Upstream Triggered Spectrum Capture

Lessons Learned from Deployment at Scale

A technical paper prepared for presentation at SCTE TechExpo24

Jonathan Leech
Principal Engineer II
Comcast
jonathan_leech@cable.comcast.com

Joseph Mitchell
Director 2, Network Maintenance & Engineering
Comcast
joseph_mitchell@cable.comcast.com

Doug Sitkin
Principal Engineer
Comcast
doug_sitkin@cable.comcast.com

Table of Contents

List of Figures

1. Introduction

As Comcast undergoes its upgrade journey from sub-split and integrated cable modem termination system (iCMTS) to mid-split remote physical layer (R-PHY) and beyond, the network management tools have undergone a parallel journey to support the new technology. This paper describes in detail the evolution of Comcast's upstream triggered spectrum capture (UTSC) tool known internally as "Yeti". The journey starts from Yeti's humble beginnings as an alternative to hardware-based swept spectrum monitors and ends at fully supporting data over cable service interface specification (DOCSIS®) 4.0 networks. The details include topics such as the fast Fourier transform (FFT), strategies for signal/noise classification, forward error correction (FEC) rates, modulation error ratio (MER), heatmap displays, and much more. The paper provides perspective from both the software development team and the field engineers. The paper explores possibilities for future work, including the areas of pattern matching using artificial intelligence (AI) and machine learning (ML), and correlation to upstream data analyzer (UDA) events. The insights and recommendations the paper makes allow other operators to benefit from Comcast's experience with UTSC.

The paper is organized according to the Yeti development timeline order and the lessons learned during each milestone are highlighted.

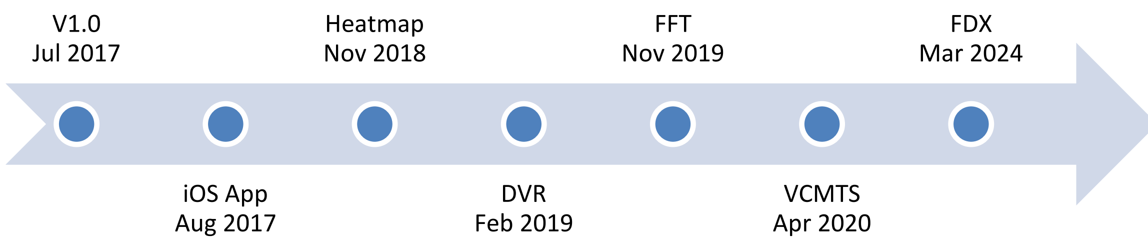

Figure 1- Yeti Development Timeline

2. Yeti Version 1.0

Before Yeti, there had been various attempts made within Comcast to create a UTSC tool, but Yeti was the first successful effort. Prior to Yeti's widespread adoption, the spectrum capture tools in use at Comcast were predominately hardware-based swept spectrum monitors, deployed in every headend. The upside of UTSC is not needing a separate platform for spectrum monitoring, and the associated cost savings in hardware, licensing, and support. Additional facility savings are found in rack space, power, cooling and physical wiring.

The first version of Yeti was released for production in July 2017. Yeti improved reliability and performance as compared to its predecessor. Yeti was deployed in a private cloud in four regional datacenters for geographic redundancy and to minimize round-trip network latency. The cable modem termination system's (CMTS's) upstream burst receiver has limited concurrency, so session management was required to accommodate many possible users. Yeti configures watches (sessions) and captures spectrum data using simple network management protocol (SNMP). As each spectrum capture is comprised of multiple steps of SNMP operations, it's important to minimize network latency to reduce the round-trip time, as it dictates the capture rate. Furthermore, since SNMP is based on user datagram protocol (UDP), minimizing the number of network hops also reduces SNMP failures. While this doesn't necessitate the extreme of deploying polling software in each headend, deploying polling software to regional data centers is an acceptable compromise. Yeti was more recently migrated from the private cloud to a public cloud provider's virtual private cloud, again leveraging regional data centers.

Lesson learned #1: Deploy to regional data centers to minimize network latency, maximize capture rate, reduce errors, and provide geographic redundancy.

Lesson learned #2: Reliability and performance are of critical importance for an application that supports field engineers and technicians.

2.1. Swept Spectrum vs FFT

Swept spectrum analyzers sweep across a frequency range, tuning to each frequency within the range and capturing power within the resolution bandwidth (RBW). Each frequency is sampled for a configurable duration referred to as dwell time. Successful detection of noise depends on the duration and periodicity of the noise as well as the RBW and dwell time. With proper configuration, swept spectrum analyzers are effective at detecting many common types of noise and interference [1].

A real-time FFT spectrum analyzer captures a continuous stream of time domain data and performs overlapping FFTs to ensure that windowing doesn't result in data loss, causing events to be missed. However, UTSC is not real-time but rather is sampled. A spectrum capture taken at 100 kilohertz (kHz) represents a capture duration of 1/100,000 of a second, or 10 microseconds. A hypothetical UTSC system that samples 10 captures per second (CPS) at 100 kHz would be only 10/100,000 of real-time or 0.01 percent. To be real-time, the same hypothetical UTSC system would need to be 10,000 times faster, or 100,000 CPS. The probability that a single-spectrum capture contains a particular noise event depends on the capture duration, CPS, as well as the duration and periodicity of the noise event. In practice, this means it may take several thousand captures over several minutes to catch some types of intermittent noise. Maximizing the capture rate reduces this amount of time.

