White Paper

The Role of Network Visibility in the Borderless Enterprise

Gidi Navon, David Melman, Moti Nisim & Daniel Maryakhin

Switching BU, Networking Group, Marvell

July 2020

Abstract

Enterprise networks are becoming "smart." The network behavior and operation are dramatically changing, becoming intent-based, artificial intelligence-based, automated and self-healing. Any brain needs sensors and eyes, and in the networking world, these sensors are called "network visibility."

In this white paper we will focus on network visibility for enterprise networks: How visibility tools are evolving to address the new smart networks; what tools are needed; and how are they used in the network planning phases and in daily operation.

Marvell's visibility tools provide unique, dynamic, accurate, comprehensive and user-friendly capabilities — tools that allow the network to operate in normal times but are also prepared for catastrophes and disasters.

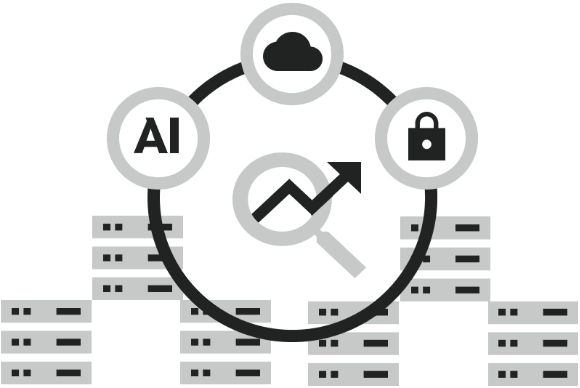
Introduction

The origins of network visibility can be traced to the carrier networks with what is called Operations, Administration and Maintenance (OAM) [1]. Carrier networks need monitoring tools as they are spread in very large geographical regions and as the carrier's clients expect a clear Service Level Agreement (SLA) for their paid services — one that is continuously monitored and guaranteed.

Enterprises now also demand exceptional visibility to their network, and it is an essential component to the success of their business. The users of the network are not only employees but also machines that cannot "open tickets." There is also no longer a single type of user. Each application can be considered a user, with very different needs and expectations from the network.

The IT department has also changed dramatically. Its members are expected to do much more with much less. They need to operate a complex, global and hybrid-cloud network from remote locations, often outsourcing to organizations specializing in network support. This cannot be done solely by humans. This is where computers and artificial intelligence (AI) can help the IT operations by creating an intent-based, predictive, automated and self-healing network. The intelligence is like any brain, and any brain needs sensors and eyes, and in the networking world, these sensors are called "network visibility."

Visibility tools need to provide continuous, up to date and accurate information about all aspects of the network. But the information provided cannot be just raw data tossed over all the time, as it is not practical to transfer, store and process all this raw data and still reach economical conclusions on how to fix and improve the network behavior in a timely manner. The art is in finding the right balance between selectively reporting what we know is important but also reporting information that we don't yet know that is important. The art is in the ability to flexibly export a wide set of metadata related to the network to a wide range of collectors.



In this white paper we will start by looking at the visibility needs of new smart enterprises, what is changing in the way they do network planning, capacity analysis and trend analysis. Then we will talk about common approaches for network visibility and for each of these approaches we will show examples of Marvell's visibility toolset offered by new members of the company's Prestera® family of devices.

Smart Enterprise Visibility Needs and Use Cases

The amount and type of network users as well as the complexity of the networks are continuously growing and changing. Let's examine these changes and how visibility tools are a pivotal part of these transitions by dividing the discussion into the network planning phase and the daily operation of the network.

Network Planning

IT managers always need to plan ahead. They must make decisions on upgrading the network from one generation to another, from one equipment vendor to another, from one Wi-Fi technology to the next, enlarging the campus, moving from a private cloud to hybrid cloud, adjusting the network to new business needs and of course, preparing for a disaster that may never come, but unfortunately does come as a “surprise” to all.

In this section we will discuss how visibility can assist in network planning; how can it allow faster and correct decision making; and how it is a crucial point for network elasticity.

When thinking about if, when and how to upgrade the network, a first step is monitoring the current usage of the network resources in different time periods of the day and the year. Depending on the organization type, different decisions may be made. Some organizations will try to make sure that even in peak usage, the network is overprovisioned, while others may accept a certain amount of congestion and packet drops of low-priority packets. IT will need to learn the trends in bandwidth growth over time, in order to be able to estimate the time when an upgrade will be needed.

A common mistake in capacity monitoring is looking at averages on long intervals. Imagine a traffic scenario over a 10GbE link, as shown in Fig 1, with quiet intervals and busy intervals. Checking the rate over long intervals of seconds will only reveal the average port utilization of 25%, giving the false pretense that the network has high margins, without noticing the peak rate. The peak rate, which is close to 100%, can easily lead to egress queue congestion, resulting in buffer buildup and higher latencies.

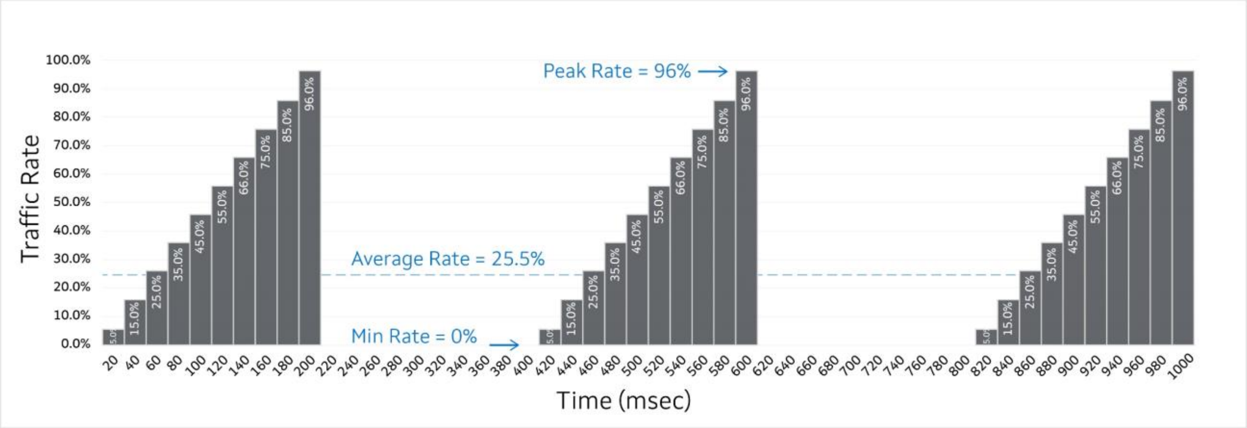


Figure 1: Port Utilization Traffic Scenario

Planning for the future, based on natural growth trends, can lead to wrong conclusions. Disruptive changes are inevitable, may they be special dates on the calendar, like heavy ecommerce dates (Black Friday, Cyber Monday and China's Bachelor day), national holidays, company live events, etc. So how can visibility help to plan for such events?

