Guide to Managing Cable Network Traffic Congestion

Addressing Capacity and Congestion with Cable Modem Termination System (CMTS) “Tweaks”

July, 2020

John J. Downey Cisco Sr. CMTS Technical Leader

With unprecedented demand on local broadband networks, operators are under pressure to prepare for congestion and capacity issues before they happen. I plan to share some tips that will help alleviate and mitigate congestion while addressing capacity concerns on cable operator DOCSIS plants.

Top Seven Steps

1. Decrease Subscribers per Service Group (SG)
   - ✓ Increase the amount of SGs and add more licensing/channels.
   - ✓ Decrease service groups (SG) down to one fiber node (FN).
   - ✓ Utilize upstream (US) segmentation. Maybe the node supports multiple downstream (DS) optical receivers and US transmitters or the US utilizes baseband digital reverse (BDR) and can be converted from 1 to 2 US segments.
   - ✓ The last option would be a physical node split, but if utilizing distributed access architectures (DAA) like Remote-PHY, possibly a 1x1 RPD can be replaced with a 2x2 or some other variant.

2. Verify no Uncorrectable Forward Error Correction (Uncorr FEC) and “Clean” Plant
   - ✓ Uncorr FEC is basically dropped packets. Regardless of modulation error ratio (MER), signal-to-noise ratio (SNR), carrier-to-noise ratio (CNR), correctable FEC (fixed packets); Uncorr FEC is the most important to your end-customer.
   - ✓ **Note**: Uncorr FEC is not only caused bad plant issues. It could be from bad timing (time offsets and MAP Advance), poor port-to-port isolation leading to signal “bleed-over”, modulation profile settings, and a myriad of other contributors.

3. Increase Capacity without Physical Node Splits or SG Changes
   - ✓ Use the highest US and DS modulation along with the largest channel widths as possible.
   - ✓ Utilize DOCSIS 3.1 where possible. ***Note**: More speed does not necessarily mean less latency! D3.1 US may exhibit even more latency with ping tests.
     - How many D3.1 CPE are needed to justify exchanging ATDMA chs for OFDMA spectrum? 10%, 25%, 50%? If 10% of your users are using 85% of your capacity, they either have D3.1 CMs already or you could identify those “heavy users” with subscriber traffic management (STM) and then give them a D3.1 CM. Assuming that 10% use 50% of your US, that could be the justification to drop 2 ATDMA chs and use that spectrum for more efficient D3.1 OFDMA. Dropping from 4 to 2 ATDMA also gives back 3 dB max Tx power for D3.0 CMs along

with less DS overhead. Two ATDMA chs give 54 Mbps aggregate and could offer a 20 Mbps service and lower. All other US offerings >20 Mbps would use D3.1.

- ✓ Allocate more spectrum for high speed data (HSD) services. This may entail using spectrum once thought to be questionable such as: roll-off, known ingress areas like CB, LTE, Aeronautic, and the very low end of the US spectrum. DS may require you to "steal" from video spectrum. Analog video reclamation should be an easy sell but converting MPEG-2 video to MPEG-4, over the top (OTT) video is the ultimate end-game.
- ✓ Utilize/exploit "Powerboost". More speed could translate to "actual" vs "perceived". Powerboost is a term/feature (name trademarked by Comcast) used for faster speeds that can affect perceived speed (DS & US). A command may need to be configured on the CMTS for DS Powerboost activation along with a very large DS Max Burst setting, like 50 MB in the cm file.
    - **Note:** DOCSIS 3.0 CMs support a TLV for per-CM Peak Rates.
    - Utilize to alleviate typical 10% over-provisioning, which is usually done to negate differences between layer 2 & 3 speed reporting.
    - Can exploit US Max Traffic Burst for US "Powerboost" as well. US Powerboost may help alleviate perception of lower speed. This can also be exploited to apply a peak-rate and less over-provisioning, which is typically done today at 10% just to "ring the bell".
    - The "jury is still out" whether this can negatively affect OTT video and other adaptive bit rate (ABR) applications.

4. Eliminate Overhead
    - ✓ More USs in a MAC domain creates more DS MAP overhead at ~.4 Mbps per US. Increasing US channel size and eliminating too many US channels will help. Another way to get less US channels per MAC domain would be to create more MAC domains. This can be achieved easily when the SG consists of two fiber nodes (FNs).
    - ✓ Also, moving to every 4th DS as Primary could save 54 Mbps per 24-ch DS bonding group (BG). The trade-off is less aggregate capacity for D2.0 CMs.
    - ✓ Remove "stale" service flows. Some inactive VoIP flows may not be torn down according to their T8 timer and can be done automatically with the CMTS command, `cable service flow activity-timeout 300.` Add this CMTS global command so flows with no activity > 300 seconds (5 minutes) are torn down if the CM/eMTA does not do it automatically.
    - ✓ Understand WIFI, VPN, etc. encapsulation overhead along with potential "bottlenecks".

5. Control Abusers and Denial of Service (DoS) Attacks
    - ✓ Cloning – DMIC, BPI+, "Hotlist".
        - The CMTS can control the same mac address on a chassis and identify potential cloned devices. It can also control CPE "appearing" behind multiple CMs, but it has not visibility across multiple chassis. Some external devices may have this visibility like the DHCP server and better suited for cloning identification.

- If a cloned device is identified, you can disable CM ranging and registration by implementing the Cisco “hotlist” command.
```
(config)#cab privacy hotlist ?
  cm            Add cm hotlist
  manufacturer  Add manufacturer hotlist
(config)#cab privacy hotlist cm ?
  H.H.H         CM mac address H.H.H
```
- **Note**: The CM could still be ranging “all the time”, but it will not even show init(r1) on the CMTS. Some could argue that it’s better to let it register and give it a cm file with network access disabled.

✓ Over-Use/Abuse
- Deep Packet Inspection (DPI) can be used to at least identify “heavy” users.
- Subscriber Traffic Management (STM) takes rate limiting and monitoring down to the least common denominator, which is bytes. There is no bias towards ports, applications, etc. It solely looks at total bytes over a certain time frame and can dynamically drop the CM’s QoS to a lower rate for a given time period.

✓ Arp Attacks, IGMP Joins?
- Arp Filters, Access Lists (ACLs), subscriber-based rate limiting (SBRL)

✓ Expiring Certs
- Allow/Deny Lists. Cablelabs’ certifications are expiring by end of 2020 and some CM certifications could render it unusable.

6. Optimize CMTS Efficiency
   ✓ Load Balancing
   ✓ D3.1 Graceful Profile Management & US/DS Resiliency/Partial Mode

7. Implement Cache Servers
   ✓ **Note**: Netflix, YouTube and other OTT video providers may drop video quality to save bandwidth and/or temporarily halt 4K video offerings.

