Worst-Case Delay Analysis for Asynchronous Traffic Shaping (ATS) in Time-Sensitive Networking (TSN)

Paul Pop, paupo@dtu.dk

October 10, 2024

Overview

- **Goal**: Compute end-to-end delay for time-sensitive streams in a TSN network using ATS.
- **Time Composability** allows us to analyze each hop individually.
- Total end-to-end delay is the sum of delays over each hop along the stream's path.
- Important for ensuring that streams meet their deadlines.

ATS Stream and Network Model

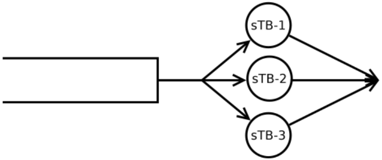
ATS Stream Model:

- Each stream f is characterized by:
  - **Path**: $path_f$
  - **Priority**: p_f
  - **Deadline**: d_f
  - **Frame Lengths**: Minimum $\breve{l}_f$, Maximum $\hat{l}_f$
  - **Reserved Data Rate**: r_f
  - **Burst Size**: $\hat{b}_f$

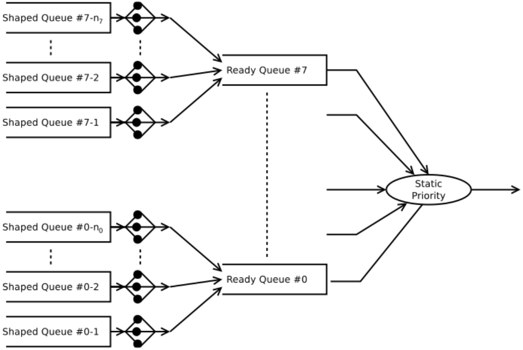
Network Model:

- Network is modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
- Vertices $\mathcal{V}$ represent switches and end-systems.
- Edges $\mathcal{E}$ represent one-way links with capacity r (bits per second).
- Streams traverse multiple hops from source to destination.
- Multiple streams may share the same links and switches.

ATS Queues (1)



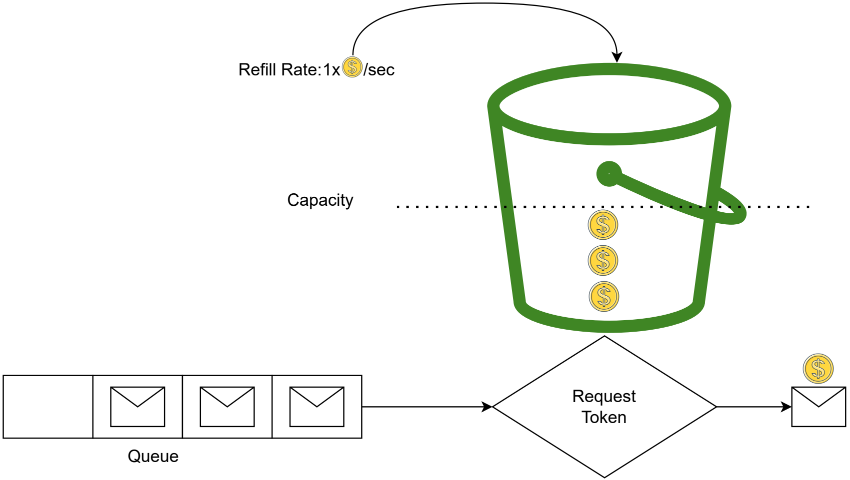
Example of a shaped queue with three streams, each with its own token bucket



ATS Queues (2)

- **Two Types of Queues in ATS:**
- **Shaped Queues**:
  - Store frames that need reshaping.
  - Frames are regulated by their own token bucket shaper.
  - Shaped queues correspond to streams sharing the same shaped queue q_I.
- **Ready Queues**:
  - Collect frames that are ready for transmission.
  - Multiple shaped queues feed into a single ready queue q_S.
  - Ready queues operate based on priorities (e.g., strict priority scheduling).
  - Frames in ready queues are eligible for transmission selection.