To prepare for the periodic "big" events, accurate monitoring and telemetry should be continuously done throughout the year. But in order to plan for one-time known events, visibility tools should provide measurements on more granular elements: Understanding the traffic behavior of a single user while utilizing a specific application can help plan for a one-time event involving 10K simultaneous users. The IT manager should prepare his network capacity to hold the traffic of 10K such users.

Most challenging is to plan for the unexpected events and the catastrophic events. However, the same visibility database of granular elements should also assist here, by planning for theoretical events based on these granular elements: For example, while the planned capacity of my hospital is for N patients, I want my network to be prepared for e.g. 10N patients.

Another use of having a granular element visibility database (i.e. per user, per application) is for enterprises planning to move the compute and storage resources to the cloud. Imagine an enterprise that has decided to move some of its critical applications to the cloud. One of the costly elements is leasing the connection to the cloud provider, but what is the rate to be leased? At least the peak rate that is currently used when the applications are on-premises. This information should be provided by the visibility tools.

While telemetry is in many cases associated with QoS (measuring rates, latencies, buffer usage and more), there are other networking parameters and resources that need to be continuously monitored. One such example is resource table usage. The introduction of thousands of new Internet of Things (IoT) sensors or organizing a big event with an unprecedented number of people may consume internal resources like forwarding database entries, IPv6 host tables and security access control lists. Such parameters may be hard to calculate as they may depend on the momentary physical location of entities that may roam using Wi-Fi connectivity.

Planning for the Multi-Cloud and Hybrid Cloud

Many organizations are moving resources to the cloud. A hybrid cloud environment means that some resources are kept in the organization while others are migrated to the cloud. A multi-cloud environment means that several cloud providers are chosen by the same organization, each for a different application or for a different global presence.

The planning of such moves and the continuous monitoring in daily operation after the migration is complete requires network visibility. Measuring the peak rates toward each application (when the applications were still installed on-premises) will define the rates to be leased towards the cloud provider. Continuous monitoring of the usage during different occasions with different workloads will allow accurate trend analysis and dynamic elasticity, as expected from self-organized networks, that may even lead to changes in selection of the cloud provider or the service towards the cloud provider.

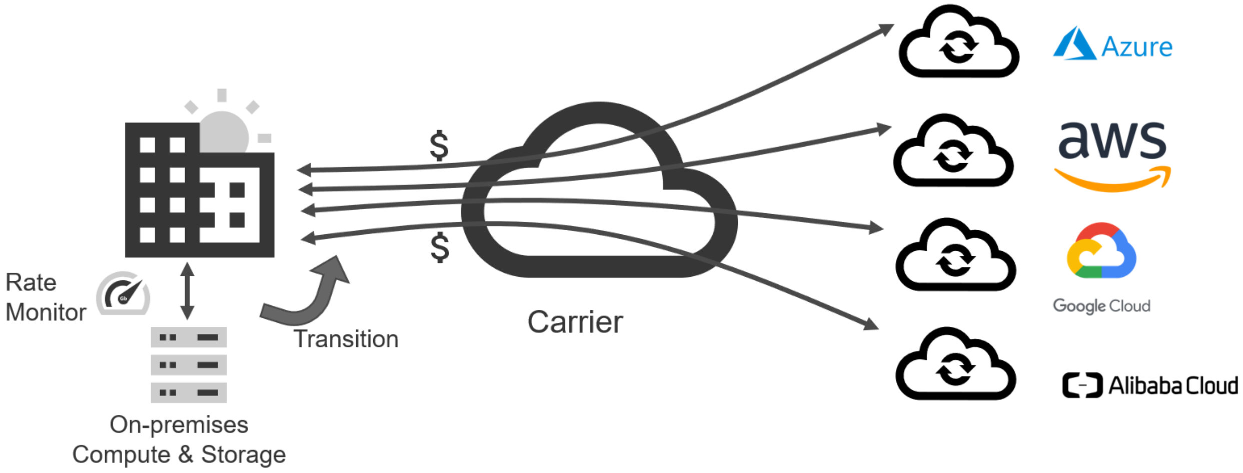


Figure 2: Cloud Migration

Daily Operation

Having the correct network equipment installed does not end the worries of the IT department. Networks need to be properly configured, adopted to new users of the network and continuously optimized for the changing requirements (optimization that may be dynamic and hopefully automated). Troubleshooting and healing of the network is another aspect of the daily operation.

Traditional monitoring tools provided very long logs on configuration errors, information on drop reasons and more. While all this should be kept and enhanced, to allow automated networks and self-healing networks, new visibility capabilities are expected.

Self-Organized Networks

The amount and type of network users is continuously growing. Each application and each IoT device can be considered a user of the network and may have very different traffic profiles and network needs. The network dynamics in some enterprises is very high, where each entering client is a new network user that needs to be handled and configured.

As the first step in a self-organized network, the visibility tools are expected to identify new users of the network, help classify them into groups and assign policies to them. Visibility tools will allow understanding of who is driving the data and ensure proper handling of each granular entity.

Self-organized networks can also be applicable to network planning and upgrade, and not just to daily operation. We can imagine a time that an automated application will issue a purchase order of a new networking device and will come with clear instructions to the IT staff, on where and how to install the new device.