CMTS Suggestions

- **CM Insertion Interval** - CM ranging opportunities
  ✓ `(config-if)#cab insertion-interval auto 120 1000` or (`60 480`)
  ✓ The Cisco scheduler has dedicated time every 60 ms for initial maintenance (IM). The number of CMs online and traffic utilization will automatically make the insertion interval change between those two numbers. Verify with the show controller command:
  - `cbr8#sh contr c1/0/2 upstream | in Insertion`
```
  Ranging Insertion Interval automatic (120 ms)
  Ranging Insertion Interval automatic (120 ms)
  Ranging Insertion Interval automatic (120 ms)
  Ranging Insertion Interval automatic (120 ms)
  Ranging Insertion Interval automatic (120 ms)
```

- ✓ **Note**: The value reported could be an average between a changing insertion interval and report a value that is not an increment of 60 as one would expect.
- ✓ It may be worth experimenting with this Insertion Interval. The cBR-8 defaults are 120 1000, but we have had success in the past with the old uBR10K defaults of 60 480. We have also used "fixed" settings (lowest of 100 ms and highest of 2000 ms) to address maintenance windows. The lower number creates more opportunities for CM registration at the expense of user traffic capacity.
- ✓ A new CM will start ranging typically around 6 to 9 dBmV in 3 dB steps until the CMTS "sees" it, which is about -20 dBmV at the CMTS. Once the CMTS "sees" it, the CM will report inti(r1) as they are doing initial maintenance (IM). This is contention time and CMs will back-off when, and if, they collide. Explained below.
- ✓ **Note**: The CM should quickly go from broadcast IM to unicast station maintenance (SM) for final ranging and report init(r2). This does not have to be 3 dB steps anymore.

- **US Range & Data Backoff & Init Tech** - Used to minimize collisions in the US
  - ✓ `cable upstream x range-backoff 3 6`
    - This can be experimented with in case CMs collide at init(r1) and have to back-off. The code allows a CM to back-off randomly between 2^0 (1) to 2^3 (8) insertion intervals (above command) for first collision. Second time collision, randomly back-off between 2^3 (8) and 2^6 (64) insertion opportunities.
    - In the case of a CMTS reboot, the insertion interval would be the lowest of every 120 msec and a bunch of CMs would be in inti(r1). Assuming collisions are happening, they would back-off randomly that first time between 1-8 opportunities, this means between 1*120 and 8*120 = a back-off anywhere from 120 msec to .96 sec.
    - **Tip:** Look for CMs stuck in init(r1) as they could cause issues by "eating up" limited IM opportunities and cause Uncorr FEC counters to appear high.
  - ✓ `cable upstream x data-backoff 3 5`
    - This is for contention Request collision back-off. The good thing about faster US service flow speeds is usually after an initial contention Request, the subsequent bandwidth (BW) Requests end up being "piggybacked" within the actual data traffic and no more possibility of collisions.

Side Note: More US utilization coupled with applications not using unsolicited grant service (UGS) (Vonage, Skype, Zoom and other BE VoIP) will increase probability of Request collisions. This could also be exacerbated by DS OTT video and its TCP acks that must be sent on the US. I suspect gaming as well.

I suspect customers with audio-only will have more contention requests as video would increase the US throughput requirements and piggybacking would probably occur more often.

Warning: These collisions could lead to laser clipping and dropped packets. This is not the case for distributed access architectures (DAA) like remote-PHY since the fiber link is digital and there would be no laser clipping.

The following commands can be used to verify BW Requests whether they are contention or piggybacked. It cannot tell when contention requests actually contend/collide. The first one is intended for a specific CM.

```
cbr8#sh int cx/y/z sid n count ver | inc BW
BWReqs {Cont,Pigg,RPoll,Other} : 8306, 3243, 0, 0
```

This second command will show per US.

```
cbr8#sh contr cx/y/z up n | in Request|Bytes
Bandwidth Requests = 2776290
Piggyback Requests = 1077964
Invalid BW Requests= 195
Bytes Requested = 256264277
Bytes Granted = 1626995783
```

If for example 500 homes were in a SG/FN and 10% are doing some sort of teleconferencing and 40% of them are doing audio-only and half of them actually have collisions. This gives 500*.1*.4*.5 = 10 potential request collisions. 10*log(10) = a 10-dB potential power spike. To add power perfectly, signals need to be the same frequency, amplitude and phase. At the US laser input, signals will be the same freq and power, but phase is based on timing/distance. CMs have time offsets to keep tight timing alignment, so phase could be aligned as well.

TIP: A trait of laser clipping is "seeing" artifacts like second and third order harmonics above the diplex filter region. One way to prove a signal is an artifact is to turn off the original, "real" signal or watch a spectrogram view, which is time in the Z axis. If artifacts disappear the same time signal below 42 MHz disappears or fluctuates, then it's a high probability that it's a harmonic or by-product of inter-mixing of signals (heterodyning). Keep in mind that sometime DS signals leak on the US, so it's actually ingress and not a harmonic. Also look below 5 MHz and make sure AM or HAM radio is not getting into your node. It's been seen in the past where a node using a special port for power insertion wasn't as efficient as believed for RF choking. Installing a power inserter on an RF leg solved the issue.

Other power spikes could be CMs coming online and ranging. A CM on a low value tap will normally only need to transmit maybe 35 dBmV and if it ranges it could go as high as 57 dBmV. Utilizing flexible solution taps (FST) with built-in EQs helps alleviate this since CMs all transmit between 40-50 dBmV and will not have a large range to ramp up.

Warning: There could also be a concern with CMs in the "hotlist" as they will still range. Whether this exacerbates the issue is unknow since they never show init(r1), but they're ramping up on every UCD and trying all day long!

- ✓ `cable upstream ranging-init-technique 2`
  - This cable interface command helps US ranging for D3.0 mtc-mode (US bonding) by eliminating contention ranging on the other USs in the US BG once the first US has ranged. The default is technique 1, which means contention IM. Tech 2 is unicast, so basically SM ranging. There have been issues with some CMs with tech 2 in 3.18 code, but tech 3 or 4 could be tried as well. It also helps with RFoG systems and also for D3.0 DS load balance (LB).

- **Throttle CM Ranging**
  - ✓ `[no] cable throttle-modem init-rate <1-1000> holdoff-time <5-100> flush-rate <100-1000>`
  - ✓ Suggested values; 32 CM/s; 45 sec; 300 CM/s
  - ✓ `show cable throttle-modem`
  - ✓ `cable upstream rate-limit-bwreq exempted-priority <priority>`
- **Prioritize Pre-registration Traffic**
  - ✓ `(config)#cable qos pre-registration us-priority [0-`**7**`]`
    - Default of 0. During CM registration it first goes through init(r1), which is contention-based ranging. Once the CMTS "hears" it, it goes to init(r2), which is unicast ranging to fine tune the levels and add Pre-EQ. Once these physical layers are complete, the CM state will report int(rc) and it can now be docsis pinged. The next state is dhcp (init(d)) and the CM must now use actual data transmissions that compete with other CM transmissions. If the dhcp discover is small, the CM could use a short grant and its associated modulation, otherwise it will use a long grant with its modulation. If other CM transmissions are higher than priority 0 and lots of US utilization, then this init(d) state may never have opportunities to be fulfilled, so set it much higher than 0!
  - ✓ DS – "`cable service flow priority`" (EDCS-1524683)
  - ✓ **Note**: Setting all BE flows > priority 0 can lead to issues.
- **US Max Power Issues**
  - ✓ `cable upstream n power-adjust continue 6`
    - Helps CMs exceeding Max Tx power to stay online.
    - **Note**: A max transmit CM will be commanded to change level every 15-20 seconds during its SM, optimally only once. Some CMs have been observed to go into the fast polling mode (every second) for 5-10 times before moving on. This is a good reason to make sure < 5% of your CMs with a marking of **!** are in this state.
    - By increasing the "continue" command to 6 dB, the CM will be permitted to stay online if the CMTS receive level is between -6 dBmV and 0 dBmV. If the level is above -1, you won't see a "!". If the level is below -6 dBmV, the CM will go offline. For systems that still have high-value taps (29 & 26 dB), this helps keep the CM online, but will produce CMs with different CNRs & MERs.
    - **Warning**: Allowing a large `power-adjust continue` to be configured can lead to CMs having a large range to overcome isolation and potentially appear on US ports where they should not! It could also allow CMs located off low value taps to range very high and create intermittent laser clipping.