Utilizing typical divide and conquer techniques to find upstream noise in the field, any increase in response time has a detrimental effect on speed to repair, encouraging the use of less productive local spectrum capture techniques. Maximizing the capture rate also reduces or removes the need to impact active services during troubleshooting.

Lesson learned #3: Maximize the spectrum capture rate to reduce troubleshooting time and customer impact.

2.2. MER and FEC

MER is the ratio of average symbol power to the magnitude of the error. MER is a similar metric to signal-to-noise ratio (SNR) and carrier-to-interference plus noise ratio (CINR). As compared to SNR and CINR, MER is more useful for troubleshooting sources of symbol error due to phase errors, micro-reflections, excessive group delay, and other impairments that interfere with the demodulation process.

While MER is computed at the symbol level, forward error correction (FEC) happens at the codeword level (multiple symbols). If error correction can overcome all the symbol errors within the codeword, the result is a correctable codeword. If unable, the result is an uncorrectable codeword and eventual packet loss.

A profile management application (PMA) adjusts profiles by lowering the modulation order, increasing the amount of error correction, or changing the codeword length to maximize the throughput of the channel while maintaining an acceptable error rate. PMA can successfully mitigate many issues, so it is important to look at not only MER and FEC rates but also the profile(s) in use for the channel [2].

Since sampled spectrum can miss events, but MER and FEC do not, updating MER and FEC at as high a rate as possible provides significant value. The update rate varies by CMTS vendor from 5 seconds to 30 seconds. On an interface with low utilization, it can take longer for the MER and FEC to reflect changes to the plant conditions. The Yeti user interface (UI) includes MER and FEC charts over time, as well as color-codes the data with a red/yellow/green scheme based on thresholds.

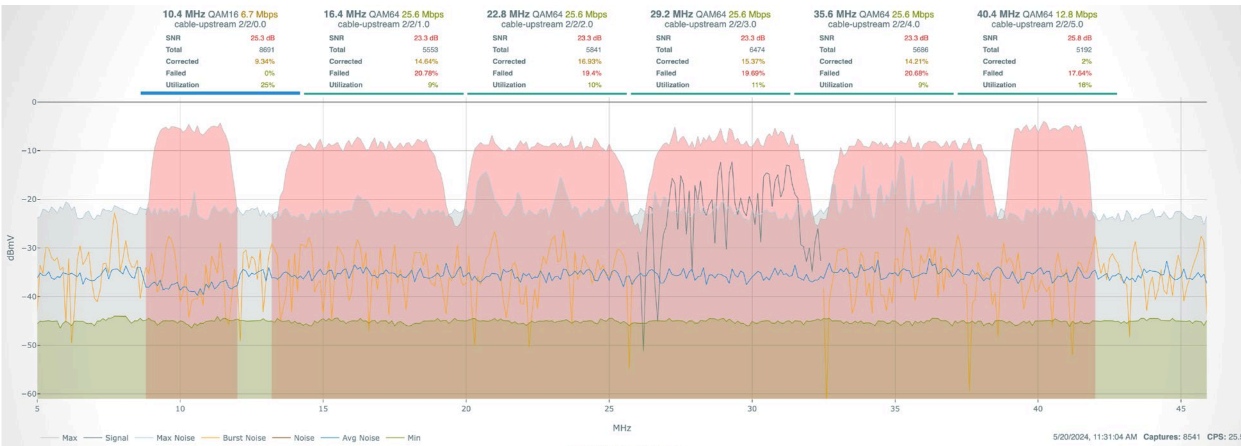

Figure 2 - MER and FEC Coloring, CPD Impairment

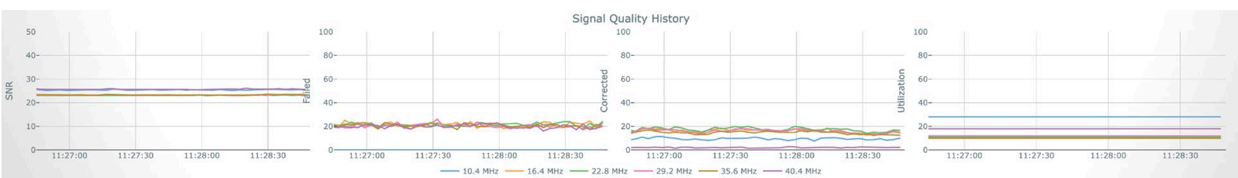

Figure 3 – MER and FEC Charts

For a group of DOCSIS 3.0 single channel quadrature amplitude modulation (SC-QAM) channels on an upstream interface, the presence and type of noise or noise-like impairments such as common path distortion (CPD), common mode disturbance (CMD), or ingress from home phoneline networking alliance (HPNA) adapters can be inferred based on MER and FEC history. Later analysis of enough spectrum captures can confirm the diagnosis. This is done in Yeti via the digital video recorder (DVR) function and external triggers from the upstream performance system. An example of CPD is shown above in Figure 2. Examples of CMD and HPNA impairments are shown below in Figure 4 and Figure 5. CPD and HPNA ingress are described in SCTE-280 [1]. CMD is described in SCTE/ANSI 249 [3].

As measurements of MER and FEC in the upstream are utilization-dependent, combining with spectrum capture provides the best possibility of immediate visualization.

