Network Telemetry Framework

Abstract

   Network telemetry is a technology for gaining network insight and
   facilitating efficient and automated network management. It
   encompasses various techniques for remote data generation,
   collection, correlation, and consumption. This document describes an
   architectural framework for network telemetry, motivated by
   challenges that are encountered as part of the operation of networks
   and by the requirements that ensue. This document clarifies the
   terminology and classifies the modules and components of a network
   telemetry system from different perspectives. The framework and
   taxonomy help to set a common ground for the collection of related
   work and provide guidance for related technique and standard
   developments.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Not all documents
   approved by the IESG are candidates for any level of Internet
   Standard; see Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   https://www.rfc-editor.org/info/rfc9232.

Copyright Notice

Table of Contents

1. Introduction

   Network visibility is the ability of management tools to see the
   state and behavior of a network, which is essential for successful
   network operation. Network telemetry revolves around network data
   that 1) can help provide insights about the current state of the
   network, including network devices, forwarding, control, and
   management planes; 2) can be generated and obtained through a variety
   of techniques, including but not limited to network instrumentation
   and measurements; and 3) can be processed for purposes ranging from
   service assurance to network security using a wide variety of data
   analytical techniques. In this document, network telemetry refers to
   both the data itself (i.e., "Network Telemetry Data") and the
   techniques and processes used to generate, export, collect, and
   consume that data for use by potentially automated management
   applications. Network telemetry extends beyond the classical network
   Operations, Administration, and Management (OAM) techniques and
   expects to support better flexibility, scalability, accuracy,
   coverage, and performance.

   However, the term "network telemetry" lacks an unambiguous
   definition. The scope and coverage of it cause confusion and
   misunderstandings. It is beneficial to clarify the concept and
   provide a clear architectural framework for network telemetry, so we
   can articulate the technical field and better align the related
   techniques and standard works.

   To fulfill such an undertaking, we first discuss some key
   characteristics of network telemetry that set a clear distinction
   from the conventional network OAM and show that some conventional OAM
   technologies can be considered a subset of the network telemetry
   technologies. We then provide an architectural framework for network
   telemetry that includes four modules, each associated with a

different category of telemetry data and corresponding procedures.
All the modules are internally structured in the same way, including
components that allow the operator to configure data sources in
regard to what data to generate and how to make that available to
client applications, components that instrument the underlying data
sources, and components that perform the actual rendering, encoding,
and exporting of the generated data. We show how the network
telemetry framework can benefit current and future network
operations. Based on the distinction of modules and function
components, we can map the existing and emerging techniques and
protocols into the framework. The framework can also simplify
designing, maintaining, and understanding a network telemetry system.
In addition, we outline the evolution stages of the network telemetry
system and discuss the potential security concerns.

The purpose of the framework and taxonomy is to set a common ground
for the collection of related work and provide guidance for future
technique and standard developments. To the best of our knowledge,
this document is the first such effort for network telemetry in
industry standards organizations. This document does not define
specific technologies.

1.1. Applicability Statement

Large-scale network data collection is a major threat to user privacy
and may be indistinguishable from pervasive monitoring [RFC7258].
The network telemetry framework presented in this document must not
be applied to generating, exporting, collecting, analyzing, or
retaining individual user data or any data that can identify end
users or characterize their behavior without consent. Based on this
principle, the network telemetry framework is not applicable to
networks whose endpoints represent individual users, such as general-
purpose access networks.

1.2. Glossary

Before further discussion, we list some key terminology and
abbreviations used in this document. There is an intended
differentiation between the terms of network telemetry and OAM.
However, it should be understood that there is not a hard-line
distinction between the two concepts. Rather, network telemetry is
considered an extension of OAM. It covers all the existing OAM
protocols but puts more emphasis on the newer and emerging techniques
and protocols concerning all aspects of network data from acquisition
to consumption.

AI: Artificial Intelligence. In the network domain, AI
            refers to machine-learning-based technologies for
            automated network operation and other tasks.

AM: Alternate Marking. A flow performance measurement
            method, as specified in [RFC8321].

BMP: BGP Monitoring Protocol. Specified in [RFC7854].

DPI: Deep Packet Inspection. Refers to the techniques that
            examine packets beyond packet L3/L4 headers.

gNMI: gRPC Network Management Interface. A network management
            protocol from the OpenConfig Operator Working Group,
            mainly contributed by Google. See [gnmi] for details.

GPB: Google Protocol Buffer. An extensible mechanism for
            serializing structured data. See [gpb] for details.

gRPC: gRPC Remote Procedure Call. An open-source high-
            performance RPC framework that gNMI is based on. See
            [grpc] for details.

IPFIX: IP Flow Information Export Protocol. Specified in
            [RFC7011].

IOAM: In situ OAM [RFC9197]. A data plane on-path telemetry
               technique.

JSON: JavaScript Object Notation. An open standard file format
               and data interchange format that uses human-readable text
               to store and transmit data objects, as specified in
               [RFC8259].

MIB: Management Information Base. A database used for
               managing the entities in a network.

NETCONF: Network Configuration Protocol. Specified in [RFC6241].

NetFlow: A Cisco protocol used for flow record collecting, as
               described in [RFC3954].

Network Telemetry: The process and instrumentation for acquiring and
               utilizing network data remotely for network monitoring
               and operation. A general term for a large set of network
               visibility techniques and protocols, concerning aspects
               like data generation, collection, correlation, and
               consumption. Network telemetry addresses current network
               operation issues and enables smooth evolution toward
               future intent-driven autonomous networks.

NMS: Network Management System. Refers to applications that
               allow network administrators to manage a network.

OAM: Operations, Administration, and Maintenance. A group of
               network management functions that provide network fault
               indication, fault localization, performance information,
               and data and diagnosis functions. Most conventional
               network monitoring techniques and protocols belong to
               network OAM.

PBT: Postcard-Based Telemetry. A data plane on-path telemetry
               technique. A representative technique is described in
               [IPPM-IOAM-DIRECT-EXPORT].

RESTCONF: An HTTP-based protocol that provides a programmatic
               interface for accessing data defined in YANG, using the
               datastore concepts defined in NETCONF, as specified in
               [RFC8040].

SMIv2:   Structure of Management Information Version 2. Defines
               MIB objects, as specified in [RFC2578].

SNMP: Simple Network Management Protocol. Versions 1, 2, and 3
               are specified in [RFC1157], [RFC3416], and [RFC3411],
               respectively.

XML: Extensible Markup Language. A markup language for data
               encoding that is both human readable and machine
               readable, as specified by W3C [W3C.REC-xml-20081126].

YANG: YANG is a data modeling language for the definition of
               data sent over network management protocols such as
               NETCONF and RESTCONF. YANG is defined in [RFC6020] and
               [RFC7950].

YANG ECA: A YANG model for Event-Condition-Action policies, as
               defined in [NETMOD-ECA-POLICY].

YANG-Push: A mechanism that allows subscriber applications to
               request a stream of updates from a YANG datastore on a
               network device. Details are specified in [RFC8639] and
               [RFC8641].

2. Background

The term "big data" is used to describe the extremely large volume of
data sets that can be analyzed computationally to reveal patterns,
trends, and associations. Networks are undoubtedly a source of big
data because of their scale and the volume of network traffic they
forward. When a network's endpoints do not represent individual
users (e.g., in industrial, data-center, and infrastructure
contexts), network operations can often benefit from large-scale data
collection without breaching user privacy.

Today, one can access advanced big data analytics capability through
a plethora of commercial and open-source platforms (e.g., Apache
Hadoop), tools (e.g., Apache Spark), and techniques (e.g., machine
learning). Thanks to the advance of computing and storage
technologies, network big data analytics give network operators an
opportunity to gain network insights and move towards network
autonomy. Some operators start to explore the application of
Artificial Intelligence (AI) to make sense of network data. Software
tools can use the network data to detect and react on network faults,
anomalies, and policy violations, as well as predict future events.
In turn, the network policy updates for planning, intrusion
prevention, optimization, and self-healing may be applied.

It is conceivable that an autonomic network [RFC7575] is the logical
next step for network evolution following Software-Defined Networking
(SDN), which aims to reduce (or even eliminate) human labor, make
more efficient use of network resources, and provide better services
more aligned with customer requirements. The IETF ANIMA Working
Group is dedicated to developing and maintaining protocols and
procedures for automated network management and control of
professionally managed networks. The related technique of
Intent-Based Networking (IBN) [NMRG-IBN-CONCEPTS-DEFINITIONS]
requires network visibility and telemetry data in order to ensure
that the network is behaving as intended.

However, while the data processing capability is improved and
applications require more data to function better, the networks lag
behind in extracting and translating network data into useful and
actionable information in efficient ways. The system bottleneck is
shifting from data consumption to data supply. Both the number of
network nodes and the traffic bandwidth keep increasing at a fast
pace. The network configuration and policy change at smaller time
slots than before. More subtle events and fine-grained data through
all network planes need to be captured and exported in real time. In
a nutshell, it is a challenge to get enough high-quality data out of
the network in a manner that is efficient, timely, and flexible.
Therefore, we need to survey the existing technologies and protocols
and identify any potential gaps.

In the remainder of this section, we first clarify the scope of
network data (i.e., telemetry data) relevant in this document. Then,
we discuss several key use cases for network operations of today and
the future. Next, we show why the current network OAM techniques and
protocols are insufficient for these use cases. The discussion
underlines the need for new methods, techniques, and protocols, as
well as the extensions of existing ones, which we assign under the
umbrella term "Network Telemetry".