Auto QoS Assignment

Application aware QoS assignment is an important objective for self-organized networks. This is a pain IT staff are struggling with. In an ideal world each network user and application will assign its packet the correct QoS tag (e.g. the IP DSCP – Differentiated Services Code Point) according to an IT policy. But how can IT trust that the applications follow the IT policy? How can IT control all the desktop applications and all the IoT devices introduced daily? The days in which VoIP traffic was originated by identifiable devices is long gone, and now the source of traffic is just an app on the laptop or mobile phone, or a voicemail on a voice gateway.

Many enterprise networks have given up on the task to properly configure the QoS inside the network and focus their efforts on the edge of the campus/branch office towards the expensive WAN links. They do so by using high-end appliances that classify the packets into applications, and then assign proper QoS to them. These appliances are both expensive and a bottleneck on the network performance, especially as a considerable amount of traffic is sent from the branch office to the cloud / private data centers.

Distributing the task of application classification to the access switches, close to the sources of the traffic, will solve both issues: Apply proper QoS inside the network and ease the burden on the such appliances, and reduce the total cost of ownership (TCO).

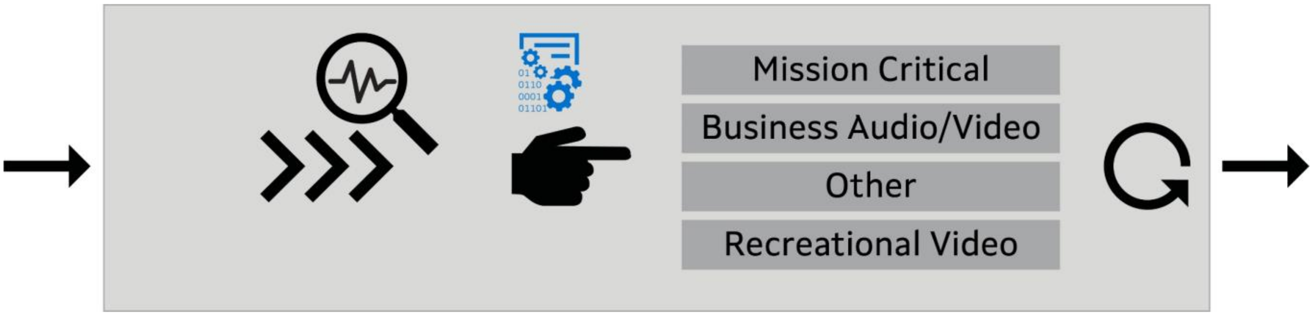


Figure 3: QoS Self-Configuration

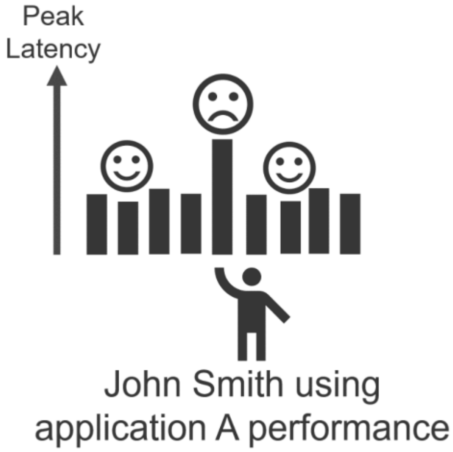
Performance Monitoring

Latency is the key performance indication users of the network care about. A classical debate is who is responsible for the sudden application slow-down. Is it the compute or is it the network? While networking people tend to say the problem is in the compute, one needs to measure and provide visibility evidence to this. If the network is causing the delay, localize the problem, identify the cause and solve the issue, hopefully before users of the network notice any disruption. We will discuss in the next chapters the different approaches for latency monitoring.

Performance monitoring also includes loss measurements. And loss in a network could be due to congestion on queues, or due to drops during the processing of the packet. These drops could be intentional drops (packet violating some security rule) or drops that we want to avoid such as those brought on by wrong configurations. In all cases, the nodes in the network need to report meaningful information on the drop reasons, number of drops, who was dropped and when.

As in other visibility capabilities, the challenge is to provide the information in a useful manner. No need to tell about every packet that was dropped or about the latency of every packet. This can be overwhelming and uncontrollable. It's better to provide the information in a context that will be helpful. For example:

- This flow experienced many drops as of this specific reason
- The latency of packets classified belong to a Zoom application that are distributed as shown in this latency histogram
- The top talkers on my networks are the backup procedures, and they are all happening at the same minute at midnight. Spreading them to different times will reduce my WAN expenses



Anomaly Detection

Detecting anomalies in the network behavior is key for identifying several events: security attacks, misuse of the network resources, and configuration errors that are causing the network to behave badly. It is also a technique in detecting network faults.

A key technology in anomaly detection is AI which in many cases uses machine learning. For detecting anomalies, one must first teach the algorithm what is considered a normal behavior. The challenge is that the normal behavior in many networks is not stable and includes patterns which are hard to anticipate. Experiencing high traffic volume in e-commerce dates is very normal, but one would need several yearly cycles to learn that, unless some human or external information is fed to the equation. In addition, the network behavior is very different in different places in the organization, and of different users of the network, so learning the normal behavior needs to be done for almost each port and each user.

In any case, for the learning process a considerable amount of information needs to be collected and streamed to the AI engine.

The information includes traffic rates associated to network users in different locations of the network. High bandwidth rates from a legitimate employee may be acceptable, but when this employee is sending traffic from the data center at night, this for sure needs to be alerted and checked.

The amount of traffic that needs to be sent, both for the training phase and for the inference stage, may end up being massive, unless some intelligence is moved to the nodes to provide visibility of just the “important stuff.”

Suspicious Flow Detection

Flows can be classified, not only to their corresponding application, but can also be identified as malicious or benign flows, just by inspecting the beginning of new flows and by examining their traffic patterns. The traffic pattern triggered by a human user accessing a corporate resource may look different than the traffic pattern of an attack initiated by a computer that tries to bring the service down.

