SCTE•ISBE CABLE-TEC EXPO® 2020
VIRTUAL EXPERIENCE » OCTOBER 12-15

2020 Fall Technical Forum
SCTE·ISBE • NCTA • CABLELABS®

Streaming Telemetry Data from the Home Network using OpenWrt Access CPE

A Technical Paper prepared for SCTE•ISBE by

Shlomo Ovadia, Ph.D.
Senior Principal Engineer
Charter Communications
14810 Grasslands Drive, Englewood, CO 80112
720-536-1686
Shlomo.ovadia@charter.com

Deependra Rawat
Sr. Software Developer
Charter Communications
14810 Grasslands Drive, Englewood, CO 80112
c-deependra.rawat@charter.com

Dan Lynch
Sr. Software Developer
Charter Communications
14810 Grasslands Drive, Englewood, CO 80112
c-daniel.lynch1@charter.com

Table of Contents

List of Figures

List of Tables

1. Introduction

In the last decade, home network architectures have become more complicated due to advances in wireless technology, and the explosion of different types of wirelessly connected devices such as Internet-of-Thing (IoT) devices, cell phones, tablets, laptops, gaming devices, etc. In fact, the U.S. smart home market is expected to show a compound annual growth rate of 16.9%, reaching 46.6B by 2024 [1]. In this home network architecture, some cable Multiple System Operators (MSOs) deploy a two-box solution for both residential and Small and Medium-size Business (SMB) customers where one box is an access CPE device (i.e., cable modem or an ONU), and the second box is a wireless router. Cable operators typically have limited information about the access CPE device's health status, and no information about the customer's home network, including what type of clients are connected to the home network, and their bandwidth usage vs. time. Such home network health and traffic information would be very useful to the Cable operators in order to enhance their customers' experience by optimizing the customer's home network traffic, prevent potential field issues, and be able to introduce new services such as customer's home network management and security. Furthermore, some Cable operators have already begun the transition towards cloud-based management of wireless routers and other devices, which is not supported by the currently field-deployed access CPE devices.

In this paper, the challenges with the current home network architecture are first explained. Then, the proposed home network architecture with an agile software stack on the access CPE device is discussed. This includes the agile software stack and components with the cloud-based management of the access CPE device to enable streaming of all the telemetry data. Third, the streaming telemetry data path, including the components and operation of the cable MSO's Streaming and Analytics platform are explained. Fourth, the organized hierarchical Grafana dashboard design with all of the different types of streamed data telemetry, including home network traffic metrics, D3.1 eMTA health metrics, DOCSIS RF info, event alarm and notifications is explained. A comparison between the OpenWrt-based streaming telemetry method in the paper and other streaming telemetry methods is then reviewed. Finally, the benefits to Cable operators using the OpenWrt-based streaming data telemetry method for access CPE devices are summarized.

2. Home Network Architecture

Cable MSOs' common customer home network is a two-box solution, consisting of a D3.1 eMTA device, which is connected to an HFC network via a coaxial cable on the WAN port, and connected to a Wi-Fi router on the LAN port via an Ethernet cable as shown in Figure 1 configuration A. For a Fiber-To-The-Home (FTTH) deployment, the access CPE device is an Optical Network Unit (ONU) connected to a Passive Optical Network (PON). A Wi-Fi router with a number of Wi-Fi APs or multiple Wi-Fi routers are typically used in larger homes for wireless mesh connectivity, resulting in improved data throughputs for all of the wirelessly connected devices in the home network. The software stack on the D3.1 eMTA device is monolithic, customized for each Silicon and OEM vendor, and does not provide any metrics about the health of the access CPE device and the wirelessly connected clients. The D3.1 eMTA monolithic software stack results in long and costly validation testing cycles before the new firmware that meets the Cable operator's requirements can be deployed. Furthermore, Cable operators are moving toward a cloud-based infrastructure for both command/control and streaming telemetry, which the current access CPE device's monolithic software architecture does not accommodate.

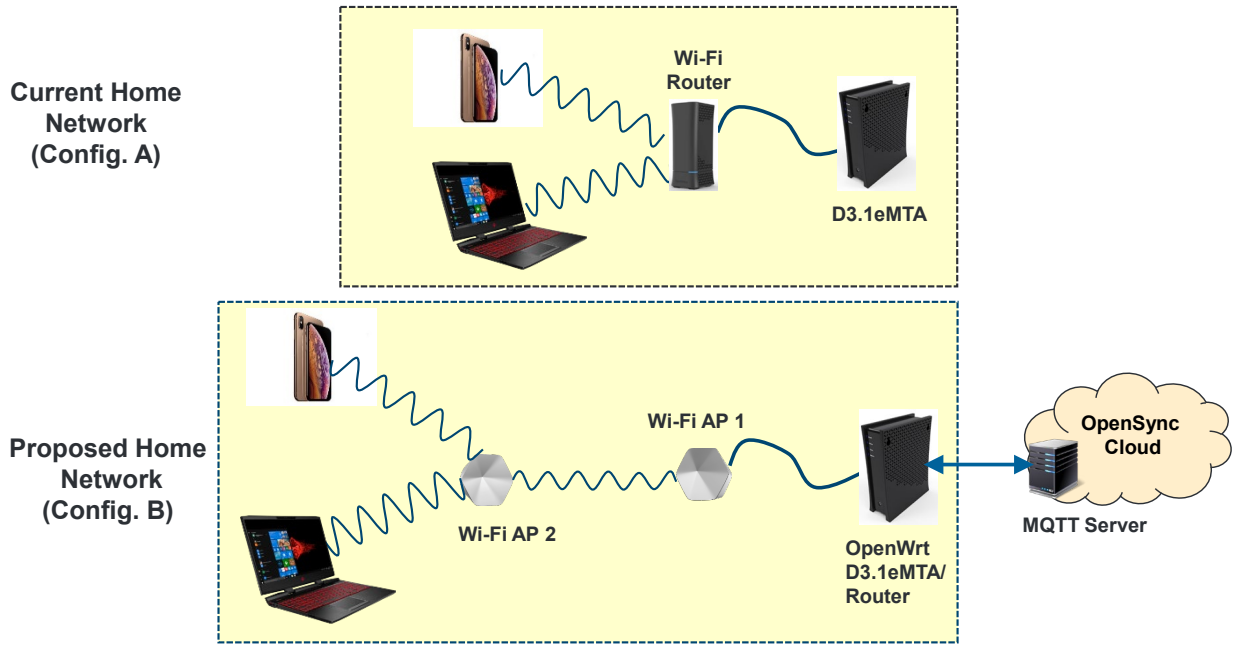


Figure 1: Current and Proposed Home Network Architecture Diagrams (Configurations A and B)

3. Access CPE OpenWrt Software Architecture

OpenWrt is an open-source project for embedded Operating System (OS) based on Linux. It was selected since it is highly-flexible open-source OS with a large ecosystem of vendors and developers that enable cable MSOs to rapidly develop new features and plugins that can also be containerized [2]. One of the key built-in benefits of OpenWrt OS is a full carrier-gradeIPv4/IPv6 routing functionality on the access CPE device with no need to redesign the hardware. Moving the routing functionality from the connected Wi-Fi router to the access CPE device enables cost optimization by using a generic Wi-Fi AP as the second box, and focusing the AP performance on the wireless connectivity in the home network.

