

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2016

Q. Wu
J. Strassner
Huawei
A. Farrel
Old Dog Consulting
L. Zhang
Huawei
March 9, 2016

Network Telemetry and Big Data Analysis
draft-wu-t2trg-network-telemetry-00

Abstract

This document focuses on network measurement and analysis in the network environment. It first defines network telemetry, describes an exemplary network telemetry architecture, and then explores the characteristics of network telemetry data. It ends with detailing a set of issues with retrieving and processing network telemetry data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	The definition of Network Telemetry	3
3.	Network Telemetry architecture	3
4.	Measurement data Characteristics	6
5.	Issues	7
5.1.	Data Fetching Efficiency	7
5.2.	Existing Network Level Metrics Inefficiency issue	7
5.3.	Measurement data format consistency issue	9
5.4.	Data Correlation issue	9
5.5.	Data Synchronization Issues	10
6.	Informative References	10
Appendix A.	Network Telemetry data source Classification	12
Appendix B.	Existing Network Data Collection Methods	12
B.1.	Network Log Collection	12
B.1.1.	Text based data collection	13
B.1.2.	SNMP Trap	13
B.1.3.	Syslog based Collection	13
B.2.	Network Traffic Collection	13
B.3.	Network Performance Collection	14
B.4.	Network Faults Collection	14
B.5.	Network Topology data Collection	14
B.6.	Other Data Collection	14
	Authors' Addresses	15

[1.](#) Introduction

Today, billions of devices can connect to the internet and VPN and establish a good ecosystem of connectivity. Our daily life also has been greatly changed with a large number of IoT applications and mobile application being built on top of it (e.g., smart tags on many daily life objects, wearable health monitoring sensors, smartphones, intelligent cars, and smart home appliances). However, the increased amount of connection of devices and the proliferation of web and multimedia services also imposes a great impact on the network. Examples include:

- o The massive scale and highly dynamic nature of the IoT applications and mobile applications (e.g., interaction with other thing at anytime and in any location)

- o The increasingly vast amounts of data gathered from the network enviroment at varying speeds, with different amounts of accuracy, and the new communication patterns created
- o The disparate types of pre- and post-processing necessary to understand the meaning and context (e.g., semantics) of measured data

Therefore the network may be subject to increased network incidents and unregulated network changes, without better network visibility or a good view of the available network resources and network topology, it is not easy to

- o schedule network resource to adapt to near real-time service demands
- o measure the network performance and assess network quality as a whole
- o provide quick network diagnosis, prove network innocence when the application quality get worse or identify what parts of the network can cause problems if a network glitch or service interruption happens.

In this document, we first define network telemetry in the context of network environment, followed by an exemplary architecture for collecting and processing telemetry data. We then explore the characteristics of network telemetry data, and end with describing a set of issues with retrieving and processing network telemetry data.

[2.](#) The definition of Network Telemetry

Network Telemetry describes how information from various data sources can be collected using a set of automated communication processes and transmitted to one or more receiving equipment for analysis tasks. Analysis tasks may include event correlation, anomaly detection,

performance monitoring, metric calculation, trend analysis, and other related processes.

