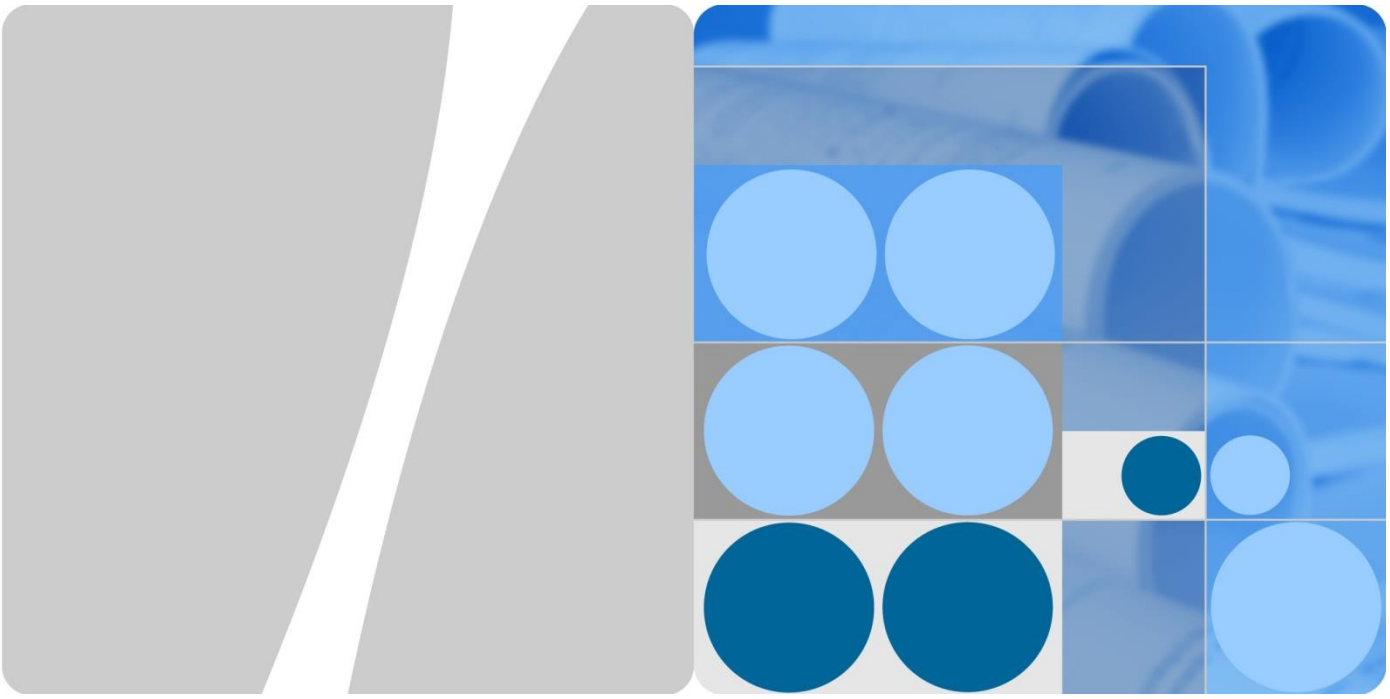


Telemetry Solution Technology White Paper

Issue 01

Date 2018-03-21

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: http://www.huawei.com

Email: support@huawei.com

Telemetry Solution Technology White Paper

Keywords

Telemetry, gRPC

Abstract

Telemetry is a technology developed to remotely collect data from physical or virtual devices. In push mode, telemetry collects data in real time, improving the usage of devices and networks when collecting data.

Background

Today's networks are bigger than ever and transmit a growing number of services. Intelligent O&M systems place more stringent requirements on the number of monitored devices, performance, and precision of network monitoring. With traditional network monitoring methods, the network management system (NMS) is not able to monitor a larger number of devices at a higher precision while minimizing the impacts on device functions and performance. The telemetry solution solves these problems.

Contents

1 Why Telemetry?

Telemetry helps to overcome the following challenges:

- Networks are continuously becoming larger, resulting in a growing number of devices to be managed.
- Networks are becoming increasingly complex, requiring the NMS to quickly locate faults and predict risks.
- If traffic microbursts occur, packets exceeding the device forwarding capability are discarded, requiring retransmission and affecting communication quality. The more microbursts occur, the worse the communication quality. Therefore, the NMS is required to promptly detect microbursts and quickly adjust the traffic.