To address the challenge explained above, an agile OpenWrt-based software stack was developed as a Proof of Concept (PoC) using D3.1 eMTA device operating in a home network architecture as shown in Figure 1 configuration B. The agile OpenWrt software stack is integrated with an OpenSync™ layer, a Silicon vendor Software Development Kit (SDK), and the Message Queue Telemetry Transport (MQTT) server architecture as shown in Figure 2. In addition, the CM and voice firmware was loaded on the access CPE device.

MQTT is a lightweight Machine to-Machine (M2M) transport communications protocol [3]. The D3.1 eMTA streams the telemetry data statistics to the Grafana dashboard via the MQTT server that is hosted in the OpenSync™ cloud, which forwards the data to the cable MSO's Streaming and Analytics platform. MQTT supports various authentications and data security mechanisms (using a script to generate security certificates). The Grafana tool was selected since it is a multi-platform open-source analytics and interactive visualization web application that users can customize to create complex monitoring dashboards [4].

The OpenSync™ cloud is composed of a Network Operations Center (NOC) and OpenSync™ controller for managing a network of OpenSync™-enabled devices. Cable operators can establish their own

OpenSync™ cloud by obtaining an OpenSync™ source-code license. The OpenSync™ cloud provides operator-friendly services, including:

- Device and firmware management
- Inventory and billing system
- Network performance control
- Onboarding and provisioning of field-deployed devices
- Telemetry reporting and data analytics
- Network operations, and customer support.

The NOC in the OpenSync™ cloud translates and communicates management commands in a single Wi-Fi AP network and in mesh multi-AP Wi-Fi network via Open vSwitch Database (OVSDB) distributed database commands. The OpenSync™ controller utilizes the OVS implementation and OpenFlow protocol for networking, and MQTT protocol for receiving state and telemetry data from OpenSync™-enabled devices. Specifically, the OpenSync™ controller provides the necessary command and control services such as:

- Network Status
- IP Address (displayed)
- Network Mask
- DHCP Status
- Parental Control
- Speed Test initiation and results
- Reset and Reboot device

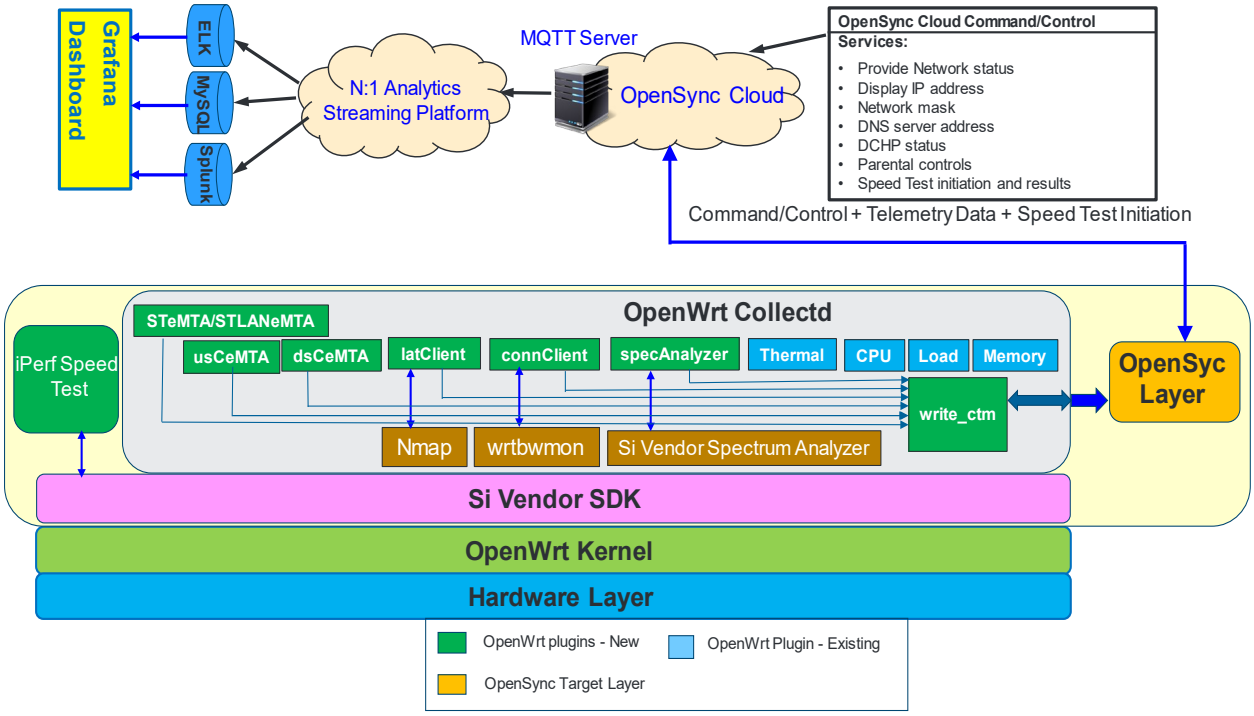


Figure 2: OpenWrt Integrated with OpenSync™ Software Architecture

In this software architecture a smart remote agent based on an OpenWrt data collector open-source software component called collectd was used as shown in Figure 2 [5]. The collectd component gathers metrics from various sources, e.g. the operating system, applications, log-files and external devices, and stores this information or makes it available over the network. Those statistics can be used to monitor systems, find performance bottlenecks (i.e. *performance analysis*) and predict future system load (i.e. *capacity planning*). The collectd component, which offers a variety of Plugins (software programs), is used to collect different types of telemetry data from a few Wi-Fi routers. The collectd component's default reporting time interval is 30 seconds, but other configurable time intervals can be selected. New capabilities and functionality (green-colored boxes in Figure 2) were added to the collectd software components. For example, the smart remote agent can be used to run a specific test such as measure the IPv4/IPv6 DOCSIS round-trip latency as explained below using the eMTALat plugin, read the collected measurement data, and streamed the data to the service operator's streaming and analytics platform. The blue-colored boxes are existing supported OpenWrt Collectd plugins that are utilized in this architecture. The collectd software components are integrated with the Silicon vendor's SDK and with the supported OpenWrt OS. It should be pointed out that other custom plugins and shell scripts can be developed, and the listed plugins below are just a sample framework.