3. Network Telemetry architecture

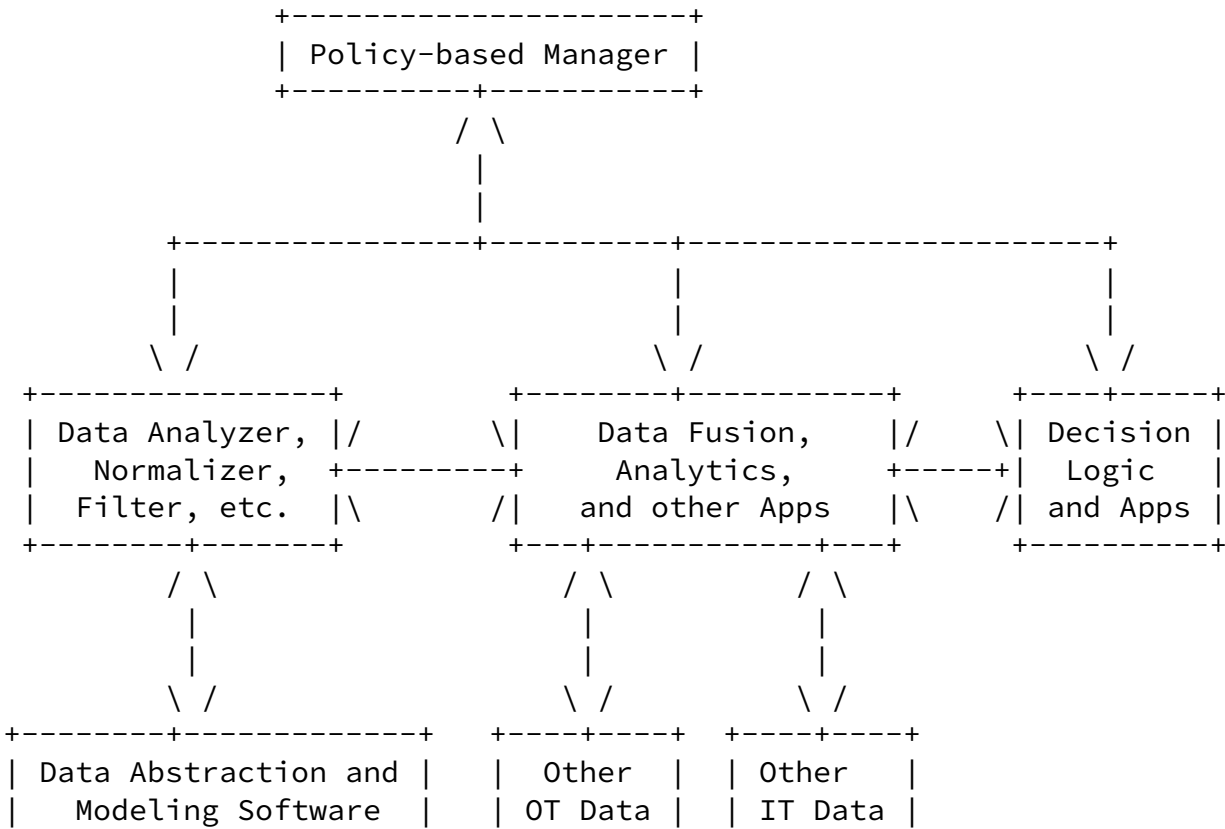
A Network Telemetry architecture describes how different types of Network Telemetry data are transmitted from different network sources and received by different collection entities. In an ideal network telemetry architecture, the ability to collect data should be independent of any specific application and vendor limitations. This means that protocol and data format translation are required, so that

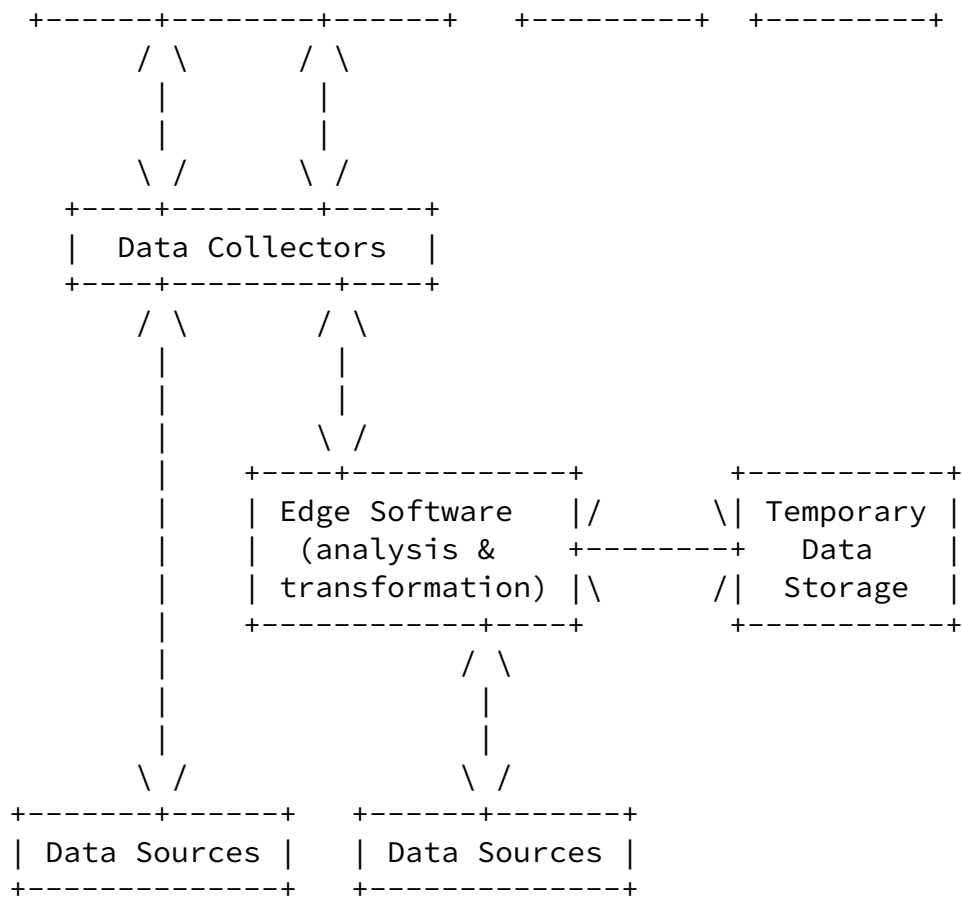
a normalized form of data can be used to simplify the various analysis and processing tasks required.