- **Note**: If the level of noise on the US is enough to distort the US level being received by the CMTS, then the CM and CMTS will go into "`power-adjust noise`" averaging mode. A "*" will be displayed next to the receive level in the `show cable modem` command. When this occurs, CMs are polled using a one second interval. By default, the percentage of "noisy" ranging responses that cause a CM to enter "`cable upstream n power-adjust noise`" mode is 30%. This percentage may be increased to alleviate excessive power level adjustments in the presence of noise.
- The following command can be used to identify CMs in max Tx power, max time offset and noise averaging mode; `cbr8#scm | in \*|!|MAC|State`

```
MAC Address    IP    I/F       MAC             Prim RxPwr  Timing
               Add             State           Sid  (dBmV)  Offset
38c8.5cb6.63ca -- C2/0/2/U1 online(pt)      15   !0.00   1209
38c8.5c09.42c0 -- C2/0/1/UB w-online(pt)    1    -1.00  !6104
6477.7d90.4368 -- C2/0/7/UB w-online(pt)    12   *0.50   1532
```

✓ `cable upstream max-channel-power-offset 6`

- The above command helps D3.0 & D3.1 CMs select the best US BG. When a D3.0 CM registers, it does so on a single channel, a reference channel, and relays its Tx level back to the CMTS. The CMTS can determine if that level will be adequate for multi-ch bonding. The "`power-adjust continue`" range isn't used for this determination.
- **Note**: Cisco has a feature that will drop from 4-ch to 2-ch (if configured) and finally single-ch and/or non-mtc mode. This depends on the Tx level supported plus it adds in the `max-channel-power-offset` calculations. This command has a default of 3 dB, but a value of 6 is recommended. If that level is not adequate for all options, then the CM resets itself. Example; CM ranges on US0 and reports 55 dBmV, CMTS wants to do 4-ch bonding and determines that 64-QAM for 4-ch US bonding has a max output of 51 dBmV + 3 max-ch-offset = 54, so CM drops to 2-ch BG, if configured.

✓ Stick with the double minislot from default like we suggest and never quadruple it. If so, more "time on the wire" will be wasted. Dropping it to the default minislot of 1 when using 6.4 MHz ch width will not save anything and could affect US concatenation and per-CM US speed.

- **Cable Interface Suggestions**

✓ `cable upstream balance-scheduling`

- The US scheduler tends to allocate more minislots in the first US in the US BG if not using this command. This is not "bad" but can affect D2.0 CM US load balancing. This command is not on by default, but highly suggested to implement. Another option would be to assign US0 to your highest, "best" US frequency and the last US in the BG to the lowest, "worst" US frequency.
- **Warning**: Do not use this command for RFoG (DPON) environments.

- ✓ `cable upstream qos fairness`
    - Implement the qos fairness cable interface command to help fairly share between D3.0 CMs and D3.1 so one or the other doesn't "starve out" the other. It's not on by default and we have seen D3.1 allocated more speed at the expense of D3.0 CMs. The command doesn't change the cross-bonding functionality. D3.1 CMs still prefer 3.1 spectrum before utilizing 2.0/3.0 spectrum (chs).

- **VoIP Call Signaling Insurance**
  - ✓ Utilize non real-time polling service (nRTPS) for call signaling. This allocates non-contention request opportunities to guarantee call signaling during high congestion. The beauty of nRTPS is it allows contention requests, if available, along with non-contention requests while RTPS is non-contention only. It also allows a priority to be configured for the flow associated with the nRTPS request.

- **Service Tiers**
  - ✓ When adding faster service tiers, be sure to delete the old, slower ones. Many people forget to delete obsolete tiers when they migrate to higher tiers.
  - ✓ **Warning**: The slow-to-fast ratio cannot be more than 1:1000. If it is, the slower rate can constrain the faster rate!
  - ✓ Make sure DS call signaling flows utilize an LLQ flow by making sure they use a non-zero max latency value. Then these slow flows will not affect the ratio limit.
  - ✓ Look at all the flows forwarded on a Wideband or Integrated interface and verify the highest rate and the lowest rate do not exceed a 1000:1 ratio.
    - Example; if offering a 1 Gbps speed, then the lowest offering should be 1 Mbps and higher.
    - It's also not a good idea to use a minimum guarantee rate for any flows (US or DS), but dynamic QoS flows like UGS are fine.

Going Forward and Planning for the Next Inevitable Event

- Implement a subscriber-based subscription model
  - ✓ For quick activation of more channels/capacity

- Have segmentable nodes
  - ✓ Future segmentation for quick activation

- Implement DAA
  - ✓ Better performance, complementary to D3.1, and a pathway to Cloud

- When available, implement D4.0 LLD features
  - ✓ PGS
  - ✓ Cisco DPS feature – `cab upstream dps`
    - ▪ Helps US latency in long CIN delay DAA, lowers latency for DS TCP
    - ▪ **Note**: Intel/TI Puma 5 CMs don't seem to benefit

Thoughts and More Info to Add

My top points.

1. Maybe D3.1 US would not only add more capacity to alleviate the issue in the first place, but provide more contention opportunities from the technology, ch width, etc.
2. Could SID cluster 2 make the matter worse since all US BE service flows would use 2 SIDs with they own Req/grant cycle.
   a. A 2 Mbps flow with 2 SIDs in essence would be 2, 1 Mbps flows and maybe more cont vs piggybacking (not exactly, but you get the point).
3. Need to validate/verify which apps cause more cont vs piggybacking; Skype, Zoom, Webex etc (with and without video), DS OTT video, Pandora and Spotify, gaming
   a. Would also be good to see (after collision occurs) if data back-off does anything since near real-time app, why would the CM resend, the app wouldn't. Maybe the CM resends, but it's basically wasted anyway. Engineering informed that spec states retry 16 times.

