How Cox Communications Implemented an Expert System for Service-First Autonomous Operations

A Technical Paper prepared for SCTE by

Dave Norris
Sr Director, Video Engineering
Cox Communications
6205-B Peachtree Dunwoody Rd
Atlanta GA, 30328
dave.norris@cox.com

Table of Contents

Title	Page Number

List of Figures

Title	Page Number

1. Introduction

At the beginning of the current video evolution to deliver live IPTV streaming to subscribers, the Cox Video Engineering team recognized they were entering a period of unprecedented change and growth. There would be many challenges coming fast to high-capacity video delivery. We knew we'd need to absorb and incorporate new methods of compressing and delivering video without dropping the preexisting systems. At the same time, we were approaching end-of-life status on our national video compression systems. All these factors drove the need for and timing of deploying a new and more flexible video compression platform.

In 2017, the Cox Video Engineering team evolved our national video encoding systems into a virtual platform. For agility, flexibility, API controllability, and ultimately economy, we selected a software video encoder running on commodity servers in dense blade-center chassis. One of our chief concerns was that deploying a virtualized video encoder might be less reliable than the legacy hardware-based appliances. Any negative impact to video stream reliability would be highly visible, driving subscriber frustration, calls, and expense.

We decided to close any performance gaps by implementing a closed-loop Expert System solution to manage all system day-1 build and day-2 operational functions.

This paper will present some of the reasons that drove the evolution to software video encoding, some unexpected motivation that expanded our scope, a few details about the phases of the implementation, and the results.

2. Motivation and Drivers

At the end of 2015, we recognized that in the next 18-24 months we'd need to replace our aging national video encoders. We realized that Cox would be expanding our delivery of MPEG-4 to commercial and residential customers, as well as growing our offering in OTT-style IPTV live streaming for both in and out-of-home uses. We opened an RFP and began the search to find our new video encoders. At that point, no single vendor offered everything we were looking for.

It quickly became apparent that the software-based vendors had the most up-to-date feature sets, and the least disruptive pathway for upgrades. Our requirements included things like: time-aligned multi-bitrate ("MBR") encoding for adaptive ("ABR") playback; accepting HEVC contribution streams from content-providers and eventually outputting content compressed in HEVC; ATSC 3 features supporting new audio CODECs and metadata; etc. The hardware-based compression vendors could meet many of the requirements but were challenged when addressing the path and timing for incorporating new features. The software-based compression vendors provided a much faster and more efficient path for incorporating new technologies and features. Thanks to Moore's Law, CPUs (and GPUs) were closing the density gaps that had stifled software video encoding for years.

As our path to software-based compression became clear, we were increasingly intrigued by the prospect of leveraging APIs and scripting to provision both the environment and the video streams on the platform. While server stand-up automation is mature, automating video compression software was new territory. Video compression devices, whether hardware or software based, traditionally use GUIs for configuration. The operator must manually enter a lot of parameters to define the inputs, outputs, customized filters and processing of the audio and video streams, ad-insertion messaging, and any other necessary metadata. This manual interface interaction is very repetitive and error-prone, particularly when done at a high volume. We recognized we needed to bring automation into the process.

3. Platform Evolution

With the decision to move to software-based encoding, it was time to select hardware for our new platform. We decided to purchase and host our own Cisco UCS blade server chassis and run them as bare metal for video encoding. Bare metal saved us CPU processing overhead vs VMs, and running warm spares enabled the fastest methods of video stream failover. We leveraged PXE boot and Spacewalk to take our hundreds of bare-metal servers from boot-up through the installation of CentOS.

4. Selecting an Automation Strategy

While it was expected that we would be scripting the API calls to load the video streams onto the encoders, we also decided to engage companies offering more advanced automation solutions. We spoke with a vendor offering an “Expert System” automation platform and quickly became motivated by the idea. Instead of just executing a script referencing a database that defines the systems configuration, the Expert System can be programmed with rules and exceptions. It’s also a form of a state machine and can take “expert” actions based on a pre-programmed understanding of the operational states of the components within its scope of authority.

To explain the differences in strategies, consider a self-driving car. It can be programmed to deliver its occupants and cargo to a destination. Scripting can do that in a closed and controlled environment like at a proving ground—just define the destination and route to move the car from here to there. But if you want that car to navigate on real roads, alongside the rest of us and arrive safely, advanced steps must be taken to adapt to the real-world issues it will encounter along the way. Advanced algorithms are necessary to account for the dynamic environment consisting of traffic, adverse road conditions, construction zones, weather, flat tires, etc. The Expert System self-driving car is programmed to adapt and react to each variable like a human expert would. Each of the many various task algorithms are employed as part of the autopilot system as needed to avoid that box in the road, stop for the school bus, or slow down over the rough roads, etc.