The Network Telemetry architecture is made up of the following three key functional components:

- o Data Source: The Data Source can be any type of network device that generates data. Examples include the management system that accesses IGP/BGP routing information, network inventory, topology, and resource data, as well as other types of information that provides data to be measured and/or contextual information to better understand the network telemetry data.
- o Data Collector: The Data Collector may be a part of a control and/or management system (e.g., NMS/OSS, SDN Controller, or OAM system) and/or a dedicated set of entities. It gathers data from various Data Sources, and performs processing tasks to feed raw and/or processed data to the Data Analyzer.
- o Data Analyzer: The Data Analyzer processes data from various data collectors to provide actionable insight. This ranges from generating simple statistical metrics to inferring problems to recommending solutions to said problems.

Figure 1 shows an exemplary architecture for network telemetry and analysis.





Network Telemetry and Analysis Architecture

- o Data Abstraction and Modeling Software. This component uses an overarching information model to define relevant terms, objects, and values that all components in the Network Telemetry Architecture can use.
- o Edge Software refers to performing compute, storage, and/or networking functions on nodes at the edges of a network. This enables processing of data to occur at or near the source of the data. Figure 4 shows that some information from some Data Sources may be sent directly to Data Collectors, while other data may be sent first to Edge Software for further processing before it is consumed by Data Collectors.
- o Policy-based Manager. This component is responsible for managing different aspects of the Network Telemetry Architecture in a

distributed and extensible manner through the use of a set of policies that govern the behavior of the system. Examples include defining rules that determine what data to collect when, where, and how, as well as defining rules that, given a specific context, determine how to process collected data.

This reference architecture assumes that Data Collectors can choose different measurement data formats to gather measurement data, and different protocols to transmit said data; the Data Abstraction and Modeling Software normalizes collected data into a common form. Both the Data Collector and the Data Analyzer may support data filtering, correlation, and other types of data processing mechanisms. In the above architecture, bi-directional communication is shown for generality. This may be implemented a number of different ways, such as using a request-response mechanism, a publish-subscribe mechanism, or even as a set of uni-directional (e.g., push and pull) requests.

[4.](#) Measurement data Characteristics

Measurement data is generated from different data sources, and has varying characteristics, including (but not limited to):

- o Measurement data can be any of network performance data, network logging data, network warning and defects data, network statistics and state data, and network resource operation data (e.g., operations on RIBs and FIBs[RFC4984]).
- o Most measurement data are monitor state data rather than configuration data. However, on occasion, network configuration data may also be included (e.g., to establish context for the measurement data).

- o In many cases, telemetry data requires real time delivery with high throughput, multi-channel data collection mechanisms.
- o In most cases, the required frequency of access to monitoring state data is extremely high.

[5.](#) Issues

[5.1.](#) Data Fetching Efficiency

Today, the existing data fetching methods (See [appendix B](#)) prove insufficiency due to the following factors:

- o The existing Network management protocol is not dedicated and also not sufficient for data collection.
 - * E.g.,NETCONF more focus on network configuration, only retrieve operational data
- o SNMP relies on Periodic fetching. Periodic fetching of data is not an adequate solution for many types of applications
 - * E.g., Applications that require frequent update to the stored data

In addition, it adds significant load on participating networks, devices, and applications

- o We increasingly rely on RPC-style interactions [[RFC5531](#)] to fetch data on demand by application. However most of applications are interested in update of the data or change to the data.
- o When data fetching protocol is selected, human readable format such as XML, JSON to encode structured data enable us to parse without knowing schema, however it lacks efficiency on the wire.