Traditional network monitoring methods, including SNMP, CLI, and syslog, cannot address the preceding challenges.

Figure 1-1 Issues faced by traditional network monitoring methods

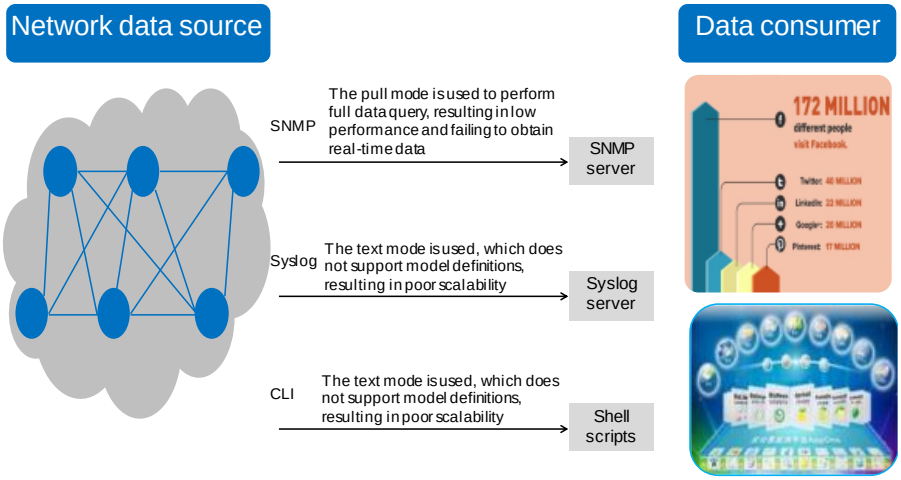


Table 1-1 Traditional network monitoring methods

Item	SNMP Get	SNMP Trap	CLI	Syslog
Working mode	Pull	Push	Pull	Push
Precision	Minutes	Seconds	Minutes	Seconds
Monitoring range	All data	Only traps	All data	Only events
Structure	MIB-defined structure	MIB-defined structure	Non-structured	Non-structured

Traditional network monitoring methods (such as SNMP Get and CLI) obtain device monitoring data, including interface traffic, in pull mode and cannot monitor a large number of network devices, limiting growth of the network. In addition, intelligent O&M systems place increasingly stringent requirements on the precision of network device data. To improve data precision, traditional network monitoring methods must increase the query frequency. This consumes more CPU resources on network devices and therefore affects device functions. Due to the network transmission delay, data on network devices can only be obtained every few minutes, failing to meet requirements for obtaining data every few seconds or even subseconds. In addition, the command output is complex to parse.

Although SNMP trap and syslog use the push mode and can push data with little delay, only traps or events can be pushed. Other data such as interface traffic cannot be pushed.

A new network monitoring method is required to implement large-scale and high-performance monitoring on network devices.

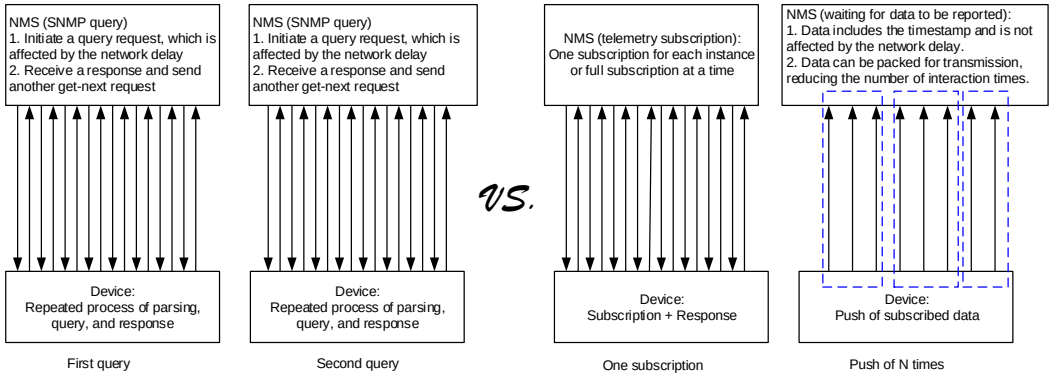
2 What Is Telemetry?