2.1. Telemetry Data Coverage

Any information that can be extracted from networks (including the
data plane, control plane, and management plane) and used to gain
visibility or as a basis for actions is considered telemetry data.
It includes statistics, event records and logs, snapshots of state,
configuration data, etc. It also covers the outputs of any active
and passive measurements [RFC7799]. In some cases, raw data is
processed in network before being sent to a data consumer. Such
processed data is also considered telemetry data. The value of
telemetry data varies. In some cases, if the cost is acceptable,
less but higher-quality data are preferred rather than a lot of low-
quality data. A classification of telemetry data is provided in
Section 3. To preserve the privacy of end users, no user packet

content should be collected. Specifically, the data objects
generated, exported, and collected by a network telemetry application
should not include any packet payload from traffic associated with
end-user systems.

2.2. Use Cases

   The following set of use cases is essential for network operations.
   While the list is by no means exhaustive, it is enough to highlight
   the requirements for data velocity, variety, volume, and veracity,
   the attributes of big data, in networks.

   * Security: Network intrusion detection and prevention systems need
      to monitor network traffic and activities and act upon anomalies.
      Given increasingly sophisticated attack vectors coupled with
      increasingly severe consequences of security breaches, new tools
      and techniques need to be developed, relying on wider and deeper
      visibility into networks. The ultimate goal is to achieve
      security with no, or only minimal, human intervention and without
      disrupting legitimate traffic flows.

   * Policy and Intent Compliance: Network policies are the rules that
      constrain the services for network access, provide service
      differentiation, or enforce specific treatment on the traffic.
      For example, a service function chain is a policy that requires
      the selected flows to pass through a set of ordered network
      functions. Intent, as defined in [NMRG-IBN-CONCEPTS-DEFINITIONS],
      is a set of operational goals that a network should meet and
      outcomes that a network is supposed to deliver, defined in a
      declarative manner without specifying how to achieve or implement
      them. An intent requires a complex translation and mapping
      process before being applied on networks. While a policy or
      intent is enforced, the compliance needs to be verified and
      monitored continuously by relying on visibility that is provided
      through network telemetry data. Any violation must be reported
      immediately - this will alert the network administrator to the
      policy or intent violation and will potentially result in updates
      to how the policy or intent is applied in the network to ensure
      that it remains in force.

   * SLA Compliance: A Service Level Agreement (SLA) is a service
      contract between a service provider and a client, which includes
      the metrics for the service measurement and remedy/penalty
      procedures when the service level misses the agreement. Users
      need to check if they get the service as promised, and network
      operators need to evaluate how they can deliver services that meet
      the SLA based on real-time network telemetry data, including data
      from network measurements.

   * Root Cause Analysis: Many network failures can be the effect of a
      sequence of chained events. Troubleshooting and recovery require
      quick identification of the root cause of any observable issues.
      However, the root cause is not always straightforward to identify,
      especially when the failure is sporadic and the number of event
      messages, both related and unrelated to the same cause, is
      overwhelming. While technologies such as machine learning can be
      used for root cause analysis, it is up to the network to sense and
      provide the relevant diagnostic data that are either actively fed
      into or passively retrieved by the root cause analysis
      applications.

   * Network Optimization: This covers all short-term and long-term
      network optimization techniques, including load balancing, Traffic
      Engineering (TE), and network planning. Network operators are
      motivated to optimize their network utilization and differentiate
      services for better Return on Investment (ROI) or lower Capital
      Expenditure (CAPEX). The first step is to know the real-time
      network conditions before applying policies for traffic
      manipulation. In some cases, microbursts need to be detected in a
      very short time frame so that fine-grained traffic control can be
      applied to avoid network congestion. Long-term planning of

network capacity and topology requires analysis of real-world
network telemetry data that is obtained over long periods of time.

* Event Tracking and Prediction: The visibility into traffic path
  and performance is critical for services and applications that
  rely on healthy network operation. Numerous related network
  events are of interest to network operators. For example, network
  operators want to learn where and why packets are dropped for an
  application flow. They also want to be warned of issues in
  advance, so proactive actions can be taken to avoid catastrophic
  consequences.

2.3. Challenges

   For a long time, network operators have relied upon SNMP [RFC3416],
   Command-Line Interface (CLI), or Syslog [RFC5424] to monitor the
   network. Some other OAM techniques as described in [RFC7276] are
   also used to facilitate network troubleshooting. These conventional
   techniques are not sufficient to support the above use cases for the
   following reasons:

   * Most use cases need to continuously monitor the network and
     dynamically refine the data collection in real time. Poll-based
     low-frequency data collection is ill-suited for these
     applications. Subscription-based streaming data directly pushed
     from the data source (e.g., the forwarding chip) is preferred to
     provide sufficient data quantity and precision at scale.

   * Comprehensive data is needed, ranging from packet processing
     engines to traffic managers, line cards to main control boards,
     user flows to control protocol packets, device configurations to
     operations, and physical layers to application layers.
     Conventional OAM only covers a narrow range of data (e.g., SNMP
     only handles data from the Management Information Base (MIB)).
     Classical network devices cannot provide all the necessary probes.
     More open and programmable network devices are therefore needed.

   * Many application scenarios need to correlate network-wide data
     from multiple sources (i.e., from distributed network devices,
     different components of a network device, or different network
     planes). A piecemeal solution is often lacking the capability to
     consolidate the data from multiple sources. The composition of a
     complete solution, as partly proposed by Autonomic Resource
     Control Architecture (ARCA) [NMRG-ANTICIPATED-ADAPTATION], will be
     empowered and guided by a comprehensive framework.

   * Some conventional OAM techniques (e.g., CLI and Syslog) lack a
     formal data model. The unstructured data hinder the tool
     automation and application extensibility. Standardized data
     models are essential to support the programmable networks.

   * Although some conventional OAM techniques support data push (e.g.,
     SNMP Trap [RFC2981][RFC3877], Syslog, and sFlow [RFC3176]), the
     pushed data are limited to only predefined management plane
     warnings (e.g., SNMP Trap) or sampled user packets (e.g., sFlow).
     Network operators require the data with arbitrary source,
     granularity, and precision, which is beyond the capability of the
     existing techniques.

   * Conventional passive measurement techniques can either consume
     excessive network resources and produce excessive redundant data
     or lead to inaccurate results; on the other hand, conventional
     active measurement techniques can interfere with the user traffic,
     and their results are indirect. Techniques that can collect
     direct and on-demand data from user traffic are more favorable.

   These challenges were addressed by newer standards and techniques
   (e.g., IPFIX/Netflow, Packet Sampling (PSAMP), IOAM, and YANG-Push),
   and more are emerging. These standards and techniques need to be
   recognized and accommodated in a new framework.

2.4. Network Telemetry

   Network telemetry has emerged as a mainstream technical term to refer
   to the network data collection and consumption techniques. Several
   network telemetry techniques and protocols (e.g., IPFIX [RFC7011] and
   gRPC [grpc]) have been widely deployed. Network telemetry allows
   separate entities to acquire data from network devices so that data
   can be visualized and analyzed to support network monitoring and
   operation. Network telemetry covers the conventional network OAM and
   has a wider scope. For instance, it is expected that network
   telemetry can provide the necessary network insight for autonomous
   networks and address the shortcomings of conventional OAM techniques.

   Network telemetry usually assumes machines as data consumers rather
   than human operators. Hence, network telemetry can directly trigger
   the automated network operation, while in contrast, some conventional
   OAM tools were designed and used to help human operators to monitor
   and diagnose the networks and guide manual network operations. Such
   a proposition leads to very different techniques.

   Although new network telemetry techniques are emerging and subject to
   continuous evolution, several characteristics of network telemetry
   have been well accepted. Note that network telemetry is intended to
   be an umbrella term covering a wide spectrum of techniques, so the
   following characteristics are not expected to be held by every
   specific technique.

   * Push and Streaming: Instead of polling data from network devices,
      telemetry collectors subscribe to streaming data pushed from data
      sources in network devices.

   * Volume and Velocity: Telemetry data is intended to be consumed by
      machines rather than by human beings. Therefore, the data volume
      can be huge, and the processing is optimized for the needs of
      automation in real time.

   * Normalization and Unification: Telemetry aims to address the
      overall network automation needs. Efforts are made to normalize
      the data representation and unify the protocols, so as to simplify
      data analysis and provide integrated analysis across heterogeneous
      devices and data sources across a network.

   * Model-Based: Telemetry data is modeled in advance, which allows
      applications to configure and consume data with ease.

   * Data Fusion: The data for a single application can come from
      multiple data sources (e.g., cross-domain, cross-device, and
      cross-layer) that are based on a common name/ID and need to be
      correlated to take effect.

   * Dynamic and Interactive: Since the network telemetry means to be
      used in a closed control loop for network automation, it needs to
      run continuously and adapt to the dynamic and interactive queries
      from the network operation controller.

   In addition, an ideal network telemetry solution may also have the
   following features or properties:

   * In-Network Customization: The data that is generated can be
      customized in network at runtime to cater to the specific need of
      applications. This needs the support of a programmable data
      plane, which allows probes with custom functions to be deployed at
      flexible locations.

   * In-Network Data Aggregation and Correlation: Network devices and
      aggregation points can work out which events and what data needs
      to be stored, reported, or discarded, thus reducing the load on
      the central collection and processing points while still ensuring
      that the right information is ready to be processed in a timely
      way.

