DOCSIS 3.1 CPE Validation

Why a software validation campaign makes sense

Date

The case for DOCSIS 3.1 CPE software validation

Quality assurance testing provides an insurance against the damages from malfunctioning systems. In insurance it's all about cost vs. risk, where risk is the probability of the issue times its impact.

Over the years, Excentis has spotted dozens of hardware, software, configuration and integration issues with severe real-life impact on both CAPEX (unneeded or wrong investments), OPEX (flood of customer calls and field operations) and revenue (customer churn and brand damage). The reported issues have blocked mass production of flawed equipment or halted planned roll-out of malfunctioning new software.

Deployment of malfunctioning equipment causes

- Customer complaints and eventually churn and bad publicity
- Increased OPEX due to service calls and truck rolls
- Increased workload across the operation: service desk agents, NOC staff, field technicians, engineers and management
- High recall costs (in case of hardware issues)
- Consultancy & debug costs; resolving field issues is both urgent and complex

For complex systems like access networks, a 100% ironclad insurance is not feasible – let alone cost-effective given the immense amount of cases and scenarios. So while the case for QA testing should be clear, the case for individual test cases must be made. That's why this test plan focusses on those tests that uncover issues with the highest probability and/or the biggest impact.

Below we highlight four cases from the draft plan.

System stability test

When new CPE hardware or software is introduced, the data and voice services offered over the [CUSTOMER]'s DOCSIS network should keep operating as expected. Unstable or degrading service (whatever the reason) has big customer and operational impact. By testing the [CUSTOMER]'s access network in an integrated way over multiple days, we spot elusive issues before they get released in the field.

This test validates the end-to-end access operation and is the only way to uncover elusive but important issues. That's why we propose to run this test before every large-scale field trial or mass deployment of new hardware or software commences.

Note: Given [CUSTOMER]'s current transition to DOCSIS 3.1, we test the new CPE on two CMTS configurations: the current EuroDOCSIS 3.0 config and the upcoming DOCSIS 3.1 config.

What we've seen

Some issues spotted while running this test in the past:

- Modems crash during the test and need a power reset
- Degradation of voice call quality after a few days
- Disruption of data and voice service due to unexpected modem resets
- Modem taking a long time to come online or require multiple power resets
- Modems reporting incorrect data and metrics, which are used for maintenance and investment decisions

Test overview

This test verifies the operation, stability and interoperability of the CPE device in a simulated real-life setup. The CPE device and its services should operate in a stable and uninterrupted manner for four days. During this test time, we make sure the CPE device remains online and handles periodic network events (like DHCP renewals, BPI+ key exchanges and load balancing) well. Data service is tested by looking at packet loss, data rates and TCP connection failures. Voice service is tested by looking at voice connection failures and dropped calls.

We automatically gather relevant data from the CMTS, CPE, provisioning system and traffic generator during the test. We report anomalies and use the gathered data to do an initial investigation. This analysis drills down to a specific network component, but a full root cause analysis is outside the test scope.

Service rates compliance test

[CUSTOMER] should comply with the data services it offers to its customers. These customers increasingly compare their offered service with what actually got through – and publicly comment on the results. While individual users run a simple one-way speed test, consumer advocacy groups and regulators may test services more thoroughly for their market reports. With this test, operators move proactively by doing such an assessment internally.

The business case for running this test consistently before any software or hardware deployment is strong, given the short duration vs. the potential marketing fall-out.

Note: This test can be run for each top-level service [CUSTOMER] is currently offering or for services that will be rolled out in the near future.

What we've seen

This test has provided these interesting results:

- While the service rate was met on IPv4, this was not the case for IPv6 traffic.
- When tested separately, upstream and downstream rates were met. However, high bidirectional rates were unstable, so service was only met intermittently.

Test overview

This test checks the data rate compliance with the offered service. Both upstream only, downstream only, upstream only and bidirectional cases are covered. Both IPv4 and IPv6 service is covered. In all these cases we measure TCP throughput similar to a speed test, but over a period of 15 minutes.

We report whether service was met and where things went wrong. The test is defined as a trigger to action; root cause analysis is not part of the scope.

Partial service test

When a DOCSIS channel goes down, modems can work with the CMTS to fall back to their remaining channels ('partial service'). Not falling back correctly has a devastating effect on TCP traffic and thus on end-user experience. Modems should also recover quickly and automatically once the channel comes back. This mechanism is a key aspect of modem resiliency, especially in deployments where such events are common.

The big impact on end-user experience and elusiveness of the issue (interplay with the CMTS can give interop problems) makes this an important test to run on every CM software update.

What we've seen

This test has brought the following issues into the spotlight:

- Modems regularly fail to recover, which results in a nearly empty upstream channel. The only way to resolve this is to disable partial service (not a good idea) or to build a complex monitoring system that manually reboots modems in partial service.
- A modem with a 10-minute upstream channel outage suffered a timing offset. Because of this, the modem fails to recover once the channel was back up and kills transmissions of other modems by sending at the wrong time.

Test overview

In the partial service test we interrupt specific channels temporarily and monitor how the modem responds. A variety of cases are covered: we bring down downstream and upstream channels, SC-QAM channels and OFDM channels, and primary and non-primary channels. For DOCSIS 3.1 modems we check whether the modem falls back to its back-up primary channel correctly – unlike ED3.0 modems who should reboot when losing their primary channel. These test cases are tested on both operational modems (who are online and active) and on modems that are coming online.

We monitor the modem status and DOCSIS messages to see whether the modem behaves as expected. When this is not the case, we show the what happened, when things went wrong and pinpoint the component at fault.

Active Queue Management test

With AQM the modem shapes upstream traffic in such a way that high-speed TCP data rates are improved. However, bad implementations or configurations can easily cause severe *negative* impact on the end-user throughput. By comparing AQM-disabled and AQM-enabled data rates, we see the optimization (if any) AQM brings and whether the optimization is needed to meet the data services offered by [CUSTOMER].

The AQM implementation has a software component, which means a test on each new software load is useful. Still, running it every single time may not be justified. It is clear however, that the test should be run when AQM is enabled for the first time (it is on by default in D3.1 products!) and even more so on non-certified devices.

Test overview

The interplay between AQM and TCP is a complex affair and depends heavily on the type of TCP traffic. That's why we look at the effect of enabling AQM on variety of real-life traffic patterns. When enabled, AQM is configured according to the [CUSTOMER]'s configuration. We expect AQM with the [CUSTOMER]'s settings to keep the TCP round-trip time in check and improve the general throughput.

We show the overall performance gain by enabling AQM and report on any remarkable results (e.g. for a specific traffic pattern). We note whether AQM is needed to meet the offered service levels and offer a recommendation to either enable or disable AQM for this software load. Anomalies or service degradations are shown, but a root cause analysis of the AQM algorithm is not part of the scope.