MARVELL™

White Paper

Network Telemetry Solutions for Data Center and Enterprise Networks

Tal Mizrahi, Vitaly Vovnoboy, Moti Nisim, Gidi Navon, Amos Soffer
Switching Group Marvell

December 2021

ABSTRACT

One of the most prominent trends in the networking community in the last few years is network telemetry - which enables accurate measurement of the network′s performance in real-time. In this white paper we will discuss how network telemetry is evolving in modern data center networks. Details will then be given of how the generic approach to network telemetry that has been taken by Marvell is providing greater visibility into network performance, plus flexible support of existing telemetry protocols, as well as the future ones emerging.

1. Introduction

Operations, Administration and Maintenance (OAM) is a well-known term that refers to a toolset for fault detection and isolation, and for performance measurement [1]. OAM protocols have been widely employed for many years, and are defined at various layers of the protocols stack. They are used for Ethernet, IP, MPLS, and for many other network protocols.

While OAM and network measurement have seen widespread implementation in carrier networks, the common perception until now was that data center networks are relatively simple in nature, and do not therefore require the complexity of OAM protocols. As data center networks have evolved, their increasing speed and complexity have made network monitoring and troubleshooting more difficult. Such challenges have brought forth the need for new network telemetry methods, capable of provide detailed real-time information about the overall performance in high-speed network infrastructure.

In this white paper we briefly touch on some of the most common network telemetry methods - traditional as well as recent. The focus will then be placed upon Marvell's approach to network telemetry, in particular the company's Prestera family of devices.

2. Network Telemetry Approaches

Before going further, we should define exactly what is meant by network telemetry. Telemetry is a process in which the performance of a network is measured at various points, and data related to this measurement is acquired by a central collector, such as an analytics server. There are various approaches to measuring the performance of a network, as will be described in this upcoming section.

Active vs. Passive Measurement

Performance measurement approaches can basically be classified as being either active or passive in form [2].

a. **Active** measurement uses dedicated control plane (OAM) messages. The performance of these messages is monitored, thereby giving an indicator of the performance of the user traffic.

b. **Passive** measurement, in contrast, does not use control plane messages, and instead monitors the performance of live user traffic.

Active and passive measurements approaches that lie at the two extremes. In practice, some of the most common measurement protocols actually rely on a hybrid approach. **Hybrid** approaches measure the user traffic via control plane messages, or through control information that is piggybacked onto data plane packets.

Passive Measurement

Passive measurement is typically applied using passive probes that monitor performance metrics and track them continuously. Monitored attributes often include packet and byte counters, queue status and latency statistics. It should be noted that passive measurement provides information that is strictly local, and does not give network-wide information about network paths or dropped packets. Nevertheless, passive measurement is a common practice - proving to be both straightforward and effective.

Active Measurement

Several measurement protocols use control plane messages to determine performance metric levels; such as packet loss, delay, delay variation and bandwidth. Ping is probably the most common and well-known application that performs active measurement. Other common measurement protocols, like the ones defined in ITU-T Y.1731 [3] and RFC 6374 [4], use OAM messages to measure packet loss and delay in a network. Figures 1 and 2 illustrate an example of active measurement. Here timestamped packets are used to compute either the one-way delay, or the two-way delay of between two switches in a network.

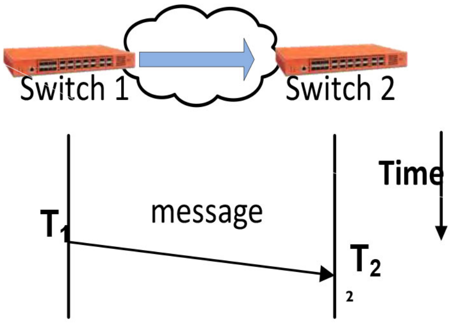


Figure 1: One-way delay measurement. A timestamped message is sent from Switch 1 to Switch 2, allowing Switch 2 to compute the one-way delay (T - T_1). Requires Switch 1 and Switch 2 to be synchronized.

