AmLight-INT: In-band Network Telemetry to support big data applications

Jeronimo Bezerra - IT Associate Director/FIU
Arturo Quintana - Sr. Software Developer/FIU
Outline

• Network Monitoring: Current Limitations and Technologies
• Introduction to Network Telemetry
• Current Network Telemetry Efforts
• Telemetry at AmLight: A Use Case
• Next Steps
Warming Up!

All networkers in the call have heard at least one of these questions from users:

• *Why am I not getting full bandwidth for my download?*

• *Why is my application achieving such a poor network performance?*

• *Where is the packet loss?*

• *May I be placed outside of the firewall?*
Current Network Monitoring Limitations and Technologies

• Answering questions about network visibility and performance and isolating faults affecting data transfers are still highly complex
  • Hardly answered in real-time.

• Example of questions network operators ask themselves that usually have incomplete answers:
  • “Where is the 1:1,000,000 packet loss?”
  • “Are there [micro] bursts happening now?”
  • “Which network queues are running at their maximum capacity?”
  • “Which flows were using the network queues when network queues were full?”
Current Network Monitoring Limitations and Technologies [2]

• Complexity exists because we are still leveraging legacy tools
  • Traceroute, ICMP, SNMP, NetFlow, RMON, and sFlow.
    • Such tools can collect statistics based on samples or on-demand tests.
    • Port-mirror and network taps operate out-of-band and provide good visibility but impose huge challenges for scalability in a multiple connections scenario.

• New protocols, such as OpenFlow and Ethernet OAM, do not capture transient events, such as a microburst, nor do they enable network operators to find the root cause of many network anomalies.
Current Network Monitoring Limitations and Technologies [3]

• To add complexity, new \textit{real-time} big-data applications are being created with very strict Service Level Agreements (SLA).

• Any attempt to track network state in \textit{real-time} could become a very complex and expensive task.
  • Polling SNMP or OpenFlow counters is not recommended in a sub-30s interval.
  • Sampling technologies usually export data after a few seconds.
  • Packet sniffing at 100G has a high CAPEX and OPEX.

• Identifying microburst or isolating packet loss in \textit{real time} is not trivial with current technology.
Introduction to Telemetry

• **Telemetry** is a very popular concept in the *medical* and *geology* environments.

• Medical telemetry refers to the machines that patients are hooked up to when they are at risk and need to be continuously monitored.

• There are many important **metrics**:
  - Examples: cardiac rhythms, blood pressure, and blood oxygen levels.

• Important **properties**:
  - Accuracy, utility of each metric that is measured, granularity, sample rate, and alerts in case of a problem.

• **Real time detection and notification is required.**
Introduction to *Network Telemetry*

- **Network telemetry is the extension of network reporting to higher granularities and sample rates combined with actionable metrics and alerting** [1]

- Network telemetry technologies define several characteristics [2]:
  - Push and Streaming: Instead of polling data from network devices, the telemetry collector subscribes to the streaming data pushed from data sources in network devices.
  - The data is normalized and encoded efficiently for export.
  - The data is model-based which allows applications to configure and consume data with ease.
  - Network telemetry means to be used in a closed control loop for network automation
  - Also known as *streaming network telemetry* or *streaming telemetry*

- Streaming network telemetry is very useful to detect microburst and queue utilization at a sub-second interval

- With all historic network state, forensic troubleshooting is enabled

Example: Microbursts vs. Telemetry
Example: Microbursts vs. Legacy Monitoring

Are you sure there was a microburst in the last minute?

