SCTE SoCal - Understanding CMTS/CM Contention Time & Req/Grant Cycle

Feb 16, 2022

John J. Downey
Sr. CMTS Technical Leader
Cisco Systems

Things that Affect CM Registration & BW Req Contention

1. Prioritize Pre-Registration Traffic
   - `(config)#cable qos pre-registration us-priority [0-7]`
   - **Note**: Setting all BE flows > priority 0 can lead to issues (US & DS)
2. Limit Contention Ranging for D3.0 CMs
   - `cab up ranging-init-technique 2` (default is 1, so try 2, 3 or 4)
   - Re-acquires Layer 3 faster after D3.0 DS LB & great for RFoG as well
3. Adjust CM Insertion Interval for more CM Ranging Opportunities
   - `cab insertion-interval auto 120 1000` or (`60 480` uBR10K defaults)
4. Minimize US Collisions w/ Range & Data Back-off Changes
   - `cab up x range-backoff 3 6` (try a 4 6) – CM registration
   - `cab up x data-backoff 3 5` (try a 4 6) – BW requests
   - Congestion exacerbates the issues
5. 17.3.1 Adds Configurable Guaranteed Contention Time - Under Cable Interface
   - `(config-if)#cable upstream min-bwreq-ops scqam x y`
   - 5% suggested with 10% if DPS enabled

Contention Time - Initial Maintenance (IM)

- `Init(R1)` for CM IM ranging every 60 ms to 1 sec depending on traffic
  - Uses IM burst (IUC3) at QPSK or 16-QAM
  - Collisions can have Uncorr FEC
  - `Range backoff` & `insertion interval` affect total opportunities
- `Init(R2)` is unicast, "keepalive", station maintenance; **not contention**
  - Uses SM burst (IUC4) at QPSK or 16-QAM
  - **Note**: SM burst also affects:
    - ✓ US MER, Levels & Pre-EQ
    - ✓ Flaplist
    - ✓ Ping Docsis
    - ✓ US Bonding Partial Mode

Typical Show Cable US Modulation Profile Output

		Type	Pre Len	FEC T-B	FEC k-B	Max B	Grd time	Last CW	S	Pre Type	RS
IUC1	req	16qam	36	0	10	0	22	no		qpsk1	no
IUC3	init	16qam	384	5	22	0	48	no		qpsk1	no
IUC4	station	16qam	384	5	22	0	48	no		qpsk1	no
IUC9	a-short	64qam	64	6	4c	6	22	yes		qpsk1	no
IUC10	a-long	64qam	64	9	e8	0	22	yes		qpsk1	no
IUC11	a-ugs	64qam	64	9	e8	0	22	yes		qpsk1	no

- First 3 bursts must be QPSK or 16-QAM
  - Suggestion has been 16-QAM when actual modulation is 16-QAM or higher for data
- Most traffic uses IUC10 because of concatenation and IUC11 for VoIP
- IUC9 mostly for small ping, CM registration like `init(d)`, BE VoIP (Vonage, Skype, Zoom?)
  - Actual usage is based on Max burst; 6 shown above means 6 minislots*48B/mini = 288B including all overhead
- Preamble is always QPSK pattern and length is in bits
- FEC T bytes adds overhead to codeword listed as k in Hex bytes = (2*T/(2*T+k)
- Gaurdtime is in Symbols and calculated based on modulation
- RS = Reed-Solomon dynamic US interleaving if activated

Contention Time - Contention BW Requests

- US BW Requests use IUC1
- Depending on Req burst mod profile settings & minislot size, this could be 1 minislot, 2, or 3
  - No FEC T bytes, so **no** Uncorr FEC when collisions occur
- During congestion, only 1 guaranteed Cont Req opportunity every 2 ms
  - `Data-backoff` affects CM random BW Request backoffs
- Piggyback Requests make things better since no collisions
  - Have seem some CMs with less piggybacking to bypass serialization time and achieve better per-CM US speed, but falls apart when US congested
- Applications that require many Cont Requests will exacerbate the issue
  - BE VoIP (Vonage, Skype, Zoom, Webex, mobile WiFi…..)
- **Note**: Many collisions can cause peak overload & lead to US analog laser clipping