* In-Network Processing: Sometimes it is not necessary or feasible
   to gather all information to a central point to be processed and
   acted upon. It is possible for the data processing to be done in
   network, allowing reactive actions to be taken locally.

* Direct Data Plane Export: The data originated from data plane
   forwarding chips can be directly exported to the data consumer for
   efficiency, especially when the data bandwidth is large and real-
   time processing is required.

* In-Band Data Collection: In addition to the passive and active
   data collection approaches, the new hybrid approach allows to
   directly collect data for any target flow on its entire forwarding
   path [OPSAWG-IFIT-FRAMEWORK].

It is worth noting that a network telemetry system should not be
intrusive to normal network operations by avoiding the pitfall of the
"observer effect". That is, it should not change the network
behavior and affect the forwarding performance. Moreover, high-
volume telemetry traffic may cause network congestion unless proper
isolation or traffic engineering techniques are in place, or
congestion control mechanisms ensure that telemetry traffic backs off
if it exceeds the network capacity. [RFC8084] and [RFC8085] are
relevant Best Current Practices (BCPs) in this space.

Although in many cases a system for network telemetry involves a
remote data collecting and consuming entity, it is important to
understand that there are no inherent assumptions about how a system
should be architected. While a network architecture with a
centralized controller (e.g., SDN) seems to be a natural fit for
network telemetry, network telemetry can work in distributed fashions
as well. For example, telemetry data producers and consumers can
have a peer-to-peer relationship, in which a network node can be the
direct consumer of telemetry data from other nodes.

2.5. The Necessity of a Network Telemetry Framework

Network data analytics (e.g., machine learning) is applied for
network operation automation, relying on abundant and coherent data
from networks. Data acquisition that is limited to a single source
and static in nature will in many cases not be sufficient to meet an
application's telemetry data needs. As a result, multiple data
sources, involving a variety of techniques and standards, will need
to be integrated. It is desirable to have a framework that
classifies and organizes different telemetry data sources and types,
defines different components of a network telemetry system and their
interactions, and helps coordinate and integrate multiple telemetry
approaches across layers. This allows flexible combinations of data
for different applications, while normalizing and simplifying
interfaces. In detail, such a framework would benefit the
development of network operation applications for the following
reasons:

* Future networks, autonomous or otherwise, depend on holistic and
   comprehensive network visibility. Use cases and applications are
   better when supported uniformly and coherently using an
   integrated, converged mechanism and common telemetry data
   representations wherever feasible. Therefore, the protocols and
   mechanisms should be consolidated into a minimum yet comprehensive
   set. A telemetry framework can help to normalize the technique
   developments.

* Network visibility presents multiple viewpoints. For example, the
   device viewpoint takes the network infrastructure as the
   monitoring object from which the network topology and device
   status can be acquired, and the traffic viewpoint takes the flows
   or packets as the monitoring object from which the traffic quality
   and path can be acquired. An application may need to switch its
   viewpoint during operation. It may also need to correlate a
   service and its impact on user experience (UE) to acquire the
   comprehensive information.

*   Applications require network telemetry to be elastic in order to
    make efficient use of network resources and reduce the impact of
    processing related to network telemetry on network performance.
    For example, routine network monitoring should cover the entire
    network with a low data sampling rate. Only when issues arise or
    critical trends emerge should telemetry data sources be modified
    and telemetry data rates be boosted as needed.

*   Efficient data aggregation is critical for applications to reduce
    the overall quantity of data and improve the accuracy of analysis.

A telemetry framework collects all the telemetry-related works from
different sources and working groups within the IETF. This makes it
possible to assemble a comprehensive network telemetry system and to
avoid repetitious or redundant work. The framework should cover the
concepts and components from the standardization perspective. This
document describes the modules that make up a network telemetry
framework and decomposes the telemetry system into a set of distinct
components that existing and future work can easily map to.

3. Network Telemetry Framework

The top-level network telemetry framework partitions the network
telemetry into four modules based on the telemetry data object source
and represents their relationship. Once the network operation
applications acquire the data from these modules, they can apply data
analytics and take actions. At the next level, the framework
decomposes each module into separate components. Each of these
modules follows the same underlying structure, with one component
dedicated to the configuration of data subscriptions and data
sources, a second component dedicated to encoding and exporting data,
and a third component instrumenting the generation of telemetry
related to the underlying resources. Throughout the framework, the
same set of abstract data-acquiring mechanisms and data types
(Section 3.3) are applied. The two-level architecture with the
uniform data abstraction helps accurately pinpoint a protocol or
technique to its position in a network telemetry system or
disaggregates a network telemetry system into manageable parts.

3.1. Top-Level Modules

Telemetry can be applied on the forwarding plane, control plane, and
management plane in a network, as well as on other sources out of the
network, as shown in Figure 1. Therefore, we categorize the network
telemetry into four distinct modules (management plane, control
plane, forwarding plane, and external data and event telemetry) with
each having its own interface to network operation applications.

```
        +-----------------------------+
        |                             |
        |      Network Operation      |<-------+
        |        Applications         |        |
        |                             |        |
        +-----------------------------+        |
             ^           ^         ^           |
             |           |         |           |
             V           V         |           V
        +------------+----------|---+  +----------+
        |            | Control  |   |  |          |
        |            | Plane    |   |  | External |
        |          <--->        |   |  | Data and |
        |            | Telemetry|   |  | Event    |
        | Management |       ^  V  |  | Telemetry|
        | Plane      +-------|-------+  |          |
        | Telemetry  |       V      |  +----------+
        |            | Forwarding   |
        |            | Plane        |
        |          <--->            |
        |            | Telemetry    |
        |            |              |
```

```
                  +--------------+---------------+
```
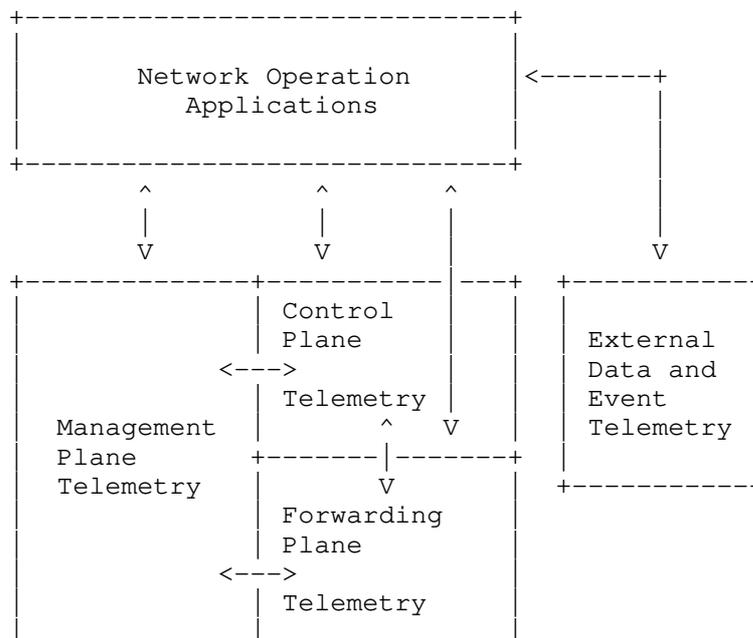
Figure 1: Modules in Layer Category of the Network Telemetry
Framework

The rationale of this partition lies in the different telemetry data
objects that result in different data sources and export locations.
Such differences have profound implications on in-network data
programming and processing capability, data encoding and the
transport protocol, and required data bandwidth and latency. Data
can be sent directly or proxied via the control and management
planes. There are advantages/disadvantages to both approaches.

Note that in some cases, the network controller itself may be the
source of telemetry data that is unique to it or derived from the
telemetry data collected from the network elements. Some of the
principles and taxonomy specific to the control plane and management
plane telemetry could also be applied to the controller when it is
required to provide the telemetry data to network operation
applications hosted outside. The scope of this document is focused
on the network elements telemetry, and further details related to
controllers are thus out of scope.

We summarize the major differences of the four modules in Table 1.
They are compared from six angles:

*   Data Object

*   Data Export Location

*   Data Model

*   Data Encoding

*   Telemetry Application Protocol

*   Data Transport Method

Data Object is the target and source of each module. Because the
data source varies, the location where data is mostly conveniently
exported also varies. For example, forwarding plane data mainly
originates as data exported from the forwarding Application-Specific
Integrated Circuits (ASICs), while control plane data mainly
originates from the protocol daemons running on the control CPU(s).
For convenience and efficiency, it is preferred to export the data
off the device from locations near the source. Because the locations
that can export data have different capabilities, different choices
of data models, encoding, and transport methods are made to balance
the performance and cost. For example, the forwarding chip has high
throughput but limited capacity for processing complex data and
maintaining state, while the main control CPU is capable of complex
data and state processing but has limited bandwidth for high
throughput data. As a result, the suitable telemetry protocol for
each module can be different. Some representative techniques are
shown in the corresponding table blocks to highlight the technical
diversity of these modules. Note that the selected techniques just
reflect the de facto state of the art and are by no means exhaustive
(e.g., IPFIX can also be implemented over TCP and SCTP, but that is
not recommended for the forwarding plane). The key point is that one
cannot expect to use a universal protocol to cover all the network
telemetry requirements.