While many anomaly detection algorithms are based on an unlabeled data set (both unsupervised and semi-supervised), malicious flow detection is usually based on labeled data of well-known attacks, and in this aspect, is closer to application classification.

Advanced switches are expected to provide the required information of any new flows to allow such classification. And, advanced switches should be able to stream this information without missing any new flows, even when the number of new flows per second is high.

Application and User Awareness

Many IT managers ask themselves, “Who is actually using my network? Who are the heavy hitters? Who are the top talkers? Which application dominates the network usage? And when?”

The answer to these questions should not be based only on the total amount of GBytes consumed e.g. per day. This is because in most cases that instantaneous use is the important metric. The impact on the network performance will be mostly affected by high-rate bursts, even if short-lived.

Providing continuous information on rates, peak rates, and rate distribution will provide valuable information in tuning the network.

However, the legitimate attempt to reduce the amount of visibility information needs to be done with caution: Providing the analytics server with the network usage of each application and, separately, the network usage for each user is probably not good enough as the collector will not be able to deduce from it the usage per application of each user. Adding to the challenge is identifying users and applications in overlay networks.

Visibility tools should provide continuous information that is granular enough in order to allow the collector to perform cross references and flexibly deduce intelligent conclusions.

Network Visibility Approaches

Networking visibility tools are evolving to address the needs and use cases discussed in the previous chapter. In this chapter we will discuss approaches for network visibility in terms of performance monitoring, monitoring the use of resources, and providing insights on the network users and applications — what is known as network flows. For each of these approaches we will also show examples of Marvell's visibility toolset, important for creating an automated self-healing network.

Performance Monitoring

Performance monitoring usually refers to tools that show latency and packet loss (if it exists). After all, networking is all about moving packets around quickly and safely. Performance being dynamic and changing means that it should continually be monitored and reported, and thus visibility tools come into play.

However, looking only at latency and loss is not enough as things can be all nice and good, but suddenly one might see dramatic degradation of performance as network resource are exhausted. Monitoring resource usage is also very important and will be discussed in the next section.

Latency in networks is built from four parts: propagation delay (according to length of fibers and copper), processing delay (which is negligible in modern switches), queuing delay and transmission delay (a.k.a. serialization delay, which depends on port speed). Each networking node is expected to report its contribution (processing + queueing + transmission) out of which queueing delay is usually the biggest contributor by far.

When looking at the queueing delay there are two approaches: measuring the size of the queues in the packet buffer (the source of the latency) or monitoring the actual latency a specific packet has encountered.

Providing the actual latency is easier to understand and easy to compare as it already takes into consideration the result of the scheduling scheme (high vs. low priority queues). However, providing the latency of each individual packet is too much information. Peak latency (if different from the min latency) is more useful to know and similarly, peak queue fill level is a good indication of congestion events that one may want to avoid.

All packets transmitted via a specific queue will basically experience the same latency. In order to reduce the amount of visibility information, some may argue that it is better to measure only latency information of the queues. This approach ignores the case that a lucky flow may arrive to a queue each time the queue is empty and will not experience the latency that other flows do using the same queue. Even when monitoring latency on queues only, reporting the latency on a per flow basis is more meaningful.

The other approach, monitoring the size of the queues in the packet buffer (and not only latency), is also useful for adjusting scheduling mechanisms, drop thresholds and LAG distribution algorithms.

Loss measurement is the second performance metric, and as discussed in the previous chapter, the reporting must be meaningful and include the drop reasons, number of drops, who was dropped and when. The common approach for reporting loss is on a per flow basis based on packets that share the same fate. For example, all packets of a flow will match the same access-list drop rule. For loss as of congestion, or as of policers, reporting the number of drops per flow will provide meaningful information on the victims of congestion and on the aggressors that violated some rate policy.

Marvell’s Advanced Performance Monitoring

In Marvell’s previous Telemetry White Papers [2], we showed various capabilities to measure and export loss and delay, active measurements using a variety of OAM protocols, in-band monitoring using INT [3] & IOAM [4] and hybrid measurements using an innovative alternate marking technique [5].

For many enterprise networks, the technique that makes most sense for performance monitoring is a passive monitoring technique which continuously tracks the latency of all queues. The valuable information that is useful for understanding trends, anomalies, and catastrophic events can be gathered by providing histograms on the average and peak latencies.

Figure 4 shows peak latency on a specific queue. The Y-axis shows the percentage of time inside a specific second the queues had the latencies shown on the X-axis. This is a live graph, in which each color represents a specific second during the day. In this example, the last six seconds are shown, and in a live scenario, the graph moves as data from a new second replaces the older data. Let’s examine the ‘red’ second (the one on the left of each latency bin): It shows that for 12% of this second, the queue latency was between 0-10usec, 15% of this second latency was between 10 and 20usec, and so on. Such data should be sent to an analytic server for anomaly detection, trend analysis and more.

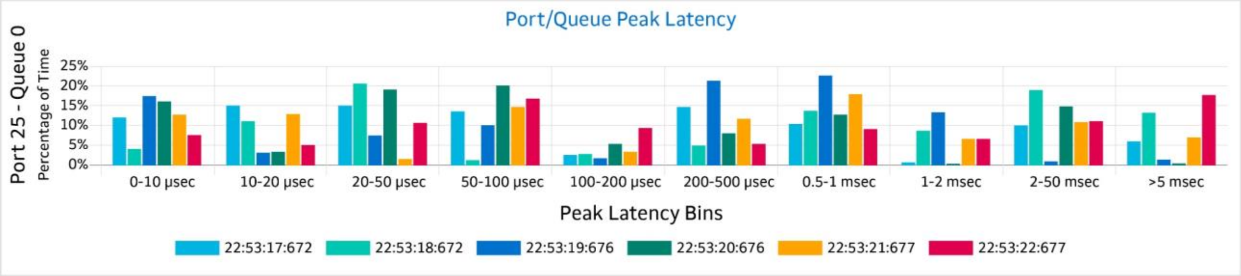


Figure 4: Peak Latency Monitoring shown in Histogram

While the above information is provided ‘only’ per queue, and ‘only’ in latency bins, it provides rich enough information that does not overwhelm the analytic server.