Green-coded collectd plugins descriptions:

1. **usCeMTA:** Software plugin to pull all the DOCSIS upstream channel information used by the D3.1 eMTA (RF level, channel frequency, etc.).
2. **dsCeMTA:** Software plugin to pull all the DOCSIS downstream channel information used by the D3.1 eMTA (RF level, channel frequency, etc.).
3. **latClient:** Software plugin that measures and reports the round-trip latency from the D3.1 eMTA to each of the wirelessly connected devices in the home network based on their IP address or MAC address as shown in Figure 1.
4. **connClient:** Software plugin that measures and reports the number of transmitted and received packets from each of the wirelessly connected devices in the home network.
5. **specAnalyzer:** Software plugin to obtain the RF downstream and upstream spectrum of the Access CPE device.
6. **eMTALat:** Software plugin that measures and reports the minimum, maximum, and average round-trip DOCSIS latency between the D3.1 eMTA and the connected CMTS. First, it initiates a trace-route command to get CMTS IPv4 and IPv6 addresses. Then, it starts ICMP request and reply commands to measure the DOCSIS latency between CM and CMTS and stores the test results in separate files for IPv4 andIPv6. The eMTALat plugin reads results from these files and send them to write_ctm plugin, which in turn sends the measured DOCSIS latency results to the OpenSync™ layer's SM (not shown in Figure 1).

Wrtbwmon:

wrtbwmon is a small and basic shell script designed to run on Linux powered routers (OpenWRT, DD-WRT, Tomato, and other routers where shell access is available). It provides per user bandwidth monitoring capabilities and generates usage reports [6].

Nmap:

Network Mapper or Nmap is a free and open source utility for network discovery and security auditing [7]. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics (nmap.org).

write_ctm collectd Plugin:
The **collectd** service calls its collection plugins over a configurable time period. The collectd daemon collects various statistics from the device for aggregation and forwarding to a desired destination. The write_ctm plugin aggregates the stats, converts them to Protobuf format, and sends the collected telemetry data to the OpenSync™ STATS Manager (SM), which in turns forwards the collected telemetry data to the OpenSync™ Queue Manager (QM). The OpenSync™ QM sends the collected telemetry data to the cable MSO's Streaming and Analytics platform via the MQTT server. It should be pointed out that collectd plugin can also send string-formatted event notification based on system defined threshold levels. The write_ctm plugin registers its write API to collectd, and fetches data from the existing plugins on the expiration of every time period. No changes are needed to the existing plugins.

The blue-coded collectd plugins (Thermal, CPU, Load, Memory) are standard open-source collectd plugins that were added to the list of collectd plugins as shown in Figure 2.

4. Access CPE OpenSync™ Software Architecture

Figure 3 shows the key OpenSync™ software architecture components integrated with OpenWrt software, and the connectivity to the OpenSync™ cloud [8]. OpenSync™ is a cloud-agnostic open-source software that consists of many managers running as separate processes and performing their specific set of tasks. The software code of the Diagnostics Manager (DM) and STATS Manager (SM) was updated for streaming the collected telemetry data and other test results such as the iPerf speed test results via the OpenSync™ cloud. Table 1 summarizes the OpenSync™ manager functionality and their status. Some of the managers listed are required for basic operation, while the other listed managers are optional, depending on the desired functionality.

Table 1: Summary of OpenSync™ Managers' Functionality and Status

Manager Name	Manager Functionality	Manager Status
Diagnostics Manager (DM)	Responsible for spawning the rest of the OpenSync™ managers and optionally monitoring them. It controls starting, stopping, restarting of the OpenSync™ managers, and monitoring the reboot status of the OVSDB. The iPerf speed test software was developed and integrated into the DM such that the speed test can be initiated from the OpenSync™ NOC, and the DS/US speed test results are sent to Grafana dashboard.	Required for basic operation
Connection Manager (CM)	Responsible for establishing the backhaul connection and keeping connectivity to the cloud.	Required for basic operation
Network Manager (NM)	Responsible for managing all network related configuration and network status reporting.	Required for basic system network configuration

Wireless Manager	Not applicable to access CPE devices. Used in Wi-Fi routers to read and updated their configuration and state tables.	Required for basic system network configuration
Queue Manager (QM)	Responsible for aggregating reports from different OpenSync™ Managers	Optional
Statistics Manager (SM)	Responsible for processing all requested wired and wireless statistics and sending results to the cloud. The configuration is done through OVSDB while MQTT is used for the data plane. All the telemetry health metrics mentioned below are collected by write_ctm component as shown in Figure 2, and are transmitted to the SM, which forwards all the collected telemetry data to the QM as shown in Figure 3.	Optional
OpenFlow Manager (OM)	If the OpenVSwitch is used on the device, then the OM is responsible for managing packet flow rules.	Optional
Log Manager (LM)	Responsible for collecting and uploading logs and system information upon the Cloud request (log pull) and for handling log severity setting for running modules.	Optional
Platform Manager (PM)	Responsible for covering specific platform features which can't be covered by other managers such as synchronization between device GUI and cloud, and cloud-managed device parental control.	Optional

4.1. iPerf Speed Test

The iPerf speed test is initiated from the OpenSync™ Network Operations Center (NOC). Submitting a speed test request from the NOC sends a message via Openflow to the access CPE device, and the speed test request is detected on the device by the speed test handler in the OpenSync™ DM. The speed test handler calls a script on the device that in turn invokes an iPerf3 speed test with a pre-defined set of arguments. The speed test is run once to collect the upstream test results, and once again to collect the downstream results. The speed test results from each test are saved to files on the Access CPE device. The STeMTA collectd plugin processes the speed test results from the files and delivers them to the MQTT server, as described in the STeMTA Plugin section. The ability to initiate the iPerf speed test from the OpenSync™ NOC and review the collected speed test results would be very helpful to the Cable operators' call center to quickly address customers' issues.

4.1.1. STeMTA Plugin:

A new speed test plugin, which is called STeMTA, was added to collectd. The STeMTA plugin calls an iPerf speedtest script to initiate the iPerf speed test on WAN port of the D3.1 eMTA. Once an iPerf speed test is completed, then the script writes the downstream and upstream speed test results to iPerf download and upload result files. The STeMTA plugin reads results from these two files and sends them to the write_ctm plugin, which in turn sends the measured speed test results to the OpenSync™ SM.

4.1.2. *STLANeMTA Plugin:*

A new Speed Test plugin, which is called STLANeMTA, was added to collectd. The STLANeMTA plugin calls iPerf LAN speedtest script to initiate the iPerf speed-test on LAN port of the D3.1 eMTA. Once iPerf speed-test is completed, the script writes the downstream and upstream speed test results to an output file. The STLANeMTA plugin reads results from this file and sends to them to the write_ctm plugin, which in turn sends the measured speed test results to OpenSync™ SM.

4.1.3. *Cloud Security*

The software architecture also includes several cloud-connectivity security features. The D3.1 eMTA uses OpenSync™ device certificates to authenticate and connect to the OpenSync™ cloud, and to connect to the MQTT server in order to stream the collected telemetry data to the Cable operator's Streaming and Analytics platform. In addition, a shell script in the device is used to monitor all Secure Shell (SSH) and TELNET connections to the D3.1 eMTA, which are reported to the Grafana dashboard.

