802.1Qbv: Performance / Complexity Tradeoffs

Rodney Cummings
National Instruments
Automotive Networking History (1 of 2)

• During FlexRay’s formation, common complaint...
  “CAN is not deterministic”
  • CAN media access is event-based, with no time sync
  • CAN latency analysis complex; requires specific system
  • CAN configuration is simple and flexible
    • Company changes its traffic... doesn’t affect other companies’ traffic
Automotive Networking History (2 of 2)

• For today’s use of FlexRay, common complaint…
  “FlexRay configuration is too complex”
  • FlexRay media access uses time-based slots
  • FlexRay latency analysis is simple; general (per slot)
  • FlexRay configuration has business implications
    • Company changes its traffic…all other companies must adapt
Determinism: Lessons Learned

- Inherent tradeoffs between...
  - Bandwidth utilization
  - Latency / jitter
  - Complexity of latency analysis
  - Complexity of configuration

- No network technology is perfect for all simultaneously

- Best technologies enable tradeoffs per application
  - Example: Mitigation for complexity of CAN latency analysis...
    use only 80% bandwidth to allow idle for deadlines
Contention of this Presentation

• Planned PARs for IEEE 802.1 AVB make Ethernet the best network technology for deterministic (automotive/industrial) applications
• 802.1Qbv enables performance / complexity tradeoffs
• If application requirements evolve, tradeoffs evolve
  • Without the need to switch to a new network technology
Tradeoffs for Reserved Traffic

- **802.1Qav (credit-based shaper) and 802.1Qat (MSRP)**
  - Bandwidth utilization 😊
    - All bandwidth not used by reserved is available for best-effort
    - Avoids long bursts of reserved so best-effort progresses
  - Latency / jitter 😅
    - Not optimal, but sufficient for many control applications
  - Complexity of latency analysis 😞
    - No general formula; requires a specific system
  - Complexity of configuration 😊
    - Adding/removing streams does not affect existing streams
- **Similar to CAN (most popular automotive network)**
Example for 802.1Qbv Tradeoffs (1 of 3)

- Example automotive requirements from AVB assumptions

- Previous presentations using this example
  - Scheduled shaper (802.1Qbv) with store&forward
  - Preemption with store&forward, and with cut-through

- Assumptions for calculations
  - Each AVB hop includes preamble and IFG
  - Each AVB hop includes internal device delay ($t_{Device}$)
    - Worst: Talker 5.12µs (512 FE bit times), Bridge 10.24µs
    - Best: Talker 0.04µs, Bridge 0.04µs
Example for 802.1Qbv Tradeoffs (2 of 3)

- Scheduled frames on Fast Ethernet (FE)
  - Maximum latency: 100μs over 5 AVB hops
  - Transmission period: 500μs
  - Maximum frames per period: 8
  - Maximum payload: 128 bytes
    - Assuming layer 2 tagged (22 bytes overhead), 150 bytes total
  - Frame time = 13.6μs
    - Frame + preamble + IFG =
      - \((150 \times 80\text{ns}) + (8 \times 80\text{ns}) + (12 \times 80\text{ns}) =\)
      - 12.0μs + 0.64μs + 0.96μs
Example for 802.1Qbv Tradeoffs (3 of 3)

• For 100µs latency, must assume talker window per frame
  • Single window in talker
  • Multiple windows in talker
Design 1: Optimal (1 of 3)

- Scheduled shaper (802.1Qbv) with cut-through
- Cut-through at 64 bytes (including preamble)
  - Store 5.12μs ingress before egress
  - Cut-through for remainder of frame: 8.48μs
Design 1: Optimal (2 of 3)

Talker ingress (DMA)

Talker egress

Bridge 1 egress

Open scheduled gate (close all others)

Guard band for best-effort interference (not to scale)

Same gates (no best-effort)

123.36μs

Window 1

Window 2

1μs after talker for 802.1AS inaccuracy

Same gates (no best-effort)

TimeA

TimeB

TimeC

1 store 1 cut

2 store 2 cut

t_{Device}

t_{Device}
Design 1: Optimal (3 of 3)

- Latency for frame 1
  - $TimeA$ (talker before cut) = $t_{DeviceTalker} + 5.12\mu s$
  - $TimeB$ (bridge before cut) = $1.0\mu s + t_{DeviceBridge} + 5.12\mu s$
  - $TimeC$ (cut of frame 1) = $8.48\mu s$
  - Frame 1 latency = $TimeA + (4 \times TimeB) + TimeC$
  - Using worst $t_{Device}$ ($t_{DeviceTalker} = 5.12\mu s$, $t_{DeviceBridge} = 10.24\mu s$)
    - $10.24\mu s + (4 \times 16.36\mu s) + 8.48\mu s = 84.16\mu s$ (< 100µs requirement)
    - Bandwidth for scheduled = 191.72µs (38%)
  - Using best $t_{Device}$ ($t_{DeviceTalker} = 0.04\mu s$, $t_{DeviceBridge} = 0.04\mu s$)
    - $5.16\mu s + (4 \times 6.16\mu s) + 8.48\mu s = 38.28\mu s$
    - Bandwidth for scheduled = 110.12µs (22%)
Tradeoffs for Optimal Design