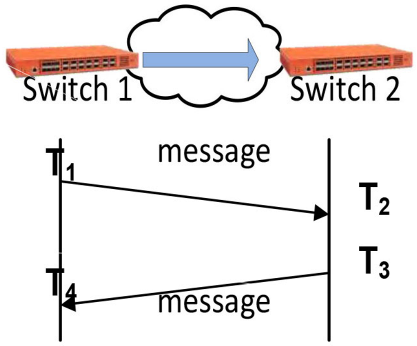


Figure 2: Two-way delay measurement. Switch 1 sends a timestamped message (with T_1) to Switch 2. Switch 2 replies with a message that includes T_1, T_2, and T_3. Switch 1 can compute the two-way delay, ($[T_4-T_1] - [T_3-T_2]$).

In-band Measurement

In-band measurement is an example of a hybrid measurement approach that has gained a lot of momentum over the last few years. The idea of in-band telemetry [5], is that each node along the path incorporates timestamps (and potentially other information) in the headers of data plane packets, allowing fine-grained measurement and congestion detection. These approaches are known as In-band Network Telemetry (INT) [6] and In-situ OAM (IOAM) [7], which are under discussion within the P4 consortium and IETF, respectively.

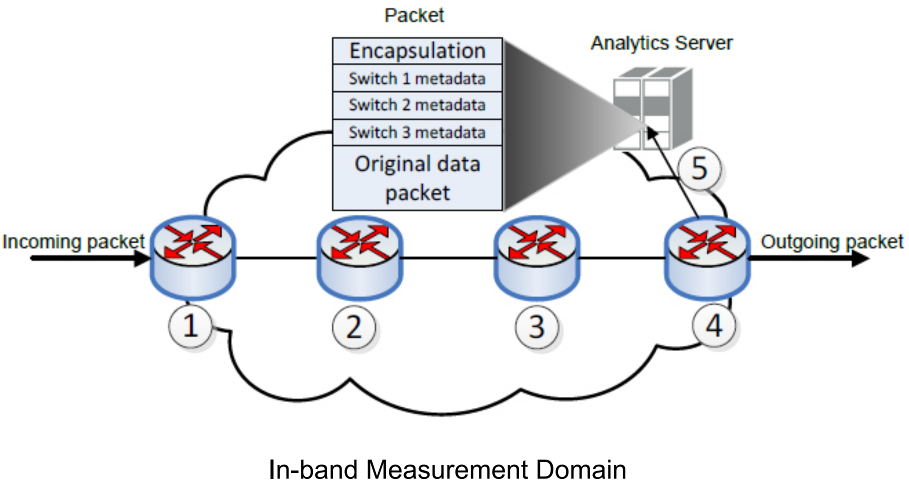


Figure 3: In-band measurement. Each switch along the path incorporates performance-related metadata, including timestamps, in the header of en-route data plane packets. The packets and metadata can be sent to an analyzer for further analysis.

Alternate Marking

Alternate marking [8] is another hybrid measurement method, that is used for measuring loss and delay between two Measurement Points (MPs) using one or two bits in the header of every packet. In a nutshell, the header of each data packet includes a binary color bit, either ‘0’ or ‘1’. The color bit divides the traffic into consecutive blocks of packets, allowing the two MPs to process each block separately. The alternating colors allow very accurate measurement of the loss and delay between the two MPs.

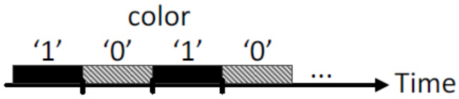


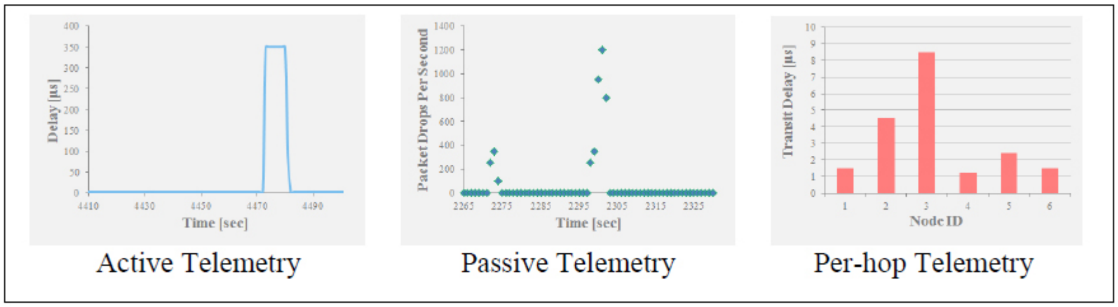
Figure 4: The alternate marking method.