Random Delay in Bandwidth Request

- CPE behind CM transmits traffic
- CM classifies traffic and looks to make BW Req if BE flow
- CM waits for random number of contention slots to pass to avoid collisions with other CMs before sending Req
  - Each US configured on CMTS with start and end data-backoff window exponent values that can range from 0 to 15 (defaults start: 3 and end: 5)
  - Data backoff window for each US begins at $2^{start} - 1$ ($2^3 - 1 = 7$ default)
  - CM bonding multiple US will sum all data backoff windows together and then select random number between 0 and this value (if bonding 4 US all using default start value of 3, then CM will select random value between 0 – 28)
  - CM will count contention opportunities on all US as they pass and wait until random number has passed before sending request on next one
- Req further delayed by US propagation & serialization before reaching CMTS

Additional Bandwidth Request Delay

- CM expects to see data grant or data grant pending (US busy) in future MAP on any of the bonded US chs (not just on US ch request was made)
- CM assumes request experienced collision if neither grant or grant pending is seen within period of time (ACK time)
- CM then increments data backoff window exponent value by 1 on each US (or keeps at end data backoff value if already reached)
- If started with default of 3, new data backoff window is now $2^4 - 1$ one each US (15)
- CM sums databack off window for all bonded US and picks random value between 0 and this value (0 - 60 if doing 4 US using this example)
- Process continues until CM receives data grant or data grant pending in MAP or until a maximum of 16 consecutive contention retries then drops data
- Highly desirable to get request in 1-2 MAPs with no collisions
- **Note**: Piggyback Reqs do not have this random delay component

Verification Commands

- BW Req size
  - `cmts#sh int cx/0/z mac-scheduler 0 map-stats | in Slots`
    `ReqSlotSz:1  ReqSlotChunkSz:4  ImSlotSz:150  SmSlotsz:6`
- If it says 2 or more, then mod profile for Req burst (IUC1) not optimized
  - If using QPSK, that would explain it
  - US service flows nRTPS & RTPS “nail-up” non-cont BW Requests, so if Req not optimized, very bad!
- Following commands can be used to verify BW Reqs whether contention or piggybacked
  - **Note**: It cannot tell when Cont Reqs actually contend/collide
- This command intended for specific CM
  - `sh int cx/0/z sid n count ver | inc BW`
    `BWReqs {Cont,Pigg,RPoll,Other} : 8306, 3243, 0, 0`
- This command will show per US
  - `sh contr cx/0/z up n | in Request|Bytes`
    `Bandwidth Requests = 2776290`
    `Piggyback Requests = 1077964`

17.3.1 Adds Configurable Guaranteed Contention Time
Under Cable Interface

- `(config-if)#cable upstream min-bwreq-ops scqam x y`
  - `X` = `0-20` percent option with 0 default
    - ✓ 0 means 1 Cont BW Req opportunity every 2 ms during full congestion
  - `Y` = `0-9` "tenths of a percent" option with 0 default
- Example; Typical 2 ms bucket contains 160 minislots
  - 2-tick minislot*6.25 us/tick = 12.5 us; 2 ms/12.5 us = 160 minislots/"bucket"
  - Current default is 1/160 = .625%
  - 5% would be 8 minislots for every 160 minislot bucket
- **Note**: Relevant when US highly utilized/congested

Preliminary Results

- 4 US ATDMA chs = 108 Mbps & 90% utilization = 97.2 Mbps
- 10% cont guarantee gave ping time ~ 8 ms
- Without guarantee & exceeding 90% utilization, latency was **85 ms**!
  - CM pinged also allocated time for ping, so higher US BE Priority helps
- **Note**: This was just 2 CMs with US traffic & pinging another or 2
- **Goal**: Test with many CMs doing Cont Requests filling up US
  - Ping with 1500B at 8 ms would be ~ 1.5 Mbps only
- Trade-off will be utilization for guaranteed contention time
  - Less calls/US possible (180 vs 198) & less speed/US (24 Mbps vs 27)

US Service Flows (D1.1 & New PGS Flow)

- Unsolicited Grant Service – UGS used for VoIP
  - After call setup, no more BW requests
  - **Note**: Still rely on DS map for proper 10 or 20 ms US grant
- Real Time Polling Service – RTPS meant for gaming or US video
- Non-Real Time Polling Service – nRTPS used for call signaling
  - Better than RTPS
    - ✓ Can utilize Cont & non-cont Reqs
    - ✓ Traffic can have priority set (0-7)
- Unsolicited Grant Service with Activity Detection - UGS-AD
  - nRTPS while call muted so BE traffic can be scheduled
- Proactive Grant Service - PGS

