP traffic analysis. The destination IP address is the floating IP address of the collector.

traffic-analysis udp export source ip 10.1.3.1

traffic-analysis udp export host ip 10.1.3.2 6000

## 2.6.2.2 Data Calculation Principles for Intelligent UDP Traffic Analysis

After device configurations are successfully delivered, the device analyzes the matched UDP packets and reports the quintuple flow table information to the FabricInsight collector. The following figure shows the fields in the flow table.

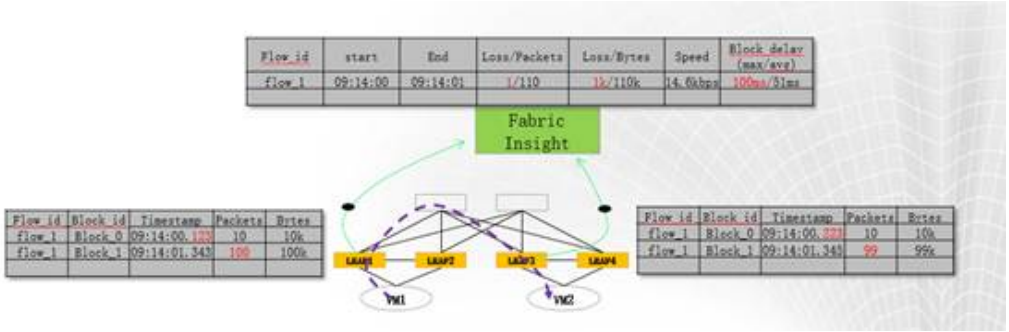
**Figure 2-26** Keys in quintuple information for creating a UDP flow table

| Key | Description |
| --- | --- |
| SPORT | Specifies the source port number of a service flow. |
| SIP | Specifies the source IP address of a service flow. Currently, only IPv4 addresses are supported. |
| DPORT | Specifies the destination port number of a service flow. |
| DIP | Specifies the destination IP address of a service flow. Currently, only IPv4 addresses are supported. |
| Protocol | UDP protocol. |

**Figure 2-27** UDP traffic characteristics that can be analyzed by the TAP

| Characteristic | Description |
| --- | --- |
| Number of packets | The TAP can count the number of UDP packets of a device. |
| Size of packets | The TAP can count the number of bytes of UDP packets of a device. |
| Timestamp. | The TAP can collect statistics on timestamps. For the same UDP flow, the timestamp increases with the growth of data volume. |
| Flow table creation time | The TAP can collect statistics on the time when a UDP flow table is created. |
| VNI | The TAP can identify VNIs of UDP packets. |

After receiving the UDP flow table reported by a device, FabricInsight calculates the flow information (including the number of packets, number of bytes, and rate) and forwarding quality (packet loss and latency) between every two neighboring devices based on the TTL.



As shown in the preceding figure, FabricInsight calculates the metrics about packet transmission between Leaf1 and Leaf3 based on the TTL sequence.

- Latency = Timestamp (Leaf3) - Timestamp (Leaf1)
- Number of packets = Maximum number of packets
- Number of lost packets = Number of packets (Leaf1) - Number of packets (Leaf3)
- Traffic = Maximum number of bytes
- Discarded traffic = Number of bytes (Leaf1) - Number of bytes (Leaf3)

NOTE

- It is recommended that IP-based clock synchronization in compliance with the IEEE 1588v2 standard be enabled. NTP clock synchronization will cause inaccurate calculation of metrics such as the TCP TTL and UDP latency.
- Devices with TD3 chips match only decapsulated VXLAN packets from VXLAN tunnels. If packets are not from a VXLAN tunnel or are not decapsulated, the devices match packets based only on common ACL rules.
- Intelligent traffic analysis can be used to analyze a specified flow (unidirectional or bidirectional) that passes through the same switch twice instead of multiple times.
- ECMP per-packet load balancing is not supported. That is, paths for packets in one quintuple session in the request and response directions must be fixed.
- The ERSPAN data and edge intelligence data of a specified flow cannot be both reported. The priority for reporting edge intelligence data is higher than that of reporting ERSPAN data.
- For details about the device models that support edge intelligence, see the specification list.

# 3 Function Constraints

This section describes the requirements for the networking, hardware configuration, and deployment of FabricInsight.

3.1 Device Types and Networking Restrictions

## 3.1 Device Types and Networking Restrictions

### Networking Restrictions

The supported networks are as follows:

- VxLAN hardware-centralized gateway network
- VxLAN hardware-distributed gateway network
- Pure IP network (IP Fabric)

Note:

(1) The underlay network is based on IP forwarding.

(2) The SVF network is not supported.