The colors are toggled periodically, so that each color is used for a fixed time interval. Hence, the color bit can be viewed as a one-bit timestamp that wraps around cyclically. Moreover, if the data packets already carry an in-band timestamp, then it is possible to use one of the timestamp bits as the color bit. For example, if the timestamp is measured in seconds, then by choosing the least significant bit of the timestamp, we get a color bit that is toggled with a one-second period.

Alternate marking uses one or two bits per packet, piggybacked onto live data traffic. Since the effect of one or two bits per packet on the network performance is negligible, alternate marking is often viewed as nearly-passive - allowing accurate measurement without using dedicated control plane messages or representing a large per-packet overhead.

3. Marvell's Network Telemetry Toolset

The Prestera family of devices were designed by Marvell with a focus on maximizing performance visibility, while providing the required flexibility and programmability needed to address emerging as well as future network telemetry protocols. The network telemetry toolset covers a wide range of measurement methods and protocols, from traditional OAM protocols to the most recent telemetry techniques.



Active Telemetry

Passive Telemetry

Per-hop Telemetry

Figure 5: The network telemetry toolset.

Use of Passive Measurement

Through Prestera, Marvell provides high visibility into the network performance using a wide set of passive monitoring mechanisms such as:

Counters - The Prestera devices support a large and flexible set of packet-based, byte-based, and drop counters. The counters may be based on various criteria, e.g., per port, per-queue, or per-flow. Furthermore, counters can be probed in one of multiple locations along the packet processing pipeline.

Burst detection and classification - One of the key challenges in high-speed networks is to detect, classify and respond to traffic bursts and network congestion. Network congestion is sometimes caused by high-bandwidth flows, which consume significant network resources for a long period of time, while in other cases the network suffers from short traffic bursts that consume a large amount of resources for a short period of time, also known as µBursts. Marvell's Prestera family of devices continuously track network traffic, thereby allowing detection of bursts, plus measurement of their size and duration over long periods of time. This means they can quickly react to situations as they arise.

Latency monitoring - A key metric of network performance is latency. Therefore, it is important to continuously track latency and maintain statistics about the maximal, minimal, and average latency. These statistics can be maintained on a per port basis, on a per {source, destination} port pair, or on a per-flow basis.

Use of Active Measurement

Marvell's generic approach to OAM [9] enables implementation of an array of different active and hybrid measurement protocols. Instead of supporting a set of protocols, Marvell's devices provide a set of generic building blocks:

- Flexible timestamping
- Flexible counting
- Keep-alive monitoring (including automatic detection of loss of connectivity)
- Automatic protection switching
- Various mirroring and sampling mechanisms

These generic building blocks provide the necessary hooks for supporting the various OAM protocols that are used for failure detection, protection switching, loss measurement and delay measurement.

Use of In-band Telemetry

One of the keys to supporting high-resolution network telemetry is flexibility and programmability.

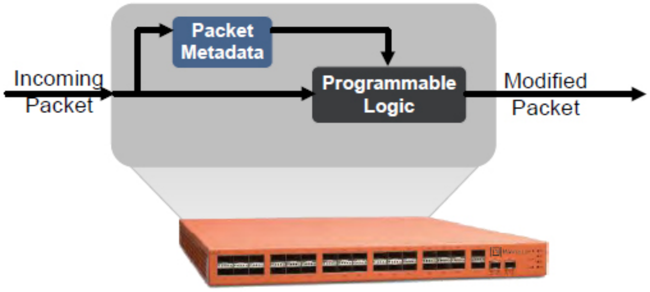


Figure 6: Programmable metadata-based packet processing.

Marvell's programmable metadata processing is illustrated in Figure 6; with every incoming packet being assigned a set of internal metadata fields. Each packet is then processed by programmable logic that uses both the packet header and the internal metadata. This flexible header editing logic enables in-flight insertion of metadata into data packets, including:

- Device ID
- Ingress and egress port ID
- Queue ID

- Queue and congestion status
- Quality of Service (QoS) attributes (such as DSCP)
- Port utilization
- Sequence number
- Timestamp
- Transit delay

Other metadata fields are also possible, such as priority-related information, or various counters and statistics.