Module	Management Plane	Control Plane	Forwarding Plane	External Data
========	==================	===============	==================	===============
Object	configuration and operation state	control protocol and signaling, RIB	flow and packet QoS, traffic stat.,	terminal, social, and environmental

			buffer and queue stat., FIB, Access Control List (ACL)	
Export Location	main control CPU	main control CPU, linecard CPU, or forwarding chip	forwarding chip or linecard CPU; main control CPU unlikely	various
Data Model	YANG, MIB, syslog	YANG, custom	YANG, custom	YANG, custom
Data Encoding	GPB, JSON, XML	GPB, JSON, XML, plain text	plain text	GPB, JSON, XML, plain text
Application Protocol	gRPC, NETCONF, RESTCONF	gRPC, NETCONF, IPFIX, traffic mirroring	IPFIX, traffic mirroring, gRPC, NETFLOW	gRPC
Data Transport	HTTP(S), TCP	HTTP(S), TCP, UDP	UDP	HTTP(S), TCP, UDP

Table 1: Comparison of Data Object Modules

Note that the interaction with the applications that consume network telemetry data can be indirect. Some in-device data transfer is possible. For example, in the management plane telemetry, the management plane will need to acquire data from the data plane. Some operational states can only be derived from data plane data sources such as the interface status and statistics. As another example, obtaining control plane telemetry data may require the ability to access the Forwarding Information Base (FIB) of the data plane.

On the other hand, an application may involve more than one plane and interact with multiple planes simultaneously. For example, an SLA compliance application may require both the data plane telemetry and the control plane telemetry.

The requirements and challenges for each module are summarized as follows (note that the requirements may pertain across all telemetry modules; however, we emphasize those that are most pronounced for a particular plane).

3.1.1. Management Plane Telemetry

The management plane of network elements interacts with the Network Management System (NMS) and provides information such as performance data, network logging data, network warning and defects data, and network statistics and state data. The management plane includes many protocols, including the classical SNMP and syslog. Regardless the protocol, management plane telemetry must address the following requirements:

*   Convenient Data Subscription: An application should have the freedom to choose which data is exported (see Section 3.3) and the means and frequency of how that data is exported (e.g., on-change or periodic subscription).

*   Structured Data: For automatic network operation, machines will replace humans for network data comprehension. Data modeling

languages, such as YANG, can efficiently describe structured data
and normalize data encoding and transformation.

*   High-Speed Data Transport: In order to keep up with the velocity
    of information, a data source needs to be able to send large
    amounts of data at high frequency. Compact encoding formats or
    data compression schemes are needed to reduce the quantity of data
    and improve the data transport efficiency. The subscription mode,
    by replacing the query mode, reduces the interactions between
    clients and servers and helps to improve the data source's
    efficiency.

*   Network Congestion Avoidance: The application must protect the
    network from congestion with congestion control mechanisms or, at
    minimum, with circuit breakers. [RFC8084] and [RFC8085] provide
    some solutions in this space.

3.1.2. Control Plane Telemetry

The control plane telemetry refers to the health condition monitoring
of different network control protocols at all layers of the protocol
stack. Keeping track of the operational status of these protocols is
beneficial for detecting, localizing, and even predicting various
network issues, as well as for network optimization, in real time and
with fine granularity. Some particular challenges and issues faced
by the control plane telemetry are as follows:

*   How to correlate the End-to-End (E2E) Key Performance Indicators
    (KPIs) to a specific layer's KPIs. For example, IPTV users may
    describe their UE by the video smoothness and definition. Then in
    case of an unusually poor UE KPI or a service disconnection, it is
    non-trivial to delimit and pinpoint the issue in the responsible
    protocol layer (e.g., the transport layer or the network layer),
    the responsible protocol (e.g., IS-IS or BGP at the network
    layer), and finally the responsible device(s) with specific
    reasons.

*   Conventional OAM-based approaches for control plane KPI
    measurement, which include Ping (L3), Traceroute (L3), Y.1731
    [y1731] (L2), and so on. One common issue behind these methods is
    that they only measure the KPIs instead of reflecting the actual
    running status of these protocols, making them less effective or
    efficient for control plane troubleshooting and network
    optimization.

*   How more research is needed for the BGP monitoring protocol (BMP).
    BMP is an example of the control plane telemetry; it is currently
    used for monitoring BGP routes and enables rich applications, such
    as BGP peer analysis, Autonomous System (AS) analysis, prefix
    analysis, and security analysis. However, the monitoring of other
    layers, protocols, and the cross-layer, cross-protocol KPI
    correlations are still in their infancy (e.g., IGP monitoring is
    not as extensive as BMP), which requires further research.

Note that the requirement and solutions for network congestion
avoidance are also applicable to the control plane telemetry.

3.1.3. Forwarding Plane Telemetry

An effective forwarding plane telemetry system relies on the data
that the network device can expose. The quality, quantity, and
timeliness of data must meet some stringent requirements. This
raises some challenges for the network data plane devices where the
first-hand data originates.

*   A data plane device's main function is user traffic processing and
    forwarding. While supporting network visibility is important, the
    telemetry is just an auxiliary function, and it should strive to
    not impede normal traffic processing and forwarding (i.e., the
    forwarding behavior should not be altered, and the trade-off
    between forwarding performance and telemetry should be well-

balanced).

*   Network operation applications require end-to-end visibility
    across various sources, which can result in a huge volume of data.
    However, the sheer quantity of data must not exhaust the network
    bandwidth, regardless of the data delivery approach (i.e., whether
    through in-band or out-of-band channels).

*   The data plane devices must provide timely data with the minimum
    possible delay. Long processing, transport, storage, and analysis
    delay can impact the effectiveness of the control loop and even
    render the data useless.

*   The data should be structured, labeled, and easy for applications
    to parse and consume. At the same time, the data types needed by
    applications can vary significantly. The data plane devices need
    to provide enough flexibility and programmability to support the
    precise data provision for applications.

*   The data plane telemetry should support incremental deployment and
    work even though some devices are unaware of the system.

*   The requirement and solutions for network congestion avoidance are
    also applicable to the forwarding plane telemetry.

Although not specific to the forwarding plane, these challenges are
more difficult for the forwarding plane because of the limited
resources and flexibility. Data plane programmability is essential
to support network telemetry. Newer data plane forwarding chips are
equipped with advanced telemetry features and provide flexibility to
support customized telemetry functions.

Technique Taxonomy: This pertains to how one instruments the
telemetry; there can be multiple possible dimensions to classify the
forwarding plane telemetry techniques.

*   Active, Passive, and Hybrid: This dimension pertains to the end-
    to-end measurement. Active and passive methods (as well as the
    hybrid types) are well documented in [RFC7799]. Passive methods
    include TCPDUMP, IPFIX [RFC7011], sFlow, and traffic mirroring.
    These methods usually have low data coverage. The bandwidth cost
    is very high in order to improve the data coverage. On the other
    hand, active methods include Ping, the One-Way Active Measurement
    Protocol (OWAMP) [RFC4656], the Two-Way Active Measurement
    Protocol (TWAMP) [RFC5357], the Simple Two-way Active Measurement
    Protocol (STAMP) [RFC8762], and Cisco's SLA Protocol [RFC6812].
    These methods are intrusive and only provide indirect network
    measurements. Hybrid methods, including IOAM [RFC9197], Alternate
    Marking (AM) [RFC8321], and Multipoint Alternate Marking
    [RFC8889], provide a well-balanced and more flexible approach.
    However, these methods are also more complex to implement.

*   In-Band and Out-of-Band: Telemetry data carried in user packets
    before being exported to a data collector is considered in-band
    (e.g., IOAM [RFC9197]). Telemetry data that is directly exported
    to a data collector without modifying user packets is considered
    out-of-band (e.g., the postcard-based approach described in
    Appendix A.3.5). It is also possible to have hybrid methods,
    where only the telemetry instruction or partial data is carried by
    user packets (e.g., AM [RFC8321]).

*   End-to-End and In-Network: End-to-end methods start from, and end
    at, the network end hosts (e.g., Ping). In-network methods work
    in networks and are transparent to end hosts. However, if needed,
    in-network methods can be easily extended into end hosts.

*   Data Subject: Depending on the telemetry objective, the methods
    can be flow based (e.g., IOAM [RFC9197]), path based (e.g.,
    Traceroute), and node based (e.g., IPFIX [RFC7011]). The various
    data objects can be packet, flow record, measurement, states, and
    signal.

3.1.4. External Data Telemetry

   Events that occur outside the boundaries of the network system are
   another important source of network telemetry. Correlating both
   internal telemetry data and external events with the requirements of
   network systems, as presented in [NMRG-ANTICIPATED-ADAPTATION],
   provides a strategic and functional advantage to management
   operations.

   As with other sources of telemetry information, the data and events
   must meet strict requirements, especially in terms of timeliness,
   which is essential to properly incorporate external event information
   into network management applications. The specific challenges are
   described as follows:

   * The role of the external event detector can be played by multiple
      elements, including hardware (e.g., physical sensors, such as
      seismometers) and software (e.g., big data sources that can
      analyze streams of information, such as Twitter messages). Thus,
      the transmitted data must support different shapes but, at the
      same time, follow a common but extensible schema.

   * Since the main function of the external event detectors is to
      perform the notifications, their timeliness is assumed. However,
      once messages have been dispatched, they must be quickly collected
      and inserted into the control plane with variable priority, which
      is higher for important sources and events and lower for secondary
      ones.

   * The schema used by external detectors must be easily adopted by
      current and future devices and applications. Therefore, it must
      be easily mapped to current data models, such as in terms of YANG.

   * As the communication with external entities outside the boundary
      of a provider network may be realized over the Internet, the risk
      of congestion is even more relevant in this context and proper
      countermeasures must be taken. Solutions such as network
      transport circuit breakers are needed as well.