(3) IP address overlapping scenarios (for example, multi-tenant and VPC scenarios) are not supported.

(4) Other networking modes such as traditional layer-2 networking (including the VLAN and STP), TRILL networking, and MPLS VPN are not supported.

### Device Configuration Restrictions

To perform ERSPAN remote mirroring, users need to use the ACL to match the traffic and match the SYN, FIN, and RST packets of the TCP flow. In addition, the ACL resources on the device are limited and the ACL matching rules are incorrect. Therefore, when policy-based routing and traffic statistics are configured on the device to use the ACL resources, users need to pay attention to the scenarios where the ACL rules conflict or ACL resources are insufficient.

# 4 Typical Application Scenarios

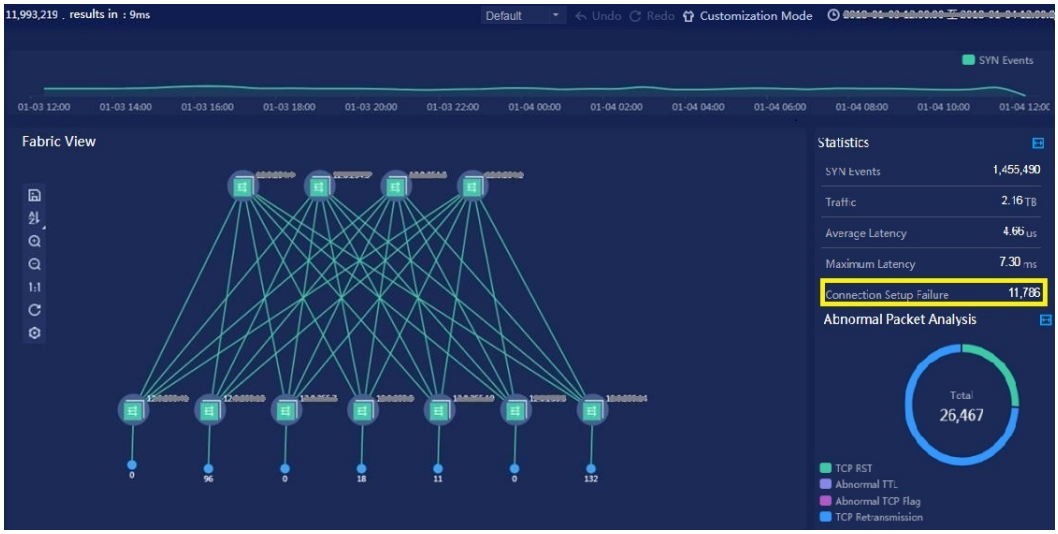
4.1 TCP Connection Setup Failure Analysis
4.2 TCP RST Packet Analysis
4.3 Proactive Prediction of Abnormal Device Metrics and Correlation Flow Analysis

## 4.1 TCP Connection Setup Failure Analysis

Generally, the TCP connection setup failure is caused by that the client receives no response from the server after the TCP SYN packet is retransmitted several times. If TCP connection setup failure occurs occasionally, it may be caused by packet loss upon network congestion, which is not a problem. However, if the TCP connection setup failure is not an occasional phenomenon and occurs in a certain rule, there may be optimization possibility.

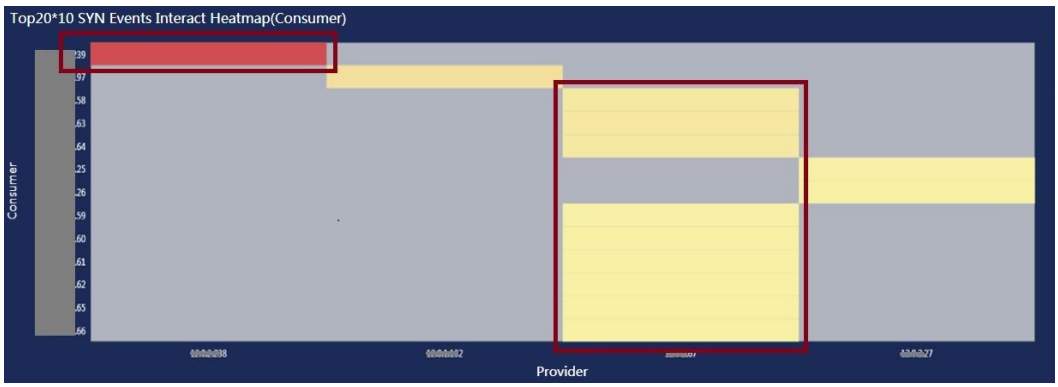
FabricInsight can identify TCP SYN packet retransmission and connection setup failures. In addition, FabricInsight provides related functions to analyze connection setup failures. The following uses a real case as an example to describe the general process of connection setup failure analysis.