Programmable header editing enables both INT and IOAM. These protocols are supported over various encapsulation protocols - including VXLAN-GPE, Geneve and NSH. Many of these encapsulation protocols include a UDP header, and thus metadata insertion requires the UDP checksum field to be updated. When inserting telemetry metadata into an en-route packet, the Prestera device can optionally perform an incremental update of the UDP checksum field [10], or update a checksum complement field (as defined in [7]).

Selective Probing

INT and IOAM provide highly granular per-packet information. The main challenge with such detailed information is to be able to analyze it in real-time. Obviously analytics servers cannot process the entire bandwidth of the data plane traffic in the network. Hence, it is important for switches to be able to selectively probe telemetry information to the analytics servers.

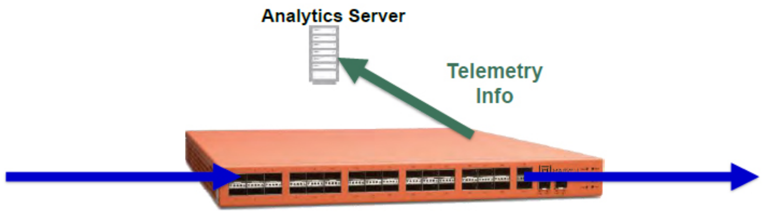


Figure 7: Selective probing of telemetry information.

Marvell's Prestera devices selectively choose a subset of the data plane packets and send their telemetry information to external analytics servers. Selective probing combines statistical sampling with congestion-detection-based sampling. Specifically, selective probing can be based on one or more of the following methods:

a. **1 out of N -**Where out of set number of packets (N) one is probed.
b. **Periodic -** Where a packet is probed every predetermined time period.
c. **Time interval -** Where within every predetermined time period, packets are probed for a short time interval. For example, packets are probed during the first 1 millisecond of every second.
d. **Congestion -** In which packets are probed when a queue is filled up beyond a predetermined threshold.
e. **Drop -** In which telemetry information is probed when a packet is dropped.

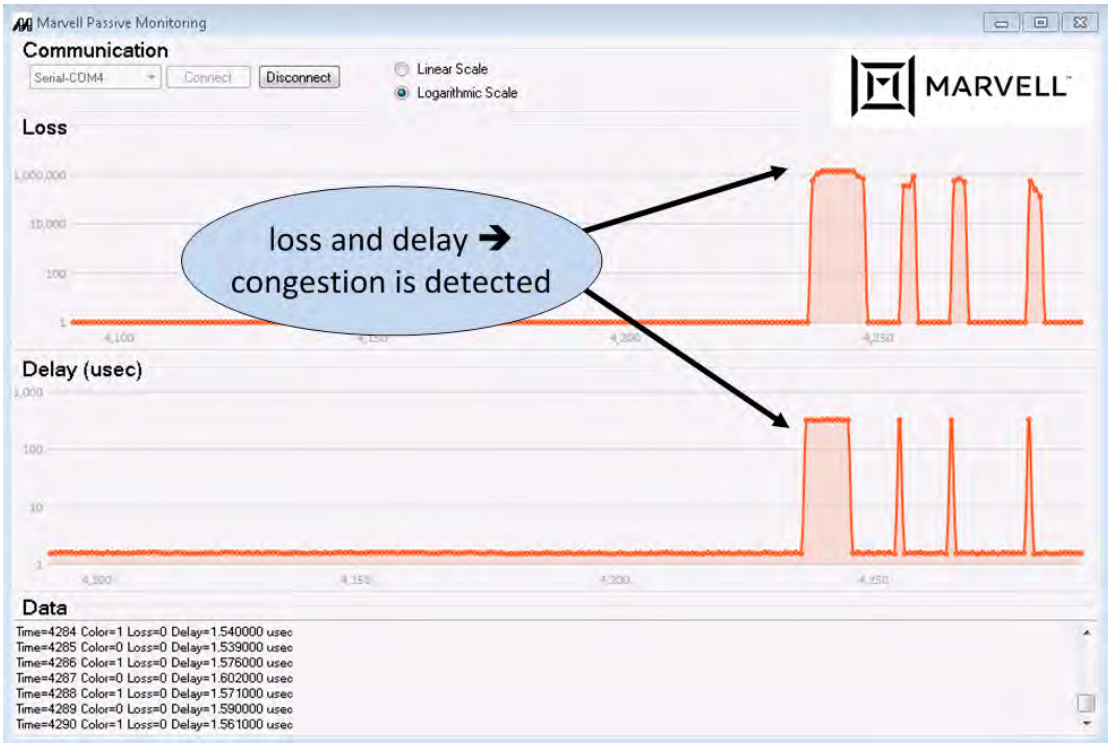
f. **Rate -** Where packets are probed when the rate of a flow exceeds a predetermined threshold.