Figure 4 – MER and FEC Coloring, CMD Impairment

Figure 5 – MER and FEC Coloring, HPNA Impairment

Lesson learned #4: Augment spectrum captures with MER and FEC captured at a high rate.

3. iOS Application

The next major milestone was the development of the iPhone operating system (iOS) application in August 2017. Having a UI optimized for the phones and tablets that field engineers use encouraged the adoption of Yeti. Despite the smaller form factor of phones, the Yeti iOS app delivers all the functionality and performance of the web version.

Development challenges included creating the heatmap visualization, as the charting library used does not include one, and re-factoring the UI to fit in the smaller form factor. We built our own heatmap visualization using two-dimensional image application programming interfaces (APIs) and used the iPhone orientation, touch gestures, tabs, overlays, and multiple screens to display the data.

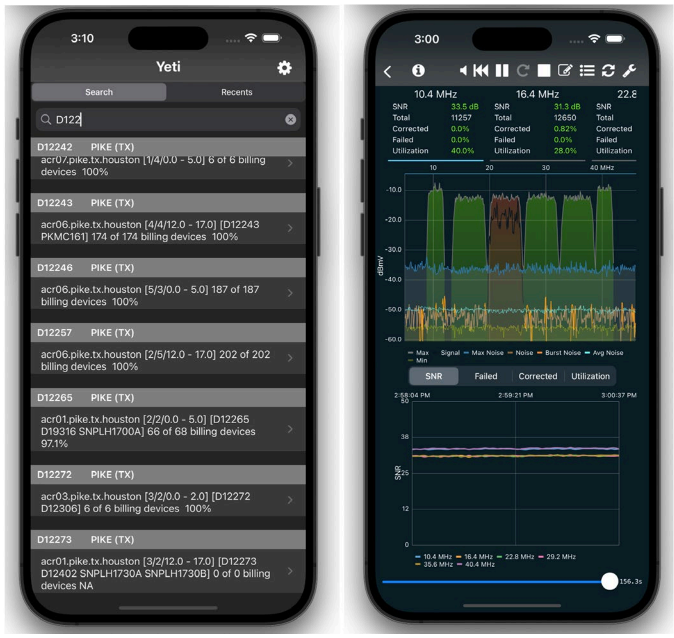

Figure 6 – iOS Application

Lesson learned #5: Make client accessible to mobile users.

4. Heatmaps and Signal / Noise Classification

In November 2018, two significant innovations were released in Yeti: heatmaps and algorithmic signal/noise classification.

4.1. Heatmaps

A heatmap, or two-dimensional histogram uses color to convey information. The value of each cell represents the number of times a spectrum capture passed through the cell, and the values are transformed to a specific color for display via a color scheme. The data for the heatmap is represented as a two-dimensional array of floating-point values. The horizontal axis represents frequency, and the vertical axis represents magnitude in decibels per micro-volt (dBmV). The resolution chosen for the heatmap values is 100 kHz by 1 dBmV. Signal values are represented via positive numbers, and noise values by negative numbers in the histogram. The values are exponentially decayed over time using a user-configurable half-life. The decay creates a bias for more recent values and creates a sense of time in the display. Because the heatmap shows every spectrum capture made, it is often a better choice for visualizing noise.

Heatmaps are not only a useful visualization technique but are also useful in the backend for data analytics, and in data transfer from the backend to the UI due to the more efficient representation as compared to the raw data. This is due to the heatmap's resolution being less than that of the raw data and not preserving the raw data's order. This enables higher spectrum capture rates and a lower, fixed data rate to the UI.

4.1.1. Color Schemes

The choice of a color scheme, also known as a color map, is important for a heatmap display to accurately convey information. A poorly designed color scheme can lead a user to inaccurate conclusions, be difficult for a visually impaired user to use, or be difficult for any user to use in specific environmental conditions. The “rainbow” color map is both commonly used and a particularly poor choice [4].

The Yeti UI allows the user to choose between two color schemes – one with a dark background and colors that brighten as the values increase, and the other with white background and colors that darken as the magnitudes increase. The schemes use blue colors to represent noise and gray shades to represent signals. The schemes are appropriate for different lighting conditions that users encounter in the field. Figure 7 and Figure 8 depict the light and dark color schemes for a CPD impairment. Figure 9 and Figure 10 further show the effectiveness of the heatmap display with CMD and HPNA impairments.