2.1 Comparison Between the SNMP Pull Mode and the Telemetry Push Mode

In the traditional pull mode, stable network device monitoring with data obtained every few seconds cannot be implemented due to various factors such as the network scale and transmission delay. In push mode, network devices automatically push data to the NMS, improving monitoring performance.

In a typical network monitoring scenario, interface traffic needs to be periodically collected from network devices. SNMP only supports sampling every few minutes. Telemetry, on the other hand, supports sampling every few seconds or even subseconds in push mode.

Figure 2-1 Comparison between the SNMP query process and the telemetry sampling process



Telemetry: eliminating repeated query

With SNMP, devices interact by alternatively sending requests and responses. If 1000 query requests are sent in the first minute, SNMP query requests are parsed 1000 times. In the second minute, SNMP query requests are parsed another 1000 times. This process is subsequently repeated.

The 1000 SNMP query requests that are sent in the first minute are the same as those sent in the second minute. Telemetry parses 1000 subscription requests in the first minute and continuously pushes data to the NMS in the following minutes, avoiding the need to parse query requests after the first minute.

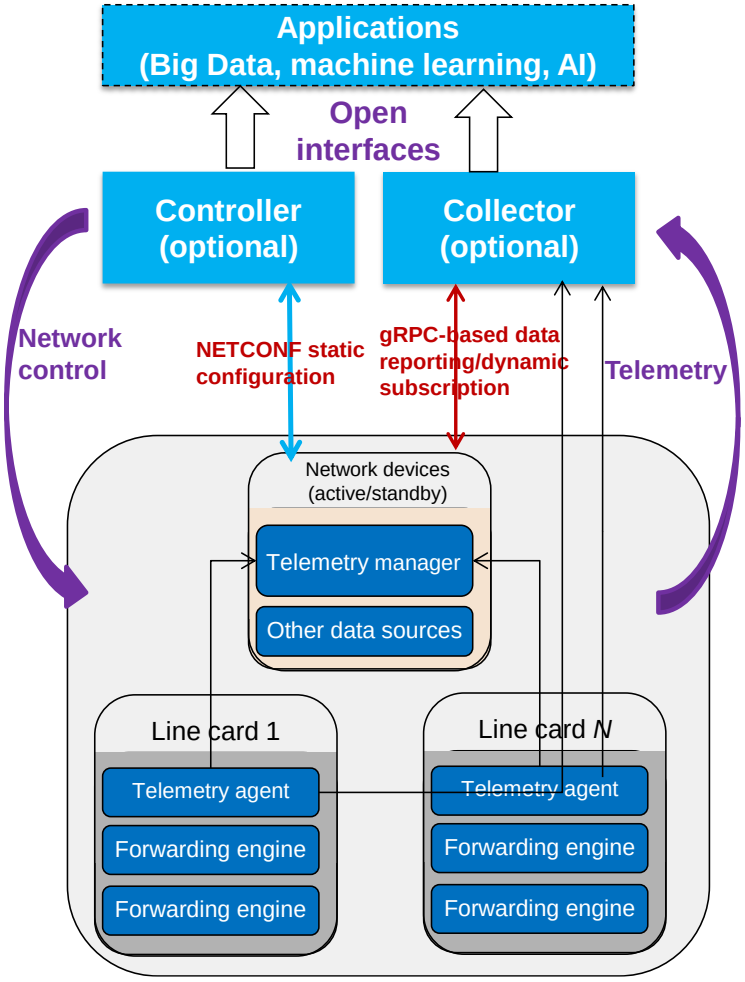
Data packaging

With SNMP, one packet is sent for each interface. Telemetry obtains data of multiple interfaces each time, packages the obtained data, and sends it to the network analyzer, reducing the number of interaction times.

2.2 Telemetry Technology

In a broad sense, telemetry is a self-closed-loop automatic O&M system.