Step 1 On the **Network** page, you can check whether connection setup failure occurs on the network. As shown in the following figure, the network is in good condition and only a few connection setup failures occur. Further analysis is required to determine whether the connection setup failure events have a certain rule.

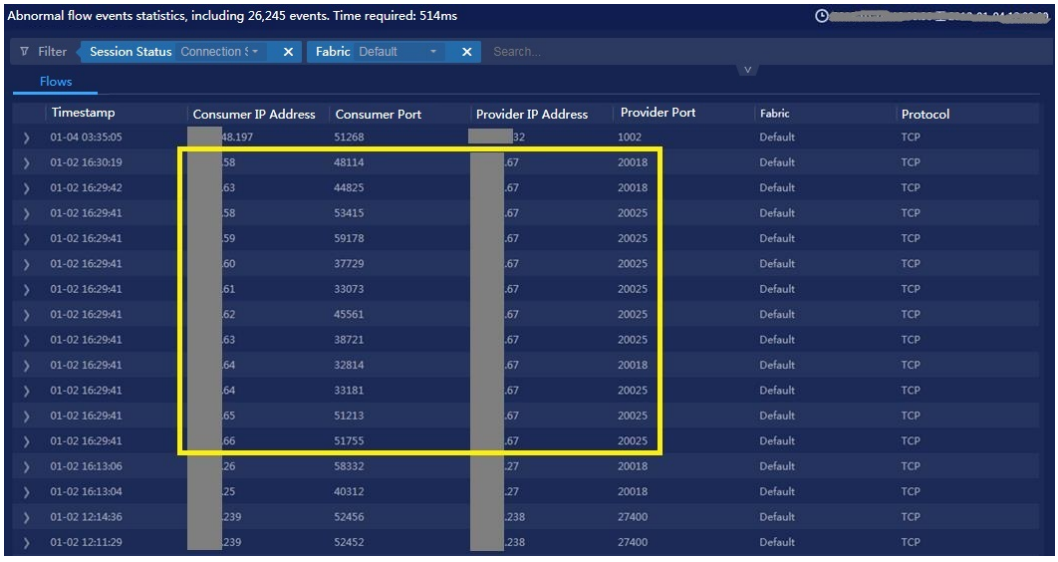


Step 2 Use the heatmap in the dashboard to analyze the connection setup failure. As shown in the following figure, connection setup failure events occur intensively. Especially, connection setup failure events occur between an IP address and multiple IP addresses.

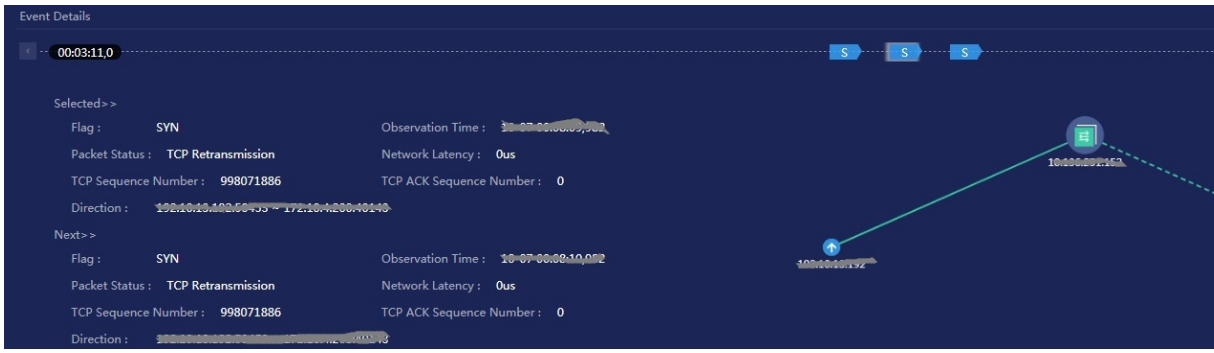



Step 3 On the **Event** page, filter connection setup failure events for detailed analysis. As shown in the following figure, the connection setup failure events are scattered from the perspective of the source IP address and are not centralized on a port from the perspective of the destination port. In addition, the connection setup failure time is within one second. Therefore, it can be preliminarily determined that the events are not closely associated with the client and destination port and are closely related only to the destination IP address.




Step 4 View details about an event. As shown in the preceding figure, the TCP connection experience four times of TCP SYN packet retransmission. Each event passes through a leaf node during transmission. Actually, the two IP addresses are across leaf nodes. If the network is normal, both the spine node and the last-hop leaf node receive packets. Therefore, it can be preliminarily determined that the fault is caused by packet loss on the network. The fault point is between the spine node and peer leaf node.