Figure 7 – Heatmap Display with Light Theme, CPD Impairment

Figure 8 – Heatmap Display with Dark Theme, CPD Impairment

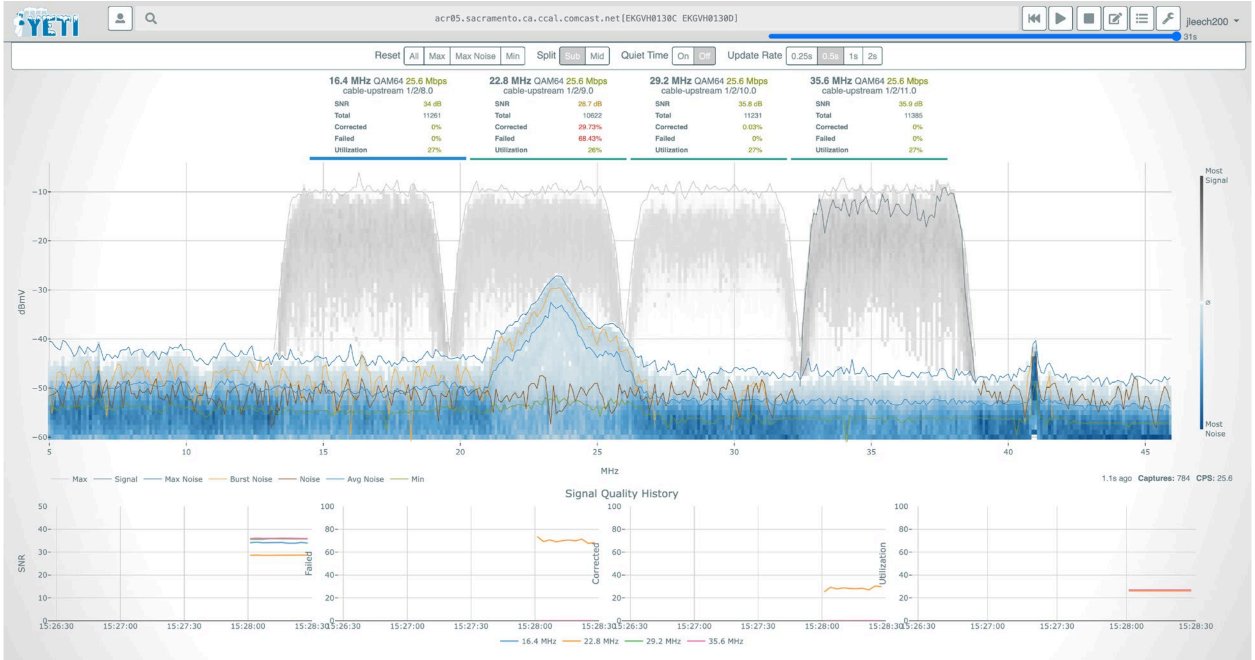

Figure 9 – Heatmap Display, CMD Impairment

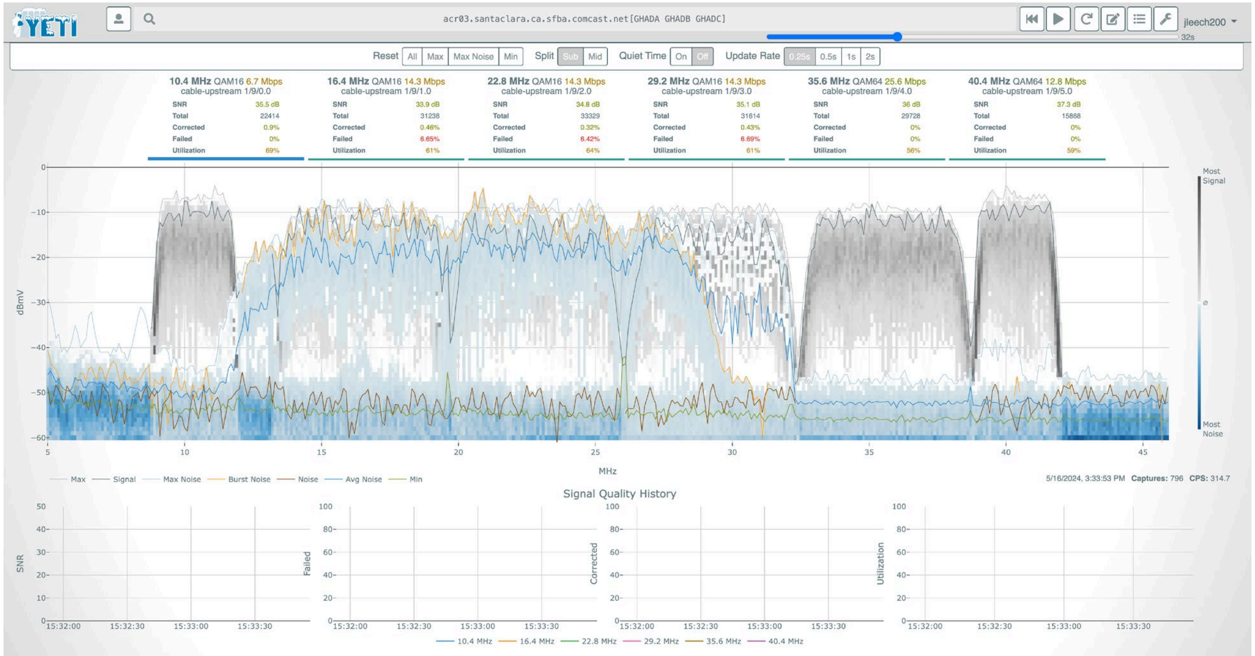

Figure 10 – Heatmap Display, HPNA Impairment

Lesson learned #6: Use a heatmap display with good color schemes.

4.1.2. Exponential Decay

The half-life dictates the rate of decay, such that a value in the heatmap will be reduced to half its original value in that time. The Yeti UI provides a user-configurable half-life setting with values ranging from 0 (off) to 64 seconds. Shorter, sub-second half-life values are useful for seeing frequent/fast-burst ingress. Longer values (or off) are useful for capturing infrequent ingress or if the engineer is not actively watching the display.

4.1.3. Data Analytics

The heatmap data structure can be used for data analytics, e.g., to compute values such as mean, standard deviation, and percentile. The same computations could be made using the raw data that populated the heatmap but would incur more computational and storage costs. There is some loss of accuracy due to the resolution of the heatmap. Some of that loss can be recovered by using linear interpolation. When exponential decay is used, it will bias the computations toward more recent values. Yeti uses the heatmap data and analytical functions in the UI for the Min and Avg traces, with or without the heatmap visualization enabled.