Token Bucket Algorithm in ATS (1)



Token Bucket Algorithm in ATS (2)

Per-Stream Token Bucket Operation:

- **For each stream f:**
  - Tokens accumulate in the bucket at rate r_f, up to capacity $\hat{b}_f$.
  - Frame transmission consumes tokens equal to frame size l_f.

Operation in Shared Shaped Queue:

- Frames from multiple streams are queued in a single **FIFO shaped queue**.
- **Head-of-Line Blocking:**
  - If the frame at the head cannot be transmitted (tokens $< l_f$), it **blocks** the queue.
  - All other frames from other streams **must wait**, even if they have enough tokens.

Purpose:

- Enforces bandwidth and burst constraints **per stream**.
- Maintains **temporal isolation** while sharing queue resources.
- Ensures compliance with the **interleaved shaping** mechanism.

Queue Assignment Rules (QARs)

Purpose of QARs:

- Ensure **temporal isolation** between streams to minimize interference.
- Facilitate **reshaping for free** by proper stream segregation.

Queue Assignment Rules:

- **QAR1**: streams f_i and f_j are **not allowed to share a shaped queue** if they are **received from different upstream nodes**.
  - Separates streams based on their **ingress ports** or **source nodes**.
- **QAR2**: streams f_i and f_j are **not allowed to share a shaped queue** if they are **received from the same upstream node** but have **different priority levels**.
  - Ensures that streams of different **priorities** from the same source are isolated.
- **QAR3**: streams f_i and f_j are **not allowed to share a shaped queue** if they are **sent by the current node** and have **different priority levels**.
  - Applies to streams **originating** at the current node with different priorities.

Understanding the Queue Assignment Rules

Clarifying the Differences between QARs:

- **QAR1** addresses streams from **different upstream nodes**:
  - Ensures that streams arriving from different sources are placed in separate shaped queues, even if they have the same priority level.
  - Prevents interference due to variations in traffic from different sources.
- **QAR2** focuses on streams from the **same upstream node** but with **different priority levels**:
  - streams received from the same upstream node are separated into different shaped queues if their priorities differ.
  - Maintains priority-based isolation among streams from the same source.
- **QAR3** is about streams **sent by the current node** with **different priority levels**:
  - Streams originating from the current node are assigned to different shaped queues based on their priority.
  - Ensures that locally generated streams do not interfere with each other across priority levels.

QAR Importance: Why are these rules needed?

- **Prevent Interference**: To minimize interference between streams, ensuring that the delay introduced by reshaping does not increase the worst-case delay.
- **Maintain Temporal Isolation**: By segregating streams based on their source and priority, we maintain temporal isolation, crucial for real-time communication.
- **Enable "Reshaping for Free"**: Satisfying these rules allows the network to reshape streams without additional delay, a property known as "reshaping for free".

How are the QARs implemented?

How are the QARs implemented?

- **Queue Allocation**:
  - Assign streams to **shaped queues** based on their source server and priority level.
  - Ensure that streams violating QARs are placed in separate queues.
- **Queue Structure**:
  - Each output port maintains **multiple shaped queues** and **ready queues** per priority level.
  - Shaped queues feed into ready queues, which are then scheduled for transmission.

How many queues are needed?

- Let n be the number of input ports (servers), and m be the number of priority levels.
- **Total Shaped Queues**: $m \times n$
- **Total Ready Queues**: m (one per priority level)
- **Total Queues per Output Port**: $m \times n + m$