Source: https://www.arista.com/assets/data/pdf/TechBulletins/AristaMicrobursts.pdf
New Telemetry Trends @ IETF and ONF

• In 2016, P4.org group create a new P4 application:
  • In-band Network Telemetry (2016)

• IETF Internet Protocol Performance Measurement (ippm) WG:
  • Proof of Transit (2016)
  • Encapsulations for In-situ OAM Data (2017)
  • Data Fields for In-situ OAM (2017)
  • Requirements for In-situ OAM (2018)

• IOAM, In-situ OAM, In-band OAM, INT, In-band Network Telemetry are used interchangeably in this presentation.
In-band Network Telemetry (INT)

- INT is an implementation to record operational information in the packet while the packet traverses a path between two points in the network:
  - Complements current out-of-band OAM mechanisms based on ICMP or other types of probe packets.
  - Basically, INT adds metadata to each packet with information that could be used later for troubleshooting activities.

- Example of information added:
  - Timestamp, ingress port, egress port, pipeline used, queue buffer utilization, WiFi link power, CPU utilization, Battery Utilization, Sequence #, and many others

- As metadata is exported **directly from the Data Plane**, Control Plane is not affected:
  - Translating: *you can track/monitor/evaluate EVERY single packet at line rate.*
Questions addressed by INT

• **How did this packet get here?**
  • The sequence of network devices a packet visited along its path.

• **Why is this packet here?**
  • The set of rules a packet matched upon at every switch along the way.

• **How long was this packet delayed?**
  • The time a packet spent buffered in every switch, to the nanosecond, from end-to-end.

• **Why was this packet delayed?**
  • The flows and applications that a packet shared *each queue with*. 
INT: How does it work?

Source: NoviFlow INT Setup Guide
Introduction to AmLight

- AmLight Express and Protect (AmLight-ExP) (NSF International Research Network Connections (IRNC) Award #1451018)
- 680Gbps of upstream capacity between the U.S. and Latin America
- Production SDN Infrastructure since 2014
- NAPs: Florida(2), Brazil(2), Chile, Puerto Rico, and Panama
- Carries Academic and Commercial traffic
- Control Plane: OpenFlow 1.0 and 1.3
- Inter-domain Provisioning with NSI
- A consortium involving FIU, NSF, RNP, ANSP, CLARA, REUNA, and AURA.
• Supporting the Large Synoptic Survey Telescope (LSST)’s requirements
  • The LSST will be installed in Chile
  • Every 27 seconds throughout the night, the telescope will take a 6.4GB picture of the sky, process it, generate transient alerts (6.3GB) from this picture, and send the 12.7GB data-set to Illinois/USA
  • From the 27-seconds window, only 5 seconds are available for data transmission
  • Multi traffic types with different priorities (db sync, control, general internet traffic)
Telemetry at AmLight: LSST Use Case [2]

• What if the LSST doesn’t manage to send its data in its 5-seconds transfer window?
  • For instance, because of packet loss, lack of capacity, lack of buffers, microburst, DoS attacks?

• If the data transfer window is missed, will AmLight engineering team be able to fix whatever it is happening before the next data transfer window (in less than 22 seconds)?

• How many windows are we going to miss if we have to troubleshoot it manually?

  ▪ *AmLight-INT Project might be the solution!*
AmLight-INT Project

• NSF IRNC: Backbone: AmLight In-band Network Telemetry (AmLight-INT), Award# OAC-1848746
  • Started in Nov 2018
• AmLight-INT Project Plan:
  • Deploy P4/INT-capable switches
  • Deploy INT Collectors (100G hosts) to collect metadata
  • Develop a new methodology to collect and export INT data in real time to feed SDN controllers and users with monitoring information
  • Create a Network Telemetry Design Pattern to be used by other R&E networks
Research challenges being addressed

• How many high-priority flows can be handled in real-time by the INT Collector?

• What is the impact caused by INT in a complex network such as AmLight-ExP (increasing MTU, extra delay)?

• How to dynamically enable INT monitoring of specific flows?

• What is the definition of real-time for AmLight and LSST?

• How to store and process multiple Gbps of telemetry data per switch?