[5.2.](#) Existing Network Level Metrics Inefficiency issue

Quality of Service (QoS) and Quality of Experience (QoE) assessment [[RFC7266](#)] of multimedia services has been well studied in ITU-T SG 12. Media quality is commonly expressed in terms of MoS (Mean Opinion Score) [[RFC3611](#)][G107]. MoS is typically rated on a scale from 1 to 5, in which 5 represents excellent and 1 represents unacceptable. When multimedia application quality becomes bad, it is hard to know whether this is network problem or application specific problem(e.g.,Codec type, Coding bit rate, packetization scheme, loss recovery technique,the interaction between transport problems and application-layer protocols). To make sure this is not network

is, network health index or network key performance Index(KPI) or key quality index(KQI) becomes important.

However, QoS/QoE assessment of network service that is dependent on or not dependent on the underlying network technology (e.g., MPLS, IP) is not well studied or defined in any body or organization. The QoS/QoE of generic network services requires a set of appropriate network performance, reliability, or other metric definitions. This may take the form of key quality and or performance indicators, ranging from high-level metrics (e.g., dropped calls) to low-level metrics (e.g., packet loss, delay, and jitter). IP service performance parameters are defined in ITU-T Y.1540 [[Y1540](#)]; however, these existing network performance metrics are proving insufficient due to several factors:

- o These transport-specific metrics are defined for specific technologies. For example, network performance parameters in Y.1540 are only designed for IP networks, and do not apply to connection- oriented networks, such as an MPLS-TP network.
- o Not all the metrics are end-to-end performance metrics at the network level. For example, the TE performance metric defined in ISIS-TE [[RFC5305](#)] is only defined for per link usage.
- o These transport specific metrics are all single objective metrics; there are no transport specific metrics defined as multi-objective metrics. For example, IP transfer Delay (IPTD) is a single-objective metric and cannot be used to measure similar and important performance behaviors such as IP packet Delay Variation[Y1541]).
- o Different services have different performance requirements. It is hard to measure network QoS to satisfy all possible services using a single metric.
- o Transport-specific metrics are not applied to the whole network, but to a specific flow passing through the network corresponding to matched QoS classes.
- o If there are multiple paths from source to destination in the IP network, then transport-specific metrics change with the path selected and it may be also hard to know which path the packet will traverse.

[5.3.](#) Measurement data format consistency issue

The data format is typically vendor- and device-specific. This also means that different commands, having different syntax and semantics characteristics that use different protocols, may have to be issued to retrieve the same type of data from different devices.

The Data Analyzer may need to ingest data in a specific format that is not supported by the Data Collectors that service it. For example, the ALTO data format used between a data source and a Data Collector generates an abstracted network topology and provides it to network-aware applications (i.e., a Data Analyzer) over a web service based API [[I-D.wu-alto-te-metrics](#)]. In this case, prefix data in the network topology information need to be generated into ALTO Network Maps, TE (topology) data needs to be generated into ALTO Cost Maps. To provide better data format mapping, ALTO Network Map and Cost MAP need to be modeled in the same way as prefix data and TE data in the network topology information. However, these data use different data formats, and do not have a common model structure to represent them in a consistent way.

This is why the architecture shown in Figure 1 has a "Data Abstraction and Modeling Software" component. This component normalizes all data received into a common format for analysis and processing by the Data Analyzer. If this component is not present, then the Data Analyzer would have to deal with m vendor devices \times n versions of software for each device at a minimum. Furthermore, different protocols have different capabilities, and may or may not be able to transmit and receive different types of data. The Data Abstraction and Modeling Software component can provide information that defines the structure of data that should be received; this can be useful for checking for incomplete collection data as well as missing collection data.

[5.4.](#) Data Correlation issue

To provide consistent configuration, reporting and representation for OAM information, the LIME YANG model [[I-D.draft-ietf-lime-yang-oam-model-01](#)] is proposed to correlate defects, faults, and network failures between the different layers and irregardless of network technologies. This helps improve efficiency of fault detection and localization, and provide better OAM visibility.

Today we see large amounts of data collected from different data sources. These data can be network log data, network event data, network performance data, network fault data, network statistics

state, network operation state. However, these data can only be meaningful if they are correlated in time and space. In particular,

useful trend analysis and anomaly detection depend on proper correlation of the data collected from the different Data Sources. In addition, Correlate different type data from different Data Sources with time or space can provide better network visibility. But such correlations is still an challenging issue.