- Window per frame in talker and bridges
  - Bandwidth utilization 😐
    - Up to 123µs of each 500µs unused (0% to 25%)
    - Preemption solves this (not related to 802.1Qbv tradeoffs)
  - Latency / jitter ☺
    - Optimal ($t_{Device}$ has biggest impact; benefits from cut-through)
  - Complexity of latency analysis 😎
    - Simple addition; general (calculate for a single frame)
    - Clearly deterministic
  - Complexity of configuration 😞
    - Multiple distinct windows in talker and each bridge
    - Change in one talker’s traffic can impact entire system
Adjusting Tradeoffs

• Assume there is a complaint about Optimal design…
  “802.1Qbv configuration is too complex”

• Application designer
  • Uses single loop for talker
    • Rates harmonic to loop (e.g. 500µs loop; rates 500µs, 1ms, 4ms, …)
  • Needs simple configuration with few interdependencies
  • Needs simple latency analysis
  • Understands tradeoffs
    • Latency/jitter may not be optimal
    • Bandwidth utilization may not be optimal
Design 2: Simple (1 of 3)

- Continue to assume cut-through
- Single window in talker
  - All scheduled frames in a burst
- Single window in bridges
  - Window represents maximum bandwidth for scheduled
  - Direction independent: same window in all bridges
- Topology independent
  - Talkers / listeners can move
    - e.g. 8 frames from 1 talker, then 1 frame from 8 different talkers, etc
  - Assume maximum number of hops (5)
Design 2: Simple (2 of 3)

Talker ingress (DMA)

Talker egress

Bridge 1 egress

Bridge 2 egress

Bridge 3 egress

Bridge 4 egress

IEEE 802.1 AVB, September 2012, Santa Cruz
Design 2: Simple (3 of 3)

- Latency for frame 8 (last in talker’s burst ingress)
  - $Time_A$ (talker before cut) = $t_{DeviceTalker} + 5.12\mu s$
  - $Time_B$ (bridge before cut) = $1.0\mu s + t_{DeviceBridge} + 5.12\mu s$
  - $Time_C$ (cut of 1 + full 2-8) = $8.48\mu s + (7 \times (t_{DeviceBridge} + 13.6\mu s))$
  - Frame 8 latency = $Time_A + (4 \times Time_B) + Time_C$
  - Using worst $t_{Device}$ ($t_{DeviceTalker} = 5.12\mu s$, $t_{DeviceBridge} = 10.24\mu s$)
    - $10.24\mu s + (4 \times 16.36\mu s) + 8.48\mu s + 166.88\mu s = 251.04\mu s$
    - Bandwidth for scheduled = 232.32\mu s (48%, 10% unused)
  - Using best $t_{Device}$ ($t_{DeviceTalker} = 0.04\mu s$, $t_{DeviceBridge} = 0.04\mu s$)
    - $5.16\mu s + (4 \times 6.16\mu s) + 8.48\mu s + 95.48\mu s = 133.76\mu s$
    - Bandwidth for scheduled = 128.6\mu s (26%, 4% unused)
Tradeoffs for Simple Design

- Window per frame in talker and bridge
  - Bandwidth utilization 😐
    - As with Optimal, up to 123µs of each 500µs unused (0% to 25%)
    - 802.1Qbv tradeoff: additional 4% to 10% always unused
  - Latency / jitter 😐
    - Not optimal, but sufficient for many control applications
    - No interference per hop; 133.76µs is close to 100µs requirement
  - Complexity of latency analysis 😊
    - Simple addition; general; clearly deterministic
  - Complexity of configuration 😊
    - Significant flexibility for traffic changes
802.1Qbv Flexibility

• Optimal and Simple are opposite ends of a spectrum

• Many points in between
  • E.g. Mix: Optimal for critical traffic, Simple for rest
  • E.g. Multiple windows in talkers, one window in bridges

• All points provide simple latency analysis
Conclusions

• 802.1Qbv scheduled traffic provides
  • Simple latency analysis
  • Tradeoffs between performance and configuration simplicity
• Reserved is an excellent option for some traffic
• Application’s network design can evolve as needed

• We’re on the right track folks!
Thank you