We had a mtg of the minds to discuss and brainstorm. There is no "hack" or work-around to increase the amount of minislots reserved every 2 msec. Here's an example of the US DOCSIS scheduler. It is a 300 msec scheduling wheel, but the 60 msec shown below is repeated 5 times.

```
cbr8#sh int c1/0/2 mac-scheduler 0 map
    Bucket vacancy table (slot_count, used_ms, vacancy_ms)
    (  1,155,  5) (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159)
    (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159)
    (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159)
    (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159)
    (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159)
    (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159) (  0,  1,159)
```

- That first 2 msec bucket is showing 155 minislots that cannot be scheduled as it is reserved for IM (modems to come online init(r1). It is repeated every 60 msec. I was hoping that this region could be used for cont requests also if no CMs ranging, but was told that's not the case. It's strictly for IM (and BE traffic if not used), which is why we suggest the insertion interval auto. If you hard-set the insertion interval, you would be nailing up time on the wire that is not used when all CMs are online.
- Each bucket after that is showing no scheduled flows (0) with 1 minislot reserved for the cont req like we have been talking about and 159 minislots open for scheduling. When I say scheduling, I mean UGS, nRTPS, RTPS. BE does not use this strict scheduling wheel, but it still uses the minislots, just not strictly in this scheduler that is used to control latency and jitter. BE can use the IM region if not used as well.
- Engineering was kicking around an idea of an eng build where that cont req reservation could be changed. Right now the code figures out how many minislots are needed for a req and that's what ends up in the wheel. We want the Req to be optimized (mod profile settings and minislot tick size) to keep it at 1 minislot, but it would be cool if we could change the wheel from auto to 6 or 10. Here's some quick math.
  - ✓ Typical settings: 6.4 MHz, 64-QAM for a-long, 16-QAM for REQ, 2-tick minislot. Data using a-long is 48B/minislot. Req using 16-QAM would amount to 32B/minislot. When you add 6B DOCSIS REQ + preamble and guardband, it fits in that 1 minislot. 1 tick = 6.25 usec, so a 2-tick minislot would be 12.5 usec. If we increase cont time to 6

minislots, that would be 6*12.5 = 75 usec/every 2 msec. .075/2*100 = 3.75%. Not bad if it fixes our issue. I just wouldn't get my hopes up for an eng build anytime soon.
    - ✓ **Note**: Potential bug found where Req burst using QPSK is not optimized and ends up needing 2 minislots.
- I then thought of an idea that I was excited about. I know a way to "nail-up" non-contention requests, nRTPS. If you had 500 CMs in the SG and every one of them got an nRTPS flow (more SIDs BTW) with 100 msec polling, that would be 10 minislots/sec * 48B/mini = 480*8b/B*500 CMs = 1.92 Mbps nailed up BW for non-contention requests. D3.0 CMs doing 4-ch US bonding only do that on 1 US not all 4. D2.0 could be distributed across the 4 US for example 50 50 50 50. 50*480*8 = 192 kbps on each US. IRT to D3.0, the nRTPS would distribute across USs as well, so the original math of 500*480*8 = 1.92 Mbps would be the overhead for the 108 Mbps aggregate!
    - ✓ Here's where it falls apart. nRTPS is assigned to a default single-ch BG and the subsequent traffic (BE) that goes with the nRTPS non-contention request is configured as well and can have higher priority, etc. The issues is, it is also relegated to that single US ch AFAIK. I wonder if we used LLQ if that would change anything.
        -
```
cbr8(config-if)#cab upstream 0 scheduling type nRTPS mode ?
docsis  docsis compliant mode
llq     low latency scheduling of periodic events
```

I searched for an old correspondence I had with another customer when they had US bonding partial issues and dropped secondary flows. They use nRTPS (old practice) for call signaling. They had CMs with bad code also.

- First, I suggested `cab upstream resiliency sf-move UGS, NRTPS, & RTPS` so if a D3.0 US bonded CM goes into US partial mode and the US that is hosting the UGS (call) is bad, the CMTS will move it to one of the other USs that is less congested and still active for that CM. The nRTPS will also select a different default singe-ch BG to move to.
- They did this, but the bad CMs would still not move their nRTPS flow, so if the CM registered and picked US0 for nRTPS and then it went to partial mode with US0 (dr) instead of (sta), no more calls could be made. An existing call would be fine since the UGS would move properly.
- The reason I bring this up, I suggested `cab upstream 0 scheduling type nRTPS mode llq` hoping that would allow it to just bond. We found out the scheduled flows like UGS, nRTPS, RTPS will always pick a default single-ch BG no matter if docsis or llq scheduling. DOCSIS mode just puts it in the buckets to control jitter < 2 msec. llq allows some jitter with the benefit of not hard-setting those non-contention requests.
- If you could identify apps, ports, services that create more contention requests and could assign them to an nRTPS service flow using llq mode and you were ok with it not US bonding, then that would be something to think about.

Here's the latest from one customer.

- ✓ `cable upstream x data-backoff 4 6`
- ✓ `cable sid-cluster-switching max-request 4`

CSCvu04892 opened to track efforts in optimization in the US-scheduler.

- DPS works good for Brcm-based CMs and Hitron PUMA 7 CMS, but not for PUMA 5 CMs. Unfortunately, PUMA 5 is end of life, so modem vendor won't provide a fix for it. We may need to do parameter tuning to help on this case.

Is the D3.0 `docsQosCmServiceUsStatsRqRetries` OID (CM must support it) counting only collisions and do those collision numbers look realistic for you? Are we able to check the collision counts on the cBR8? I guess not individually by modem…

Poll D2.0 CM via: `docsIfCmServiceRqRetries`

If that really is retries, then it can only be attributed to a collision if we assume a back-off had to happen. What if the App didn't require or need a retry? I suspect the CM would still resend regardless if the resent Req was even used since layer 2/3 and the App is layer 4-7. We have actual collision counters on the CMTS, correct? We wouldn't know who (CM) they came from, but we can identify "noise" during that contention time.

The code calculates how many minislots are needed for a Cont Req and that's the reserved amount per 2 msec bucket (map). You can confirm with this command:

```
cbr8#sh int c1/0/2 mac-scheduler 0 map-stats | in Slots

     ReqSlotSz:1  ReqSlotChunkSz:4  ImSlotSz:150   SmSlotsz:6
```

If they need 2, then their mod profile for the Req burst (IUC 1) is not optimized and probably making things worse. If using QPSK, that would explain it.

I just had another thought. Here's an example with some assumptions.

- If the app(s) sends every 20 msec (req/grant) like VoIP and all contention requests.
- 100 customers on at same time in SG.
- 80% utilized of 160 minislots per 2 msec bucket(map) = 32 minislots available for contention requests, but because using QPSK for IUC 1, each Req = 2 minislots leading to 16 customers can send.
- Assuming 10 get through and the other 6 have collisions (besides power addition) they must back off between 2^0 (1) and 2^3(8), I assume 2 msec maps. Leading to a back-off anywhere from 2 msec (next map) to 16 msec.
- Since DOCSIS doesn't allow outstanding requests, what happens if the app has the next frame to send and the Req that backed off was for info that is old? Whose job is to flush old info.
- Also, if the CMTS honors the Req, but the CM/app decided it's too late and drops it, does the CMTS count that as "used" time? Almost like UGS that is nailed up or scheduled.
- Could this explain what we are seeing where utilized is extremely high, but actual traffic doesn't add up. It could be manifesting from these apps and like a snowball effect.
- It's fine at first, but when utilization gets high and lots of contention requests that back-off and then not needed lead to even worse circumstances.

Maybe I'm way off.

- I believe data backoff isn't doing much because of the types of applications that are generating the US traffic.

