Modeling and Simulating Time-Sensitive Networking

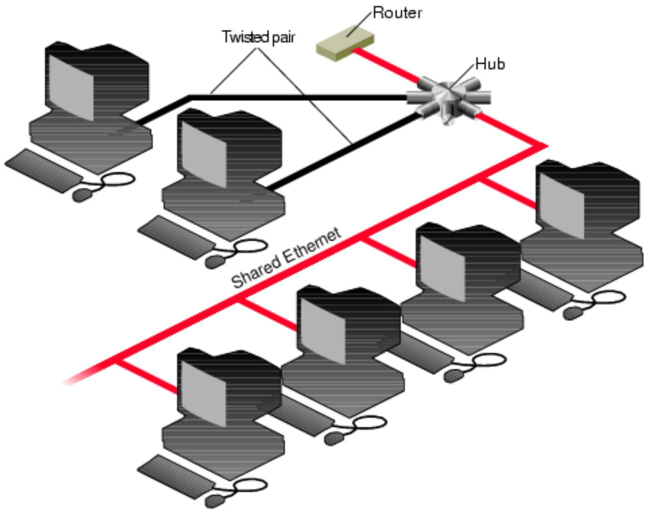
Harri Laine

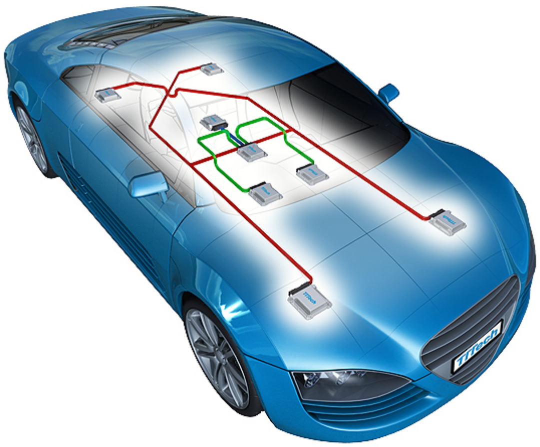
DTU Compute
Department of Applied Mathematics and Computer Science

Overview

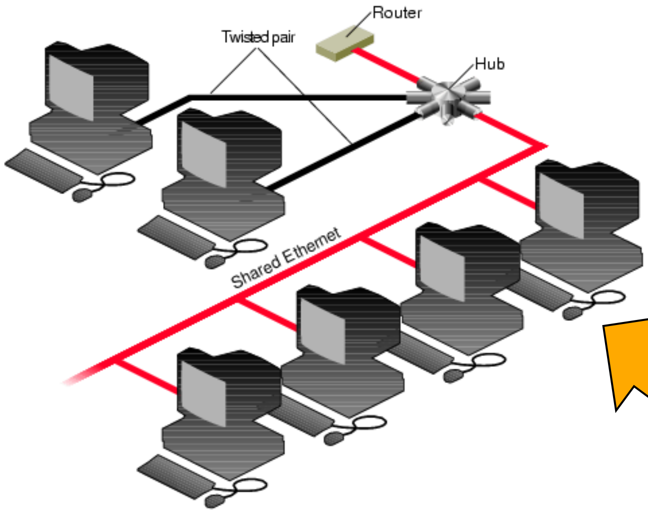
- Introduction
- Time-Sensitive Networking
  - Background
  - Goals
  - Architecture
- Simulator
  - Design
  - Implementation
- Evaluation
- Conclusion