Figure 2-2 Telemetry self-closed-loop system



- Analyzer: analyzes and processes monitoring data on the network.
- (Optional) Controller: controls and manages devices on the network.
- (Optional) Collector: receives and stores the monitoring data reported by network devices.
- Network device: receives configurations from the NMS, samples the data source specified by the NMS, and pushes the sampling data to the collector.

Self-closed-loop process

1. The NMS sends data sampling instructions to network devices. Network devices sample data and push it to the NMS.
2. The NMS analyzes the received data and adjusts the network configuration.
3. Network devices push real-time data to the NMS. The NMS checks the adjustment effect. In this way, a closed-loop automatic O&M system is formed.

This document describes telemetry in its narrow sense, namely, the telemetry mechanism on the device side. It samples data based on models, encodes the sampling result based on GPB, and pushes the data to the collector through gRPC (UDP will also be supported in the future).

3 Telemetry Solution

3.1 Model-driven Telemetry

Data on network devices is described using proprietary YANG models (customized by vendors) or standard YANG models (defined by standards organizations such as IETF).

Data on network devices is organized and presented based on model definitions. The NMS and network devices interact with each other through model definitions.

Model-driven telemetry refers to the process in which users or the NMS uses model definitions to notify network devices of the data to be sampled. After collecting the required data, the network devices report the data organized based on the model definitions to the NMS.

Figure 3-1 Hierarchical device model

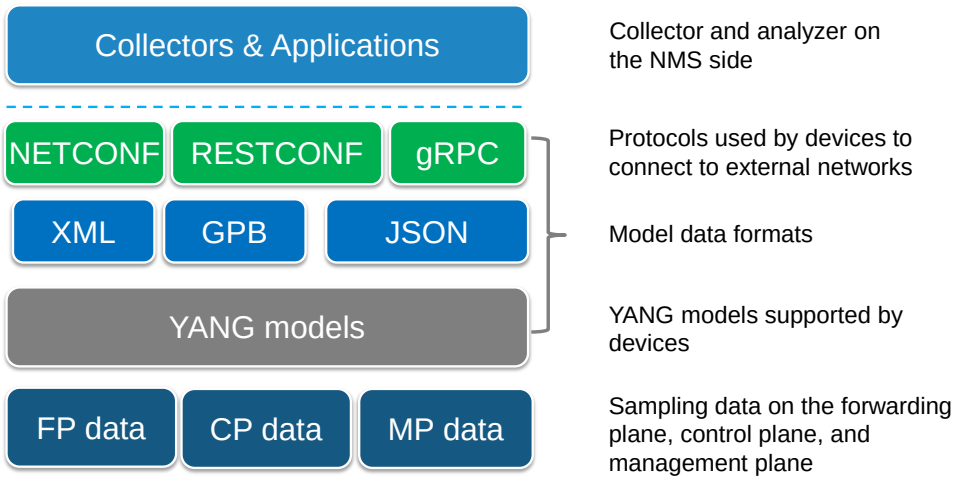


Figure 3-1 shows the hierarchical device model. The lowest part is the internal device model and is not visible from external networks. Network device data is visible from external networks only after it is mapped through YANG models.

Currently, only proprietary YANG models defined by Huawei are supported. Later, YANG models defined by standards organizations such as IETF and OpenConfig will also be supported.

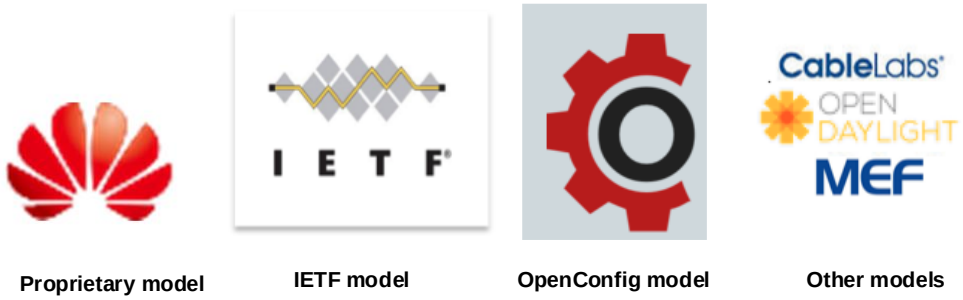
After network device data is organized based on YANG models, the data is encoded through XML, JSON, or GPB to interact with external networks using connection protocols.