- OTT video is TCP based causing many acks. This may or may not be causing Cont REQs, but depends on actual US traffic rate and buffering to create piggybacking requests. A collision here would/could generate a resend and data back-off may help.
  - ✓ BTW, Sirius/XM radio is UDP, but Pandora and Spotify are TCP based, I believe.
- Zoom, Skype, webex gotomtg, etc are BE traffic and not using UGS via the eMTA. The fact that many people are home using these apps all day creates lots of US requests (since not UGS). I suspect customers doing audio-only are creating contention requests and no piggybacking or limited piggybacking.
  - ✓ Because of the nature of "real-time" audio, there would be no reason to resend a request that is dropped, so no data back-off. Even if there is, the next Req that gets through would be useless, time has moved on.
- This should not be hard to test and prove. Track IUC 1 (request bursts) and SID cont vs piggyback Req counters while doing controlled apps.
- To make matters worse, customers in the US are seeing analog laser clipping from what we believe to be power addition from all these collisions.
  - ✓ Good selling point for DAA and R-PHY as digital fiber does not "clip".
- I have another trick up my sleeve that would be cool to prove and another selling point. Will D3.1 US not exhibit this issue as bad since the contention time may have more opportunities vs SC-QAM?
- I was even trying to think of a way to use nRTPS to "nail" non-contention requests, but it would create single-ch flows and not bond anyway.
- I would want to keep the Req at 1 minislot and increase the amount of cont minislots available every 2 msec like you mentioned.
- Typical settings: 6.4 MHz, 64-QAM for a-long, 16-QAM for REQ, 2-tick minislot.
- Data using a-long is 48B/minislot
- Req using 16-QAM would amount to 32B/minislot. When you add 6B DOCSIS REQ + preamble and guardband, it fits in that 1 minislot.
- 1 tick = 6.25 usec, so a 2-tick minislot would be 12.5 usec. If we increase cont time to 6 minislots, that would be 6*12.5 = 75 usec/every 2 msec. .075/2*100 = 3.75%. Not bad if it fixes our issue.
- This is just for experiment to check if increasing BWREQ minslots will help. In a channel with high utilization it will produce the similar MAP message to guarantee at least 6 minislots of BWREQ every 2msec instead of 1 minislot. The following is the alternative code change to achieve the goal of "I would want to keep the Req at 1 minislot and increase the amount of cont minislots available every 2 msec like you mentioned."

In function **mapmaker_serve_fcms** ()

```
map_ievarp =  UBR_MAP_GET_IEVARS(mapvarp->pch, mapvarp->lch);

uint16_t bwreq_mslots = 6 * map_ievarp->req_mslots;
if ((!scqam_tafdm &&
     mapmaker_no_im_jitter (mapvarp, bwreq_mslots)) ||
    (scqam_tafdm &&
     mapvarp->minislots_available >= bwreq_mslots)){
     mapmaker_srv_grant_slot(map_pchvarp, mapvarp,
                             BCAST_SID,
```

```
                                            REQ_IUC,
                                            bwreq_mslots,
                                            &ie_desc);
                mapvarp->ci_fcms += bwreq_mslots;
        }
```

Customer Questions:

1. Did you add more capacity to alleviate it?
2. Did you activate D3.1?
3. Did you change padding or levels into laser and/or CMTS config?

- We are experimenting with customers by changing data back-off from 3 5 to 5 6 to see if more random values for the first back-off will help. My thoughts and concerns are:
    1. High utilization leading to only one or 2 cont req opportunities every 2 msec map
    2. The type of apps with mostly cont req and no piggybacking (small somewhat infrequent frames)
    3. Lots of flows simultaneously
    4. No amount of back-off will make a difference since the collisions just keep happening regardless
- We are also troubleshooting an issue where some CMs seem to get more then they need. One of the engineers told me about this command:
    - ✓ `config)#cab us-scheduler bwr-drop 20 100`
    - ✓ It's not on by default, so maybe it can be experimented with. We are seeing in a few markets where the minislot allocation (utilization) is extremely high and the actual traffic doesn't add up. This may be a bad CM, a DoS attack, etc. Either way, I'm sure this is exacerbating the issue since it is making the US fully utilized leading to less cont time. I don't have a lot of info on this yet.
- I also found out that the default mod profile of 221 using QPSK for the req burst does not seem optimized and forcing 2 minislots instead of 1. 16-QAM is fine. I had to drop the guardtime from 22 sym to 18 to fit it back into 1 minislot and preamble changes had no effect. This may be another data point to see if a QPSK Req is better suited. Using ATDMA and QPSK will change the preamble to QPSK0, which is lower power than QPSK1 preamble pattern and it will be longer in time than 16-QAM. Just brainstorming.

```
Slot-6-0#test cable stats pull-map-bldr <md> <us-ch> 0      <<< We can focus on
single US channel, repeat command several times (also collect mac-scheduler command
at similar time)

sw_stats_map_bldr_test_req_sender: got map-builder stats
      map_stats_md_idx = 0
      map_stats_md_lch_0_idx = 1
      req_num_slots = 2349590582
```

And info below:

```
S007#sh cab card 1/0 us-phy scqam-channel ?
fec-summary  Show all channels FEC errors
iuc-stats    Show per IUC counters
map-stats    Show MAP counters/stats

sh cab card 1/0 us-phy scqam-channel iuc-stats 4 0 0 ?
```

```
all   All IUCs

show cable card 2/0 us-phy channel 1 1 0 get
Channel Counters for Mg1, 1/0:
+-------+----------+-------+----------+-------+-------+----------+----------+----------+-----+
|IUC    |  Grants  |Collide|  No      |Phy    |No     |    Good  | Corrected|Uncorrectd| SNR |
|       |          |       |  Energy  |Errors |Preambl|    FEC   | FEC      |FEC       |     |
+-------+----------+-------+----------+-------+-------+----------+----------+----------+-----+
|1-Req  |1715228290|10994  |1499703919|0      |4144011|0         |0         |0         |0 . 0|
```

This field probably indicates contention requests since piggybacked requests wouldn't use IUC 1 and would be embedded with real date using IUC 10 most likely. IUC 1 is used for contention requests but could be "nailed up" as well if they have nRTPS /RTPS US service flows typically used for VoIP call signaling.

Maybe we should look at:

```
cbr8#sh interfaces cable 1/0/2 upstream 0
MAC domain upstream impairment report: 0x0
Cable1/0/2: Upstream 0  is up
     Description: UC1/0/2:U0
     Received 30 broadcasts, 0 multicasts, 635430 unicasts
     1724 discards, 1948 errors, 0 unknown protocol
     635460 packets input
     Codewords: 16295259 good 1079356 corrected 13353 uncorrectable
     0 noise, 0 microreflections
     Total NON-MTC Modems On This Upstream Channel : 0 (0 active)
     Total MTC Modems On This Upstream Channel : 6 (6 active)
     Segments: 7035239 valid, 1724 discarded

cbr8#sh int integrated-Cable 1/0/2:0
Integrated-Cable1/0/2:0 is up, line protocol is up
  Hardware is CMTS IC interface, address is 00a2.89d1.eceb (bia 00a2.89d1.eceb)
  MTU 1500 bytes, BW 37500 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation MCNS, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1887131 packets output, 186737206 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

I have this written down in an old doc:
errors: The total of a whole range of errors, many of which don't matter. The errors that this value adds up are

Upstream errored frames that for some reason aren't classified as HCS or CRC errored.
Upstream frames with HCS problems.
Upstream frames with CRC errors.
Uncorrectable codewords received.
Collisions in the bandwidth request IUC.