# 4.2 TCP RST Packet Analysis

## TCP RST Packet Introduction

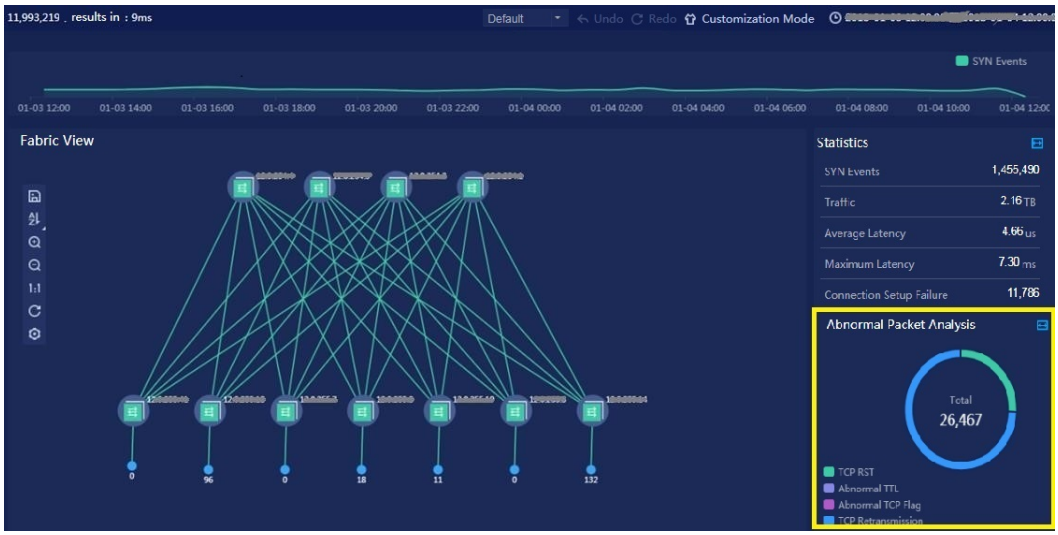
The TCP RST events may be caused by TCP RST attack or improper application implementation. The events may even be normal. Generally, TCP RST packets are generated in the following scenarios:

- No process is listening on the destination port when the TCP connection request arrives at the port.
- A TCP connection is torn down abnormally. When a TCP connection is torn down through FIN packets, the FIN/ACK and ACK packets need to be exchanged twice. When a TCP connection is torn down through RST packets, the packets need to be sent once only. Therefore, an application may send RST packets to quickly tear down a TCP connection.
- The connection is half closed. When one of the two parties of the TCP interaction still receives data on a closed TCP connection, TCP RST packets are generated. For example, the client initiates a connection teardown request and sends a FIN packet to the server. After sending the FIN packet, the client waits for the FIN packet returned by the server. However, if the client receives the last data packet (PSH) sent by the server before the FIN packet arrives at the client, the client immediately sends a TCP RST packet to the server to notify the server that the current connection needs to be reset.

FabricInsight can be used to analyze TCP RST packets on the network and identify the normal and abnormal TCP RST packets. The following uses an example to describe the analysis process of TCP RST packets.

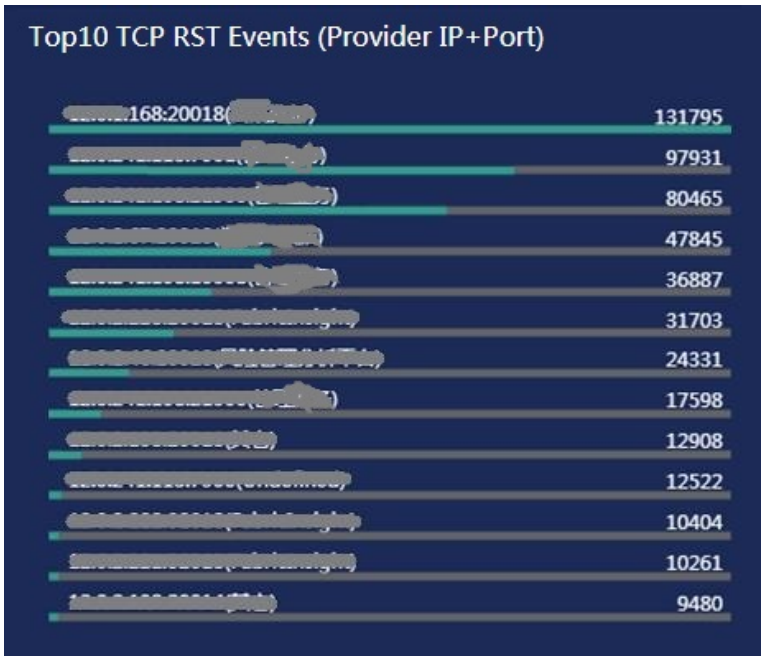
## Case 1: TCP RST packets are generated due to improper application implementation mechanism.