4.1.3.1. Mean

The mean, or average of the raw data, is computed by the sum/count. In the heatmap, each column represents the distribution of dBmV values for a specific frequency. Summing the values in the column reproduces the count from the original raw data. Summing the values multiplied by the dBmV (optionally converted to linear) reproduces the sum of the raw data.

4.1.3.2. Standard deviation

Standard deviation can be computed similarly as the mean computation above.

4.1.3.3. Percentiles

The nth percentile of raw data is computed by sorting the original values in ascending order and taking the (count * n / 100) th value from the sorted list. The 0th percentile is the min and the 100th percentile is the max. With many raw values, this can be computationally expensive to do the sorting step, but not so with a heatmap. Sum the values in the column (count), start at 0, and work upwards computing a running sum until the value reaches the target (count * n / 100). Optionally, use linear interpolation to produce an intermediate value within the cell's vertical range.

Averaging a range of percentiles can undo the quantization of a hardware FFT and approximate the result of a min-hold of a swept-spectrum analyzer. In Figure 11, the horizontal bands at -60 and -50 dBmV show the quantization. There are no intermediate values returned by the hardware FFT between -60 and -50 dBmV. The steps between values become smaller as power increases. The Min trace (army green color) is computed by averaging the 0th – 50th percentiles of noise.

Figure 11 – Noise Floor Quantization

Lesson learned #7: Use analytical functions on the heatmap data in the UI and backend.

4.2. Signal / Noise Classification

4.2.1. Quiet Time

Unlike hardware-based solutions, the upstream burst receiver also has the benefit of fine-grained scheduling capabilities. As such, one of the trigger modes allows for the spectrum to be captured while no bursts are scheduled, also known as a quiet time. Spectrum captured during a configured quiet time interval ensures that no modem upstream bursts are present for the duration of the capture on a single DOCSIS 3.0 upstream SC-QAM channel. Modems may or may not be transmitting on the other channels during the capture, and noise often crosses channel boundaries. Modems may also transmit when they are not supposed to, and specific versions of CMTS implementations may have bugs in the quiet time functionality. A previous version of Yeti allowed the user to select and change a single quiet time channel. In its current version, Yeti configures a quiet time watch on each SC-QAM channel and performs captures on each in a round-robin fashion. The Yeti software then performs an algorithmic signal/noise classification (see 4.2.2 below) to create the illusion of a global quiet time setting across all channels. This simplifies the UI, allows different users to have different settings on the same watch, and allows the user to change the quiet time setting during DVR playback.

DOCSIS 3.1 specification further specifies a quiet probe capture on an entire orthogonal frequency-division multiple access (OFDMA) channel, as well as specific spectrum capture triggers [5]. Yeti only leverages the free-running capability of UTSC.

4.2.2. Algorithmic

While Yeti does spectrum captures on SC-QAM channels in quiet time to ensure that some captures include noise, it relies exclusively on algorithmic classification for the final determination. The specific algorithm is beyond the scope of this paper. The algorithm uses power levels in the data and guard bands along with a heatmap data structure and analytical functions (see Section 4.1.3, above). The algorithm's effectiveness can be reduced by conditions such as extremely high utilization and noise at elevated levels and/or in the guard bands. The algorithm works well in practice but could be improved by using time in-phase quadrature (IQ) data and more advanced digital signal processing (DSP) techniques.

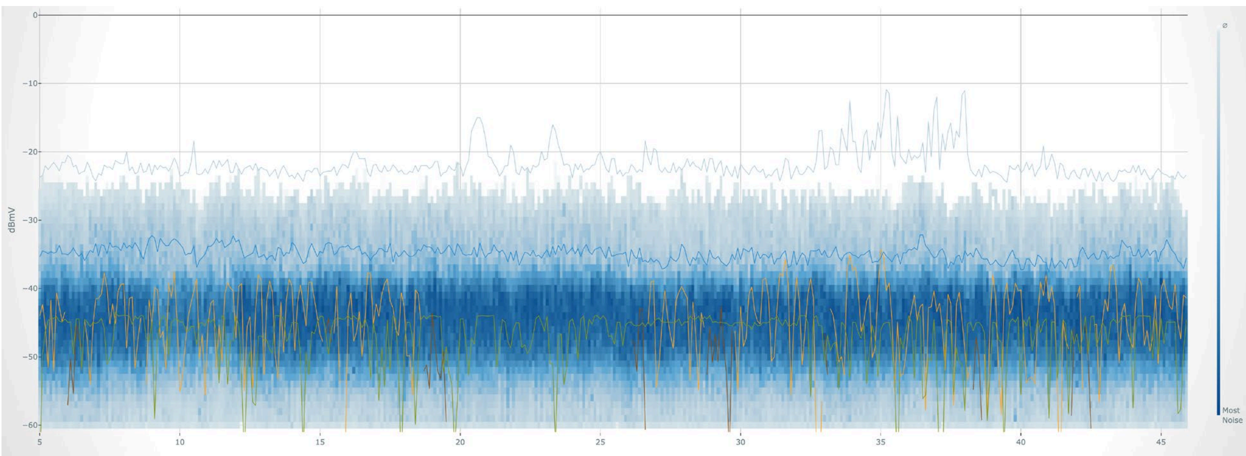

Figure 12 – Noise Only Display After Signal / Noise Classification

Lesson learned #8: Augment quiet time captures with algorithmic signal/noise classification.