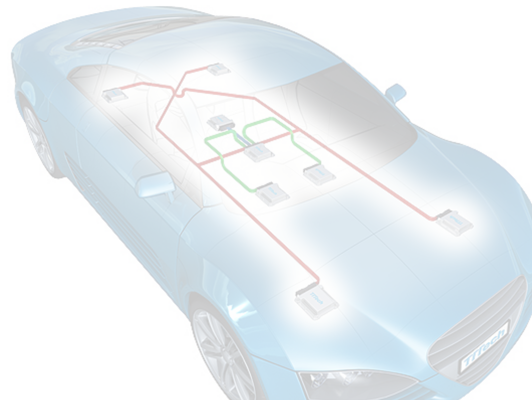
Introduction – Devices that use network





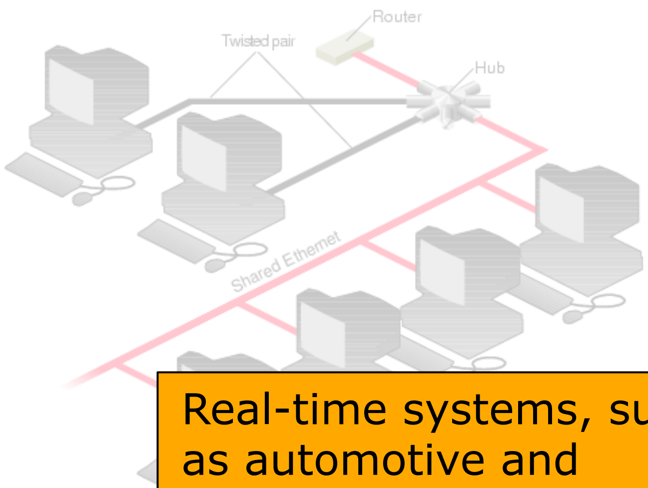
Introduction – Consumer electronics

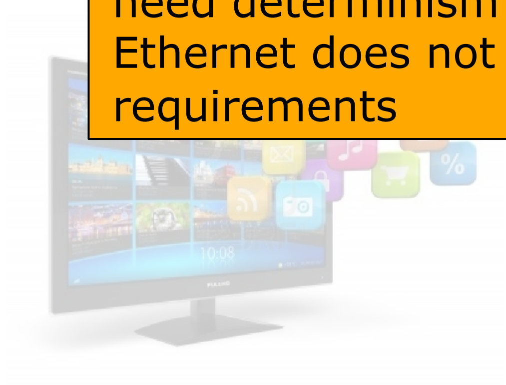




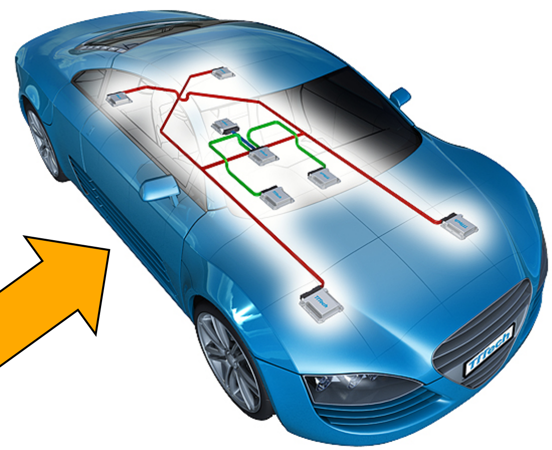
No need for determinism: Ethernet fulfills requirements

Introduction – Real-time systems

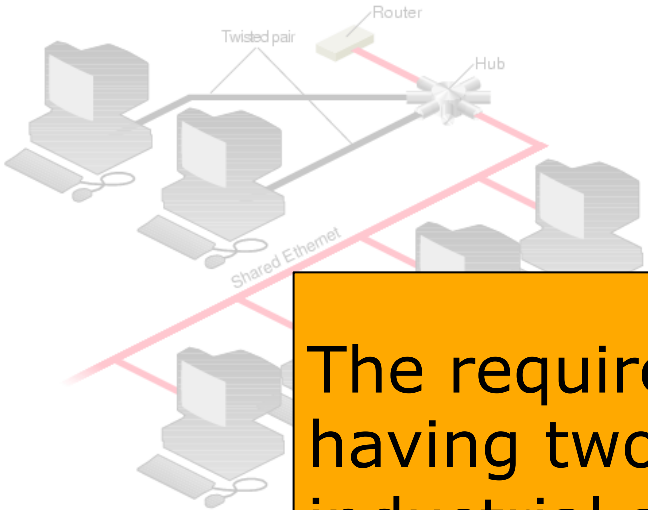


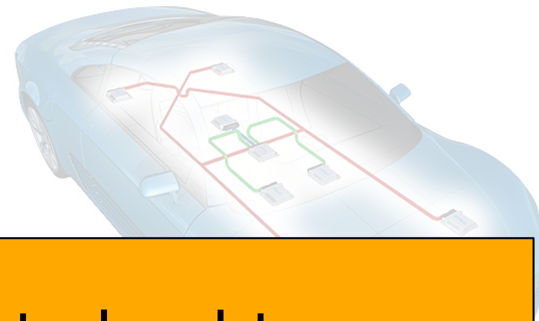


Real-time systems, such as automotive and industrial applications, need determinism: Ethernet does not fulfill requirements