Non-Real Time Polling Service - nRTPS

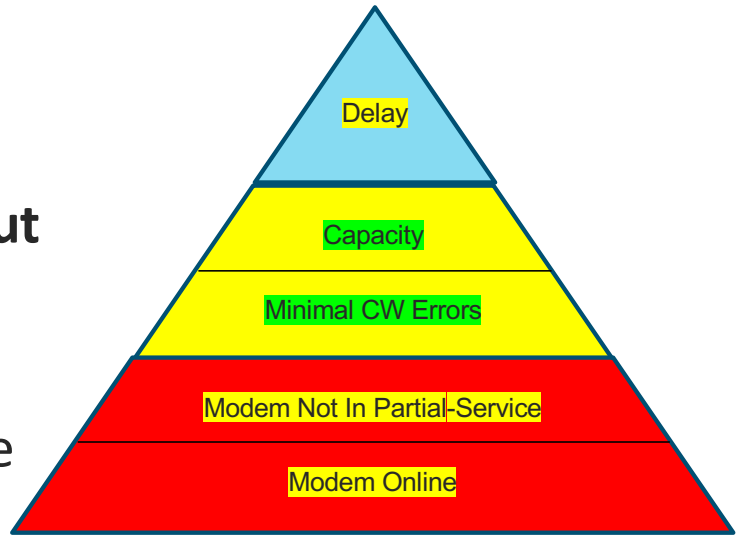
- Can “nail-up” non-contention requests, so no collisions or back-off
- Example: 500 CMs in SG and every one of them got an nRTPS flow (more SIDs BTW) with 100 ms polling, that would be 10 minislots/sec * 48B/mini = 480*8b/B*500 CMs = 1.92 Mbps “nailed up” BW for non-contention requests
- D3.0 CMs doing 4-ch US bonding only do that on 1 US not all 4
  - nRTPS would distribute across USs, so 1.92 Mbps would be overhead for entire 108 Mbps aggregate!
- Here’s where it falls apart
  - nRTPS is assigned to default single-ch BG
  - Subsequent traffic (BE) that goes with nRTPS non-contention request is configured as well, can have higher priority, but it is also relegated to that single US ch
- LLQ mode allows some jitter with benefit of not hard-setting those non-contention requests like DOCSIS scheduler mode, but traffic still assigned to single US
  - ```
    cbr8(config-if)#cab upstream 0 scheduling type nRTPS mode ?
      docsis  docsis compliant mode
      low latency scheduling of periodic events
    ```
- PGS+ solves this issue
    ```

Req/Grant Cycle

- Request in minislots includes all overhead
  - Usually for concatenated frame using a-long burst (IUC10)
- US grant fragmented into 1800 B each; includes phy overhead
- Piggyback Req avoids collisions, but adds serialization time
  - Early Req Extraction (ERE) would be good, but not sure where piggyback resides IRT fragment
- DS Map every 2 ms, but Map Advance and serialization time affect US speed
  - Turn-around time could be every 2nd, 3rd or 4th Map (4, 6, 8 ms)
- Ch width & minislot size in ticks decides max concat & BW Req of 255 minislots
  - 255*.9 to cover phy overhead gives max concat ethernet frame of 229 minislots
  - 2-tick minislot with 6.4 MHz ch width using 64-QAM equates to 48B minislot
  - Max concat of 48*229 = 11 kB every 8 ms = 11000*8*1/.008 = 11 Mbps
    - ✓ From spreadsheet with fragmentation taken into account: 10.6 Mbps
- D3.0 MTC-mode with continuous concatenation and fragmentation (CCF) avoids this Req/Grant cycle conundrum associated with single US ch CMs

Consider Impact of Changes

- Force higher order modulation to increase channel capacity **but** CM sees high CW errors
- Reduce MAP advance timer to minimize delay **but** CM could go offline due to late MAP messages
- Minimize time interleaving to reduce delay **but** CM now sees CW errors caused by impulse noise impairments
- Eliminate data backoff random delay before bandwidth request **but** get constant collisions leading to very long US delays & lost data



Miller Hierarchy of Modem Needs

How To Decrease Latency?

- Enable DOCSIS Predictive Scheduling (DPS)
- Consider other US scheduling methods other than “best effort”
- Do not allow US chs to approach full capacity
- If US channels nearing full capacity, configure 5% of ch for Cont Req
- Static Map Advance may be desirable
- IRT OFDMA, need to balance channel capacity with latency
  - 50 kHz subcarrier spacing has less latency **but** more overhead than 25 kHz spacing
  - Smaller frame size (K – symbols per frame) has less latency **but** also less time interleaving so less robust to impulse impairments
  - Higher pilot pattern overhead per minislot with small frame size **so** reduced ch capacity