5. DVR

Yeti introduced DVR functionality in February 2019. Via the UI using the familiar DVR metaphor, users can pause, rewind, fast-forward, etc. External triggers ensure recordings are made when issues are happening. The DVR reduces troubleshooting time as technicians can know in advance what they are looking for. Distinct types of issues may involve different repair agencies (fix agents) and troubleshooting steps. For example, fixing HPNA issues often involves finding the source from a recently disconnected subscriber, whereas the source of CPD is typically at the output end of an amplifier [1]. Likewise, CMD noise is often from a current subscriber with a particular model of modem coupled with a loose connector.

The raw DVR data in Yeti can be accessed via an API for further analysis by other tools and teams. For example, the Yeti development team, with the data science team, used Yeti DVR data to pre-qualify mid-split spectrum before activation.

Figure 13 below shows DVR listings with their accompanying thumbnail images and the UI elements of the DVR function.

Figure 13 – Yeti DVR Functionality

The DVR functionality, in addition to providing insight into the type of impairment, introduces a historical record. This history can be examined for patterns, allowing the dispatch of the correct fix agent at the proper time of day, particularly helpful for intermittent nighttime impairments.

Lesson learned #9: Capture and record spectrum when issues are happening.

Lesson learned #10: Provide programmatic access to the raw DVR data for later analysis.

6. FFT

Yeti externalized the FFT processing step in November 2019.

The hardware FFT on the burst receiver is configured to minimize errors in de-modulation with a setback to the analog front-end gain. As a result of the hardware FFT, there can be quantization of the values in the lower range of the noise floor. There can also be a reduction in the dynamic range. Both limit the visibility of noise. If possible, getting the data in the time domain in IQ format and doing the FFT in software is advantageous as it will remove the quantization and increase the dynamic range. Not all CMTS vendors support time IQ format.

Several efficient FFT software implementations exist and there is no performance concern using them in the backend. If saved in the DVR recordings for later processing, the raw time IQ data can be used for advanced analysis. One example of advanced analysis is determining the exact frequency of a short-wave radio signal by using zero-padding before the FFT step. Another example is characterizing the amplitude, duration, and periodicity of burst noise using time/frequency analysis.