g. **Alternate marking -** Where packets can be probed based on a marking bit within the header (see further details below).

Alternate Marking

Marvell’s Prestera offering supports full-wire-speed alternate marking for loss and delay measurement. The programmable header editing functionality of these devices allows any header field to be used as the marking field, supporting double marking, single marking and multiplexed marking.

Marvell’s alternate marking implementation uses TimeFlips. A TimeFlip [11] is a ternary content-addressable memory (TCAM) lookup that uses the current time as a match criterion in the TCAM. This approach allows Prestera devices to flexibly support a wide range of possible measurement periods, from a few milliseconds to several minutes.



loss and delay → congestion is detected

```
Time=4284 Color=1 Loss=0 Delay=1.540000 usec
Time=4285 Color=0 Loss=0 Delay=1.539000 usec
Time=4286 Color=1 Loss=0 Delay=1.576000 usec
Time=4287 Color=0 Loss=0 Delay=1.602000 usec
Time=4288 Color=1 Loss=0 Delay=1.571000 usec
Time=4289 Color=0 Loss=0 Delay=1.590000 usec
Time=4290 Color=1 Loss=0 Delay=1.561000 usec
```

Figure 8: Loss and delay measurement using alternate marking. The horizontal axis represents time (seconds), and the vertical axis represents the delay in microseconds (bottom graph), and the number of packets lost per second (top graph).

Selective Probing using Alternate Marking

What happens if detailed per-hop telemetry information needs to be collected, as performed in INT or in IOAM, but without the data plane overhead of piggybacking this information onto data packets? One way to achieve this is to *mark* specific packets or specific flows, thus allowing the switches along the path to detect the marked packets, and export their required telemetry information.

This method requires just a single marking bit in each data packet. For example, if the ingress node sets the marking bit in one packet per second, the rest of the switches along the path detect the marked packet, and export telemetry information about the marked packet. Thus, telemetry information will be exported to the analytics server only for the marked packets, allowing the Server to correlate the information received from the different switches along the path. Alternatively, the marking bit can be used to mark a specific flow that is temporarily experiencing performance issues, indicating that telemetry information should be exported for this flow.

4. Marvell's Telemetry Software Suite

Marvell offers a Telemetry and Monitoring (TAM) software suite that enables customers to monitor their network and determine how traffic is being handled by the device in real-time. This suite provides major benefits to network operators such as:

- Better characterization of congestion events according to the different statistics
- The ability to correlate network congestion events with servers activities
- Monitoring network health and identifying the severity of traffic events

Marvell's suite provides an offloading service to the application CPU or to the network controller, alleviating the need to collect statistics for a large number of events.

The TAM suite provides a high abstraction layer that enables both passive measurement and in-band measurement (e.g. INT). It allows the configuration of what counters to measure, tracking of the device buffer counters, maintaining of snapshots, measurement of µBurst durations, generation of histograms based on the measured statistics, setting of threshold crossing notifications, exporting of telemetry information to an analytics server, and numerous other functions.

At the heart of the suite lies a software Telemetry Agent (as shown in Figure 9) which runs on the switch device and leverages Marvell's embedded smart monitoring engines. The Telemetry Agent talks with the analytics application that typically runs in a stand-alone analytics server or as an add-on in the SDN controller or orchestration software.