[5.5](#). Data Synchronization Issues

When retrieving data from Data Sources or Data Collectors, synchronization the same type of data between data source and data collector or between data collector and data analyzer is a complicated thing.

- o Arrange src and dst synchronized, especially when multiple source feed one data collector, or multiple data collector feed one data analyzer
- o Aggregate data from different data source and synchronize the data to the data analyzer is also not easy task.

The reference architecture of Figure 1 defines a "Policy-based Manager" to manage the set of data that are collected how, when, where, and by which devices. This component provides mechanisms that help ensure that needed information is collected by the appropriate components of the Network Telemetry Architecture. It also facilitates the synchronization of different components that make up the Network Telemetry Architecture, since these are likely distributed throughout one or more networks.

It also provides a mechanism for the Data Analyzer, or other applications (e.g., the "Data Fusion, Analytics, and other Apps", as well as the "Decision Logic and Apps" components in Figure 1) to provide information to the Policy-based Manager in the form of feedback (e.g., see [I-D.[draft-strassner-anima-control-loops-01](#)]).

[6](#). Informative References

- [G107] ITU-T, "The E-model: a computational model for use in transmission planning", ITU-T Recommendation G.107, June

2015.

[I-D.ietf-idr-ls-distribution]

Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", [draft-ietf-idr-ls-distribution-13](#) (work in progress), October 2015.

Wu, et al.

Expires September 10, 2016

[Page 10]

Internet-Draft

Network Telemetry and Big Data

March 2016

[I-D.ietf-idr-te-pm-bgp]

Wu, Q., Previdi, S., Gredler, H., Ray, S., and J. Tantsura, "BGP attribute for North-Bound Distribution of Traffic Engineering (TE) performance Metrics", [draft-ietf-idr-te-pm-bgp-02](#) (work in progress), January 2015.

[I-D.ietf-lime-yang-oam-model]

Senevirathne, T., Finn, N., Kumar, D., Salam, S., Wu, Q., and Z. Wang, "Generic YANG Data Model for Connection Oriented Operations, Administration, and Maintenance(OAM) protocols", [draft-ietf-lime-yang-oam-model-02](#) (work in progress), February 2016.

[I-D.strassner-anima-control-loops]

Strassner, J., "The Use of Control Loops in Autonomic Networking", [draft-strassner-anima-control-loops-00](#) (work in progress), October 2015.

[I-D.wu-alto-te-metrics]

Wu, W., Yang, Y., Lee, Y., Dhody, D., and S. Randriamasy, "ALTO Traffic Engineering Cost Metrics", [draft-wu-alto-te-metrics-06](#) (work in progress), April 2015.

[RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", [RFC 3611](#), DOI 10.17487/RFC3611, November 2003, <<http://www.rfc-editor.org/info/rfc3611>>.

[RFC4984] Meyer, D., Ed., Zhang, L., Ed., and K. Fall, Ed., "Report from the IAB Workshop on Routing and Addressing", [RFC 4984](#), DOI 10.17487/RFC4984, September 2007, <<http://www.rfc-editor.org/info/rfc4984>>.

- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", [RFC 5305](#), DOI 10.17487/RFC5305, October 2008, <<http://www.rfc-editor.org/info/rfc5305>>.
- [RFC5531] Thurlow, R., "RPC: Remote Procedure Call Protocol Specification Version 2", [RFC 5531](#), DOI 10.17487/RFC5531, May 2009, <<http://www.rfc-editor.org/info/rfc5531>>.
- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", [RFC 5693](#), DOI 10.17487/RFC5693, October 2009, <<http://www.rfc-editor.org/info/rfc5693>>.

Wu, et al.

Expires September 10, 2016

[Page 11]

Internet-Draft

Network Telemetry and Big Data

March 2016

- [RFC7266] Clark, A., Wu, Q., Schott, R., and G. Zorn, "RTP Control Protocol (RTCP) Extended Report (XR) Blocks for Mean Opinion Score (MOS) Metric Reporting", [RFC 7266](#), DOI 10.17487/RFC7266, June 2014, <<http://www.rfc-editor.org/info/rfc7266>>.
- [Y1540] ITU-T, "Internet protocol data communication service - IP packet transfer and availability performance parameters", ITU-T Recommendation Y.1540, March 2011.
- [Y1541] ITU-T, "Network performance objectives for IP-based services", ITU-T Recommendation Y.1541, December 2011.