We saw a ton of potential in the Expert System approach. Properly designed, this could enable capabilities well beyond simply provisioning thousands of streams automatically. It could become an autonomous operator of the platform after that provisioning. We selected the StratOS Expert System from Sea Street Technologies, and partnered with them in the development process.

5. Development

We created a flat database file that included every parameter needed to build our hundreds of video streams into the encoders. For efficiency, we utilized look-up tables for the highly repetitive audio and video settings based on classes of outputs in MPEG-2 and 4 like SD, HD 1080i, HD 720P-60, single program AC3 audio, primary and AC3 audios, AAC audio, etc.

That database would be loaded into the Expert System to kick off the automatic build activities. Any future additions, changes, or deletions would be conducted in the same way. Once read into the Expert System, it automatically modifies whatever necessary to reflect any changes to the latest version of the database.

At Cox we use the TeleStream (formerly Tektronix) Sentry product family as video probes. The Sentries monitor and alert on issues that impact audio and video quality. These can be conditions like audio silence or excessive loudness, loss of stream, frozen video, tiling or blocky video artifacts, among many other conditions. When any Sentry sounds an alarm condition, SNMP traps are sent to collectors in the Network Operations Center and are then forwarded to the NOC personnel for triage and escalation to the correct fix agents.

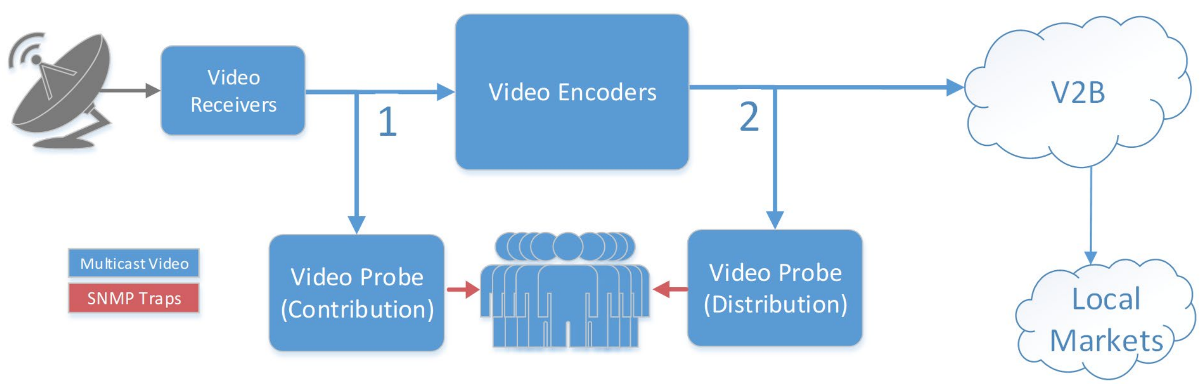

Figure 1 – Humans respond to SNMP traps from the Video Probes on the Contribution and Distribution sides of the video encoders

We integrated the Contribution and Distribution probes SNMP alerting capabilities into the Expert System by adding it as an additional destination for traps and then modeling how it should respond as they arrive. This allows the system to be aware of the health and status of the streams at both the input and output sides of the video encoders.

We also documented the steps a human expert fix agent would take based on the good or bad status reports from the video probes. The Expert System was programmed to monitor for traps on the distribution streams, and when an alarm condition exists, to check for the same traps on any or all the available redundant contribution streams. With knowledge of stream health on both sides of the video encoders it's controlling, our Expert System automatically makes informed "expert" decisions on what to do, in sequenced steps, to restore from the outage or impaired condition.

To define the actions an expert human operator would take during break-fix activities, we created a decision matrix consisting probe-error feedback that defines the problem conditions, and the first few steps the expert would take under those circumstances to restore the stream. For example:

- Output bad, all inputs good→ restart the encoding service
  - If still bad→ mute the primary output and rebuild the stream in spare encoder pool, then remove the stream instance from the primary encoder
- Output bad, primary input bad, secondary input good→ toggle input
  - If still bad→ restart the encoding service
  - If still bad→ mute the primary output and rebuild the stream in spare encoder pool, then remove the stream from the primary encoder
- Output bad, all inputs bad→ do nothing, let the alarms flow to the NOC Operators