```
cbr8#sh int c1/0/2 sid 7 count ver
Sid                             : 7
Request polls issued            : 0
BWReqs {Cont,Pigg,RPoll,Other} : 8306, 3243, 0, 0
Grants issued                   : 11550
Packets received                : 22702
Bytes received                  : 21878473
Queue-indicator bit statistics : 0 set, 0 granted
Total Codewords rx              : 116740
Good Codewords rx               : 108610
Corrected Codewords rx          : 8118
Uncorrectable Codewords rx      : 12
Concatenated headers received  : 0
Fragmentation headers received : 0
Fragmentation headers discarded: 0
```

I like this command for per-CM cont vs piggyback as we could use it to test the theory of which applications create more contention requests.

Also

```
cbr8#sh contr c1/0/2 up 0
Controller 1/0/2 upstream 0  AdminState:UP OpState: UP
  atdma mode enabled
  Frequency 10.000 MHz, Channel Width 6.400 MHz, Symbol Rate 5.120 Msps
  Modulation Profile Group 224
  Modulations (64-QAM) - A-short 64-QAM, A-long 64-QAM, A-ugs 64-QAM
  Mapped to connector 2 and receiver 0
  Bind to Cable1/0/2 US0
  US phy MER(SNR)_estimate for good packets - 38.1620 dB
  Spectrum Group is overridden
  Nominal Input Power Level 0 dBmV

  PHY Dev status: UP
  PHY: us errors 0  us recoveries 0 (enp 0)
  PHY: TSS late 0  discontinuous 0
  PHY: TSS mis-match 0  not-aligned 0
  PHY: TSS missed snapshots from phy 612

  UCD LCH state: RUN_STEADY
  UCD change count = 8
  UCD Tx Counts = 17197380
  UCD Internal Tx Counts = 0
```

```
  Bandwidth Requests = 2776290
  Piggyback Requests = 1077964
  Invalid BW Requests= 195
  Bytes Requested = 256264277
  Bytes Granted = 1626995783
  Ranging Insertion Interval automatic (120 ms)
```

```
CHNDCAPC09#show controllers c9/0/6 up 0 | i Bandwidth.Requests|Invalid.BW
  Bandwidth Requests = 4133066821
  Invalid BW Requests= 1155244    <<< 0.028%, highest of the group
```

I also wrote this to Cox who are tracking this path.

- Monitor Cont %
  - `CMP1CMTS007#show int c1/0/5 mac-scheduler | i Avg`

```
Avg upstream channel utilization(%data grants) : 50%
Avg upstream channel utilization in 30 sec : 45%
Avg percent contention slots : 49%
Avg percent initial ranging slots : 1%
Avg percent minislots lost on late MAPs : 0%
Avg percent guardband slots : 0%
Avg upstream channel utilization(%data grants) : 41%
Avg upstream channel utilization in 30 sec : 46%
Avg percent contention slots : 58%
Avg percent initial ranging slots : 1%
Avg percent minislots lost on late MAPs : 0%
Avg percent guardband slots : 0%
```

a. We don't show scheduled traffic here and must use the admission control command to get UGS/nRTPS/RTPS utilization.

- `cbr8#sh cab admission-control interface c1/0/2 upstream 0`

```
   Interface  Cable1/0/2
   Upstream # 0
   Upstream Bit Rate (bits per second) = 30720000
     Sched Table Rsv-state: Grants 0, Reqpolls 0
     Sched Table Adm-state: Grants 0, Reqpolls 0, Util 0%
     UGS    : 0 SIDs, Reservation-level in bps 0
     UGS-AD : 0 SIDs, Reservation-level in bps 0
     RTPS   : 0 SIDs, Reservation-level in bps 0
     NRTPS  : 0 SIDs, Reservation-level in bps 0
     BE     : 22 SIDs, Reservation-level in bps 0
   Maximum AC reservable bandwidth is not configured
```

b. If (`data_grant+cont+sched`) doesn't add up to 99%, then track specific CMs and figure out why.

c. If `cont` low and **does** add up properly, then not much you can do since heavy US traffic is "real".

d. Work-around to laser clipping is install RPD 😊. Seriously though, setting 3-6 dB lower into laser (with expected lower MER) may be an option as well. Can be done with pad movement and/or CMTS config.

```
CMP1CMTS007#show controller integrated-Cable 1/0/5 cou rf-channel
```

```
Controller  RF    MPEG        MPEG         MPEG    Sync        MAP/UCD     User     QAM
            Chan  Packets     bps          Mbps    Packets     Packets     Mbps     Util
                  Tx                               Tx          Tx                   Percentage
1/0/5       0     24304176125 39481867     39.48    188178648   3985520830 33.83    77.03
1/0/5       1     24258099426 39470436     39.47    188178648   3985499093 33.84    77.00
```

More US utilization along with Vonage, Skype, Zoom and other BE VoIP applications (not UGS) will increase probability of Req collisions. These collisions could lead to laser clipping (not for R-PHY).

I ran some numbers to see if the modulation scheme used for the Req IUC along with the minislot size will dictate the efficient usage of BW. Experimenting with minislot 1 instead of 2 with 6.4 MHz ch width is assessed below.

```
cbr8#sh contr upstream-Cable 1/0/2 u 0
USPHY OFDMA support: FULL
Controller 1/0/2 upstream 0  AdminState:UP OpState: UP
  atdma mode enabled
  Frequency 10.000 MHz, Channel Width 6.400 MHz, Symbol Rate 5.120 Msps
  Modulation Profile Group 224
  Modulations (64-QAM) - A-short 64-QAM, A-long 64-QAM, A-ugs 64-QAM
  Mapped to connector 2 and receiver 0
  Bind to Cable1/0/2 US0
  US phy MER(SNR)_estimate for good packets - 36.7020 dB
  Spectrum Group is overridden
  Nominal Input Power Level 0 dBmV
  part_id=0x3142, rev_id=0xC0, rev2_id=0x00
  Range Load Reg Size=0x2C
  Request Load Reg Size=0x07
  Minislot Size in number of Timebase Ticks is = 2
  Minislot Size in Symbols = 64
  Minislot Size in Bytes = 48
cbr8#sh cab  modulation-profile upstream-Cable 1/0/2 u 0
Mod  Docsis-Mode  IUC      Type  Pre Diff FEC  FEC  Scrmb  Max Guard Last Scrmb Pre    Pre   RS
                                 len enco T    k    seed   B   time  CW         offst  Type
                                          BYTE BYTE        siz size  short
224  atdma        request 16qam 32  no   0x0  0x10 0x152  0   22    no   yes   0      qpsk1 no
224  atdma        initial 16qam 384 no   0x5  0x22 0x152  0   48    no   yes   0      qpsk1 no
224  atdma        station 16qam 384 no   0x5  0x22 0x152  0   48    no   yes   0      qpsk1 no
224  atdma        a-short 64qam 64  no   0x6  0x4C 0x152  6   22    yes  yes   0      qpsk1 no
224  atdma        a-long  64qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0      qpsk1 no
224  atdma        a-ugs   64qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0      qpsk1 no
```

Because the Req Burst uses 16-QAM, the B/minislot would be 4/6*48 = 32 B.

The actual Req would be 6B DOCSIS + 32 bit preamble/8b/B, but always a QPSK pattern, so *2 (QPSK is double time vs 16-QAM) + 22 sym of guardtime where 16-QAM is 4b/sym, so 22*4/8 = 11B for a grand total of 6+8+11 = 25 B and would fit into that 1, 32B minislot. So, dropping to 1 tick probably won't save us anything since it would be a 16B minislot using 16-QAM and just end up using 2 minislots anyway. You could argue that 25B gives us 7 extra bytes to use before we fill that minislot, but the Req has no FEC T bytes and all we could really do is make the preamble and/or guardtime bigger.

The question would be, what would happen if using QPSK for the Req Burst? A 2-tick minislot would be 48B for 64-QAM, but 48/3 = 16B for QPSK.