The above information can be translated, in most cases, to be on a per flow basis, for all flows using these queues. This will allow tracking and understanding the network performance of e.g. a specific user using a specific application, and clearly understand whether the network is at fault or not when things are not running smoothly.

Resource Monitoring

Monitoring switch resources is critical for anticipating situations in which resources are becoming fully consumed and performance is about to degrade.

The switch resources can be its forwarding tables, internal measuring units like the switch counter pool, its packet buffer resources and the use of the Ethernet links themselves. As we discussed, understanding how utilized a port is, is key for network planning, and upgrade decisions from e.g. 1G to 2.5G copper links.

The approaches for reporting resource usage depends on how dynamic the changes are. Port utilization, for example, continuously changes, but cannot be and does not need to be continuously reported. A common approach for such fast-changing parameters is to report information in rate histograms.

For example, Figure 5 shows a report for a specific second, and details the percentage of time inside this second that the port was between different port-utilization bins. This diagram shows, that while on a second-average, the utilization is about 2.5%. One might think that the utilization is very low, however, in actuality during 13% of the time in this second, the utilization is more than 80%.

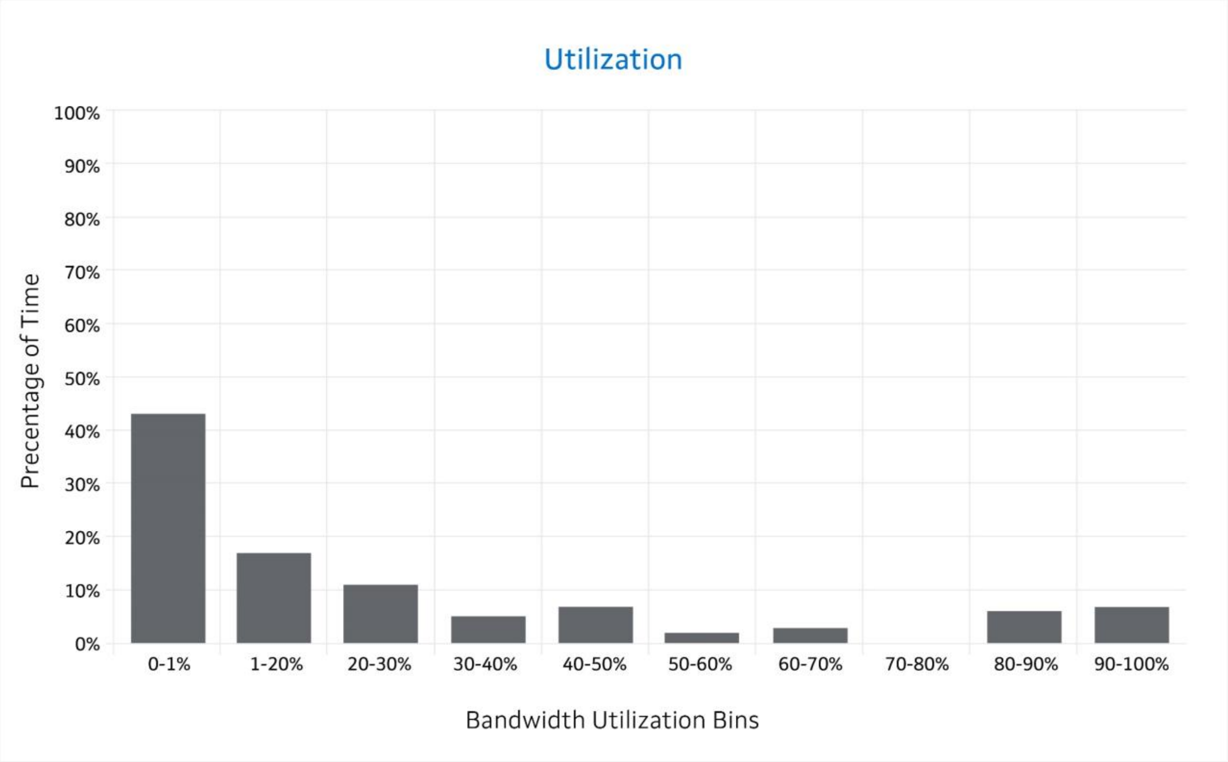


Figure 5: The use of Histograms for Monitoring

This technique of histograms can be used for other fast changing parameters like buffer usage and more. This intelligent and reduced data can be easily streamed to an analytic server to perform trend analysis for anticipating situations in which resources are becoming fully consumed and performance is about to degrade.

Marvell's Advanced Resource Monitoring

Marvell's resource monitoring techniques include the ability to continuously monitor the rates of all queues in the system, and calculate minimum, average and peak queue rates in sub-msec time intervals.

Figure 6 shows a graph of the minimum, average and peak rates (expressed as a percentage of the port's speed), for a specific queue over time. While the shown samples are slow (every 1 sec), the underlying measurements are fast in order to measure the correct values. Note that slow monitoring techniques would have not detected the correct peak rate and would have created the false notion that peak rate is much lower.