Step 1 Use the network function to view the network topology and view the total number of abnormal events and the proportion of various abnormal events. As shown in the following figure, the proportion of TCP RST events is the largest. A large number of TCP connections on the network are reset. Further analysis is required to determine the specific causes.



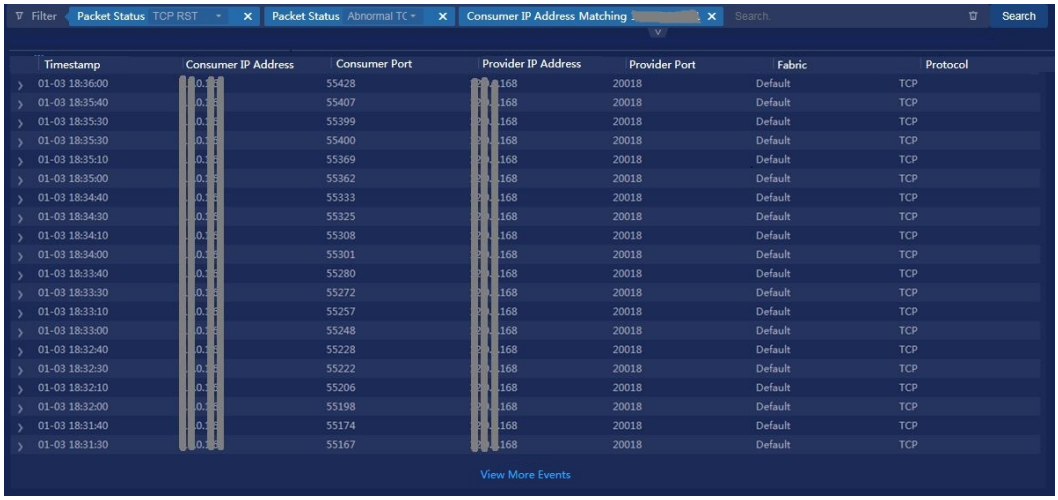

Step 2 Analyze the IP address with the largest number of RST events through the top N RST events in the dashboard. As shown in the following figure, TCP RST events are evenly distributed in the request direction and the average number of RST events of top 10 IP addresses is between 5000 and 5300. However, almost all TCP RST events are distributed on the first IP address in the response direction.


Step 3 Analyze the combination of IP address and port number with the maximum number of TCP RST events by analyzing the number of top TCP RST events on the dashboard. As shown in the following figure, the port with the maximum number of RST events is 21008 in the first row.

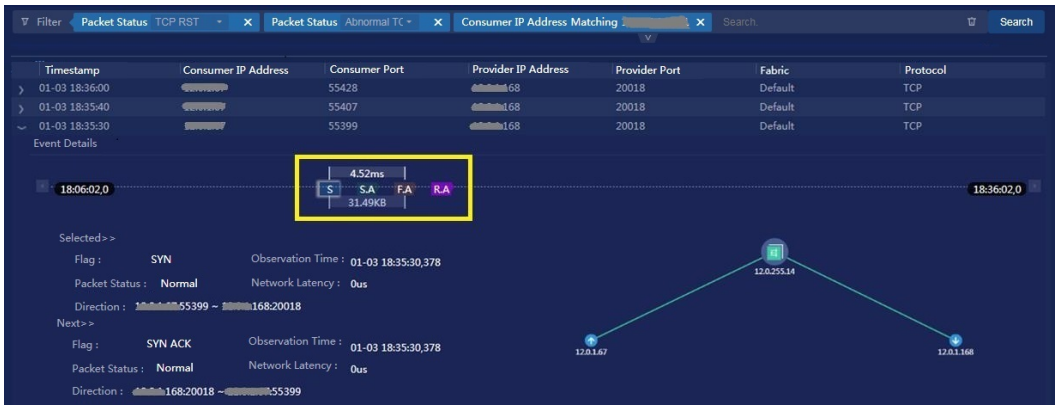



Step 4 On the **Event** page, view details about the interaction between two VMs to determine whether an RST event is normal.

The following figure shows the interaction between a VM and XXX.168:20018. A TCP connection is reset about every 10 to 20 seconds. The RST event has certain regularity from the time dimension.




View details about another event. It is found that the connection lasts only a few milliseconds. In addition, the connection is actively reset by the client. After initiating a request to reset the connection (sending a FIN packet), the client sends a RST&ACK packet. Therefore, the cause is that the client receives a data packet before receiving a FIN packet from the server.