[Appendix A](#). Network Telemetry data source Classification

Data Source Category	Information
Network Data	Usage records
	Performance Monitoring Data
	Fault Monitoring Data
	Real Time Traffic Data
	Real Time Statistics Data
	Network Configuration Data

	Provision Data
Subscriber Data	Profile Data Network Registry Operation Data Billing Data
Application Data derived from interfaces, channels, software, etc.	Traffic Analysis Web, Search, SMS, Email Social Media Data Mobile apps

[Appendix B](#). Existing Network Data Collection Methods

[B.1](#). Network Log Collection

There are three typical Log data Collection methods:

- o Text based Collection

Wu, et al.

Expires September 10, 2016

[Page 12]

Internet-Draft

Network Telemetry and Big Data

March 2016

- o SNMP Trap
- o Syslog based Collection

[B.1.1](#). Text based data collection

Text base Log data is designed for low speed network. The amount of IoT data can not be too large. It only can be parsed by the network personnel with experience to define such kind of Log. The log data can be transferred either by Email or via FTP. The difference between using Email and using FTP are:

- o The volume of data transferred by FTP can be much larger than via Email.
- o FTP based collection is active data collection while Email based collection is passive data collection

[B.1.2.](#) SNMP Trap

SNMP Trap is a notification mechanism which enables an agent to notify the management system of significant events by way of an unsolicited SNMP message. In case there are large number of devices and each device has large number of objects, SNMP Trap is more efficient to get the data than polling information from every object on every device.

[B.1.3.](#) Syslog based Collection

Syslog protocol is used to convey event notification messages and allows the use of any number of transport protocols for transmission of syslog messages. It is widely used in the network device((e.g., switch, router) .

[B.2.](#) Network Traffic Collection

Network Traffic Collection is a process of exporting network traffic flow information from routers, probes and other devices. It doesn't care operation state on the network device but traffic flow characteristic on the links between any two adjacent network device. Take IPFIX as an example, it is widely adopted in the router and switch to get IP traffic flow information for the network management system.

[B.3.](#) Network Performance Collection

Network performance collection is a process of exporting network performance information from routers, probes and other devices. The network performance information can be applied to the quality, performance, and reliability of data delivery services and applications running over network. It is also applied to traffic contract agreed by the user and the network service provider. Measurement mechanism defined in IPPM WG and OAM technology and OAM tools can be used to perform performance measurement.

[B.4.](#) Network Faults Collection

Network fault collection is a process of exporting network fault, failure, warning, defects from router, probes and other devices. It usually adopts OAM technology, OAM tools, OAM model (e.g., SNMP MIB or NETCONF YANG model) to localize fault and pinpoint fault location. However OAM YANG model is mainly focused on configure OAM functionality on the network element, how to use OAM YANG model to collect more data, e.g., warning, failure, defects and how to use these data needs to be further standardized.

[B.5.](#) Network Topology data Collection

For network topology data collection, routing protocols are important collection method, since every router need to propagate its information throughout the whole network. In addition, we can use NMS/OSS to get network topology data if they have access to network topology database or routing protocols.

Network Topology data comprise node information and link information. It can be collected in two typical ways, if the network topology data is within one IGP area or one AS, we can use ISIS protocol or OSPF to gather them and write into RIB or topology database, and then we can use I2RS protocol to read these network topology data; if the network topology data is beyond one IGP area and span across several domains, we can use BGP-LS [[I-D.ietf-idr-ls-distribution](#)][I-D.ietf-idr-te-pm-bgp] to collect network topology data in different domain and aggregated them in the central network topology database.

[B.6.](#) Other Data Collection

To collect and process large volume of data in real time or in near real time to detect subtle event and aid failure diagnosis, we can choose some other data fetching efficient tools, e.g., Facebook's Scribe, Chukwa built on top of Hadoop File subsystem to parse out structured data from some of the logs and load them into a datastore.

101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

John Strassner
Huawei
2230 Central Expressway
San Jose, CA, CA
USA

Email: john.sc.strassner@huawei.com

Adrian Farrel
Old Dog Consulting

Email: adrian@olddog.co.uk

Liang Zhang
Huawei

Email: zhangliang1@huawei.com