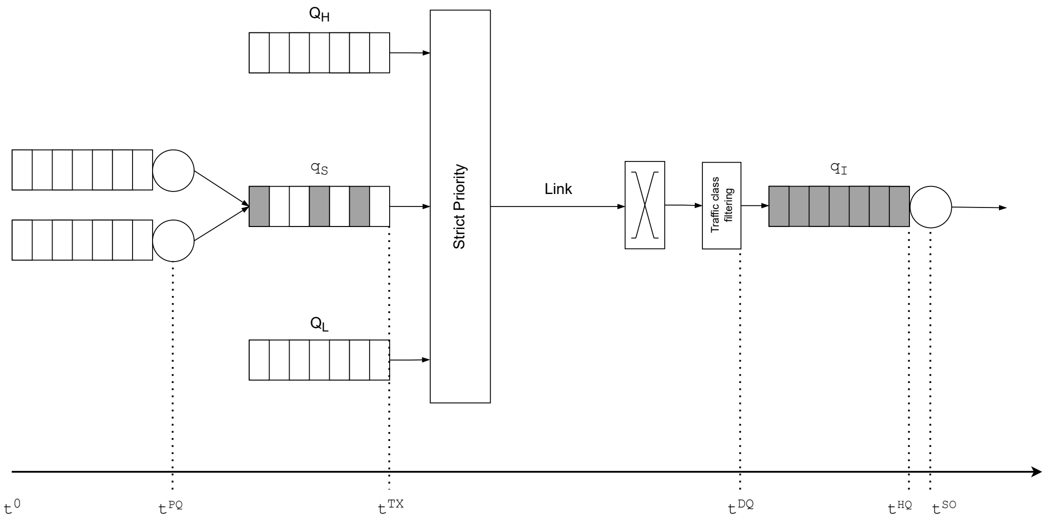
Simple Assignment of Frames to Shaped Queues

- **Switch Configuration:**
  - n ingress ports, m priority levels (e.g., 8 levels in Ethernet).
  - Total of m ready queues (one per priority level).
  - Total of $m \times n$ shaped queues (n per priority level).
- **Assignment Rules:**
  1. For each incoming frame f:
     - Determine the priority level p of frame f.
     - Identify the ingress port i where frame f arrived.
     - Assign frame f to shaped queue $S[p][i]$.
- **Explanation:**
  - Frames from different ingress ports are placed in separate shaped queues (satisfies QAR1).
  - Frames of different priority levels from the same ingress port are placed in different shaped queues (satisfies QAR2).
  - Frames to be transmitted at different priority levels are separated (satisfies QAR3).

ATS Operation in a Switch

- 1. **Traffic Classification**:
  - Incoming frames are classified based on their priority and stream identification.
  - Determines which shaped queue a frame should enter.
- 2. **Shaping with Token Buckets**:
  - Each shaped queue applies token bucket shaping to regulate traffic.
  - Ensures that each stream conforms to its allocated rate r_f and burst $\hat{b}_f$.
- 3. **Transmission Selection**:
  - Frames from ready queues are selected for transmission.
  - Typically based on priority scheduling.
  - Ensures higher priority streams are served first.

Per-Hop Delay Calculation: Times Considered (1)



Per-Hop Delay Calculation: Times Considered (2)

- t^0: The start of the busy period.
- t^{PQ}: The arrival time of frame f in its ready queue q_S.
- t^{TX}: The time when frame f starts transmission from q_S.
- t^{DQ}: The arrival time of frame f in the shaped queue q_I at the next node.
- t^{HQ}: The time when frame f becomes the head of q_I.
- t^{SO}: The time when frame f leaves q_I (after shaping delay), ready for transmission selection.

Per-Hop Delay Calculation Overview

- The total per-hop delay $d_f^{PQ,SO}$ for a frame f consists of:
  - **Queueing Delay in Ready Queue** ($d^{PQ,TX}$)
  - **Transmission Delay** ($d^{TX,DQ}$)
  - **Shaping Delay at Next Node** ($d^{DQ,SO}$)
- Total per-hop delay is:

$$d_f^{PQ,SO} = d^{PQ,TX} + d^{TX,DQ} + d^{DQ,SO}$$

Overall Per-Hop Delay Formula

Upper Bound on Per-Hop Delay for Frame f:

$$d_f^{PQ,SO} \leq \max_{j \in I} \left(\frac{\hat{b}_H + \hat{b}_{C(j)} + \hat{b}_j - \breve{l}_j + \hat{l}_L}{r - \hat{r}_H} + \frac{\breve{l}_j}{r} \right)$$