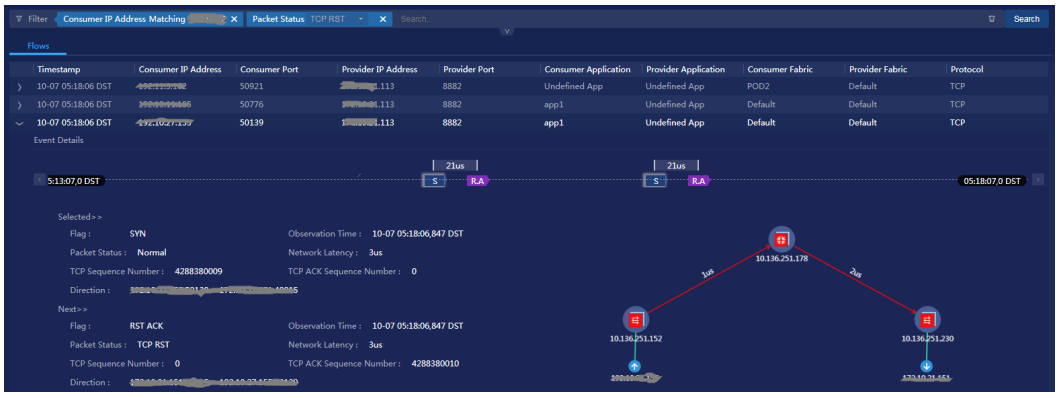



Summary: A TCP connection to port 20018 is reset about every 10 to 20 seconds and the connection lasts only a few milliseconds. The SYN, SYN&ACK, and FIN&ACK packets all exist on the event details page, indicating that the TCP connection is set up normally. In addition, the TCP connection teardown is actively initiated by the client. After initiating the connection teardown, the client receives data packets from the server. As a result, RST packets are generated. Since the short connections have a fixed interval, the problem may be caused by some implementation mechanisms of the application. In this case, the problem is caused by improper heartbeat mechanism implementation of the application.

## Case 2: RST packets are generated due to application migration.

In addition to the TCP RST exception analysis described in the previous case, the environment also has the following exception: After the client sends a SYN packet, the server directly responds with an RST packet. For details about how to find the RST event, see the analysis procedure described in case 1. Here, you can directly filter the corresponding RST event based on the combination of the IP address and event status on the flow event page.

Multiple clients initiate TCP connections to port 8882 on server 113. However, the server directly responds with only RST packets. In this case, the service corresponding to port 8882 on server 113 may be faulty. As a result, the corresponding port is not listened. Or, the service has been removed from server 113. Finally, O&M personnel confirms that the application service corresponding to port 8882 has been migrated from server 113 to another server. However, the client did not synchronize the information. As a result, TCP RST packets are generated.



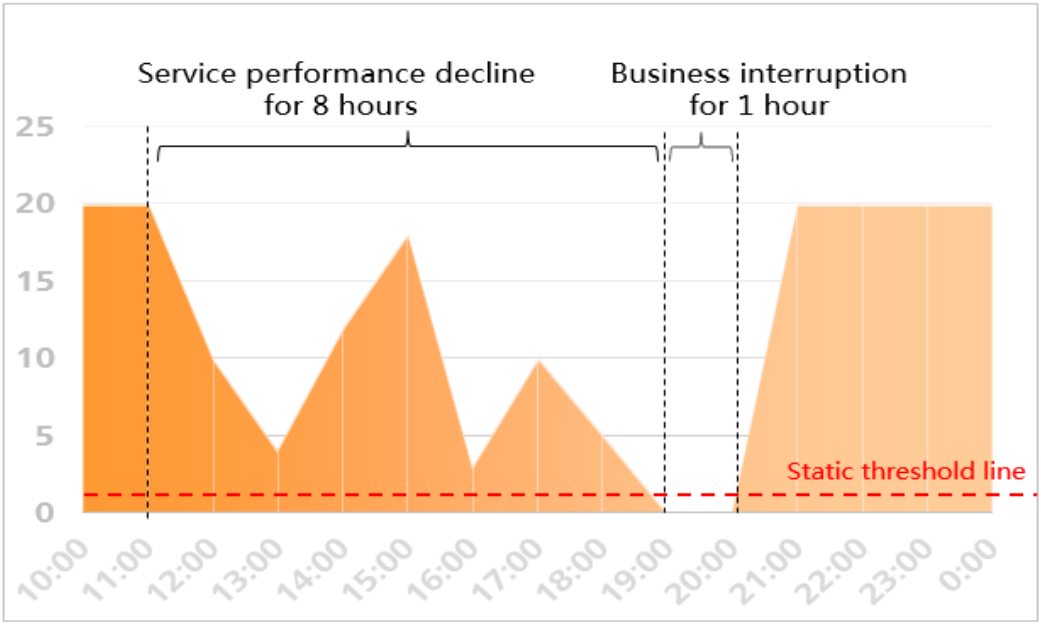

Summary: When a TCP connection has only SYN and RST events, it can be determined that the connection is abnormal. O&M personnel need to assist in analyzing the RST packet generation cause.