```
cbr8#sh cab modu 221
Mod  Docsis-Mode  IUC      Type  Pre Diff FEC  FEC  Scrmb  Max Guard Last Scrmb Pre    Pre   RS
```

```
                              len enco T    k     seed   B   time  CW          offst Type
                                       BYTE BYTE         siz size  short
221  atdma        request qpsk  36  no  0x0  0x10 0x152  0   22    no   yes   0     qpsk0 no
```

Every Req would be 6+36/8+22*2/8 = 6+4.5+5.5 = 16. = 1 minislot. nRTPS call signaling uses this as well.

I can even check it in code to make sure the settings are optimized with:

```
cbr8#sh int c1/0/2 mac-scheduler 0 map-stats | in Slots
     ReqSlotSz:1  ReqSlotChunkSz:4  ImSlotSz:150   SmSlotsz:6
```

Warning: Running new code (16.12.1y) I'm seeing more preamble (maybe 40 bits) than what is configured and I needed to drop the guardtime from 22 sym to 18 to get it back to 1 minislot.

So, moral of the story, stick with the double minislot like we suggest and never quadruple it or more time on the wire will be wasted. Dropping to the default minislot of 1 when using 6.4 MHz ch width will not save us anything and could affect US concatenation and per-CM US speed.

If the issue were a req/grant issue, then we could add more safety into the map advance. I have been suggesting, make sure map advance is not too aggressive and >= 2500.

```
Sh controller cx/y/x upstream | in Dyn.
```

As long as this Is >=2500, there should be no reason to change the "safety" on the map advance. You could lower the safety if it's much higher than 2500 map advance. Map advance config is under the cable interface, so the safety will affect all the USs in that mac domain. I know DS laser clipping and dropped maps will affect US UGS, but US issues will also obviously drop UGS. Sh cab hop cx/y/z threshold (history).

We have come to the conclusion a long time ago that some USs after the initial US ranges seem to report (cosmetically) double, but it usually doesn't affect anything. Look at scm "mac" ver. The US with the same initial and current time offsets is usually the first US ranged and the others may have this cosmetic issue. BTW, you can use ping docsis "mac" ver and it will ping the US that was ranged first. We have a whole paper on map advance, time offsets and one on CIN design for R-PHY.
http://stugots.cisco.com/rr/BNE-KnowledgeBase/Articles/Map_Advance&Time_Offsets_8-2-11.doc
http://stugots.cisco.com/rr/BNE-KnowledgeBase/Remote_Phy/DOCSIS_Timing_In_CIN_1-2020.docx

Entire suggested/recommended config and the customer can decide what not to implement.

```
cab modulation-profile 222 atdma request 0 16 0 22 qpsk scrambler 152 no-diff 32 fixed qpsk0 1 2048
cab modulation-profile 222 atdma initial 5 34 0 48 qpsk scrambler 152 no-diff 384 fixed qpsk0 1 2048
cab modulation-profile 222 atdma station 5 34 0 48 qpsk scrambler 152 no-diff 384 fixed qpsk0 1 2048
cab modulation-profile 222 atdma a-short 3 76 12 22 qpsk scrambler 152 no-diff 64 shortened qpsk0 1 2048
cab modulation-profile 222 atdma a-long 9 232 0 22 qpsk scrambler 152 no-diff 64 shortened qpsk0 1 2048
cab modulation-profile 222 atdma a-ugs 9 232 0 22 qpsk scrambler 152 no-diff 64 shortened qpsk0 1 2048

cab modulation-profile 223 atdma request 0 16 0 22 16qam scrambler 152 no-diff 32 fixed qpsk1 1 2048
cab modulation-profile 223 atdma initial 5 34 0 48 16qam scrambler 152 no-diff 384 fixed qpsk1 1 2048
cab modulation-profile 223 atdma station 5 34 0 48 16qam scrambler 152 no-diff 384 fixed qpsk1 1 2048
cab modulation-profile 223 atdma a-short 4 76 7 22 16qam scrambler 152 no-diff 64 shortened qpsk1 1 2048
cab modulation-profile 223 atdma a-long 9 232 0 22 16qam scrambler 152 no-diff 64 shortened qpsk1 1 2048
cab modulation-profile 223 atdma a-ugs 9 232 0 22 16qam scrambler 152 no-diff 64 shortened qpsk1 1 2048

cab modulation-profile 224 atdma request 0 16 0 22 16qam scrambler 152 no-diff 32 fixed qpsk1 1 2048
cab modulation-profile 224 atdma initial 5 34 0 48 16qam scrambler 152 no-diff 384 fixed qpsk1 1 2048
cab modulation-profile 224 atdma station 5 34 0 48 16qam scrambler 152 no-diff 384 fixed qpsk1 1 2048
```

```
cab modulation-profile 224 atdma a-short 6 76 6 22 64qam scrambler 152 no-diff 64 shortened qpsk1 1 2048
cab modulation-profile 224 atdma a-long 9 232 0 22 64qam scrambler 152 no-diff 64 shortened qpsk1 1 2048
cab modulation-profile 224 atdma a-ugs 9 232 0 22 64qam scrambler 152 no-diff 64 shortened qpsk1 1 2048
```

```
controller Upstream-Cable 2/0/0
us-channel 0 frequency 16000000
us-ch 0 channel-width 6400000 6400000
us-channel 0 threshold snr-profiles 24 19
us-channel 0 threshold corr-fec 0
us-channel 0 threshold hysteresis 4
```

! Suggested configs for dynamic modulation.

```
us-channel 0 docsis-mode atdma
us-channel 0 minislot-size 2
```

! We suggest double from default as it affects amount of concatenation and per-CM US speed. Don't quadruple it though as it will make it less efficient.

```
us-channel 0 modulation-prof 224 223 222
```

! I like to promote 3-level dynamic modulation.

```
us-channel 0 equalization-coefficient
no us-channel 0 shutdown
```

```
interface Cable2/0/0
load-interval 30
```

! Default is 300, but used to affect utilization reporting for US LB. Newer code has its own 30 sec avg.

```
down Integrated-Cable 2/0/0 rf-ch 0
down Integrated-Cable 2/0/0 rf-ch 4
```

! Would like to see customers start dropping some DSs as primary. Use D3.1 OFDM DS as primary if "legacy" partial mode used, otherwise I suggest RBGs & OFDM as secondary-only.

```
up 0 Upstream-Cable 2/0/0 us-channel 0
up 1 Upstream-Cable 2/0/0 us-channel 1
up 2 Upstream-Cable 2/0/0 us-channel 2
up 3 Upstream-Cable 2/0/0 us-channel 3
cab up 0 power-adjust continue 6
cab up 1 power-adjust continue 6
cab up 2 power-adjust continue 6
cab up 3 power-adjust continue 6
```

! Helps D1.1/2.0 CMs register with US Tx power level issues and also referenced for all CMs during normal keep-alive/SM polling. Default is 4 dB.

```
cab up balance-scheduling
```

! Allow better distribution of minislot allocation across all chs in the US BG; without the command (default) the first US (like US0) will appear more utilized.