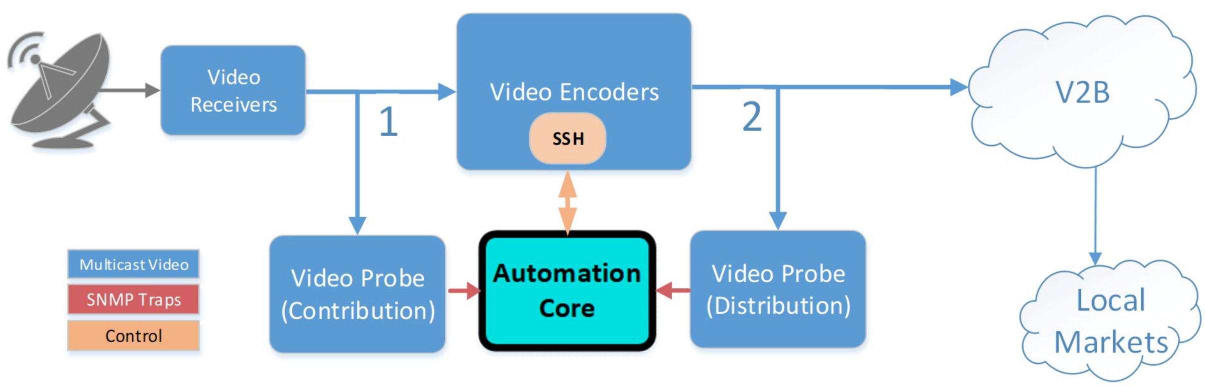

Figure 2 – The Expert System Automation Core Responds to SNMP Traps

6. "Service-First" Self-Assured Autonomous Operations Platform

A Service-First solution is one that focuses on the health of the service over the health of the individual elements. Frequently, automation platforms focus on ensuring each element is healthy but aren't aware of the health of the service itself end-to-end. Working with Sea Street Technologies, we used their StratOS platform to create an Expert System that focuses on the health and remediation of the service as the first priority, before it takes steps to fix the elements.

For example, in our solution when a fault is detected with a video stream, the Expert System will first focus on fixing it by restarting the encoder service or moving it to a spare encoder within seconds of the fault. Only after the stream is reported to be healthy again does the Expert System return to fix the faulted encoder.

The Expert System is designed to manage each video stream from initial creation to daily operations. When new video streams are brought online, the Expert System will autonomously create and configure each one and then start to monitor and independently operate each one indefinitely. This is done through "Objectives," which are continuously running objects that collect health data, execute business logic, and take actions. Instances of an Objective are created for every encoded stream and the instances all run in concert with each other, working to ensure the health of all streams and underlying elements. This method of self-assurance has enabled us to improve the reliability of our video streams, significantly improve MTTR, and reduce the need for additional staffing.

7. Launched!

In 2017, we launched our new software encoder platform under Expert System control. We entered the launch maintenance activity with all the servers online and provisioned only with CentOS. The platform-defining database was loaded into the StratOS Expert System. StratOS interpreted the database and took over the maintenance, loading and licensing the video encoding software before it began to automatically provision approximately 650 streams in the first data center.

The full configuration was completed in a less than an hour. A manual configuration of the same platform would have taken a few months of maintenance-window activities. Since launch, our Expert System autonomous operation platform has been routinely restoring errored streams in seconds with machine speed and accuracy, typically before the NOC is even alerted to the service impact.

8. Continous Improvement

The modular architecture of our Expert System makes the process more straightforward for adding features or modifying parameters. For example, we were able to easily add repair-retry logic to the system for use after extended contribution stream outages.

We've also added a second video encoder vendor's software to our Expert System. This has proven the operational environment to be independent of any single video encoding solution. Our preexisting business and operations logic was simply inherited by the new vendor-specific API resource driver module. Should Cox decide to modify our video encoding automation business rules or operations logic, those changes would immediately be applied to all the Objectives, regardless of the underlying software encoding platform.

9. How's it Working?

After years of successful operation, this has been a trailblazing and fantastic success! Bell Labs was hired to evaluate the state of automation in use at Cox and described our Expert System autonomous operation platform as the most advanced of its kind they had ever seen. They conducted an evaluation of some of the benefits and concluded the following:

- Reduced Eng effort: Fewer Maintenance Windows
  - **.4 Full Time Employee ("FTE") saved (efficiency) per year**
- Reduced Eng effort: Incident management
  - 100 in-scope incidents per week
    - **38 minutes saved per incident (T2)**
    - **15 Minutes saved per incident (T3)**
    - **2.2 FTE saved (avoidance) per year**
- Reduced Eng effort: Platform Process savings
  - **Cut time to process by 90%**
  - **1.9 FTE saved (efficiency) per year**
- Reduced Eng effort: Software release upgrade
  - **.5 FTE saved (efficiency) per year**
- Team avoided FTE growth
  - **5.0 FTE saved (avoidance) per year**

Per Bell Labs calculations, our Expert System has saved Cox over $10 Million, and continues to provide savings of $2 Million per year.

Abbreviations

ABR	Adaptive Bitrate
API	Application Programming Interface
GUI	Graphical User Interface
HEVC	High Efficiency Video Coding ("MPEG-5")
MBR	Multi-Bitrate Encoding
MTTR	Mean Time to Repair
NOC	Network Operations Center
SNMP	Simple Network Management Protocol
V2B	"Video to the Backbone" -Video distribution via the Cox backbone
VM	Virtual Machine
GUI	Graphical User Interface