Telemetry uses GPB to encode data that is based on YANG models and pushes the encoded data to the collector through gRPC. GPB is a binary encoding protocol and has a transmission efficiency higher than the traditional XML and JSON encoding protocols.

3.2 YANG Models Supported by Telemetry

Figure 3-2 shows the YANG models defined by vendors and standards organizations for network devices.

Figure 3-2 Various YANG models



- Proprietary model: YANG models developed and released by each service module of Huawei
- IETF model: standard models defined by IETF
- OpenConfig model: model developed with Google as the main instigator
- Other models: YANG models defined by other vendors, for example, Cisco YANG model

Huawei devices use Huawei proprietary models to send data. In addition, Huawei devices support third-party standard models. Huawei will preferentially support YANG models defined by IETF followed by those defined by OpenConfig.

3.3 Telemetry Modes

Telemetry supports the following triggering modes:

- Dial-out: A telemetry sampling command is configured on a device, which proactively connects and pushes data to the collector.
- Dial-in: The collector proactively connects to a device and requests the device to sample data.

3.3.1 Dial-out

On the device side, telemetry supports the dial-out mode. Users or the NMS uses the CLI or NETCONF to configure the telemetry function, including the data to be sampled, destination to which the sampled data is reported, and sampling period.

After the configuration is complete, the device samples data, sets up a connection to the destination, and pushes data.

If the connection between the device and the destination is interrupted, the device attempts to connect to the destination and push data again. However, data sampled during the reconnection is lost.

In dial-out mode, configuration restoration is supported. After devices are restarted, telemetry resumes the sampling tasks, and devices sample data and push the data to the destination.

The following is an example of the dial-out configuration process:

1. Configure a sampling path group.

Command	Description
[*HUAWEI]telemetry	Enters the telemetry configuration view.
[*HUAWEI-telemetry] sample enable	Enables telemetry sampling.
[*HUAWEI-telemetry] sensor-group mySensor	Creates a sampling path group.
[*HUAWEI-telemetry-sensor-group-mySensor]sensor-path Huawei-devm:cpuinfos/cpuinfo/	Configures a sampling path.

2. Configure a destination group.

Command	Description
[*HUAWEI-telemetry] destination-group myDest	Creates a destination group.
[*HUAWEI-telemetry-destination-group-myDest] ipv4-address 1.1.1.1 port 333 protocol grpc no-tls	Configures the destination address.

3. Configure a subscription and associate it with the sampling path and destination group.

Command	Description
[*HUAWEI-telemetry] subscription mySub	Creates a subscription.
[*HUAWEI-telemetry-subscription-mySub]sensor-group mySensor sample-interval 100	Configures the sampling path group and sampling period.
[*HUAWEI-telemetry-subscription-mySub]destination-group myDest	Configures the destination group.
[*HUAWEI-telemetry-subscription-mySub]commit	Commits the configuration.

3.3.2 Dial-in

The NMS proactively sets up a gRPC connection to devices and delivers the gRPC request, instructing the devices to periodically sample data sources and send the sampled data to the specified destination. If no destination is specified, the data is sent to the NMS.

If the gRPC connection is interrupted, dial-in is canceled, and data is not sampled and pushed.

Configuration restoration is not supported.

The dial-in mode is implemented based on the gRPC connection. The RPC request is defined based on the openconfig-rpc.yang model. The following is an example of the tree format:

```
+---x subscribe
|  +---w input
|  |  +---w request-id?               uint64
|  |  +---w destination* [destination-address destination-port]
|  |  |  +---w destination-address    inet:ip-address
|  |  |  +---w destination-port       uint16
|  |  +---w path* [path]
|  |  |  +---w path              oc-rpc-types:openconfig-path
|  |  |  +---w exclude-filter?   string
|  |  +---w sample-interval?          uint64
|  |  +---w heartbeat-interval?       uint64
|  |  +---w suppress-redundant?       boolean
|  |  +---w originated-qos-marking?   inet:dscp
|  |  +---w encoding?                 identityref
|  +--ro output
|     +--ro subscription-id?   uint32
|     +--ro request-id?        uint64
|     +--ro response-code?     identityref
|     +--ro message?           string
+---x cancel
|  +---w input
|  |  +---w request-id?        uint64
|  |  +---w subscription-id?   uint32
|  +--ro output
|     +--ro request-id?      uint64
|     +--ro response-code?   identityref
|     +--ro message?         string
```

Configure the subscription relationship.