   Organizing both internal and external telemetry information together
   will be key for the general exploitation of the management
   possibilities of current and future network systems, as reflected in
   the incorporation of cognitive capabilities to new hardware and
   software (virtual) elements.

3.2. Second-Level Function Components

   The telemetry module at each plane can be further partitioned into
   five distinct conceptual components:

   * Data Query, Analysis, and Storage: This component works at the
      network operation application block in Figure 1. It is normally a
      part of the network management system at the receiver side. On
      one hand, it is responsible for issuing data requirements. The
      data of interest can be modeled data through configuration or
      custom data through programming. The data requirements can be
      queries for one-shot data or subscriptions for events or streaming
      data. On the other hand, it receives, stores, and processes the
      returned data from network devices. Data analysis can be
      interactive to initiate further data queries. This component can
      reside in either network devices or remote controllers. It can be
      centralized and distributed and involve one or more instances.

   * Data Configuration and Subscription: This component manages data
      queries on devices. It determines the protocol and channel for
      applications to acquire desired data. This component is also
      responsible for configuring the desired data that might not be
      directly available from data sources. The subscription data can
      be described by models, templates, or programs.

*   Data Encoding and Export: This component determines how telemetry
    data is delivered to the data analysis and storage component with
    access control. The data encoding and the transport protocol may
    vary due to the data export location.

*   Data Generation and Processing: The requested data needs to be
    captured, filtered, processed, and formatted in network devices
    from raw data sources. This may involve in-network computing and
    processing on either the fast path or the slow path in network
    devices.

*   Data Object and Source: This component determines the monitoring
    objects and original data sources provisioned in the device. A
    data source usually just provides raw data that needs further
    processing. Each data source can be considered a probe. Some
    data sources can be dynamically installed, while others will be
    more static.

```
                +----------------------------------------+
                +--------------------------------------+ |
                |                                      | |
                |    Data Query, Analysis, & Storage   | |
                |                                      | +
                +-------+++ --------------------------+
                        |||                  ^^^
                        |||                  |||
                        ||V                  |||
                    +--+V-----------------+++-----------+
                    +-----V---------------+-----------+ |
                    +--------------------+-------+---------+ |
                    | Data Configuration |       |         | ||
                    | & Subscription     | Data Encoding   | ||
                    | (model, template,  | & Export        | ||
                    |  & program)        |                 | ||
                    +--------------------+-----------------+ ||
                    |                                        ||
                    |        Data Generation                 ||
                    |        & Processing                     ||
                    |                                        ||
                    +---------------------------------------  ||
                    |                                        ||
                    |    Data Object and Source            | |-+
                    |                                      | -+
                    |                                      |
                    +--------------------------------------+
```
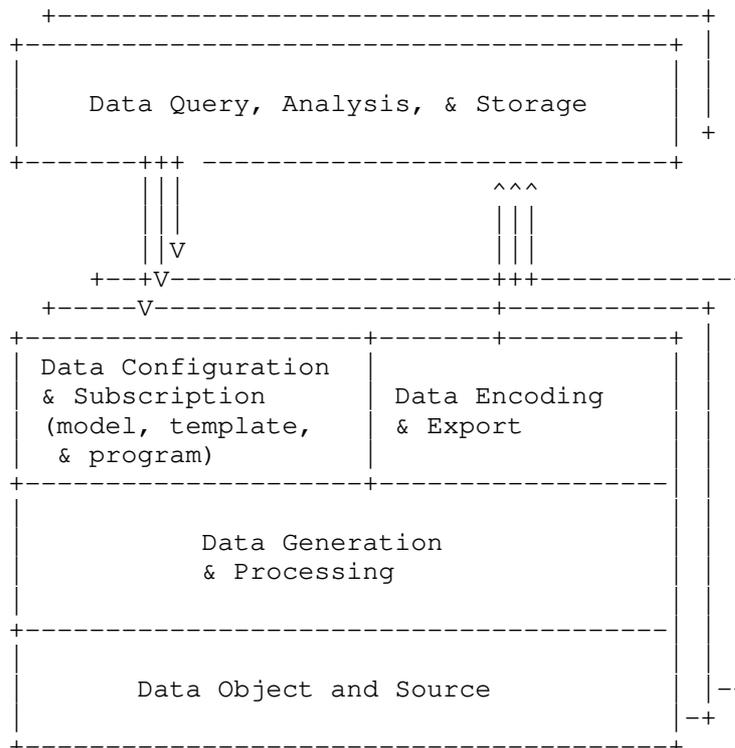
            Figure 2: Components in the Network Telemetry Framework

3.3. Data Acquisition Mechanism and Type Abstraction

   Broadly speaking, network data can be acquired through subscription
   (push) and query (poll). A subscription is a contract between
   publisher and subscriber. After initial setup, the subscribed data
   is automatically delivered to registered subscribers until the
   subscription expires. There are two variations of subscription. The
   subscriptions can be predefined, or the subscribers are allowed to
   configure and tailor the published data to their specific needs.

   In contrast, queries are used when a client expects immediate and
   one-off feedback from network devices. The queried data may be
   directly extracted from some specific data source or synthesized and
   processed from raw data. Queries work well for interactive network
   telemetry applications.

   In general, data can be pulled (i.e., queried) whenever needed, but
   in many cases, pushing the data (i.e., subscription) is more
   efficient, and it can reduce the latency of a client detecting a
   change. From the data consumer point of view, there are four types
   of data from network devices that a telemetry data consumer can
   subscribe or query:

*   Simple Data: Data that are steadily available from some datastore

or static probes in network devices.

* Derived Data: Data that need to be synthesized or processed in the
   network from raw data from one or more network devices. The data
   processing function can be statically or dynamically loaded into
   network devices.

* Event-triggered Data: Data that are conditionally acquired based
   on the occurrence of some events. An example of event-triggered
   data could be an interface changing operational state between up
   and down. Such data can be actively pushed through subscription
   or passively polled through query. There are many ways to model
   events, including using Finite State Machine (FSM) or Event
   Condition Action (ECA) [NETMOD-ECA-POLICY].

* Streaming Data: Data that are continuously generated. It can be a
   time series or the dump of databases. For example, an interface
   packet counter is exported every second. The streaming data
   reflect real-time network states and metrics and require large
   bandwidth and processing power. The streaming data are always
   actively pushed to the subscribers.

The above telemetry data types are not mutually exclusive. Rather,
they are often composite. Derived data is composed of simple data;
event-triggered data can be simple or derived; and streaming data can
be based on some recurring event. The relationships of these data
types are illustrated in Figure 3.

```
+----------------------+      +-----------------+
| Event-Triggered Data |<----+ Streaming Data  |
+-------+---+----------+      +-----+---+-------+
        |   |                       |   |
        |   |                       |   |
        |   |   +--------------+    |   |
        |   +-->| Derived Data |<--+    |
        |       +------+------ +        |
        |              |                |
        |              V                |
        |       +--------------+        |
        +------>| Simple Data  |<------+
                +--------------+
```
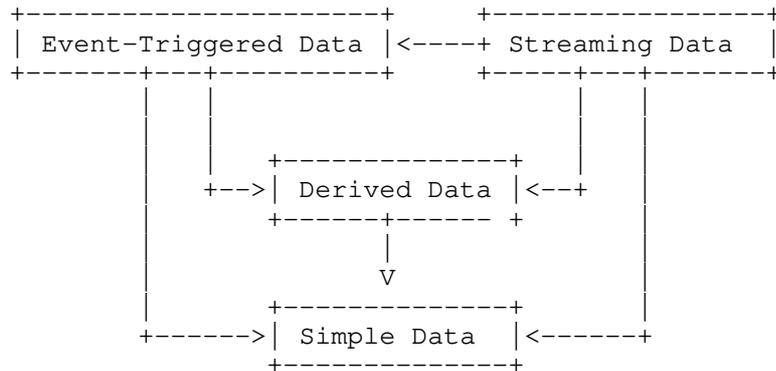
                 Figure 3: Data Type Relationship

Subscription usually deals with event-triggered data and streaming
data, and query usually deals with simple data and derived data. But
the other ways are also possible. Advanced network telemetry
techniques are designed mainly for event-triggered or streaming data
subscription and derived data query.

3.4. Mapping Existing Mechanisms into the Framework

The following table shows how the existing mechanisms (mainly
published in IETF and with the emphasis on the latest new
technologies) are positioned in the framework. Given the vast body
of existing work, we cannot provide an exhaustive list, so the
mechanisms in the tables should be considered as just examples.
Also, some comprehensive protocols and techniques may cover multiple
aspects or modules of the framework, so a name in a block only
emphasizes one particular characteristic of it. More details about
some listed mechanisms can be found in Appendix A.

	Management Plane	Control Plane	Forwarding Plane
data configuration and subscribe	gNMI, NETCONF, RESTCONF, SNMP, YANG-Push	gNMI, NETCONF, RESTCONF, YANG-Push	NETCONF, RESTCONF, YANG-Push
data	MIB, YANG	YANG	IOAM,

```
|  generation    |                 |                 | PSAMP,      |
|  and process   |                 |                 | PBT, AM     |
+----------------+-----------------+-----------------+-------------+
|  data encoding | gRPC, HTTP, TCP | BMP, TCP        | IPFIX, UDP  |
|  and export    |                 |                 |             |
+----------------+-----------------+-----------------+-------------+
```

                      Table 2: Existing Work Mapping

   Although the framework is generally suitable for any network
   environments, the multi-domain telemetry has some unique challenges
   that deserve further architectural consideration, which is out of the
   scope of this document.