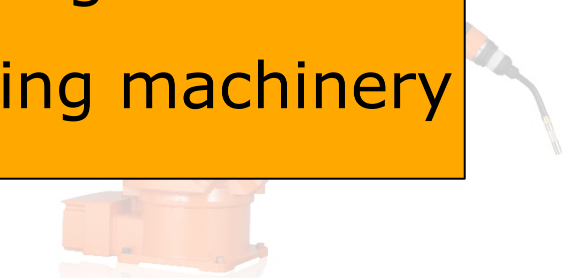
Introduction – Problem





The requirements lead to having two networks in industrial applications!

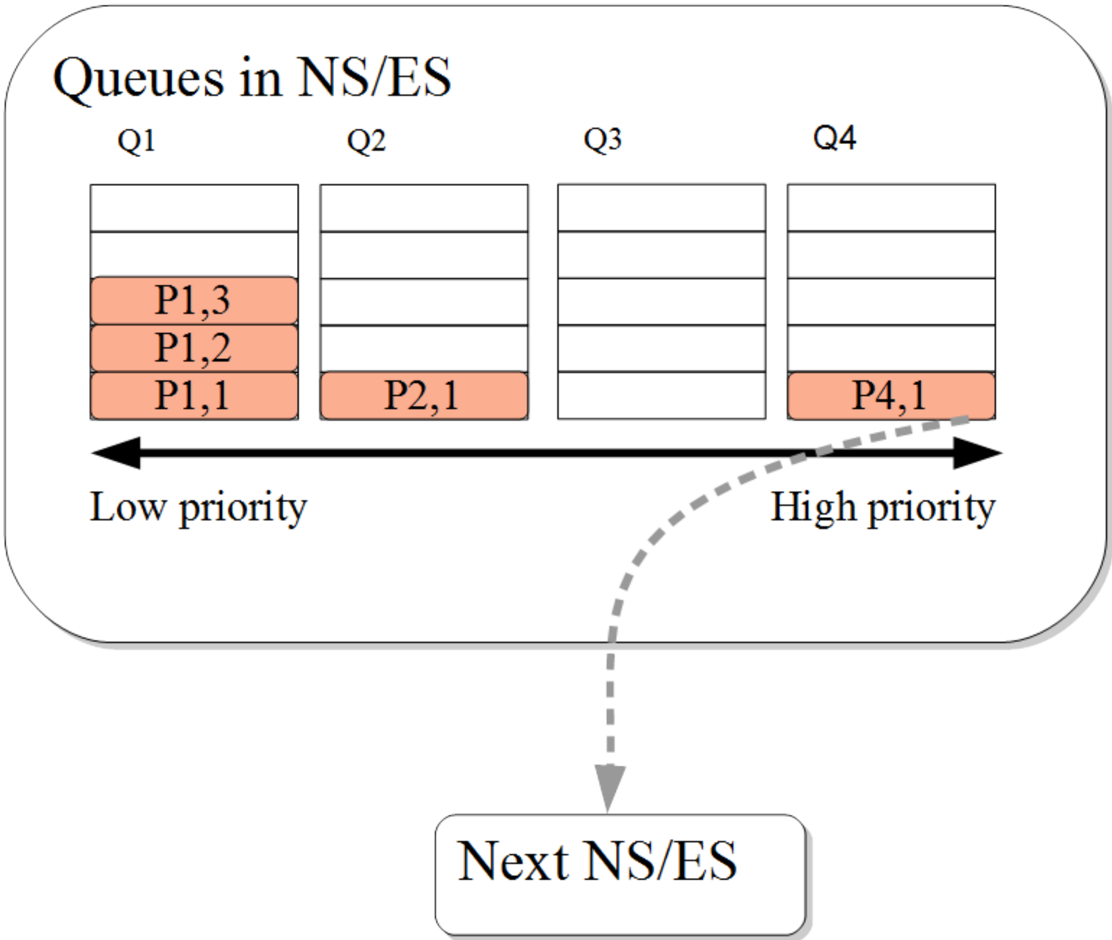
- one for connecting PCs
- one for controlling machinery