Configure the destination group.

Configure the sampling paths.

Configure the sampling interval.

Cancel the subscription relationship.

Cancel the sampling task of a specific subscription ID.

3.3.3 gRPC Connection

The format described in the .proto file is used to implement the gRPC connection.

- PROTO files defined by RPC
  - **huawei-grpc-dialout.proto**: defines the RPC interface provided by a device that functions as a client to push data.
  - **huawei-grpc-dialin.proto**: defines the RPC interface provided by a device that functions as the server.
- Telemetry header definition file

  huawei-telemetry.proto: defines the data header for reporting data sampled by telemetry, including key information such as the sampling path and sampling timestamp.
- Service data file

  huawei-*app*.proto: defines the service app data format. An example is **huawei-devm.proto**.

Connection when a device functions as a gRPC server

When a device functions as a gRPC server, the NMS can proactively subscribe to dial-in and send the sampled data back the same way. The collector proactively initiates either of the following RPC requests to devices:

- Subscription request: /huawei_dialin.gRPCConfigOper/Subscribe
- Unsubscription request: /huawei_dialin.gRPCConfigOper/Cancel

The RPC interface is defined in the **huawei-grpc-dialin.proto** file.

Connection when a device functions as a gRPC client

When a device functions as a gRPC client, sampled data can be pushed. Devices proactively initiate the RPC request /huawei_dialout.gRPCDataservice/dataPublish. The RPC interface is defined in the **huawei-grpc-dialout.proto** file.

The header is described in the **huawei-telemetry.proto** file, and the content is described by the **huawei-*app*.proto** file.

3.4 Key Telemetry Technologies

3.4.1 Hardware-based Data Collection

Telemetry data is collected through hardware-based query, implementing real-time and high-speed data collection.

3.4.2 Data Encoding

GPB is used to encode the collected data. The encoding efficiency of GPB is 2 to 5 times that of JSON while the data size after encoding is only 1/3 to 1/2 of that encoded using JSON. This improves telemetry data throughput performance and saves CPU and bandwidth resources.

3.4.3 Sending Protocol

Google's open-source gRPC framework is used to carry data based on HTTP2. HTTP2 supports the push mode and has a higher speed and bandwidth usage than HTTP1.

The open-source gRPC framework supports multiple languages. The server and client can use different programming languages. Seamless interconnection can be implemented if the same .proto file is used, simplifying the interconnection between the collector and devices and greatly reducing interconnection costs.

3.4.4 High Availability

Telemetry service configurations are saved when the system is restarted or an active/standby switchover is performed. After the restart or active/standby switchover is complete, telemetry reloads its configurations and continues to run.

3.4.5 Second- and Subsecond-Level Sampling

Telemetry supports data collection every few seconds or subseconds.

3.4.6 AAA Authentication

The dial-in mode supports AAA authentication, ensuring device security.

3.4.7 TLS Encryption

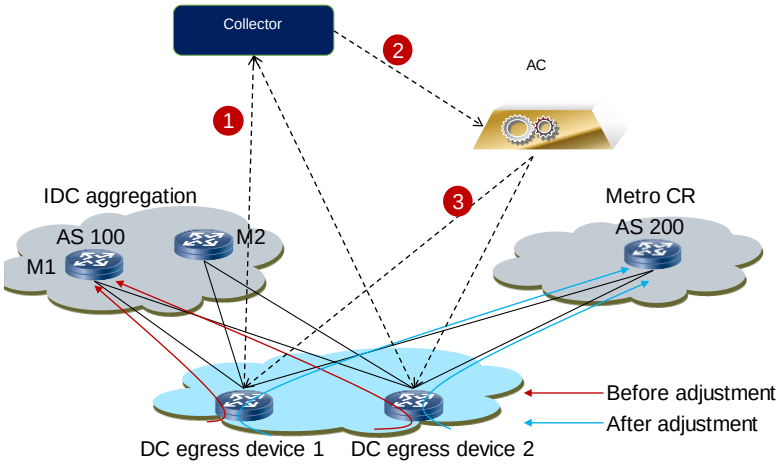
Both the dial-in and dial-out modes of telemetry support TLS 1.2 encryption, ensuring the security of data transmission channels.