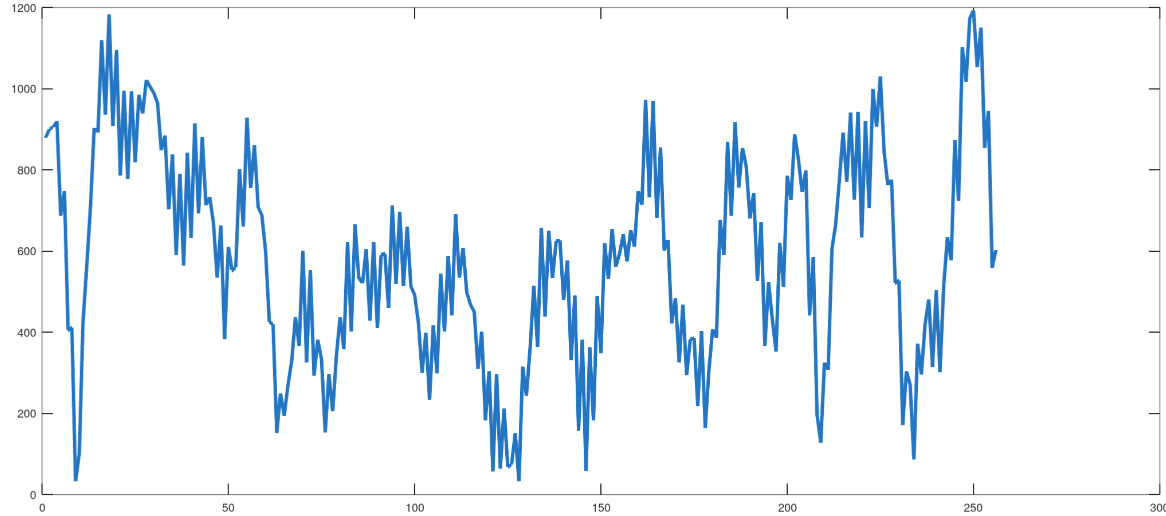

Figure 14 – Amplitude of Time Domain IQ Data

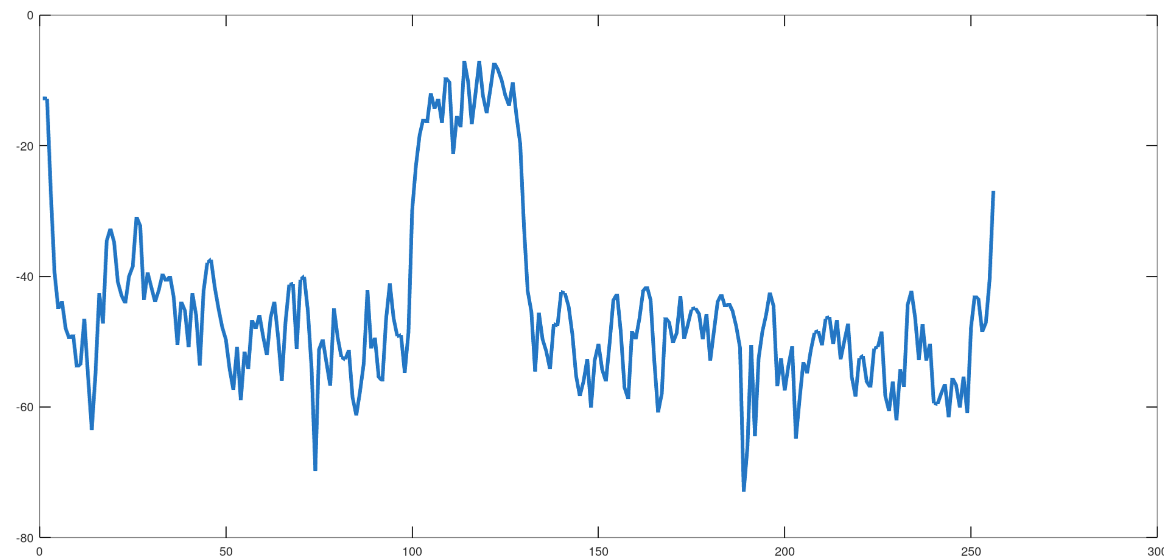

Figure 15 – Frequency Domain Amplitude after FFT

Lesson learned #11: Capture spectrum in time IQ format and do the FFT in software to increase noise floor visibility.

Lesson learned #12: Store the raw time IQ data in the DVR recordings for advanced analysis.

7. vCMTS and Remote PHY

Yeti first added support for virtual cable modem termination systems (vCMTS) and R-PHY in April 2020. From a field support perspective, Yeti support for the new platform was a high priority, as without it there was no alternative other than sending engineers to the field with meters to troubleshoot noise and ingress. Swept-spectrum analyzers are not feasible for R-PHY deployments where the CMTS physical layer is in a node in the outside plant. From the software development perspective, some of the challenges for vCMTS were software reliability, vendor-specific differences, calibration, out-of-order time IQ data, and network connectivity.

With vCMTS and R-PHY, Yeti gets the spectrum captures in a streaming fashion and additive round-trip latency is much less of a concern. The capture rate is solely dictated by the settings. The Yeti development team doubled the capture rate for vCMTS compared to iCMTS. Additionally, on iCMTS, spectrum captures for multiple watches on the same iCMTS share the resources and capture rate. For example, if the capture rate is 20 CPS, a single watch gets the full 20 CPS, but two simultaneous watches get 10 CPS each. Additional watches lower the CPS of each watch accordingly. vCMTS does not have this limitation. Each watch gets the full capture rate. The benefit of higher capture rates to troubleshooting efforts cannot be understated. It is still advantageous to send the large spectrum capture payloads shorter distances and to have geographic redundancy for the backend for reliability.

Despite the increased capture rate, intermittent noise events can still be missed due to the bursty nature of upstream noise. For a DOCSIS 3.1 OFDMA channel, per mini-slot MER data does not miss noise events and shows the precise signature of noise at 400 kHz resolution. Such data may be collected by a PMA system [2]. Due to the funnel effect, this data will not aid in noise localization for most cases. However, the noise signatures present in the spectrum capture and/or per mini-slot MER data can be correlated to UDA data for noise localization [6].

Lesson learned #13: Incorporate per mini-slot MER data on OFDMA to visualize noise.

8. DOCSIS 4.0 and FDX

Yeti introduced support for DOCSIS 4.0 and full duplex (FDX) in March 2024. Development challenges included supporting four ports with unique spectrum captures at each port, supporting up to six upstream channels, determining the best combination of FFT settings, and calibration.

The hardware supports a maximum FFT capture width of 512 MHz, of which only about 80% is usable. This is less than the full width of the FDX spectrum range. As such, Yeti breaks the FDX spectrum into two halves with three FDX OFDMA channels each. Only one half is visible at a time.

Calibration is important. Incorrect calibration values returned to Yeti by the vCMTS can cause issues with noise floor visibility and signal/noise classification. The issues can be mitigated by overriding the calibration or expanding the dynamic range of the spectrum view.

Although the four ports are combined at the burst receiver, the spectrum for each is sampled individually before combining. This allows for noise localization but poses a challenge to field engineers doing the troubleshooting due to the additional system complexity and troubleshooting time. Also, external DVR triggers need to coordinate a recording for each port sequentially.