- Where:
  - I: Set of streams sharing the same shaped queue with f

Intuition Behind the Formula

- **The formula provides an upper bound** on the delay experienced by frame f over one hop.
- It accounts for:
  - **Interference** from higher priority and same priority streams.
  - **Shaping delays** due to the token bucket mechanism at the next node.
  - **Transmission delays** based on frame sizes.
- By considering the worst-case scenario among all interfering frames $j \in I$, we ensure a safe upper bound.

Queueing Delay in Ready Queue ($d^{PQ,TX}$)

$$d^{PQ,TX} \leq \frac{\hat{b}_H + \hat{b}_C + \hat{l}_L}{r - \hat{r}_H}$$

- **Notes**:
  - Represents the time to clear the backlog before f can be transmitted.
  - Considers the worst-case scenario: all higher and same priority frames are ahead of f.
- The queueing delay is the time needed to transmit all *higher* and *same* priority frames already queued ahead of frame f, plus any *lower* priority frame currently being sent, before f frame can be transmitted.
- Remember how the ready queues handle different stream priorities in FIFO and the available link bandwidth after higher priority traffic.

Understanding $d^{PQ,TX}$

- **Numerator** $(\hat{b}_H + \hat{b}_C + \hat{l}_L)$:
  - $\hat{b}_H$: Total burst size of higher priority streams.
  - $\hat{b}_C$: Total burst size of same priority streams (excluding f).
  - $\hat{l}_L$: Maximum frame length of lower priority streams (in transmission when f arrives).
- **Denominator** $(r - \hat{r}_H)$:
  - Available Bandwidth: The effective bandwidth left for transmitting f after higher priority traffic has consumed its reserved bandwidth.
  - r: Link capacity.
  - $\hat{r}_H$: Total reserved rate of higher priority streams.

Transmission Delay ($d^{TX,DQ}$)

$$d^{TX,DQ} = \frac{\hat{l}_f}{r}$$

Understanding $d^{TX,DQ}$

- **Parameters**:
  - $\hat{l}_f$: Maximum frame length of frame f.
  - r: Link capacity.
- **Interpretation**:
  - Time it takes to transmit frame f over the link at full capacity.
- **Assumption**:
  - Propagation delay is negligible.

Clarifying the Inclusion of $d^{DQ,SO}$ in $d_f^{PQ,SO}$

- Previously, we expressed the per-hop delay as the sum of individual delay components:
$$d_f^{PQ,SO} = d^{PQ,TX} + d^{TX,DQ} + d^{DQ,SO}$$
- However, in our per-hop delay formula, $d^{DQ,SO}$ is already included directly in $d_f^{PQ,SO}$.
- The per-hop delay formula accounts for the worst-case shaping delay at the next node ($d^{DQ,SO}$) when calculating $d_f^{PQ,SO}$:
$$d_f^{PQ,SO} \le \max_{j \in I} \left(\frac{\hat{b}_H + \hat{b}_{C(j)} + \hat{b}_j - \breve{l}_j + \hat{l}_L}{r - \hat{r}_H} + \frac{\breve{l}_j}{r} \right)$$
- This formula incorporates all components of delay, including:
  - **Queueing and Shaping Delays** ($d^{PQ,TX}$ and $d^{DQ,SO}$)
  - **Effective Bandwidth** after higher priority traffic
  - **Transmission Delay** ($d^{TX,DQ}$)
- Therefore, we do not need to separately calculate $d^{DQ,SO}$ when using the per-hop delay formula; it is already included in $d_f^{PQ,SO}$.