# 4.3 Proactive Prediction of Abnormal Device Metrics and Correlation Flow Analysis

**Scenario**

Services are interrupted in a DC. It is found that performance metrics deteriorate several hours before the service interruption. However, traditional O&M cannot provide an accurate and reasonable threshold. As a result, the system does not determine that the service is abnormal until a service complaint is reported.

**Figure 4-1** Service interruption caused by device performance metric deterioration

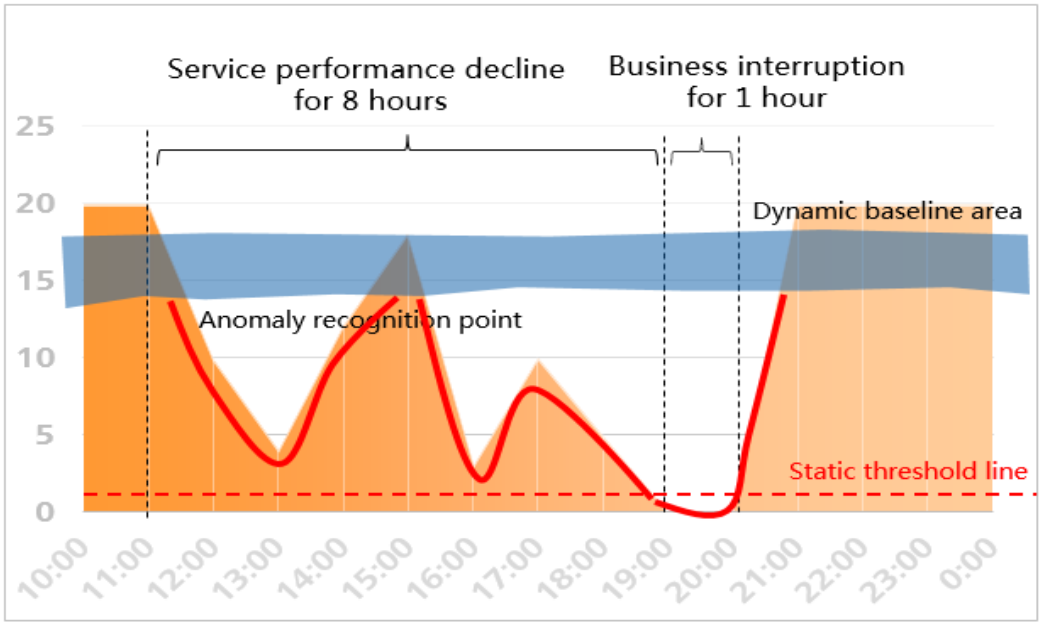



As shown in the preceding figure, the metric data of the measurement object is relatively stable before 11:00 and after 21:00. Starting from 11:00, the metric data deteriorates and services are interrupted at 19:00. Traditional O&M methods use static thresholds to identify metric threshold alarms. However, static thresholds have many problems. For example, the thresholds cannot be properly defined, and service metric changes cannot be proactively identified. In this case, you cannot determine whether a fault is a normal behavior or an abnormal behavior and predict abnormal metrics before the threshold is exceeded.

The problem can be solved based on dynamic baseline and exception detection AI algorithms.

**Figure 4-2** Abnormal detection based on the dynamic baseline can identify network exceptions in advance.




After the dynamic baseline is introduced, FabricInsight can identify network metric deterioration before service interruption. As shown in the preceding figure, FabricInsight can identify the metric baseline exception at about 11:30. You can use the analysis results provided on FabricInsight to rectify faults in advance, preventing service interruption.

**Daily Analysis Procedure:**

In this version, FabricInsight creates CPU/memory usage baselines for all connected CE devices and boards, and creates baselines of the number of received/sent packets for interfaces of physical links by default. For details about the supported metrics and data, see section 2.3.2.