```
cab up ranging-init-technique 2
```

! Default is 1, but a setting of 2-4 could help alleviate contention ranging for D3.0 CMs and also creates faster layer 3 traffic restart when doing DS LB between D3.0 BGs.

```
cab up max-channel-power-offset 6
```

! Helps D3.0/3.1 CMs use biggest US BG when they register and have US Tx max power issues. Default is 3 dB.

```
cab up ranging-poll t4-multiplier 2
```

! My personal suggestion to get faster updates of CM US MER, levels, pre-eq and time offsets. If doing 4-ch US bonding, the default T4 multiplier is 0, which means auto which means 4 in this case leading to SM updates for each US at 20*4 = 80 sec.

```
cab upstream resiliency sf-move RTPS
cab up resiliency sf-move NRTPS
cab up resiliency sf-move UGS
```

! Highly suggested to make sure secondary service flows retained when CM goes into US bonding partial mode.

```
cab up resiliency data-burst snr 24 ufec 1 cfec 0 hysteresis 4
```

! US partial mode based on user-configurable thresholds and data burst MER vs legacy partial mode feature based on SM burst and not configurable.

```
cab upstream bonding-group 2004
  upstream 0
  upstream 1
  upstream 2
  upstream 3
  attributes 80000000
cab upstream bonding-group 2002
  upstream 1
  upstream 2
```

! I personally think 1, 2-ch BG embedded in the 4 is enough and I selected the 2 middle/best chs for freq considerations.

```
  attributes 80000000
cab bundle 1
cab map-advance dynamic 800 600
```

! Dependent on actual analog fiber distance. Presos on CIN suggestions and map advance on bne web page.

```
cab sid-cluster-group num-of-clust 2
```

! We may need to step this back to default of dynamic in case of SID depletion, but different topic and slides on bne web page under misc section.

```
cab sid-cluster-switching max-req 2
cab cm-status enable 3 9-10 16 20-24
```

! If you want battery and/or energy management features enabled

```
cab reduction-mode mta-battery enable
cab reduction-mode energy-manage enable
cable upstream qos fairness
```

! Highly recommend if D3.1 US and cross-bonding OFDMA with SC-QAM to fairly share with D3.0 CMs.

We also allocate time for ingress cancellation, but that would be miniscule. 10/sec*1000 sym = 10,000 sym/sec and you're running a 5.12 Msym/sec ch width (6.4), which means only .01/5.12*100 = .2% overhead.

2 tick minislot = 12.5 usec. 160 minislot map/bucket = 160*12.5 = 2 msec. I know our scheduling wheel is shown in 150 msec and was told it's actually a 300 msec scheduling wheel, but can't see how

that would affect it unless we have to keep the 60 msec IM lined up or get back to 1 sec, but that doesn't seem to add up either.

I also forgot about how we time-align the initial maintenance slots for all the USs in a mac domain.

b. This means more USs = more slop to align. Not sure how much that really is though.
c. Would be cool if we just did this for the FN and not the mac domain, but then we would have to broadcast UCDs based on FN as well.
d. We can split primaries across 2 FNs if we make 2 mac domains.
e. We even have the capability (or used to) to assign specific USs to specific DSs. You could keep the one mac domain and just assign USs to certain DSs to control UCD broadcast. Might not solve the IM time alignment though and still 1 mac domain, SID space exhaustion, ….
f. So, best solution could be; less USs in mac domain and mac domains as 1:1 instead of 1:2 (assuming code supports more SGs!)

Update some new test results, this is another testbed which has 120 active modems on 1 UBG (4 channels), Only 6 modems transmit traffic, no ds load:

Upstream load	Avg Latency (us)	Min Latency (us)	Max Latency (us)	Avg Jitter (us)	Max Jitter (us)
50%	20,339.62	7,062	52,636.96	767.42	15,240.24
80%	16,236.86	6,719.54	38,128.80	488.13	11,811.52
100%	14,614.51	6,284.60	35,035.44	394.79	10,768.44

All 120 modems transmit traffic, no ds load:

Upstream load	Avg Latency (us)	Min Latency (us)	Max Latency (us)	Avg Jitter (us)	Max Jitter (us)
50%	241,838.93	22,565.76	847,357.44	11,436.83	144,917.12
80%	156,621.66	15,430.68	563,484.16	7,577.77	106,176.64
100%	128,687.29	12,254.16	420,446.72	6,005.74	129,662.72

```
CBR8-PingJiang#scm c0c6.872f.a478
MAC Address    IP Address      I/F           MAC                 Prim  RxPwr  Timing Num I
                                             State               Sid   (dBmV) Offset CPE P
c0c6.872f.a478 112.59.129.109  C2/0/0/UB     w-online(pt)        2056  1.00   1817   1   N

CBR8-PingJiang#show interfaces cable 2/0/0 mac-scheduler | inc uti
     Avg upstream channel utilization(%data grants) : 55%
     Avg upstream channel utilization in 30 sec : 47%
     Avg upstream channel utilization(%data grants) : 36%
     Avg upstream channel utilization in 30 sec : 46%
     Avg upstream channel utilization(%data grants) : 39%
     Avg upstream channel utilization in 30 sec : 44%
     Avg upstream channel utilization(%data grants) : 46%
     Avg upstream channel utilization in 30 sec : 44%

CBR8-PingJiang#show cable card 2/0 us-phy channel 1 1 0 get
Channel Counters for Mg1, 1/0:
```

IUC	Grants	Collide	No Energy	Phy Errors	No Preambl	Good FEC	Corrected FEC	Uncorrectd FEC	SNR
1-Req	**1715228290**	**10994**	**1499703919**	**0**	**4144011**	**0**	**0**	**0**	**0 . 0**
2-ReqD	0	0	0	0	0	0	0	0	0 . 0
3-Init	643206	0	640932	0	0	1460	5	809	35.44
4-Maint	295501	0	408	0	0	295093	0	0	38.93
5-Short	0	0	0	0	0	0	0	0	0 . 0
6-Long	0	0	0	0	0	0	0	0	0 . 0
9-AShrt	29713597	0	19	0	0	29713578	0	0	38.27
10-ALng	295294677	0	168	0	0	935721016	0	0	38.40
11-AUGS	0	0	0	0	0	0	0	0	0 . 0

***NOTE:** All MIB IUC counters are accumulated periodically by SW.

CBR8-PingJiang#show cable card 2/0 us-phy channel 1 1 0 get

Channel Counters for Mg1, 1/0:

IUC	Grants	Collide	No Energy	Phy Errors	No Preambl	Good FEC	Corrected FEC	Uncorrectd FEC	SNR
1-Req	**1719113399**	**10996**	**1503007585**	**0**	**4151411**	**0**	**0**	**0**	**0 . 0**
2-ReqD	0	0	0	0	0	0	0	0	0 . 0
3-Init	644313	0	642039	0	0	1460	5	809	35.45
4-Maint	295938	0	408	0	0	295530	0	0	38.93
5-Short	0	0	0	0	0	0	0	0	0 . 0
6-Long	0	0	0	0	0	0	0	0	0 . 0
9-AShrt	29790048	0	19	0	0	29790029	0	0	38.27
10-ALng	295804113	0	168	0	0	936794333	0	0	38.40
11-AUGS	0	0	0	0	0	0	0	0	0 . 0

***NOTE:** All MIB IUC counters are accumulated periodically by SW.