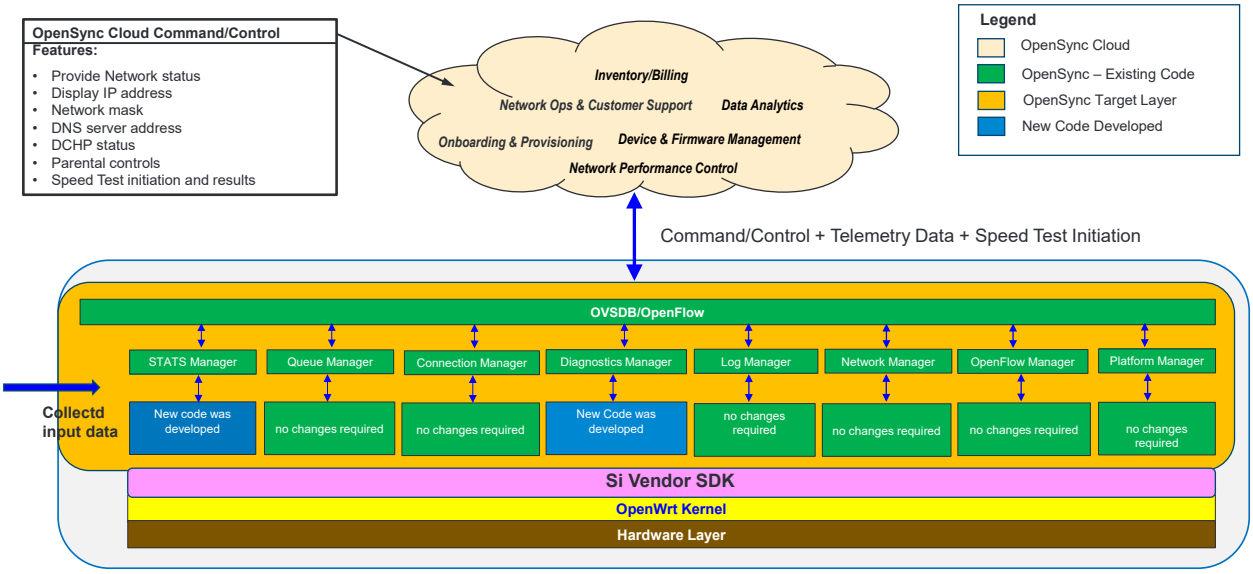


Figure 3: OpenSync™ Software Architecture with the Connectivity to the OpenSync™ Cloud

5. Streaming Telemetry Data Path

Figure 4 shows the telemetry data path (current implementation) from the OpenWrt D3.1 eMTA to the Grafana dashboard via the cable MSO's Streaming and Analytics platform. The collected telemetry data is streamed in Protobuf format to the MQTT server hosted on the OpenSync™ cloud, and then to the cable MSO's Streaming and Analytics platform.

Cable MSO's future network architecture separates the control plane and the data plane as intelligence is no longer resident on hardware devices but rather on the network's software driven controllers where network analytic models can act on traffic behaviors, services flows, and configuration state to predict and

respond in near real-time to the networks changing demands. The network architecture's data plane includes the Cog platform, Data Distribution Bus (DDB), and Data Governance platform. The Cog platform is a data engineering platform that builds enriched data sets called Analytics Data Sets (ADSs), which is represented by the First Normal Form (1NF). These ADSs are distributed across the operator's network, and are used for data modeling and Machine Learning (ML). The DDB, which is shown in Figure 4, is the initial point of data ingestion driven by Apache Kafka [9]. The DDB is also the initial system, where all data is classified as a data asset. The Data Governance platform provides the framework for decisions and accountabilities within the corporate structures to manage and protect the data assets [10]. Any raw data that is not governed as dictated by the data governance standards is transformed for compliance with the standards. As the Cable operators are moving their network's control plane to the cloud, their data plane is maintained in various edge locations deeper into the network, or in this case, the customer's home.

The D3.1 eMTA telemetry data in Protobuf format is ingested by the Kafka Connector source for MQTT, which is part of the data plane of the cable MSO's Streaming and Analytics platform as shown in Figure 4. The Kafka Brokers received the converted telemetry data from Protobuf to Apache Avro™ format, and transmit the telemetry data to the Kafka Connector Sinks [11]. The telemetry data is ingested by different data analytic tools, depending on the type of data. For example, Elasticsearch (ELK) tool ingests time-series data, while MySQL tool ingests relational data before the telemetry data is displayed on the customized Grafana dashboard.

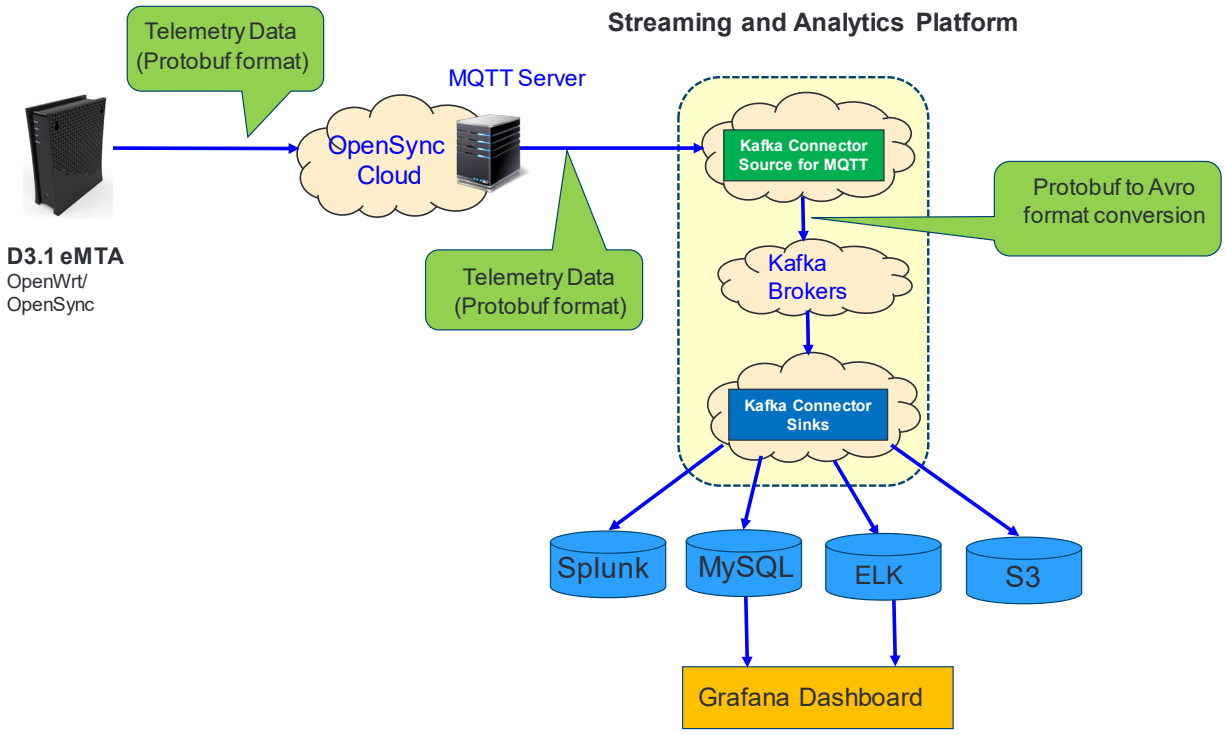


Figure 4: Telemetry Data Path from the Access CPE Device to the Grafana Dashboard