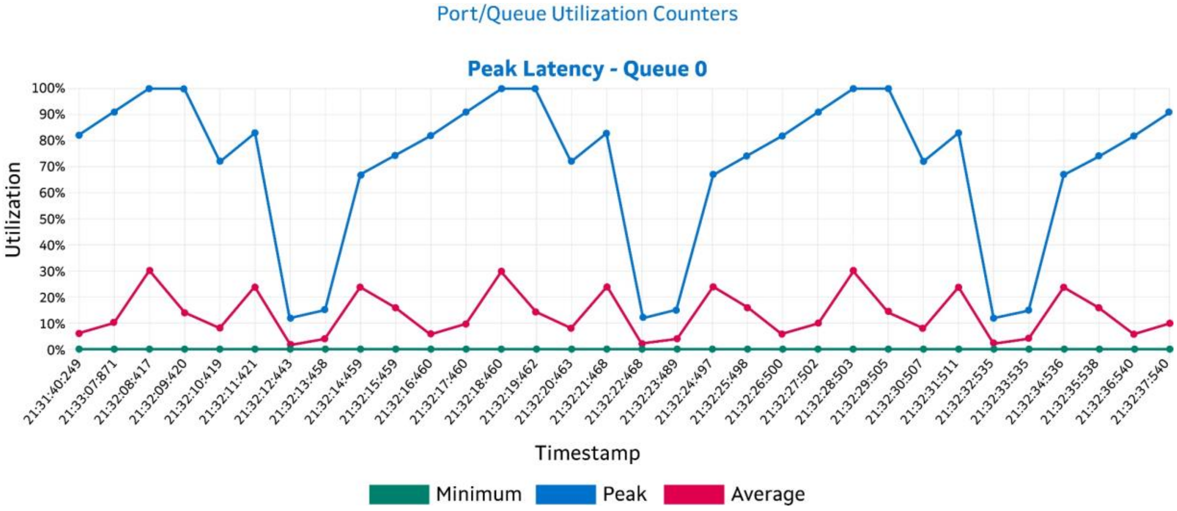


Figure 6: Monitoring Minimum, Average and Peak Resource Usage

In addition, the fast-changing rates inside each second, are placed in rate histograms that reveal the percentage of time, inside each second, a queue was transmitting at a specific rate interval. Figure 7 shows such a histogram of a specific port. The million per second data points (rates of thousand queues in sub-msec intervals) are optionally reduced to only ten thousand data points per second, allowing continuous monitoring of the entire network all the time, with meaningful information.

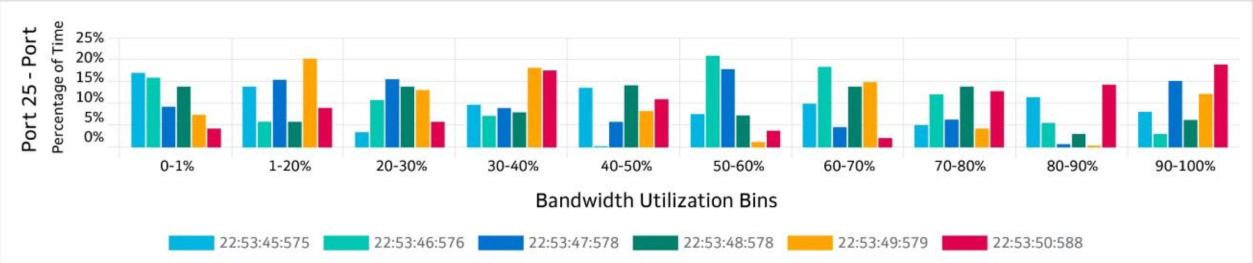


Figure 7: Resource Monitoring shown in Utilization Histogram.

Flow Monitoring

In the trade-off of providing visibility information for every packet and providing course /aggregated information on nodes, ports and other shared resources, the most common approach is providing information on flows.

Flows are not only a good balance between the two, but also a very user-friendly entity to relate to, as they provide visibility into the network usage of specific users and specific applications.

There is no one definition of what is a "flow." RFC 5470 [7] provides a definition for a flow that is very broad and offers the flexibility for each user to define the "flow" that is of interest. Most define a flow as the packets that have the same 5-tuples: source and destination IP, protocol, and source and destination UDP/TCP ports as it basically monitors each TCP/UDP session. Others prefer monitoring only 2-tupples (source and destination IP). The additional fields of a flow, like the Layer 2 addresses, the priority fields, VXLAN tags and more, don't need to be part of the key and are considered as attributes of the flow.

Providing visibility information on 5-tuples, allows a collector to later calculate course metrics on 2-tuples, so it is better to collect this fine-grained information. However, this also means that the number of monitored flows is much higher, and so is the rate of new flows per second.

IP Flow Information eXport (IPFIX)

The mechanism to export flow information from the switch is via a protocol, defined by the IETF called IPFIX [7]. Its main components are a metering process that creates new flow records, updates existing records with flow record statistics, detects flow expiration, and passes the flow records to the exporting process. All this is done for packets passing an observation point (e.g. a switch port). The Exporter that is also located at the switch sends the flow records to one or more collectors that store all the records from all devices and can perform deeper analysis and presentations.

The flow record can hold and an endless set of flow attributes, or information elements (IEs). The IETF defined about 500 such attributes [8] and each vendor selects which to support. In addition, vendors can define their own vendor-specific attributes. Solutions differentiate themselves by the type and number of attributes they are capable of providing for each flow.

Among the attributes one can find basic things like the ingress and egress port, when the flow started, how frequently sampled, and how many packet and bytes sent (since the flow was started and since the last report). Some of the attributes can be gathered by examining the beginning of each flow while others require continuous monitoring.

Usually the IPFIX Exporter sends the flow records periodically to the collector, in a rate defined by the network operator according to the capabilities of the exporter, the capabilities of the collector and the amount of bandwidth the network operator wants to spend on this task. Usually the rate is relatively low, every few seconds or even minutes. Modern and dynamic networks require much more frequent export and more intelligent attributes about the flow behavior.

Flow Application Awareness

An important attribute of a flow is the name of the application using this flow. While many applications reveal themselves by using well-known TCP/UDP ports, many new applications prefer to deliberately not use the well-known ports. Many modern applications and protocols are just running on top of http/https, creating the challenge to classify them individually.

A technique called DPI (Deep Packet Inspection) is used to dig into the payload of the packets to reveal the application. But the most modern technique is to look at the http SNI (Server Name Indication) defined as part of TLS (Transport Layer Security). While SNI was created to solve certificate issues in multi-hostname environments, it can be used for application classification also of encrypted https sessions as the SNI field is sent on the clear.