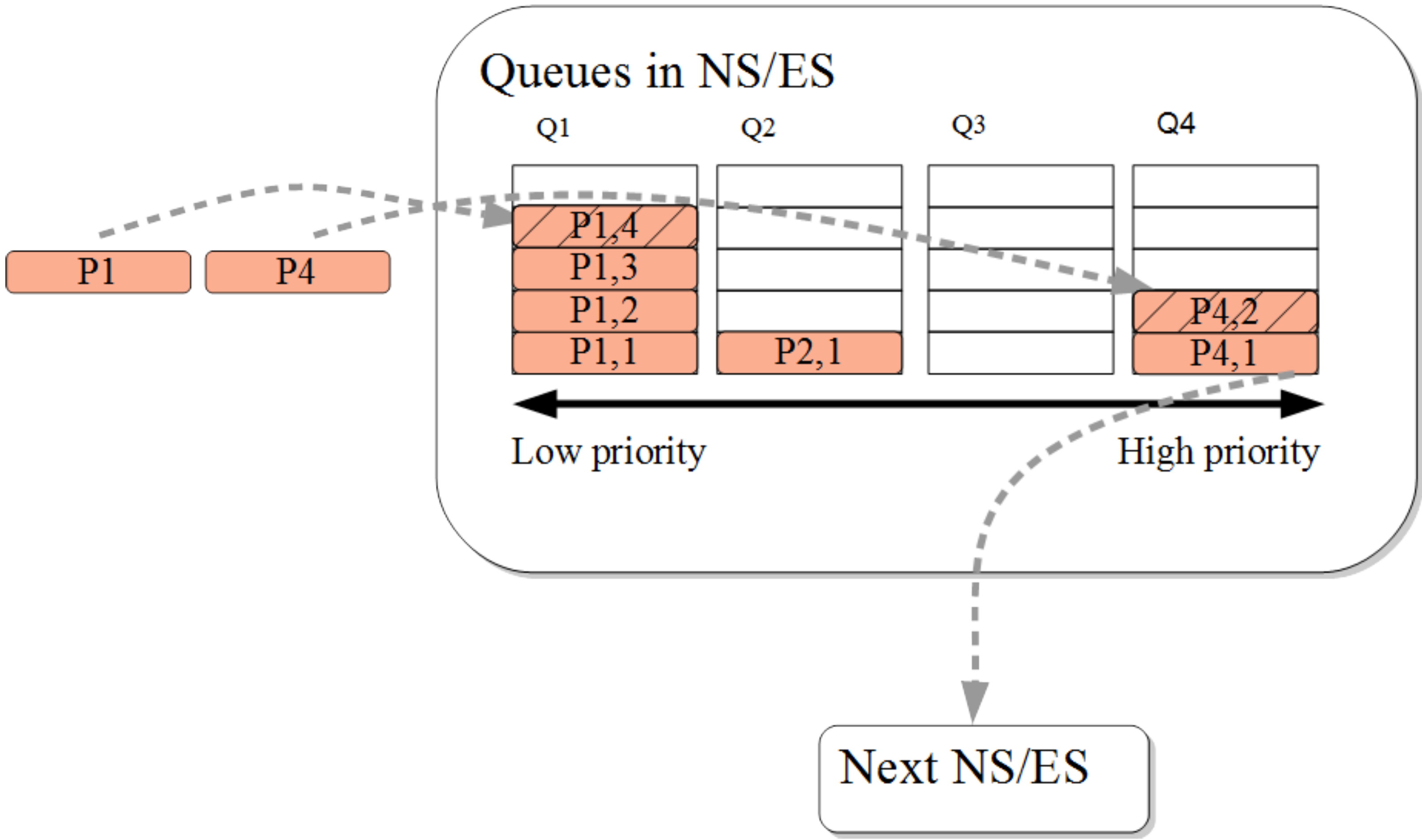
Time-Sensitive Networking – Background

- Prioritization
  - Allows important data to take preference on transmission
  - Priority tag included in each frame
  - Audio/video (AV) transmission and other real-time traffic gets lower latencies
- Credit-based shaper
  - Aimed for AV-streaming
  - Prevents high priority data to block low priority data
  - Prevents “bursts” of data in the network