6. Grafana Dashboard Design

The design goal for the customized Grafana dashboard is to concisely present all the different types of telemetry data to the Cable operator's care agent, and to focus the agent attention to any reported failures and unhealthy device metrics. Figure 5 shows, for example, the organized hierarchical color-coded Grafana dashboard with the key components, level 1 health metrics, and their operational status based on pre-determined threshold levels. In addition, the geographical map of the customer location is shown along with reported data telemetry results. The organized Grafana dashboard includes the following information:

A. D3.1 eMTA Router system information such as:
   - CPU utilization (%) in a given time period
   - Free system memory (%) in a given time period
   - System load (%) in a given time period
   - Networking information such as IP address, network mask, DHCP status, etc.
   - Instantaneous and average system temperature in a given time period
   - Average, minimum, and maximum round-trip IPv4 and IPv6 latency in a selected period of time to the CMTS

B. Home network traffic from all the wirelessly connected devices via the Pods or Access Points:
   - IPv4/IPv6 of the wirelessly connected client in home network
   - Number of transmitted and received packets for each device
   - IPv4/IPv6 round-trip latency between the D3.1 eMTA and each of the connected clients

C. Cable modem Downstream/Upstream channel information, including:
   - Downstream channel information (i.e., channel ID, channel type, lock status, channel bonding status, received power level, SNR/MER, channel center frequency, channel width, modulation profile, etc.) – see Figure 9.
   - Upstream channel information (i.e., channel ID, Transmit power level, channel center frequency, channel width, channel bonding status, etc.)

D. RF downstream and upstream spectrum information – downstream/upstream RF signal power (dBmV) vs. frequency (MHz).

E. Downstream/Upstream speed test results on the WAN port (i.e., iPerf server in the cable MSO's cloud) and the LAN port (i.e., between iPerf server running on D3.1 eMTA and the connected home network's client).

F. Security notifications and alarms information, including:
   - Security notifications: for example, if someone is trying to temper with the Access CPE via unauthorized access to the device's management and control GUI via SSH. In this case, the color-coded green status of the security notifications and alarms dashboard component would change to either a color-coded orange or red, indicating an increased security risk.
   - Collected metrics alarms where one or more red thresholds were violated. For example, if the temperature of the device is significantly elevated, and the device is about shut-down or go into energy saving mode.

G. Customer location map, providing the cable MSO's care agent information where the customer is located within the cable MSO's service area footprint.

H. Voice health metrics, including:
   - Phone line number
   - phone status (on/off hook)
   - phone line IPv4 and IPv6 addresses

- Voice call start time, end time, call duration, call failed
- phone line registration status

I. External Battery Backup Unit (EBBU) metrics, including:
- Manufacturer identity, HW model number, software agent version, battery status
- EBBU output voltage, and estimated remaining charge capacity,
- Alarm description (On battery, Low battery, Depleted battery, EBBU shutdown in pending, EBBU shutdown is eminent)

J. Access CPE device information, including:
- Cable modem (CM) HW version, SW version, MAC address, serial number
- IPv4 address and IPv6 address
- CM system uptime
- CM security status/type
- CM connectivity state status/type

K. Access CPE device reboot information, including:
- Last device reboot event time, date, and count
- Last device reboot description

L. IPv4/IPv6 DOCSIS round-trip latency, including:
- IPv4 and IPv6 minimum, average, and maximum round-trip latency to the connected CMTS in a given period of time

M. Device event logs, including any device configuration events, DHCP warnings, DOCSIS events, etc.

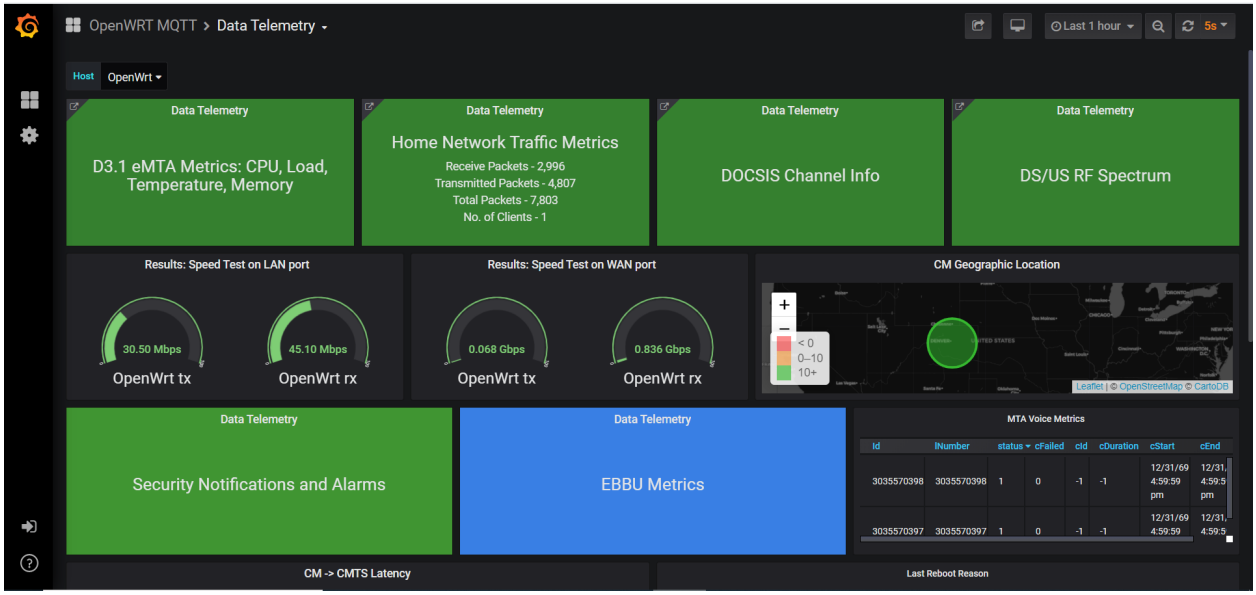


Figure 5:Hierarchical Color-Coded Grafana Dashboard with the Key Telemetry Components

Level 2 telemetry data shows the status of each of the reported health metrics in a given period of time. Figure 6, for example, shows the reported status of the D3.1 system parameter, including system load (%), CPU utilization (%), system temperature (°C), and free memory in the last hour. Each of these reported metrics has a different set of threshold levels to indicate its healthy status. For example, a healthy CPU utilization is below 75%, and it is color-coded green, while unhealthy CPU utilization is above 85%, and it

is color-coded red. CPU utilization between 75% up to 85% is color-coded orange. If all the level 2 D3.1 system parameters are healthy, then the D3.1 eMTA system component (level 1) turns green.