Per-hop Shaping Delay ($d^{PQ,SO}$)

$$d^{PQ,SO} \leq \max_{j \in I} \left(\frac{\overbrace{\hat{b}_H + \hat{b}_{C(j)} + \hat{b}_j - \breve{l}_j}^{\text{Accumulated burst}} + \overbrace{\hat{l}_L}^{\text{Lower frame trans.}}}{\underbrace{r - \hat{r}_H}_{\text{Remaining bandwidth}}} + \overbrace{\frac{\breve{l}_j}{r}}^{\text{Trans. delay}} \right)$$

Intuition:

This delay represents the worst-case time frame f spends waiting due to bursts from higher and same priority streams, lower priority frame transmission, and the transmission of a minimal-sized interfering frame j in the shaped queue, considering how the shaped queues process frames.

Understanding $d^{PQ,SO}$ (Part 1)

Interference from Other Streams:

- I: Set of streams sharing the shaped queue with f.
- For each $j \in I$, we consider the delay they can introduce.

Accumulated Burst ($\hat{b}_H + \hat{b}_{C(j)} + \hat{b}_j - \breve{l}_j$):

Captures the total burstiness from higher and same priority streams that can be ahead of f in the queue, including the effective burstiness of stream j.

- $\hat{b}_H$: Total burst size of higher priority streams.
- $\hat{b}_{C(j)}$: Total burst size of same priority streams (excluding j). We exclude f because including it would incorrectly suggest that f contributes to its own delay in this term.
- $\hat{b}_j - \breve{l}_j$: Effective burstiness of stream j.
  - The initial frame of j (of size $\breve{l}_j$) is already in the **Transmission Delay** term. Subtracting $\breve{l}_j$ from $\hat{b}_j$ gives us the remaining burst that can cause additional queuing delay for f.
  - $\hat{b}_j$: Burst size of interfering stream j.
  - $\breve{l}_j$: Minimum frame length of stream j.

Understanding $d^{PQ,SO}$ (Part 2)

Lower Frame Transmission ($\hat{l}_L$):
Represents the maximum size of a lower priority frame that might be in transmission when f arrives, causing initial blocking.

- $\hat{l}_L$: Maximum frame length of lower priority streams.

Remaining Bandwidth ($r - \hat{r}_H$):
Denotes the effective bandwidth available for transmitting f after accounting for higher priority traffic.

- r: Link capacity.
- $\hat{r}_H$: Total reserved rate of higher priority streams.

Understanding $d^{PQ,SO}$ (Part 3)

Transmission Delay ($\frac{\breve{l}_j}{r}$):

Accounts for the time to transmit the minimal frame length of interfering stream j.

- $\breve{l}_j$: Minimum frame length of interfering stream j.
- r: Link capacity.

Why do we need to consider transmission delays in the Shaping Delay?

- Note that even though the shaping queues feed into the ready queues and transmission is via priority selection, the frames ahead of f in the shaping queue can still cause delays due to their transmission times.
- Imagine you're in a checkout line at a grocery store (the shaping queue). Even if you're ready to pay (have enough tokens), you can't proceed to the cashier (ready queue) until the customers ahead of you have been served. If the customers ahead have large carts (long transmission times), you have to wait longer before it's your turn.

Total Per-Hop Delay Summary

Total per-hop delay for frame f:

$$d_f^{PQ,SO} = d^{PQ,TX} + d^{TX,DQ} + d^{DQ,SO}$$

- By calculating each component, we obtain an upper bound on the delay.
- This delay can be used to verify if the stream meets its per-hop deadline constraints.
- Summing per-hop delays over all hops gives the end-to-end delay.

Key Takeaways

- ATS allows deterministic delay analysis for time-sensitive streams.
- Time composability enables per-hop delay calculation.
- By using the provided formulas, we can calculate an upper bound on the per-hop delay.
- Proper configuration of token bucket parameters $(r_f, \hat{b}_f)$ ensures that streams meet their deadlines.
- Delay analysis aids in network design, admission control, and ensuring QoS.