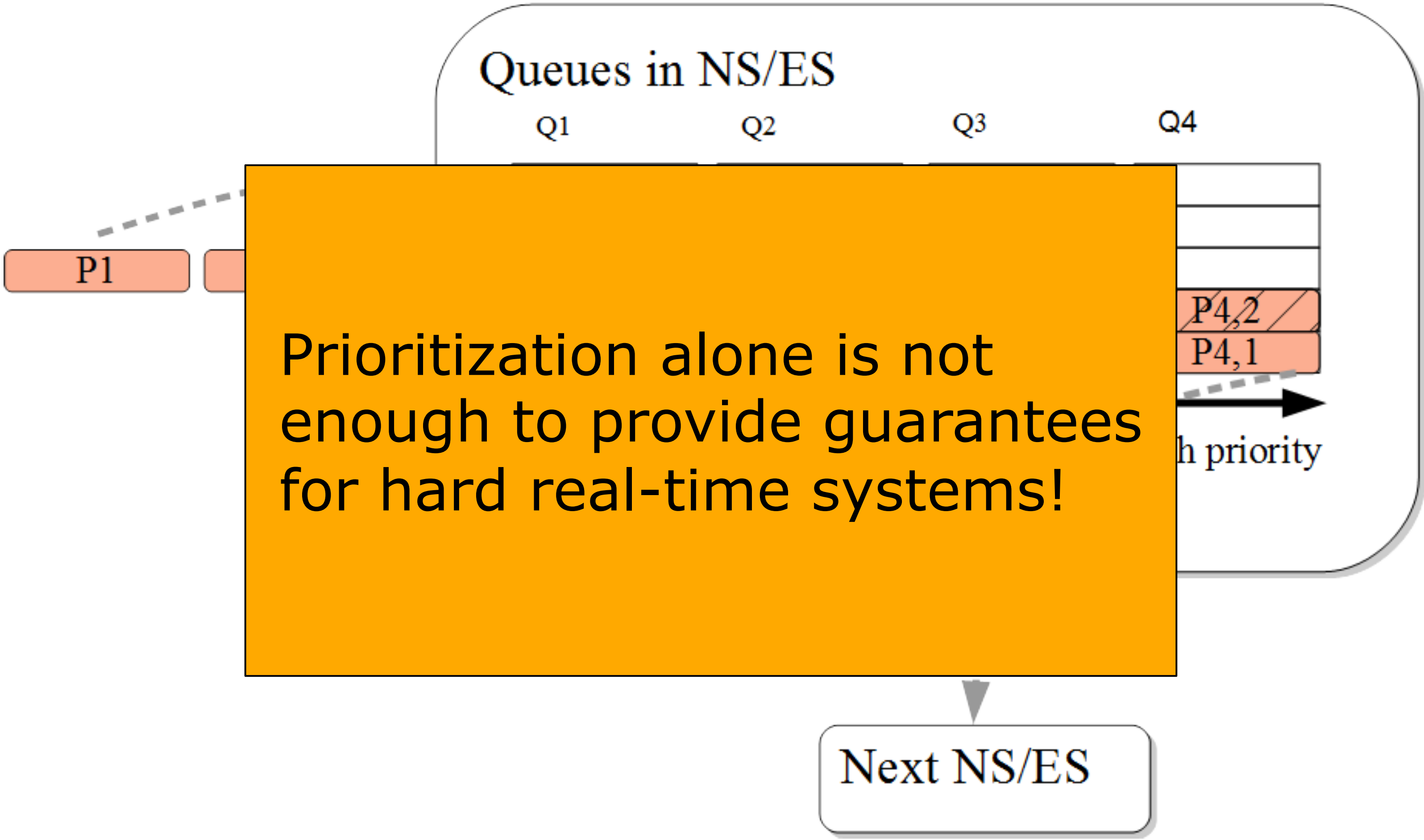
Time-Sensitive Networking – Prioritization