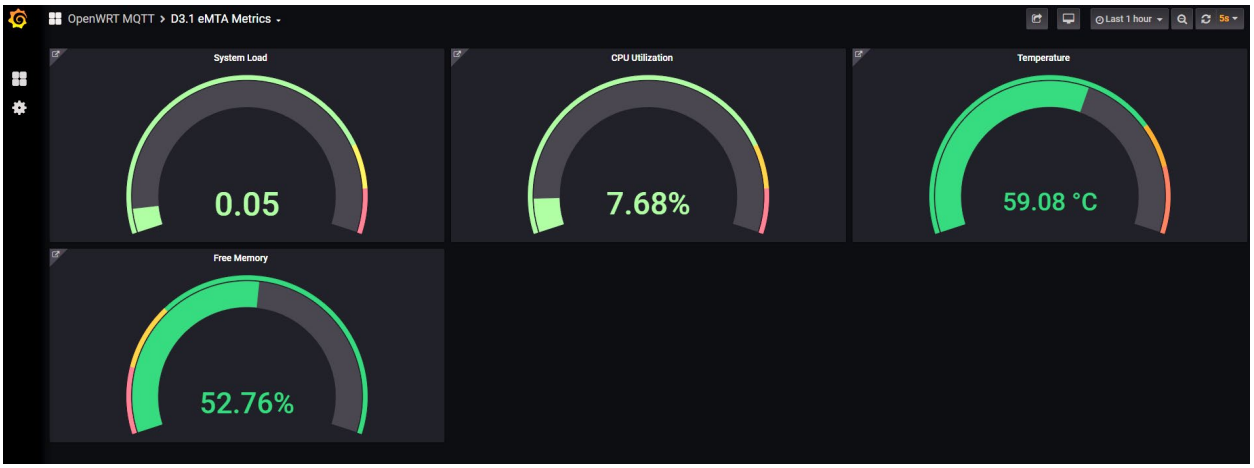


Figure 6: Reported Status Level 2 Access CPE Metrics in the Last Hour

Level 3 telemetry data shows the time behavior of each of the selected metrics in any given time frame. This type of telemetry reporting can be important when deeper insight into the reported metrics is needed to diagnose an issue or abnormal behavior of the D3.1 eMTA device. Figure 6 shows, for example, the reported level 3 telemetry data the time behavior of the CPU utilization reported in the last hour. The CPU utilization data is collected based on a selected time interval, which is 5 seconds in this example. The CPU utilization in this example is low, and varied between about 2.5% to 22.5%. Note that the average system load and CPU utilization are two different things. The system load is a measurement of how many tasks are waiting in a kernel run queue (not just CPU time but also disk activity) over the selected time period. The CPU utilization is a measure of how busy the CPU is during the selected time period.

This type of telemetry data is particularly useful since ML models can be executed on the selected customer data based on the collected historical data to determine if the current reported issue previously occurred, when it occurred, and provide suggested guidelines to the cable MSO's care agent how to mitigate this issue, particularly if this issue previously observed with other customers.

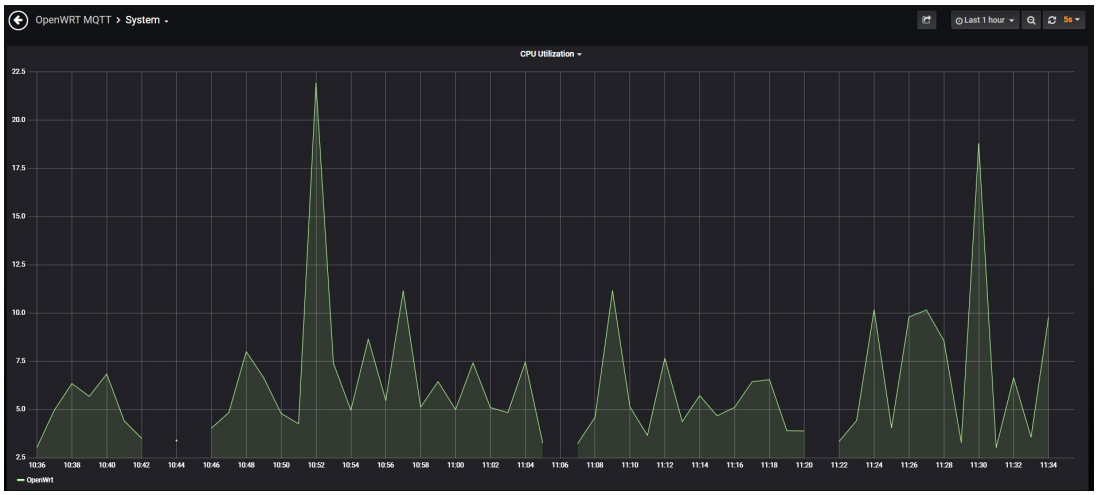


Figure 7: CPU Utilization of Access CPE vs. Time (Level 3) Reported in the Last Hour

One of the challenges for the operator's care agent is to gain visibility into the customer's home network (Figure 1) for network optimization and debugging field issues. This challenge is met by having the access CPE device function as a router, and connected to a Wi-Fi AP via Ethernet cable. Figure 8 shows, for example, the home network traffic parameters (level 2), including the number of transmitted and received packets by each wirelessly connected client in the home network based on their IP address or MAC address reported in the last hour. Instead of using the connected client's IP address or MAC address, the actual client's identification can be displayed on the Grafana dashboard with the integration of a device fingerprinting agent such as a Cujo Artificial Intelligence (AI) agent [12]. The client identification includes device name, device vendor, device model number, device type, device OS, etc. This can be a very useful feature for customers trying to diagnose their home network traffic. For example, customers can identify if there is a specific client that consumes most of the bandwidth in the home network, and/or make changes to their home network configuration.

In addition, the average round-trip latency between the D3.1 eMTA and each of the wirelessly connected client in the last hour is reported. Level 3 home network traffic data can be obtained by selecting a specific client in the home network based on their IP address. For example, the number of transmitted and/or received packets vs. time by the selected client over the selected time interval can be displayed on the Grafana dashboard.