4. Evolution of Network Telemetry Applications

   Network telemetry is an evolving technical area. As the network
   moves towards the automated operation, network telemetry applications
   undergo several stages of evolution, which add a new layer of
   requirements to the underlying network telemetry techniques. Each
   stage is built upon the techniques adopted by the previous stages
   plus some new requirements.

   Stage 0 - Static Telemetry: The telemetry data source and type are
      determined at design time. The network operator can only
      configure how to use it with limited flexibility.

   Stage 1 - Dynamic Telemetry: The custom telemetry data can be
      dynamically programmed or configured at runtime without
      interrupting the network operation, allowing a trade-off among
      resource, performance, flexibility, and coverage.

   Stage 2 - Interactive Telemetry: The network operator can
      continuously customize and fine tune the telemetry data in real
      time to reflect the network operation's visibility requirements.
      Compared with Stage 1, the changes are frequent based on the real-
      time feedback. At this stage, some tasks can be automated, but
      human operators still need to sit in the middle to make decisions.

   Stage 3 - Closed-Loop Telemetry: The telemetry is free from the
      interference of human operators, except for generating the
      reports. The intelligent network operation engine automatically
      issues the telemetry data requests, analyzes the data, and updates
      the network operations in closed control loops.

   Existing technologies are ready for Stages 0 and 1. Individual
   applications for Stages 2 and 3 are also possible now. However, the
   future autonomic networks may need a comprehensive operation
   management system that works at Stages 2 and 3 to cover all the
   network operation tasks. A well-defined network telemetry framework
   is the first step towards this direction.

5. Security Considerations

   The complexity of network telemetry raises significant security
   implications. For example, telemetry data can be manipulated to
   exhaust various network resources at each plane as well as the data
   consumer; falsified or tampered data can mislead the decision-making
   process and paralyze networks; and wrong configuration and
   programming for telemetry is equally harmful. The telemetry data is
   highly sensitive, which exposes a lot of information about the
   network and its configuration. Some of that information can make
   designing attacks against the network much easier (e.g., exact
   details of what software and patches have been installed) and allows
   an attacker to determine whether a device may be subject to
   unprotected security vulnerabilities.

   Given that this document has proposed a framework for network
   telemetry and the telemetry mechanisms discussed are more extensive
   (in both message frequency and traffic amount) than the conventional
   network OAM concepts, we must also anticipate that new security

considerations that may also arise. A number of techniques already
exist for securing the forwarding plane, control plane, and
management plane in a network, but it is important to consider if any
new threat vectors are now being enabled via the use of network
telemetry procedures and mechanisms.

This document proposes a conceptual architectural for collecting,
transporting, and analyzing a wide variety of data sources in support
of network applications. The protocols, data formats, and
configurations chosen to implement this framework will dictate the
specific security considerations. These considerations may include:

* Telemetry framework trust and policy models;

* Role management and access control for enabling and disabling
   telemetry capabilities;

* Protocol transport used for telemetry data and its inherent
   security capabilities;

* Telemetry data stores, storage encryption, methods of access, and
   retention practices;

* Tracking telemetry events and any abnormalities that might
   identify malicious attacks using telemetry interfaces.

* Authentication and integrity protection of telemetry data to make
   data more trustworthy; and

* Segregating the telemetry data traffic from the data traffic
   carried over the network (e.g., historically management access and
   management data may be carried via an independent management
   network).

Some security considerations highlighted above may be minimized or
negated with policy management of network telemetry. In a network
telemetry deployment, it would be advantageous to separate telemetry
capabilities into different classes of policies, i.e., Role-Based
Access Control and Event-Condition-Action policies. Also, potential
conflicts between network telemetry mechanisms must be detected
accurately and resolved quickly to avoid unnecessary network
telemetry traffic propagation escalating into an unintended or
intended denial-of-service attack.

Further study of the security issues will be required, and it is
expected that the security mechanisms and protocols are developed and
deployed along with a network telemetry system.

6. IANA Considerations

   This document has no IANA actions.

7. Informative References

   [gnmi]   Shakir, R., Shaikh, A., Borman, P., Hines, M., Lebsack,
               C., and C. Marrow, "gRPC Network Management Interface",
               IETF 98, March 2017,
               <https://datatracker.ietf.org/meeting/98/materials/slides-
               98-rtgwg-gnmi-intro-draft-openconfig-rtgwg-gnmi-spec-00>.

   [gpb]   Google Developers, "Protocol Buffers",
               <https://developers.google.com/protocol-buffers>.

   [grpc]   gRPC, "gPPC: A high performance, open source universal RPC
               framework", <https://grpc.io>.

   [IPPM-IOAM-DIRECT-EXPORT]
               Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F.,
               Bhandari, S., Ed., Sivakolundu, R., and T. Mizrahi, Ed.,
               "In-situ OAM Direct Exporting", Work in Progress,
               Internet-Draft, draft-ietf-ippm-ioam-direct-export-07, 13

                    October 2021, <https://datatracker.ietf.org/doc/html/
                    draft-ietf-ippm-ioam-direct-export-07>.

[IPPM-POSTCARD-BASED-TELEMETRY]
                    Song, H., Mirsky, G., Filsfils, C., Abdelsalam, A., Zhou,
                    T., Li, Z., Mishra, G., Shin, J., and K. Lee, "In-Situ OAM
                    Marking-based Direct Export", Work in Progress, Internet-
                    Draft, draft-song-ippm-postcard-based-telemetry-12, 12 May
                    2022, <https://datatracker.ietf.org/doc/html/draft-song-
                    ippm-postcard-based-telemetry-12>.

[NETCONF-DISTRIB-NOTIF]
                    Zhou, T., Zheng, G., Voit, E., Graf, T., and P. Francois,
                    "Subscription to Distributed Notifications", Work in
                    Progress, Internet-Draft, draft-ietf-netconf-distributed-
                    notif-03, 10 January 2022,
                    <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-
                    distributed-notif-03>.

[NETCONF-UDP-NOTIF]
                    Zheng, G., Zhou, T., Graf, T., Francois, P., Feng, A. H.,
                    and P. Lucente, "UDP-based Transport for Configured
                    Subscriptions", Work in Progress, Internet-Draft, draft-
                    ietf-netconf-udp-notif-05, 4 March 2022,
                    <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-
                    udp-notif-05>.

[NETMOD-ECA-POLICY]
                    Wu, Q., Bryskin, I., Birkholz, H., Liu, X., and B. Claise,
                    "A YANG Data model for ECA Policy Management", Work in
                    Progress, Internet-Draft, draft-ietf-netmod-eca-policy-01,
                    19 February 2021, <https://datatracker.ietf.org/doc/html/
                    draft-ietf-netmod-eca-policy-01>.

[NMRG-ANTICIPATED-ADAPTATION]
                    Martinez-Julia, P., Ed., "Exploiting External Event
                    Detectors to Anticipate Resource Requirements for the
                    Elastic Adaptation of SDN/NFV Systems", Work in Progress,
                    Internet-Draft, draft-pedro-nmrg-anticipated-adaptation-
                    02, 29 June 2018, <https://datatracker.ietf.org/doc/html/
                    draft-pedro-nmrg-anticipated-adaptation-02>.

[NMRG-IBN-CONCEPTS-DEFINITIONS]
                    Clemm, A., Ciavaglia, L., Granville, L. Z., and J.
                    Tantsura, "Intent-Based Networking - Concepts and
                    Definitions", Work in Progress, Internet-Draft, draft-
                    irtf-nmrg-ibn-concepts-definitions-09, 24 March 2022,
                    <https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-
                    ibn-concepts-definitions-09>.

[OPSAWG-DNP4IQ]
                    Song, H., Ed. and J. Gong, "Requirements for Interactive
                    Query with Dynamic Network Probes", Work in Progress,
                    Internet-Draft, draft-song-opsawg-dnp4iq-01, 19 June 2017,
                    <https://datatracker.ietf.org/doc/html/draft-song-opsawg-
                    dnp4iq-01>.

[OPSAWG-IFIT-FRAMEWORK]
                    Song, H., Qin, F., Chen, H., Jin, J., and J. Shin, "A
                    Framework for In-situ Flow Information Telemetry", Work in
                    Progress, Internet-Draft, draft-song-opsawg-ifit-
                    framework-17, 22 February 2022,
                    <https://datatracker.ietf.org/doc/html/draft-song-opsawg-
                    ifit-framework-17>.

[RFC1157]   Case, J., Fedor, M., Schoffstall, M., and J. Davin,
                    "Simple Network Management Protocol (SNMP)", RFC 1157,
                    DOI 10.17487/RFC1157, May 1990,
                    <https://www.rfc-editor.org/info/rfc1157>.

[RFC2578]   McCloghrie, K., Ed., Perkins, D., Ed., and J.

               Schoenwaelder, Ed., "Structure of Management Information
               Version 2 (SMIv2)", STD 58, RFC 2578,
               DOI 10.17487/RFC2578, April 1999,
               <https://www.rfc-editor.org/info/rfc2578>.

   [RFC2981]   Kavasseri, R., Ed., "Event MIB", RFC 2981,
               DOI 10.17487/RFC2981, October 2000,
               <https://www.rfc-editor.org/info/rfc2981>.

   [RFC3176]   Phaal, P., Panchen, S., and N. McKee, "InMon Corporation's
               sFlow: A Method for Monitoring Traffic in Switched and
               Routed Networks", RFC 3176, DOI 10.17487/RFC3176,
               September 2001, <https://www.rfc-editor.org/info/rfc3176>.