4 Telemetry Application Scenarios

4.1 Traffic Optimization

Telemetry collects data from a large number of monitoring devices and sends the collected data to the network analyzer for comprehensive analysis and decision-making. The network analyzer adjusts device configurations based on the decision and presents the device status after the adjustment in quasi-real-time mode.

Figure 4-1 Telemetry-based traffic optimization



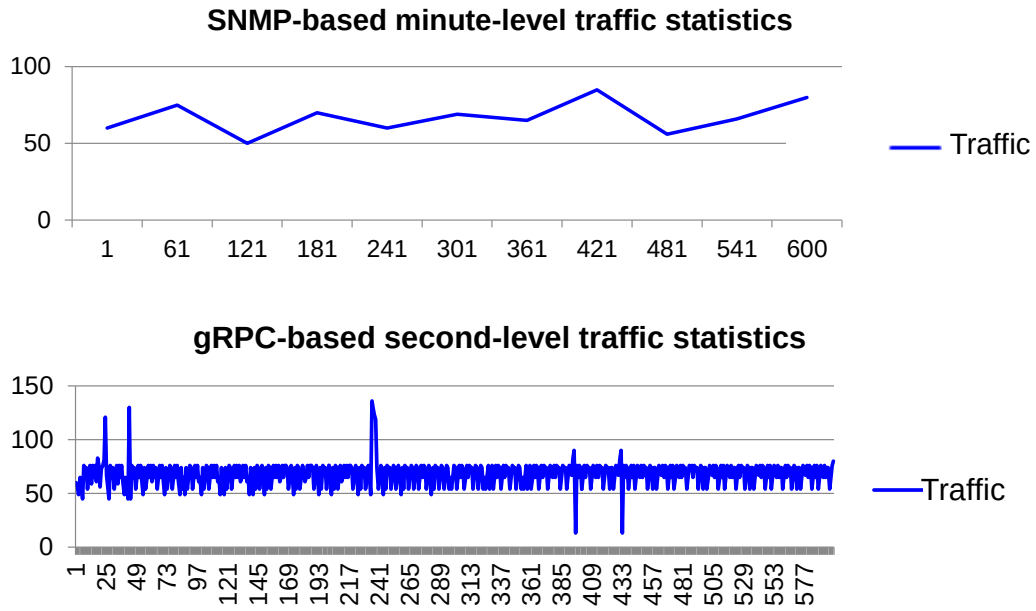
In Figure 4-1, if traffic of the DC egress devices is transmitted through the red links and the optimal path changes at a specified time, telemetry collects data and pushes it to the network analyzer. The network analyzer performs analysis, makes a decision, and sends configurations to the DC egress devices to change the next hop from M1 to M3 and transmit traffic through the blue links. Feedback on the traffic after the adjustment can be immediately provided to the network analyzer.

This solution achieves much less delay in changing traffic paths and providing feedback to the network analyzer than the SNMP solution. This makes users unaware of the traffic path switching (users can perceive a long delay in the SNMP solution) and facilitates O&M.

4.2 Microburst Detection

Network traffic is generally measured by the average period of 5 minutes and seems to be stable. However, at lower levels of granularity, for example, milliseconds, many microbursts occur in the actual traffic and cannot be perceived by Get operations in the SNMP solution.

Figure 4-2 Microburst detection



The traffic statistics collected every few minutes using the SNMP solution show that the traffic is stable and no network exception occurs. The traffic statistics collected every few seconds using gPRC show that there are microbursts. Telemetry supports high precision sampling and is capable of detecting microbursts.

5 Summary

Telemetry is a new network monitoring method that collects network device data at a higher precision, providing the basis for network fault locating and network traffic optimization. Telemetry enables the evolution from network quality analysis to Big Data analytics, facilitating intelligent O&M.

A Reference

http://openconfig.net/