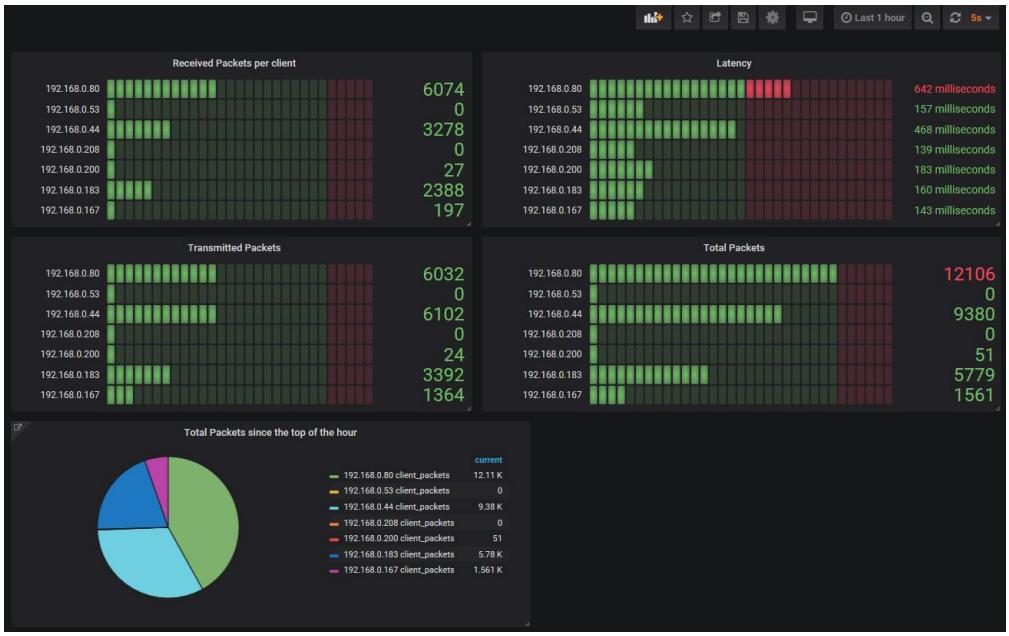
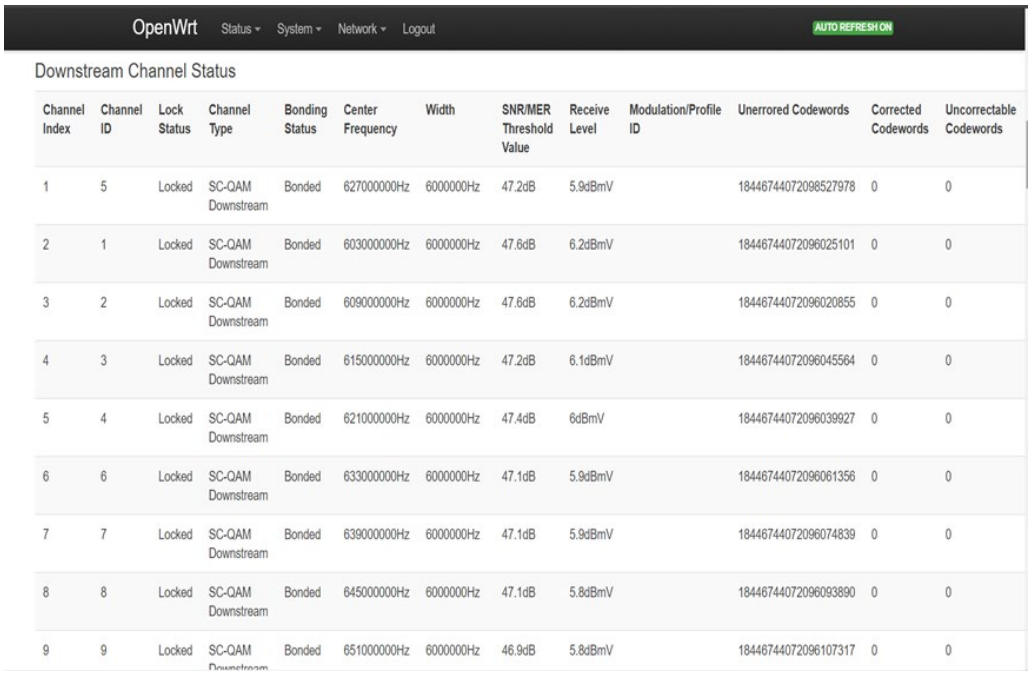


Figure 8: Home Network Traffic Parameters for All the Wirelessly Connected Client in the Home Network

Monitoring the DOCSIS channel information is also important to ensure robust operation of the CM at the customer location. Figure 9 shows, for example, the downstream DOCSIS channel information status, including the channel lock status, channel type, channel bonding status, channel center frequency, channel width, SNR threshold, and received power level, unerrored codewords, number of corrected codewords and uncorrectable codewords. Having access to such monitored historical data with ML models can significantly help to identify troubled CMs in the field compared with other CMs connected to the same CMTS.



Channel Index	Channel ID	Lock Status	Channel Type	Bonding Status	Center Frequency	Width	SNR/MER Threshold Value	Receive Level	Modulation/Profile ID	Unerrored Codewords	Corrected Codewords	Uncorrectable Codewords
1	5	Locked	SC-QAM Downstream	Bonded	627000000Hz	6000000Hz	47.2dB	5.9dBmV		18446744072098527978	0	0
2	1	Locked	SC-QAM Downstream	Bonded	603000000Hz	6000000Hz	47.6dB	6.2dBmV		18446744072096025101	0	0
3	2	Locked	SC-QAM Downstream	Bonded	609000000Hz	6000000Hz	47.6dB	6.2dBmV		18446744072096020855	0	0
4	3	Locked	SC-QAM Downstream	Bonded	615000000Hz	6000000Hz	47.2dB	6.1dBmV		18446744072096045564	0	0
5	4	Locked	SC-QAM Downstream	Bonded	621000000Hz	6000000Hz	47.4dB	6dBmV		18446744072096039927	0	0
6	6	Locked	SC-QAM Downstream	Bonded	633000000Hz	6000000Hz	47.1dB	5.9dBmV		18446744072096061356	0	0
7	7	Locked	SC-QAM Downstream	Bonded	639000000Hz	6000000Hz	47.1dB	5.9dBmV		18446744072096074839	0	0
8	8	Locked	SC-QAM Downstream	Bonded	645000000Hz	6000000Hz	47.1dB	5.8dBmV		18446744072096093890	0	0
9	9	Locked	SC-QAM Downstream	Bonded	651000000Hz	6000000Hz	46.9dB	5.8dBmV		18446744072096107317	0	0

Figure 9: Downstream DOCSIS Channel Information Status

7. Comparison with Other Streaming Telemetry Methods

There are other streaming telemetry data methods for access CPE devices. One of the streaming methods is based on the Internet Protocol Detail Record Streaming Protocol (IPDR/SP). The Cable Modem Termination System (CMTS) is using the IPDR/SP via CableLabs-defined schemas to collect customer data usage via billing records from specific service flows used by the CMs and export them to IPDR Collectors [13]. IPDR/SP utilizes the concept of templates in order to eliminate the transmission of redundant information such as field identifiers and typing information on a per data record basis.