Time-Sensitive Networking – Prioritization



Time-Sensitive Networking – Prioritization



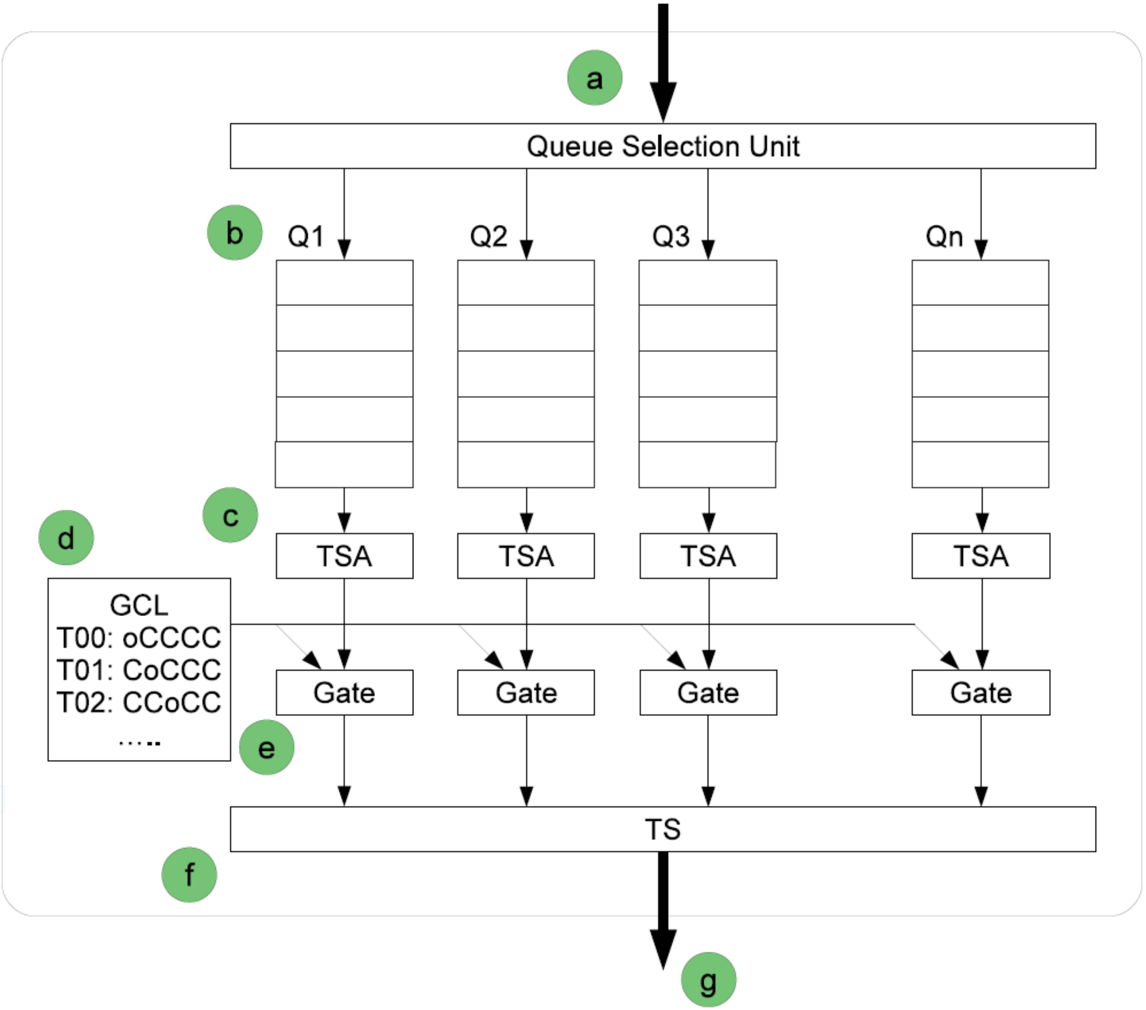
Prioritization alone is not enough to provide guarantees for hard real-time systems!

Time-Sensitive Networking – Goals

- Provide determinism in Ethernet networks
  - Protected windows using Gate Control Lists
  - Cut costs of having two networks
  - Cut costs of specialized networks
- Increase efficiency in Ethernet networks
  - Preemption
  - Credit-based shaper

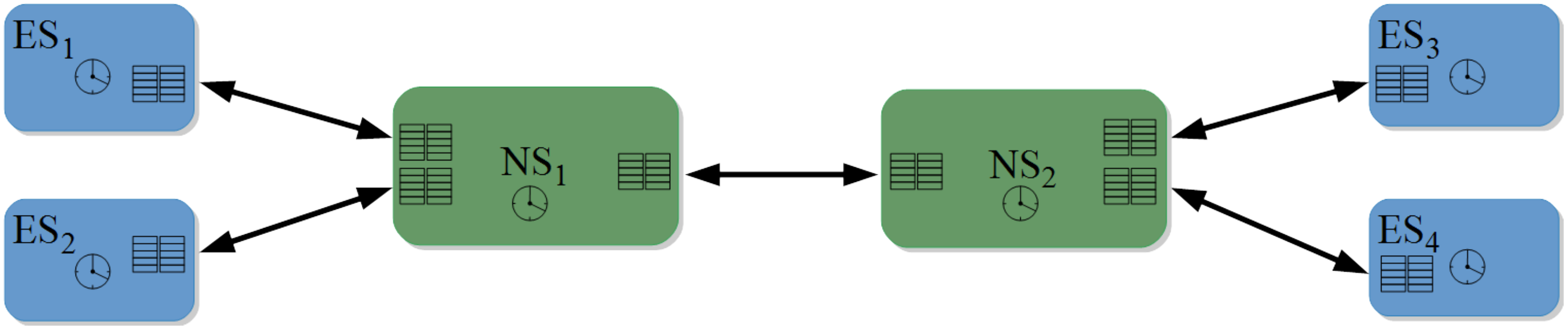
Time-Sensitive Networking – Architecture