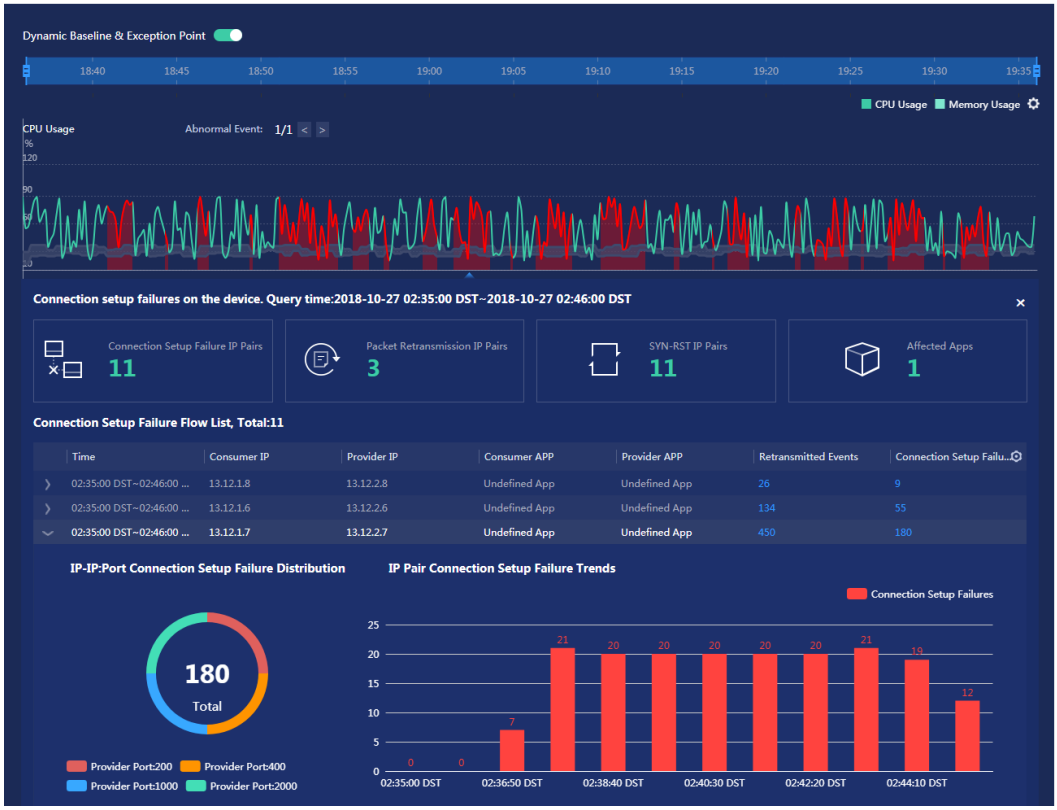
**Step 1** Choose **Telemetry** from the main menu. The Telemetry page is displayed. Information is displayed by resource types such as device, board, interface, queue, and optical module on different tab pages on the Telemetry page. Take the **Device** tab page as an example. After you select a metric (CPU/memory usage), FabricInsight sorts top devices in descending order based on the metric. The sorted results are displayed in the area distribution chart. You can select one or more devices to perform data correlation analysis for the metric. The metric statistics trend chart of the selected device is displayed on the page.

**Step 2** Click the exception button at the upper part of the area distribution chart. The system displays the measurement objects with baseline exceptions in the query time range. You can also select one or more devices for correlation analysis. In addition to the metric statistics trend chart of the selected device, the dynamic baseline and exception detection data are also displayed on the page. You can quickly locate the time when the baseline exception occurs by clicking the left or right exception switch button at the upper part of the trend chart.


**Step 3** Check whether the status of the device, board, or interface with baseline exception is normal to prevent service interruption caused by metric deterioration. You can click a device to go to the device profile page and view detailed metrics of the device.

**Step 4** Click an exception to view the exception occurrence time and flow behavior data that passes through the device or interface and has connection setup failures one minute before and after the occurrence time, and evaluate whether the device baseline exception affects service flows. On the **Device**/**Board** tab page, the system associates flows that pass through the device and have connection setup failures by default. On the **Interface** tab page, if the current device supports the ERSPAN enhancement feature and has the feature enabled (the packet forwarding route can be accurate to physical links), the system automatically queries flows that pass through the interface and have connection setup failures. Otherwise, the system still collects and displays flows that pass through the device and have connection setup failures.




**Step 5** Click the bar chart of flow connection setup failures. The **Flow Event** page is displayed. Query information by the 2-tuple information, event timestamp, and connection setup failure status. The **Flow Event** page displays the data after filtering. In this case, you can analyze the hop (device) where the packet is terminated based on the packet forwarding route. If the last hop of the packet is the current baseline exception device, there is a high probability that connection setup failure is caused by the device.

**----End**

NOTE

1. In this version, dynamic baseline and exception detection are created only for some metrics of devices, boards, and interfaces with physical links. Therefore, the dynamic baseline and baseline exception data cannot be correlatively viewed on the queue and optical module tab pages on the **Telemetry** page.