Specifically, IPDR/SP Subscriber Account Management Interface Specification (SAMIS) Type 1 schema is probably the most common IPDR schema in use by the cable MSOs. SAMIS Type 1 uniquely identifies the specific CM attributes and service flow's attributes serviced by the CM. Currently, these billing records are collected every 15 minutes by the CMTS, which forward the collected records to the IPDR Collectors hosted in the regional data centers before the data is ingested by the cable MSO's Streaming and Analytics platform. Expanding the CMTS usage of the IPDR/SP to collect all of the various performance and health metrics, alarms, and notifications from each field-deployed CM would overburden each CMTS with huge amounts of data. To estimate the magnitude of this issue, it is assumed that each CM streams ≈ 62.5MB telemetry data every 24 hours based on the current implementation, and each CMTS is connected to ≈ 20k CMs. Consequently, each CMTS would receive about 1.25TB of telemetry data every 24 hours. Thus, this approach is burdensome since the CMTS do not process or make decisions on the enormous amount of collected telemetry data. Furthermore, new IPDR schemas for non-DOCSIS parameters such as voice metrics would need to be defined, standardized, and integrated with the access CPE firmware.

Model-Driven Telemetry (MDT) is another modern method for continuously streaming operational data from network devices such as the access CPE using a push model. Applications need to subscribe to a set of the access CPE device's Yet Another Next Generation (YANG) data models over standard protocols,

and push the collected telemetry data from the device when a change has occurred. MDT is not new to the cable industry as several Cable operators have already implemented and deployed MDT data collection and monitoring systems in their network [14]. Implementing MDT on access CPE devices requires the development a new software layer and components for YANG data models. In addition, a common set of standard Application Programming Interfaces (APIs) need to be defined for integration with an OpenWrt-based access CPE software stack. Furthermore, no detailed telemetry comparison analysis of performance vs. cost has been done to justify such an MDT-based development effort by the Cable operators.

In contrast, the OpenWrt and OpenSync™-based streaming telemetry method uses an agile lightweight and efficient smart agent based on an open-source code with customized plugins using the Silicon vendor's APIs. The proposed approach is to segregate the telemetry data such that all customer's billing records continue to be provided using the IPDR/SP, while all of the other access CPE device's performance and health metrics are directly transmitted to the cable MSO's Streaming and Analytics platform via the OpenSync™ cloud. Consequently, the CMTSs are not overburdened with huge amounts telemetry data.

8. Conclusion

In this paper, an agile OpenWrt software stack integrated with OpenSync™ layer and Silicon vendor SDK with carrier-grade IPv4/IPv6 routing functionality was developed on a common existing access CPE hardware. The connectivity of this access CPE device to the OpenSync™ cloud provides a standardized command and control method for networking services as well as operator-friendly services such as onboarding and provisioning on field-deployed devices, device firmware management, network operations and customer support, billing and inventory support. A smart remote agent was developed and integrated with the OpenWrt software stack that enables the access CPE device to stream various types of telemetry data to the cable MSO's Streaming and Analytics platform via the MQTT server hosted on the OpenSync™ cloud for analysis, and displays the collected data on a hierarchical color-coded Grafana dashboard. The streaming telemetry data consists of a wide variety of information, including:

- Access CPE system information
- CM device information
- Home network traffic information from all the wirelessly connected clients
- DS/US DOCSIS channel information
- DS/US RF spectrum output
- Event and alarms information for the collected metrics
- Speed test results on both the WAN and LAN ports
- Voice metrics information
- EBBU status information

Comparison with other streaming telemetry methods such as IPDR/SP and MDT reveal various challenges with the implementation of these methods. For example, expanding the IPDR/SP usage by the CMTS to collect all of the various performance and health metrics, alarms, and notifications would overburden each CMTS with huge amounts of data that it does not process or make decisions on is not an attractive approach. Implementing MDT on access CPE devices requires the development a new software layer and components for YANG data models without clear benefits.

Finally, as shown in this paper, the adoption of an OpenWrt-based streaming telemetry offers clear benefits to Cable operators. First, it enables cloud-based management of the access CPE devices and direct streaming of the telemetry data to the cable MSO's Streaming and Analytics platform via the OpenSync™ cloud without overburdening the CMTS. Second, the availability of the telemetry data to the operator's

care agent is likely to enhance customer satisfaction by reducing the number of truck rolls as field issues are resolved more quickly. Third, collaborations among different Cable operators will enable the industry to standardize the OpenWrt-based software architecture with the streaming telemetry across different types of CPE hardware platforms. Furthermore, the standardized agile software stack is expected to accelerate the development and deployment of new revenue-generating services.

Acknowledgment

The authors would like to acknowledge technical support from Jay Liew and Philip Anderson and management support from Ahmad Ansari and Matt Petersen at Charter Communications.

Abbreviations

Table 2: Abbreviations Table

Acronym	Stand For
AP	Access Point
API	Application Programming Interface
CM	Cable Modem
CPE	Customer Premise Equipment
CPU	Central Processing Unit
CMTS	Cable Modem Termination System
DDB	Data Distribution Bus
DOCSIS	Data over Cable System Interface Specification
EBBU	External Battery Backup Unit
eMTA	Embedded Media Terminal Adapter
FTTH	Fiber To The Home
ICMP	Internet Control Message Protocol
IPDR	Internet Protocol Detail Record
LAN	Local Area Network
MDT	Model Driven Telemetry
MER	Modulation Error Ratio
ML	Machine Learning
MQTT	Message Queue Telemetry Transport
MSO	Multiple System Operator
NOC	Network Operations Center
OEM	Original Equipment Manufacturer
ONU	Optical Network Unit
OS	Operating System
OVSDB	Open vSwitch Data Base
QM	Queue Manager
RF	Radio Frequency
SAMIS	Subscriber Account Management Interface Specification
SDK	Software Development Kit
SM	STATS Manager
SNR	Signal to Noise Ratio
WAN	Wide Area Network

YANG	Yet Another Next Generation

Bibliography & References

[1] U.S. Smart Homes, Statista. https://www.statista.com/outlook/279/109/smart-home/united-states

[2] https://openwrt.org/

[3] ISO/IEC 20922:2016, Information Technology – Message Queuing Telemetry Transport (MQTT) v3.1.1, https://www.iso.org/standard/69466.html.

[4] https://grafana.com/docs/grafana/latest/features/dashboard/dashboards/

[5] https://collectd.org/

[6] https://openwrt.org/docs/guide-user/services/network_monitoring/wrtbwmon

[7] https://www.opensync.io/

[8] https://svn.nmap.org/nmap/COPYING

[9] https://kafka.apache.org/

[10] K.Wende, A Model for Data Governance – Organising Accountabilities for Data Quality Management, Association for Information Systems(2007). https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1079&context=acis2007

[11] https://avro.apache.org/docs/current/

[12] https://cujo.com/agent/

[13] Data-Over-Cable Service Interface Specifications (DOCSIS) 3.1, CCAP Operations Support System Interface Specification, CM-SP-CCAP-OSSIv3.1-I16-190917.

[14] P. Sowinski, A. Smith, and T. Liu, Remote PHY 2.0, the Next Steps for Remote PHY Technology, SCTE Technical Papers (2019).