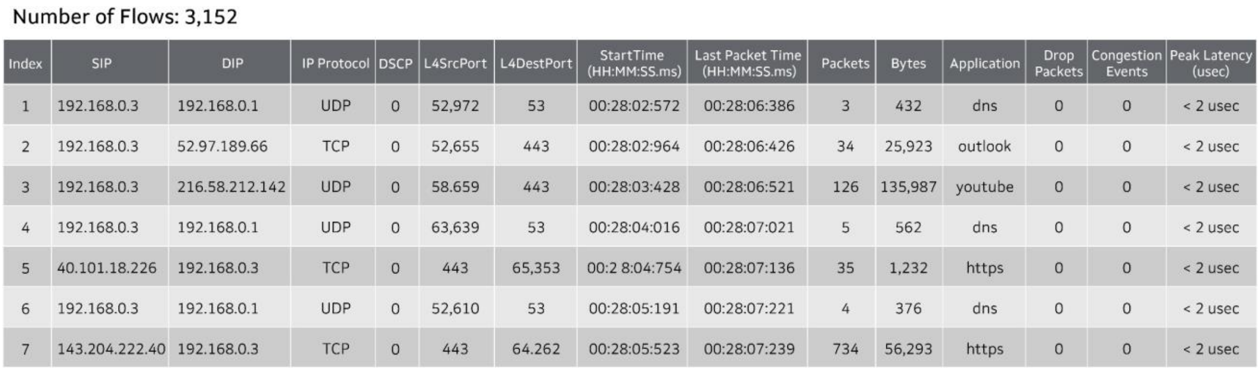
Some are suggesting encrypting the SNI field as well, in a solution called eSNI [9] though the mainstream adoption of this approach is yet to be seen. For sure, this will become a challenge for application classification.

Studies have shown that by inspecting the beginning of new flows, the connection set-up procedure, and the size and gaps between the first packets of a flow, one can identify the application of a flow, even for encrypted connections. More advanced techniques can even single out malicious flows from benign flows by inspecting the same set of parameters.

Marvell's Advanced Flow Monitoring

Marvell's advanced flow monitoring solutions provide a combination of smart, scalable and fast flow monitoring technologies with the ability to immediately learn new flows as they are born. By examining the first packets of each new flow, together with the exact arrival time and packet size (thus also the gaps between the packets), one can identify the application behind each flow.

Figure 8 illustrates a possible flow monitoring table that shows data about each flow passing the switch. This includes information about its application, the number of drop events and the number of congestion events (that may not even lead to any drop). Peak latency can be derived from the peak latency measured on the queue used by this flow, thus providing predictive indication for possible drops in the future.



Number of Flows: 3,152

Index	SIP	DIP	IP Protocol	DSCP	L4SrcPort	L4DestPort	StartTime (HH:MM:SS.ms)	Last Packet Time (HH:MM:SS.ms)	Packets	Bytes	Application	Drop Packets	Congestion Events	Peak Latency (usec)
1	192.168.0.3	192.168.0.1	UDP	0	52,972	53	00:28:02:572	00:28:06:386	3	432	dns	0	0	< 2 usec
2	192.168.0.3	52.97.189.66	TCP	0	52,655	443	00:28:02:964	00:28:06:426	34	25,923	outlook	0	0	< 2 usec
3	192.168.0.3	216.58.212.142	UDP	0	58.659	443	00:28:03:428	00:28:06:521	126	135,987	youtube	0	0	< 2 usec
4	192.168.0.3	192.168.0.1	UDP	0	63,639	53	00:28:04:016	00:28:07:021	5	562	dns	0	0	< 2 usec
5	40.101.18.226	192.168.0.3	TCP	0	443	65,353	00:2 8:04:754	00:28:07:136	35	1,232	https	0	0	< 2 usec
6	192.168.0.3	192.168.0.1	UDP	0	52,610	53	00:28:05:191	00:28:07:221	4	376	dns	0	0	< 2 usec
7	143.204.222.40	192.168.0.3	TCP	0	443	64.262	00:28:05:523	00:28:07:239	734	56,293	https	0	0	< 2 usec

Figure 8: Flow Monitoring with Additional Metadata

In-band Telemetry

In-band telemetry was widely discussed in Marvell's previous Telemetry White Papers [2]. In a nutshell, it is a mechanism in which each node, along the path of a packet, adds timestamps and other information to the headers of the user's packets. So, when the packet arrives to its destination, it includes valuable information on which nodes it went through, and how much time it spent in each node on its way [5][6]. In-band telemetry gained popularity in data centers, and it is yet to be adopted by enterprise networks.

There are some aspects in the in-band telemetry solution that are holding back its adoption.

First is the fact that it alters the user's packet and enlarges it and consequently, there is a risk that the packet may not arrive to its destination as originally sent. For this there are two proposed solutions:

One is that the last node – the node that is connected to the destination user — strips the extra telemetry data, sends this telemetry data to a collector, and sends the original packet only to its destination. But still, inside the network, packets are changed and may result in different behavior.

Another solution is that the telemetry data is not added to the user's packet, but to a clone packet that resembles the original packet, thus following its path and sharing the same fate and performance as the original packet; monitoring the clone is same as monitoring the original packet. This solution adds more packets to the network, so usually configured not to clone every packet but only a sample of packets from each flow.

The information gathered by the egress node can be directly streamed to an analytic server and can also be incorporated into a flow record database, adding additional meaningful data.