   [RFC3411]   Harrington, D., Presuhn, R., and B. Wijnen, "An
               Architecture for Describing Simple Network Management
               Protocol (SNMP) Management Frameworks", STD 62, RFC 3411,
               DOI 10.17487/RFC3411, December 2002,
               <https://www.rfc-editor.org/info/rfc3411>.

   [RFC3416]   Presuhn, R., Ed., "Version 2 of the Protocol Operations
               for the Simple Network Management Protocol (SNMP)",
               STD 62, RFC 3416, DOI 10.17487/RFC3416, December 2002,
               <https://www.rfc-editor.org/info/rfc3416>.

   [RFC3877]   Chisholm, S. and D. Romascanu, "Alarm Management
               Information Base (MIB)", RFC 3877, DOI 10.17487/RFC3877,
               September 2004, <https://www.rfc-editor.org/info/rfc3877>.

   [RFC3954]   Claise, B., Ed., "Cisco Systems NetFlow Services Export
               Version 9", RFC 3954, DOI 10.17487/RFC3954, October 2004,
               <https://www.rfc-editor.org/info/rfc3954>.

   [RFC4656]   Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M.
               Zekauskas, "A One-way Active Measurement Protocol
               (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006,
               <https://www.rfc-editor.org/info/rfc4656>.

   [RFC5085]   Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual
               Circuit Connectivity Verification (VCCV): A Control
               Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085,
               December 2007, <https://www.rfc-editor.org/info/rfc5085>.

   [RFC5357]   Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J.
               Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)",
               RFC 5357, DOI 10.17487/RFC5357, October 2008,
               <https://www.rfc-editor.org/info/rfc5357>.

   [RFC5424]   Gerhards, R., "The Syslog Protocol", RFC 5424,
               DOI 10.17487/RFC5424, March 2009,
               <https://www.rfc-editor.org/info/rfc5424>.

   [RFC6020]   Bjorklund, M., Ed., "YANG - A Data Modeling Language for
               the Network Configuration Protocol (NETCONF)", RFC 6020,
               DOI 10.17487/RFC6020, October 2010,
               <https://www.rfc-editor.org/info/rfc6020>.

   [RFC6241]   Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
               and A. Bierman, Ed., "Network Configuration Protocol
               (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
               <https://www.rfc-editor.org/info/rfc6241>.

   [RFC6812]   Chiba, M., Clemm, A., Medley, S., Salowey, J., Thombare,
               S., and E. Yedavalli, "Cisco Service-Level Assurance
               Protocol", RFC 6812, DOI 10.17487/RFC6812, January 2013,
               <https://www.rfc-editor.org/info/rfc6812>.

   [RFC7011]   Claise, B., Ed., Trammell, B., Ed., and P. Aitken,
               "Specification of the IP Flow Information Export (IPFIX)
               Protocol for the Exchange of Flow Information", STD 77,
               RFC 7011, DOI 10.17487/RFC7011, September 2013,

                    <https://www.rfc-editor.org/info/rfc7011>.

   [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an
              Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May
              2014, <https://www.rfc-editor.org/info/rfc7258>.

   [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y.
              Weingarten, "An Overview of Operations, Administration,
              and Maintenance (OAM) Tools", RFC 7276,
              DOI 10.17487/RFC7276, June 2014,
              <https://www.rfc-editor.org/info/rfc7276>.

   [RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext
              Transfer Protocol Version 2 (HTTP/2)", RFC 7540,
              DOI 10.17487/RFC7540, May 2015,
              <https://www.rfc-editor.org/info/rfc7540>.

   [RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A.,
              Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic
              Networking: Definitions and Design Goals", RFC 7575,
              DOI 10.17487/RFC7575, June 2015,
              <https://www.rfc-editor.org/info/rfc7575>.

   [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with
              Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799,
              May 2016, <https://www.rfc-editor.org/info/rfc7799>.

   [RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP
              Monitoring Protocol (BMP)", RFC 7854,
              DOI 10.17487/RFC7854, June 2016,
              <https://www.rfc-editor.org/info/rfc7854>.

   [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",
              RFC 7950, DOI 10.17487/RFC7950, August 2016,
              <https://www.rfc-editor.org/info/rfc7950>.

   [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF
              Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,
              <https://www.rfc-editor.org/info/rfc8040>.

   [RFC8084] Fairhurst, G., "Network Transport Circuit Breakers",
              BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017,
              <https://www.rfc-editor.org/info/rfc8084>.

   [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage
              Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085,
              March 2017, <https://www.rfc-editor.org/info/rfc8085>.

   [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data
              Interchange Format", STD 90, RFC 8259,
              DOI 10.17487/RFC8259, December 2017,
              <https://www.rfc-editor.org/info/rfc8259>.

   [RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli,
              L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi,
              "Alternate-Marking Method for Passive and Hybrid
              Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321,
              January 2018, <https://www.rfc-editor.org/info/rfc8321>.

   [RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard,
              E., and A. Tripathy, "Subscription to YANG Notifications",
              RFC 8639, DOI 10.17487/RFC8639, September 2019,
              <https://www.rfc-editor.org/info/rfc8639>.

   [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications
              for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641,
              September 2019, <https://www.rfc-editor.org/info/rfc8641>.

   [RFC8671] Evens, T., Bayraktar, S., Lucente, P., Mi, P., and S.
              Zhuang, "Support for Adj-RIB-Out in the BGP Monitoring
              Protocol (BMP)", RFC 8671, DOI 10.17487/RFC8671, November

                    2019, <https://www.rfc-editor.org/info/rfc8671>.

   [RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple
              Two-Way Active Measurement Protocol", RFC 8762,
              DOI 10.17487/RFC8762, March 2020,
              <https://www.rfc-editor.org/info/rfc8762>.

   [RFC8889] Fioccola, G., Ed., Cociglio, M., Sapio, A., and R. Sisto,
              "Multipoint Alternate-Marking Method for Passive and
              Hybrid Performance Monitoring", RFC 8889,
              DOI 10.17487/RFC8889, August 2020,
              <https://www.rfc-editor.org/info/rfc8889>.

   [RFC8924] Aldrin, S., Pignataro, C., Ed., Kumar, N., Ed., Krishnan,
              R., and A. Ghanwani, "Service Function Chaining (SFC)
              Operations, Administration, and Maintenance (OAM)
              Framework", RFC 8924, DOI 10.17487/RFC8924, October 2020,
              <https://www.rfc-editor.org/info/rfc8924>.

   [RFC9069] Evens, T., Bayraktar, S., Bhardwaj, M., and P. Lucente,
              "Support for Local RIB in the BGP Monitoring Protocol
              (BMP)", RFC 9069, DOI 10.17487/RFC9069, February 2022,
              <https://www.rfc-editor.org/info/rfc9069>.

   [RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi,
              Ed., "Data Fields for In Situ Operations, Administration,
              and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197,
              May 2022, <https://www.rfc-editor.org/info/rfc9197>.

   [W3C.REC-xml-20081126]
              Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E., and
              F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth
              Edition)", World Wide Web Consortium Recommendation REC-
              xml-20081126, November 2008,
              <https://www.w3.org/TR/2008/REC-xml-20081126>.

   [y1731]   ITU-T, "Operations, administration and maintenance (OAM)
              functions and mechanisms for Ethernet-based networks",
              ITU-T Recommendation G.8013/Y.1731, August 2015,
              <https://www.itu.int/rec/T-REC-Y.1731/en>.

Appendix A. A Survey on Existing Network Telemetry Techniques

   In this non-normative appendix, we provide an overview of some
   existing techniques and standard proposals for each network telemetry
   module.

A.1. Management Plane Telemetry

A.1.1. Push Extensions for NETCONF

   NETCONF [RFC6241] is a popular network management protocol
   recommended by IETF. Its core strength is for managing
   configuration, but it can also be used for data collection.
   YANG-Push [RFC8639] [RFC8641] extends NETCONF and enables subscriber
   applications to request a continuous, customized stream of updates
   from a YANG datastore. Providing such visibility into changes made
   upon YANG configuration and operational objects enables new
   capabilities based on the remote mirroring of configuration and
   operational state. Moreover, a distributed data collection mechanism
   [NETCONF-DISTRIB-NOTIF] via a UDP-based publication channel
   [NETCONF-UDP-NOTIF] provides enhanced efficiency for the NETCONF-
   based telemetry.

A.1.2. gRPC Network Management Interface

   gRPC Network Management Interface (gNMI) [gnmi] is a network
   management protocol based on the gRPC [grpc] Remote Procedure Call
   (RPC) framework. With a single gRPC service definition, both
   configuration and telemetry can be covered. gRPC is an open-source
   micro-service communication framework based on HTTP/2 [RFC7540]. It

provides a number of capabilities that are well-suited for network
telemetry, including:

*   A full-duplex streaming transport model; when combined with a
    binary encoding mechanism, it provides good telemetry efficiency.

*   A higher-level feature consistency across platforms that common
    HTTP/2 libraries typically do not provide. This characteristic is
    especially valuable for the fact that telemetry data collectors
    normally reside on a large variety of platforms.

*   A built-in load-balancing and failover mechanism.

A.2. Control Plane Telemetry

A.2.1. BGP Monitoring Protocol

   BMP [RFC7854] is used to monitor BGP sessions and is intended to
   provide a convenient interface for obtaining route views.