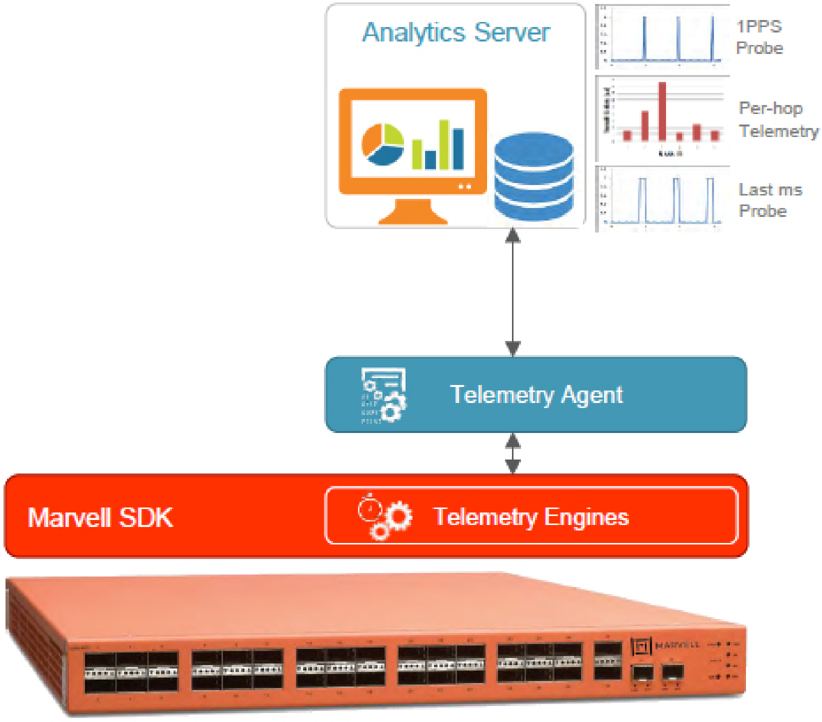


Figure 9: Marvell's Telemetry and Monitoring (TAM) Software Suite.

Marvell provides various ways for software Telemetry Agents to access the silicon telemetry engines (see Figure 10). The most common one is the Marvell Software Development Kit (SDK) for the Prestera family. Another alternative is Marvell's Forwarding Plane Abstraction (FPA), an open software Application Programming Interface (API) based on the work of the Open Networking Foundation (ONF), that is designed as a library on top of Marvell's SDK. The FPA is more commonly used by native SDN or OpenFlow management. Another option is the Switch Abstraction Interface (SAI), a vendor-independent API for controlling forwarding elements, such as a packet processors, in a uniform manner.

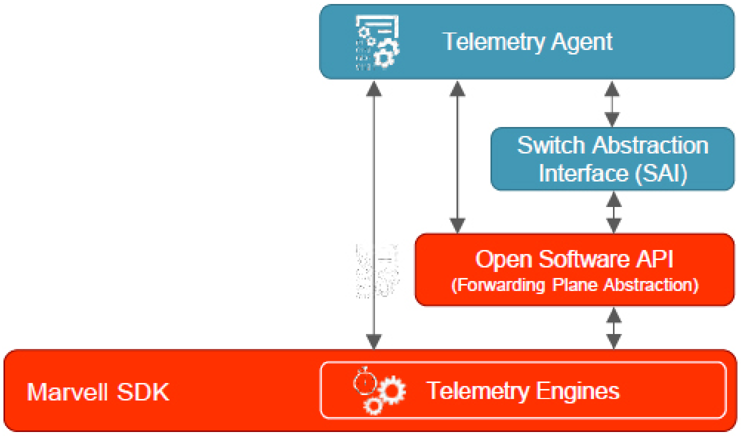

Telemetry Agent
Switch Abstraction Interface (SAI)
Open Software API
(Forwarding Plane Abstraction)
Marvell SDK
Telemetry Engines

Figure 10: Marvell's APIs for the Telemetry Agent.

A typical use case in data center networks would be employing OpenStack to collect telemetry information from Top-of-the-Rack (ToR) switches and other networking devices. OpenStack is an open source software for creating private and public clouds that controls large pools of compute, storage, and networking resources throughout a data center. It includes the Ceilometer data collection service for collecting and storing instrumentation and monitoring-related data in an OpenStack environment, and the popular Oslo messaging library that provides APIs for implementing client-server remote procedure calls and for emitting and handling event notifications.