- Gates are used in outbound port to block and allow data to flow
- Gate Control List is used to manage gate states (open/closed)



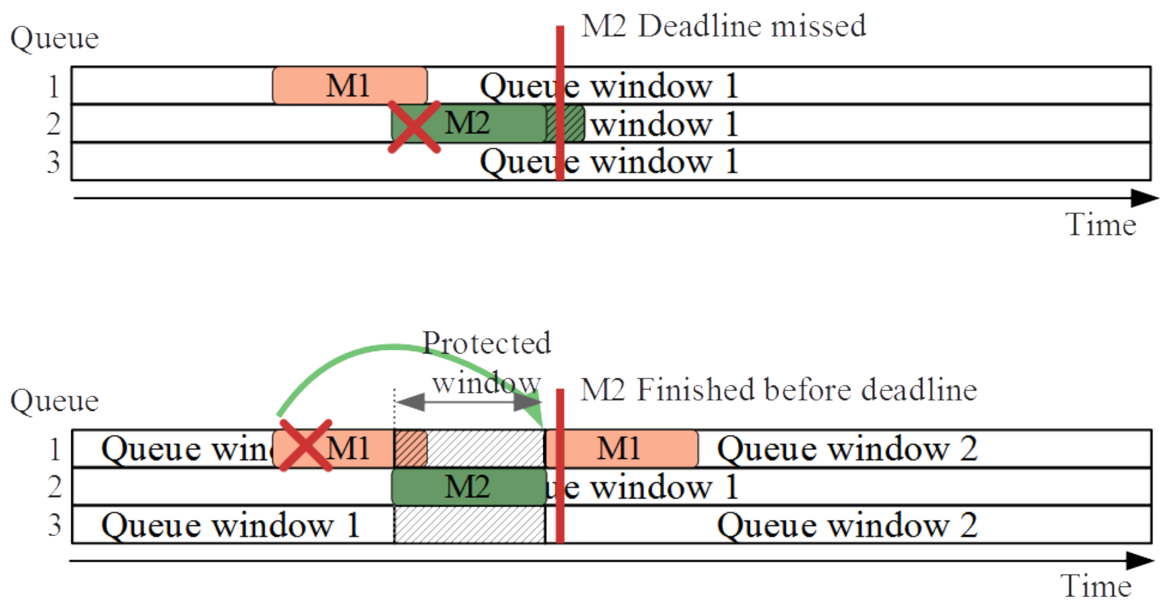
Time-Sensitive Networking – Architecture

- Architecturally simple
- Wires, clocks, queues
  - Clocks are synchronized in the network