   BGP routing information is collected from the monitored device(s) to
   the BMP monitoring station by setting up the BMP TCP session. The
   BGP peers are monitored by the BMP Peer Up and Peer Down
   notifications. The BGP routes (including Adj_RIB_In [RFC7854],
   Adj_RIB_out [RFC8671], and local RIB [RFC9069]) are encapsulated in
   the BMP Route Monitoring Message and the BMP Route Mirroring Message,
   providing both an initial table dump and real-time route updates. In
   addition, BGP statistics are reported through the BMP Stats Report
   Message, which could be either timer triggered or event-driven.
   Future BMP extensions could further enrich BGP monitoring
   applications.

A.3. Data Plane Telemetry

A.3.1. Alternate-Marking (AM) Technology

   The Alternate-Marking method enables efficient measurements of packet
   loss, delay, and jitter both in IP and Overlay Networks, as presented
   in [RFC8321] and [RFC8889].

   This technique can be applied to point-to-point and multipoint-to-
   multipoint flows. Alternate Marking creates batches of packets by
   alternating the value of 1 bit (or a label) of the packet header.
   These batches of packets are unambiguously recognized over the
   network, and the comparison of packet counters for each batch allows
   the packet loss calculation. The same idea can be applied to delay
   measurement by selecting ad hoc packets with a marking bit dedicated
   for delay measurements.

   The Alternate-Marking method needs two counters each marking period
   for each flow under monitor. For instance, by considering n
   measurement points and m monitored flows, the order of magnitude of
   the packet counters for each time interval is n*m*2 (1 per color).

   Since networks offer rich sets of network performance measurement
   data (e.g., packet counters), conventional approaches run into
   limitations. The bottleneck is the generation and export of the data
   and the amount of data that can be reasonably collected from the
   network. In addition, management tasks related to determining and
   configuring which data to generate lead to significant deployment
   challenges.

   The Multipoint Alternate-Marking approach, described in [RFC8889],
   aims to resolve this issue and make the performance monitoring more
   flexible in case a detailed analysis is not needed.

   An application orchestrates network performance measurement tasks
   across the network to allow for optimized monitoring. The
   application can choose how roughly or precisely to configure
   measurement points depending on the application's requirements.

Using Alternate Marking, it is possible to monitor a Multipoint
Network without in-depth examination by using Network Clustering
(subnetworks that are portions of the entire network that preserve
the same property of the entire network, called clusters). So in the
case where there is packet loss or the delay is too high, the
specific filtering criteria could be applied to gather a more
detailed analysis by using a different combination of clusters up to
a per-flow measurement as described in the Alternate-Marking document
[RFC8321].

In summary, an application can configure end-to-end network
monitoring. If the network does not experience issues, this
approximate monitoring is good enough and is very cheap in terms of
network resources. However, in case of problems, the application
becomes aware of the issues from this approximate monitoring and, in
order to localize the portion of the network that has issues,
configures the measurement points more extensively, allowing more
detailed monitoring to be performed. After the detection and
resolution of the problem, the initial approximate monitoring can be
used again.

A.3.2. Dynamic Network Probe

A hardware-based Dynamic Network Probe (DNP) [OPSAWG-DNP4IQ] provides
a programmable means to customize the data that an application
collects from the data plane. A direct benefit of DNP is the
reduction of the exported data. A full DNP solution covers several
components including data source, data subscription, and data
generation. The data subscription needs to define the derived data
that can be composed and derived from raw data sources. The data
generation takes advantage of the moderate in-network computing to
produce the desired data.

While DNP can introduce unforeseeable flexibility to the data plane
telemetry, it also faces some challenges. It requires a flexible
data plane that can be dynamically reprogrammed at runtime. The
programming Application Programming Interface (API) is yet to be
defined.

A.3.3. IP Flow Information Export (IPFIX) Protocol

Traffic on a network can be seen as a set of flows passing through
network elements. IPFIX [RFC7011] provides a means of transmitting
traffic flow information for administrative or other purposes. A
typical IPFIX-enabled system includes a pool of Metering Processes
that collects data packets at one or more Observation Points,
optionally filters them, and aggregates information about these
packets. An Exporter then gathers each of the Observation Points
together into an Observation Domain and sends this information via
the IPFIX protocol to a Collector.

A.3.4. In Situ OAM

Classical passive and active monitoring and measurement techniques
are either inaccurate or resource consuming. It is preferable to
directly acquire data associated with a flow's packets when the
packets pass through a network. IOAM [RFC9197], a data generation
technique, embeds a new instruction header to user packets, and the
instruction directs the network nodes to add the requested data to
the packets. Thus, at the path's end, the packet's experience gained
on the entire forwarding path can be collected. Such firsthand data
is invaluable to many network OAM applications.

However, IOAM also faces some challenges. The issues on performance
impact, security, scalability and overhead limits, encapsulation
difficulties in some protocols, and cross-domain deployment need to
be addressed.

A.3.5. Postcard-Based Telemetry

The postcard-based telemetry, as embodied in IOAM Direct Export (DEX)

[IPPM-IOAM-DIRECT-EXPORT] and IOAM Marking
[IPPM-POSTCARD-BASED-TELEMETRY], is a complementary technique to the
passport-based IOAM [RFC9197]. PBT directly exports data at each
node through an independent packet. At the cost of higher bandwidth
overhead and the need for data correlation, PBT shows several unique
advantages. It can also help to identify packet drop location in
case a packet is dropped on its forwarding path.

A.3.6. Existing OAM for Specific Data Planes

   Various data planes raise unique OAM requirements. IETF has
   published OAM technique and framework documents (e.g., [RFC8924] and
   [RFC5085]) targeting different data planes such as Multiprotocol
   Label Switching (MPLS), L2 Virtual Private Network (VPN), Network
   Virtualization over Layer 3 (NVO3), Virtual Extensible LAN (VXLAN),
   Bit Index Explicit Replication (BIER), Service Function Chaining
   (SFC), Segment Routing (SR), and Deterministic Networking (DETNET).
   The aforementioned data plane telemetry techniques can be used to
   enhance the OAM capability on such data planes.

A.4. External Data and Event Telemetry

A.4.1. Sources of External Events

   To ensure that the information provided by external event detectors
   and used by the network management solutions is meaningful for
   management purposes, the network telemetry framework must ensure that
   such detectors (sources) are easily connected to the management
   solutions (sinks). This requires the specification of a list of
   potential external data sources that could be of interest in network
   management and matching it to the connectors and/or interfaces
   required to connect them.

   Categories of external event sources that may be of interest to
   network management include:

   * Smart objects and sensors. With the consolidation of the Internet
      of Things (IoT), any network system will have many smart objects
      attached to its physical surroundings and logical operation
      environments. Most of these objects will be essentially based on
      sensors of many kinds (e.g., temperature, humidity, and presence),
      and the information they provide can be very useful for the
      management of the network, even when they are not specifically
      deployed for such purpose. Elements of this source type will
      usually provide a specific protocol for interaction, especially
      one of the protocols related to IoT, such as the Constrained
      Application Protocol (CoAP).

   * Online news reporters. Several online news services have the
      ability to provide an enormous quantity of information about
      different events occurring in the world. Some of those events can
      have an impact on the network system managed by a specific
      framework; therefore, such information may be of interest to the
      management solution. For instance, diverse security reports, such
      as Common Vulnerabilities and Exposures (CVEs), can be issued by
      the corresponding authority and used by the management solution to
      update the managed system, if needed. Instead of a specific
      protocol and data format, the sources of this kind of information
      usually follow a relaxed but structured format. This format will
      be part of both the ontology and information model of the
      telemetry framework.

   * Global event analyzers. The advance of big data analyzers
      provides a huge amount of information and, more interestingly, the
      identification of events detected by analyzing many data streams
      from different origins. In contrast with the other types of
      sources, which are focused on specific events, the detectors of
      this source type will detect generic events. For example, during
      a sports event, some unexpected movement makes it fascinating, and
      many people connect to sites that are reporting on the event. The
      underlying networks supporting the services that cover the event

can be affected by such situation, so their management solutions should be aware of it. In contrast with the other source types, a new information model, format, and reporting protocol is required to integrate the detectors of this type with the management solution.

Additional detector types can be added to the system, but generally they will be the result of composing the properties offered by these main classes.

A.4.2. Connectors and Interfaces

For allowing external event detectors to be properly integrated with other management solutions, both elements must expose interfaces and protocols that are subject to their particular objective. Since external event detectors will be focused on providing their information to their main consumers, which generally will not be limited to the network management solutions, the framework must include the definition of the required connectors for ensuring the interconnection between detectors (sources) and their consumers within the management systems (sinks) are effective.

In some situations, the interconnection between external event detectors and the management system is via the management plane. For those situations, there will be a special connector that provides the typical interfaces found in most other elements connected to the management plane. For instance, the interfaces could accomplish this with a specific data model (YANG) and specific telemetry protocol, such as NETCONF, YANG-Push, or gRPC.

Acknowledgments

Contributors

The other contributors of this document are Tianran Zhou, Zhenbin Li, Zhenqiang Li, Daniel King, Adrian Farrel, and Alexander Clemm.

Authors' Addresses

Haoyu Song
Futurewei
United States of America
Email: haoyu.song@futurewei.com

Fengwei Qin
China Mobile
China
Email: qinfengwei@chinamobile.com

Pedro Martinez-Julia
NICT
Japan
Email: pedro@nict.go.jp

Laurent Ciavaglia
Rakuten Mobile
France
Email: laurent.ciavaglia@rakuten.com

Aijun Wang
China Telecom
China
Email: wangaj3@chinatelecom.cn