Use of OpenStack can increase networking visibility in the operation of the underlay network by either pulling instrumentation data from the Telemetry Agent or have the data pushed by the Telemetry Agent in an asynchronous manner. The Telemetry Agent in that case queries telemetry information from the silicon telemetry engines using Marvell's SDK and sends statistics reports to the Ceilometer collector application running on the OpenStack controller.

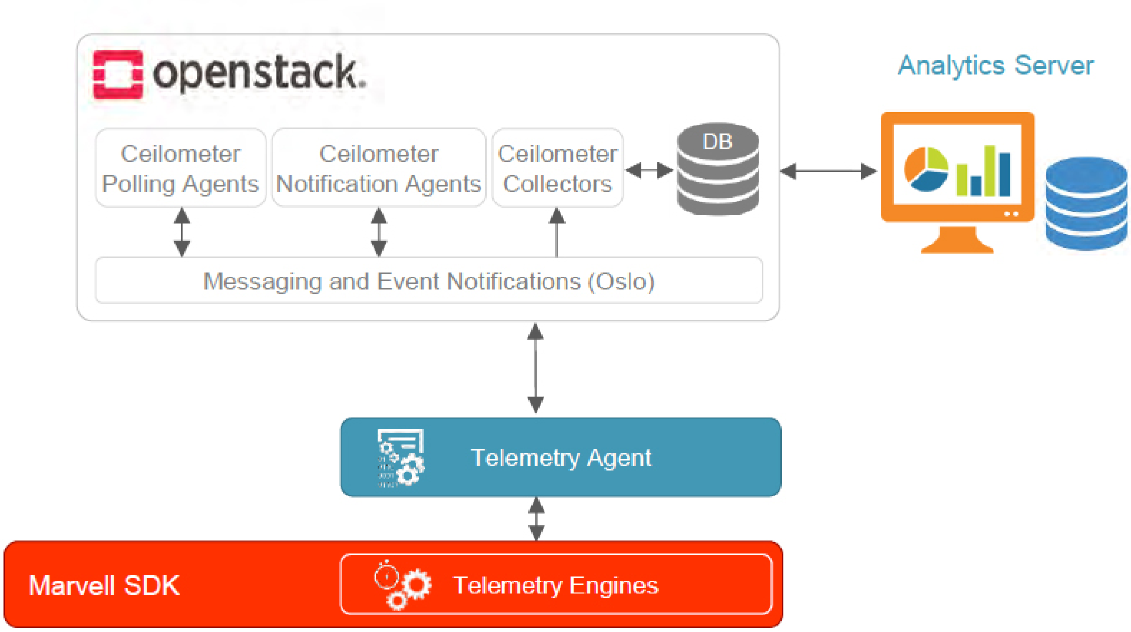

openstack.
Ceilometer Polling Agents
Ceilometer Notification Agents
Ceilometer Collectors
DB
Messaging and Event Notifications (Oslo)
Analytics Server
Telemetry Agent
Marvell SDK
Telemetry Engines

Figure 11: Running the Telemetry Agent using OpenStack.

The data written to the Ceilometer database contains information gathered from the networking device - such as buffer counters, queue utilization, timestamping, µBurst durations, etc. Using real-time visualization and monitoring platforms operators can analyze their network's health, reduce packet loss, increase network performance, and improve the design of the physical network infrastructure.

5. Conclusion

Network telemetry has become a fundamental factor for network operators and vendors over the past decade, and the team at Marvell expect that it will continue to be the center of attention, as data center network scales continue to increase and 5G network technologies evolve. Furthermore, the constant shift of critical data into the cloud has created an even stronger dependency on the health and performance of the network, raising the need to continuously monitor and track it. Consequently, the ability to monitor the performance and health of the network, to detect congestion issues, failures and anomalies, and to respond to them in real-time has become a key component in every network. Marvell is an active participant in leading standard organizations and in open source organizations that define network telemetry technologies. Network telemetry is a key feature in Marvell’s portfolio of switching products, and will continue to be pivotal in future product introductions.