• How to share the network state with the users?
AmLight-INT Project

• Collaboration between FIU and NoviFlow to expand AmLight SDN network towards an INT-capable domain

• Characteristics of the NoviFlow WB5132 switches @ AmLight:
  • Barefoot Tofino chip:
    • Provides a software-based SDN evolution path to P4-Runtime
    • 32 x 100G (high throughput: 3.2 Tbps)
    • NoviWare supports OpenFlow 1.3 (also 1.4 and 1.5) with BFD and LAG

• NoviFlow has already released five NOS versions to enable INT
  • P4/INT specification being followed
    • Nothing is proprietary or strictly created to support the LSST project
First Results (June)

- Wireshark Dissector created by NoviFlow (figure)

- AmLight-INT Collector v0.1:
  - Developed using Python 3.7
  - Receives Telemetry Reports from switches
  - Parses and sends to a RabbitMQ queue to be consumed
  - Saves Telemetry Reports to disk
First Results – Queue 0’s Jitter
First Results – Queue 0’s Occupancy

Network Telemetry @ AmLight // CI Engineering Brown Bag – Oct 18th, 2019
Last Results (October)

- AmLight-INT Collector’s QueueTop application consumes INT data and display realtime monitoring of the network’s queues

- Topology on the right created to enable experimentation
  - All links and devices are 100G
  - Novi03 switch port 04 has a bottleneck: Node 03 and Node 04 are sending data to their peers.
  - Let’s see what happens next...
One Source - One Destination - TCP - ~60Gbps

- Node 04 sending data using TCP to Node 02 at ~50Gbps
- No other traffic
- Top:
  - All Queues are using 114-115 cells (or 9K bytes)
- Bottom:
  - Hop Delay around 1 microsend (except for Novi03 that ADDs INT header)
Two Sources - Two Destinations - TCP - ~80Gbps

- Node 04 sending data using TCP to Node 02 at ~50Gbps
- Node 03 sending data using TCP to Node 01 at ~25Gbps
- Shared interface/queue on Novi03 port 4

- Top:
  - Now Novi03 uses 1026 cells

- Bottom:
  - Hop Delay at Novi03 around 9 microsend (add_int_metadata and queueing)
Two Sources - Two Destinations - TCP – 100% output utilization

• Node 04 trying to send as much data using TCP as possible to Node 02
• Node 03 trying to send as much data using TCP as possible to Node 01
• Shared interface/queue on Novi03 port 4
  • Top:
    • Now Novi03 uses 3306 cells (or 264KB)
  • Bottom:
    • Hop Delay at Nov03 around 28 microsends (add_int_metadata and queueing)

**Question:** What has happened to Novi01 and Novi04 queues??? Under investigation.
Next Steps

• Understanding the behavior seen so far:
  • With the current tools, we will test our theories.

• Improve INT Collector’s performance:
  • Currently, a 99Gbps flow with 9000 Bytes packets generates around 5 Gbps of telemetry.
  • Using Metronome P4 NICs at the INT Collectors

• Next tools:
  • Integration with InfluxDB and Elastic for network visualization/historical data.
  • All tools will be available as Open Source code through the AmLight Github account soon:
    • http://github.com/amlight.

• Presentation:
  • December 10th 4:10PM at the 2019 Internet2 Technology Exchange.
The Team

- FIU team:
  - Arturo Quintana – Sr. Software Developer
  - Julio Ibarra – PI
  - Jeronimo Bezerra – Senior Personal

- University of Passo Fundo, Brazil:
  - Use of P4 Metronome NICs at AmLight
  - Alisson Borges Zanetti – MSc. student
  - Pedro Eduardo Camera – MSc. student
  - Prof. Dr. Ricardo de Oliveira Schmidt (UPF)
  - Prof. Dr. Marco Antônio Sandini Trentin (UPF)

- Kytos E-Line integration
  - Kytos developers at State University of Sao Paulo/SPRACE and Academic Network of Sao Paulo/ANSP
    - Beraldo Leal, Antonio Francisco, Humberto Diógenes, Rogerio Motitsuki, and others
AmLight-INT: In-band Network Telemetry to support big data applications

Jeronimo Bezerra AmLight Chief Network Engineer
Arturo Quintana Sr. Software Developer