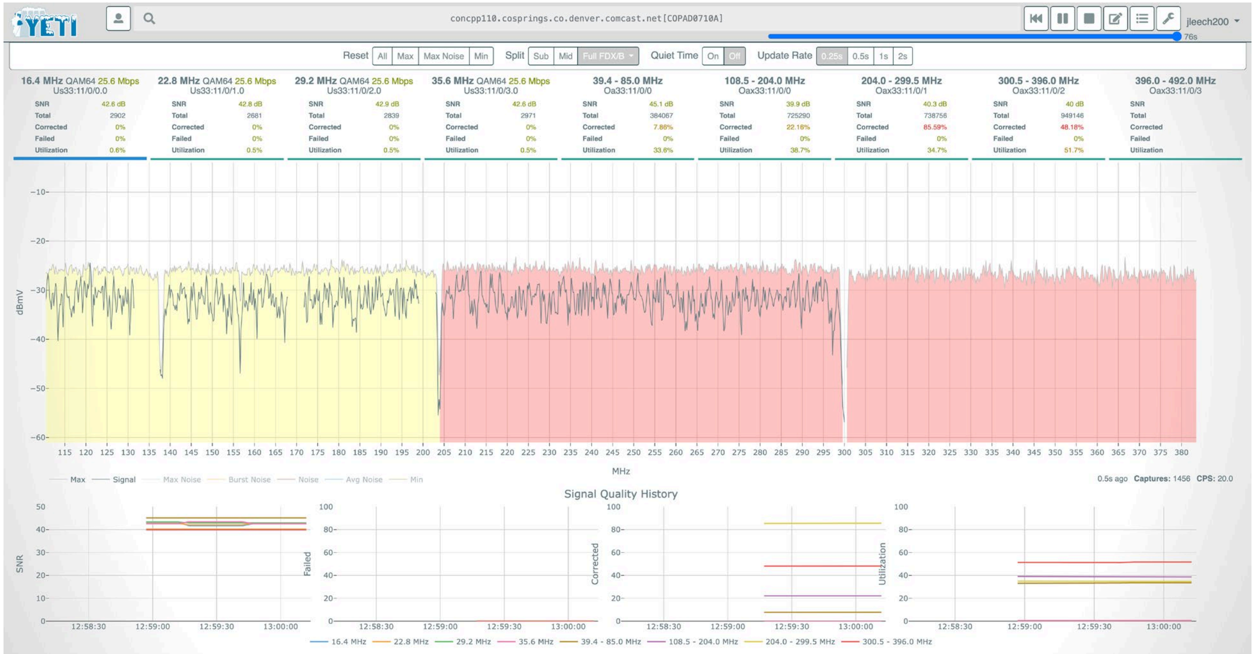

Figure 16 – Yeti FDX

Lesson learned #14: Maximize the capture width.

9. Future Work

Several opportunities exist for future features and enhancements.

9.1. Artificial Intelligence and Machine Learning

One opportunity is to automate the classification of various plant impairments using AI and ML techniques such as pattern recognition, supervised learning, and neural networks. AI models can be built for common impairments and refined with input from users. Pattern matching can be applied to single-spectrum captures or aggregate data from the heatmaps. Alerts and events can be created and correct fix agents automatically dispatched. In fact, with the unified DOCSIS 4.0 chip an AI engine exists in each amplifier, node, and customer premises equipment (CPE) device that can run these models within the network. This would decrease time to detect issues and can be used to balance data volumes being sent to the cloud.

9.2. UDA Correlation

Noise sources can be localized by performing automated correlation from noise signatures in spectrum capture to noise signatures in UDA.

9.3. FDX Amplifier

FDX amplifiers can be used as an additional source for spectrum captures to aid in the localization of noise sources.

10. Conclusion

We learned many important lessons over the seven years of Yeti development and field use. The major milestones included heatmaps, algorithmic signal/noise classification, externalized FFT, and support for vCMTS and FDX. Several opportunities exist for future work and enhancements, primarily in automated detection and correlation. We believe other operators can benefit from our lessons learned and apply them to their own UTSC efforts.

Abbreviations

AI	artificial intelligence
API	application programming interface
CINR	carrier to interference plus noise ratio
CMD	common mode disturbance
CMTS	cable modem termination system
CPD	common path distortion
CPE	customer premises equipment
CPS	captures per second
dBmV	decibels per micro-volt
DOCSIS	data over cable service interface specification
DSP	digital signal processing
DVR	digital video recorder
FDX	full duplex
FEC	forward error correction
FFT	fast Fourier transform
HPNA	home phoneline networking alliance
iCMTS	integrated cable modem termination system
iOS	iPhone operating system
IQ	in-phase quadrature
kHz	kilohertz
MER	modulation error ratio
ML	machine learning
OFDMA	orthogonal frequency-division multiple access
PMA	profile management application
RBW	resolution bandwidth
R-PHY	remote physical layer
SC-QAM	single channel quadrature amplitude modulation
SNMP	simple network management protocol
SNR	signal to noise ratio
UDA	upstream data analyzer
UDP	user datagram protocol
UI	user interface
UTSC	upstream triggered spectrum capture
vCMTS	virtual cable modem termination system

Bibliography & References

1. *"SCTE 280 Understanding and Troubleshooting Cable RF Spectrum", 2022, SCTE.*
2. *"PMA Improvements – Strategies Employed for Faster Mitigation, Increased Capacity, and Cost Savings", SCTE Cable-Tec Expo 2022, https://www.scte.org/documents/5854/FTF22_WLINE10_Leech_3854.pdf, J. Leech, A. Martushev.*
3. *"ANSI/SCTE 249 Test Method Common Mode Disturbance", https://www.scte.org/documents/793/ANSI_SCTE20249202018.pdf, 2018, SCTE.*
4. *"Rainbow Color Map (Still) Considered Harmful", D Borland, IEEE Computer Graphics and Applications Volume 27 Issue 2, March 2007.*
5. *"Data-Over-Cable Service Interface Specifications DOCSIS® 3.1, Physical Layer Specification, CM-SP-PHYv3.1-I02-EC", 2014, CableLabs.*
6. *"PNM Upstream Data Analysis: A Practical Solution for Automatically Locating Upstream Noise and Ingress", https://www.scte.org/documents/6360/3583_Wolcott_5152_paper.pdf, SCTE Cable-Tec Expo 2023, L. Wolcott, R. Gonsalves, J. Leech.*