About the Authors

Tal Mizrahi, PhD

Feature Definition Architect

Tal Mizrahi is a feature definition architect at Marvell. With over 15 years of experience in networking, network security and ASIC design, Tal has served in various positions in the industry, including system engineer, team leader and, for the past 10 years, an architect for Marvell's networking product line. Tal received his BSc., MSc. and Ph.D. in Electrical Engineering from the Technion, Israel Institute of Technology. Tal is an author of over 40 published patents, and over 25 academic publications. He is also an active participant in the Internet Engineering Task Force (IETF).

Vitaly Vovnoboy

Principal Software Architect

Vitaly is a principal architect at Marvell. With over 20 years of experience in networking, software and system design, Vitaly has served in various positions in the industry, including team leader, software department manager and, for the past 7 years, a software architect for Marvell's networking product line. Vitaly received his MSc. in Software and Applied Mathematics from the Moscow State University of Transport (MIIT). Vitaly is an author of published patents and academic publications. He is also an active participant in open software projects, including OCP SAI/SONiC and OpenSwitch.

Moti Nisim

Head of Software and System Architecture

With over 15 years of experience in networking, including leading technical research projects, architecture design, and close work with Tier 1 customers, standard committees and institutes, Moti has served in various positions in the industry, including Chief Architect for 10 years, IP Services Manager and Engineering Team Leader. He was an editor and active contributor in the Metro Ethernet Forum (MEF) and also a technical lead in projects funded by the Chief Scientist in Ministry of Economy of Israel and European Union's Research and Innovation. Prior to that Moti did a duty service in MAMRAM, the Israeli Defense Forces' central computing and networking unit. Moti received his B.A. degree in Computer Science and Management from the Open University of Israel and he holds a Practical Engineering Diploma in Computer Engineering from the Technion Institute.

Gidi Navon

System Architect

Gidi Navon is a member of the Networking CTO team at Marvell. In his role, Gidi is defining new networking devices and software solutions for cloud infrastructure products. Specifically he is driving Network Telemetry solutions for Marvell's Switching portfolio. Gidi joined Marvell 5 years ago, after holding senior product and architectural positions at Nokia Siemens Networks for 7 years, defining carrier packet platforms. Previous to that, he held various system architecture position in leading silicon and system companies. Gidi received his Bachelor of Science in Electrical Engineering from the Technion Israel Institute of Technology and his MBA from Tel-Aviv University. He holds multiple patents in the field of networking and computer communication.

Amos Soffer

Application Team Manager

Amos Soffer is an Application Team manager at Marvell. In this role, Amos introduces Marvell's Ethernet technologies and helps customers in building systems including software and hardware solutions. Amos joined Marvell 15 years ago, after holding senior roles in TDSoft and Telrad Telco. Amos received his Bachelor of Science in Aeronautics Engineering from the Technion, Israel Institute of Technology.

References

[1] 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>.

[2] 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>.

[3] ITU-T, "OAM functions and mechanisms for Ethernet based Networks", ITU-T Recommendation G.8013/Y.1731, August 2015.

[4] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <http://www.rfc-editor.org/info/rfc6374>.

[5] C. Kim, A. Sivaraman, N. Katta, A. Bas, A. Dixit, and L. J. Wobker, “In-band network telemetry via programmable dataplanes,” in ACM SIGCOMM Symposium on SDN Research (SOSR), 2015.

[6] C. Kim et al., “In-band network telemetry (INT),” P4 consortium, 2015.

[7] Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and D. Bernier, “Data Fields for In-situ OAM”, draft-ietf-ippm-ioam-data (work in progress), 2017, <https://tools.ietf.org/html/draft-ietf-ippm-ioam-data>.

[8] Fioccola, G., 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, 2018, <http://www.rfc-editor.org/info/rfc8321>.

[9] T. Mizrahi, I. Yerushalmi, "The OAM Jigsaw Puzzle", technical white paper, Marvell, 2011. http://www.marvell.com/switching/assets/Marvell_OAM_Puzzle_001_white_paper.pdf

[10] Rijsinghani, A., Ed., "Computation of the Internet Checksum via Incremental Update", RFC 1624, DOI 10.17487/RFC1624, May 1994, <http://www.rfc-editor.org/info/rfc1624>.

[11] Mizrahi, T., Rottenstreich, O. and Y. Moses, “TimeFlip: Scheduling Network Updates with Timestamp-based TCAM Ranges”, IEEE INFOCOM, 2015.