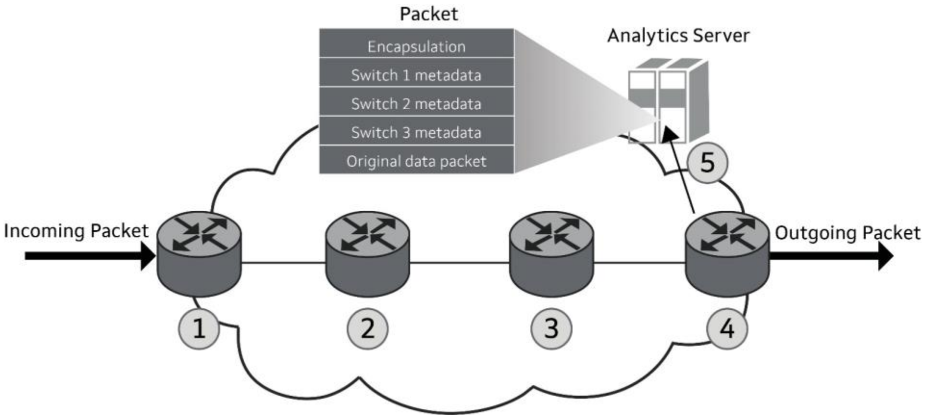


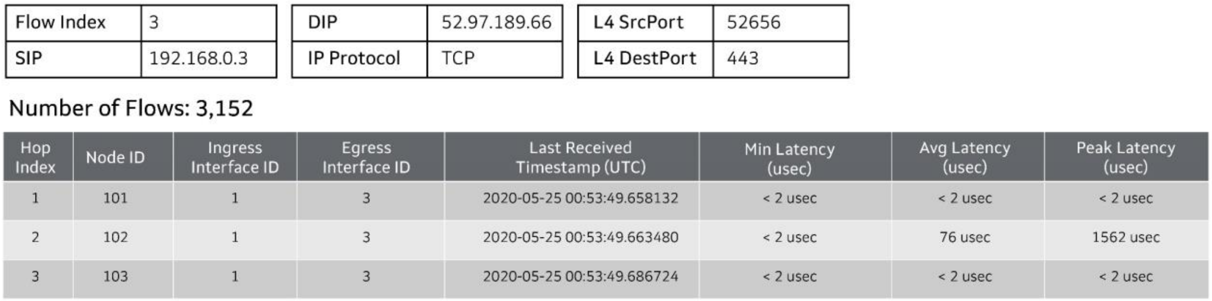
Figure 9: In-band Telemetry

Marvell's Advanced In-Band Telemetry

Marvell's capabilities for in-band telemetry is explained in the Telemetry White Papers [2]. In this section, we will discuss Marvell's view and solutions for using in-band telemetry in enterprise networks.

Marvell's approach is that the access switches that are connected to the users and servers will process multiple packets with in-band telemetry information, calculate some metrics and provide this data per flow. Gathering info from all access switches and using the unique capabilities of in-band telemetry, one can create a comprehensive view, showing for each flow: where it came from, which switches it passed on its way, how much time did it spend on each node and more. The increased use of mobility, and the use of complex hybrid and multi-cloud environments makes this information valuable more than ever before.

Figure 10 shows the information gathered from a single access node for a specific flow. It shows which nodes the flow passed through, and what was the minimum, average and peak time spent at each node.



Flow Index	3	DIP	52.97.189.66	L4 SrcPort	52656
SIP	192.168.0.3	IP Protocol	TCP	L4 DestPort	443

Number of Flows: 3,152

Hop Index	Node ID	Ingress Interface ID	Egress Interface ID	Last Received Timestamp (UTC)	Min Latency (usec)	Avg Latency (usec)	Peak Latency (usec)
1	101	1	3	2020-05-25 00:53:49.658132	< 2 usec	< 2 usec	< 2 usec
2	102	1	3	2020-05-25 00:53:49.663480	< 2 usec	76 usec	1562 usec
3	103	1	3	2020-05-25 00:53:49.686724	< 2 usec	< 2 usec	< 2 usec

Figure 10: Flow Tracking Based on In-band Telemetry

Summary

In this white paper, we showed how network visibility tools serve as the eyes of modern enterprise networks and enable self-organized and self-healing networks.

Marvell's visibility tools provide unique, dynamic, accurate, and comprehensive — but not overwhelming — capabilities. These tools include peak latency monitoring, resource utilization monitoring that anticipates problems before they happen, histogram tools that summarize information as meaningful data without overwhelming the collectors, and advanced flow monitoring that doesn't miss any short-lived flow and provides enhanced data on the network users by deploying in-band telemetry techniques. Utilizing these tools will ensure continuous visibility especially when IT managers are resting their eyes.

About the Authors

Gidi Navon

Principal System Architect

Gidi Navon is a member of the Switching CTO team at Marvell. In his role, Gidi defines new networking devices and software solutions, and drives the introduction of new technologies into Marvell's infrastructure products. Specifically, he is responsible for leading initiatives focused on network visibility solutions for the switching portfolio. Gidi joined Marvell eight years ago, after holding senior product and architectural positions at Nokia Siemens Networks for seven years, defining carrier packet platforms. Previous to that, he held various system architecture positions at leading silicon and system companies. Gidi received his Bachelor of Science degree 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.

David Melman

System Architect Technical Director

David Melman has been with Marvell's switching architecture team for the last 20 years, leading the feature definition of the Prestera family of packet processors. He is active in the IETF standards organization and is a co-author of multiple IETF drafts. David has authored many patents in the area of networking silicon features. He holds a Bachelor of Science degree in Computer Science from UCLA.

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 Nisim 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 Bachelor of Arts 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.

Daniel Maryakhin

Software Engineer

Daniel Maryakhin is a software engineer in the "Tools and Infrastructure" team of the switching platform at Marvell, a position he has held for the last four years. His main responsibility is developing modern tools to help engineers and customers operate, debug and monitor switch devices in a clear and easy way, focusing on the visual aspect and user experience. He's also responsible for the applications shown in many of Marvell's customer demos. Prior to that, Daniel served as a programmer in "Ofek", Israeli Air Force's computing unit, after finishing the MAMRAM course. He received his Bachelor of Science degree in Computer Science from the University of Haifa.

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) Tal Mizrahi, Vitaly Vovnoboy, Moti Nisim, Gidi Navon, Amos Soffer, "Network Telemetry Solutions for Data Center and Enterprise Networks", March 2018 <https://www.marvell.com/content/dam/marvell/en/public-collateral/switching/marvell-telemetry-white-paper-2018-03.pdf>

3) C. Kim et al., "In-band network telemetry (INT)," P4 consortium, 2015.

4) 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>.

5) 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>.

6) 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.

7) G. Sadasivan, N. Brownlee, B. Claise, J. Quittek, “Architecture for IP Flow Information Export”, RFC 5470, March 2009, <https://www.rfc-editor.org/info/rfc5470>

8) IP Flow Information Export (IPFIX) Entities http://www.iana.org/assignments/ipfix/ipfix.xhtml

9) Encrypted Server Name Indication for TLS 1.3 https://tools.ietf.org/html/draft-ietf-tls-esni-06

July 2020