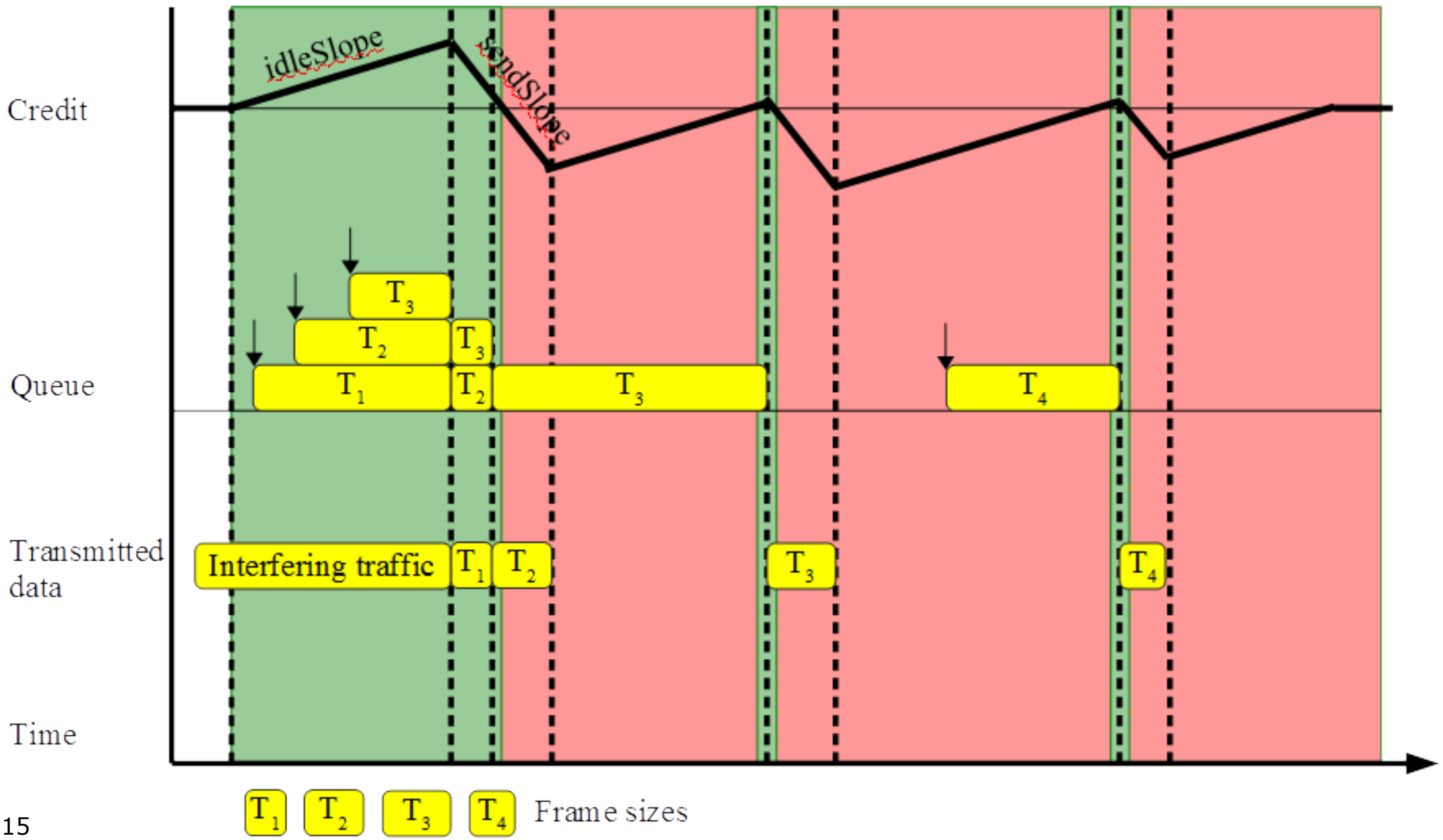
Time-Sensitive Networking – Protected windows

- Synchronized GCLs can be used to create protected windows
- Protected windows create clear communication channel for time-critical data
- Makes the network predictable
  - Suitable for hard real-time systems



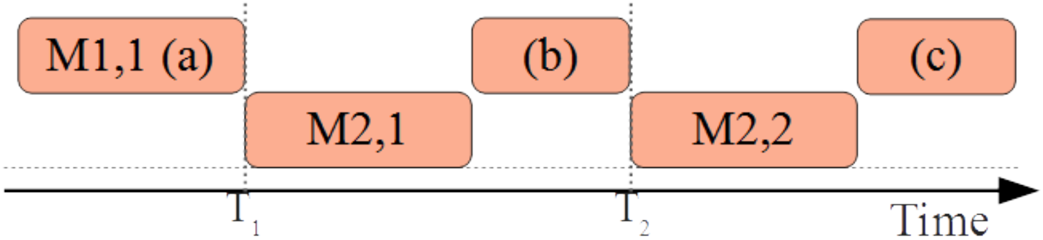
Time-Sensitive Networking – Credit-based shaper

- Restricts high priority data getting all the bandwidth
- Prevents “bursts” in network



Time-Sensitive Networking – Preemption

- Allows frames to be stopped and continued later
  - Favors high priority data, but utilizes network more efficiently
  - Low priority data can be send even just before a protected window



Simulator – Design

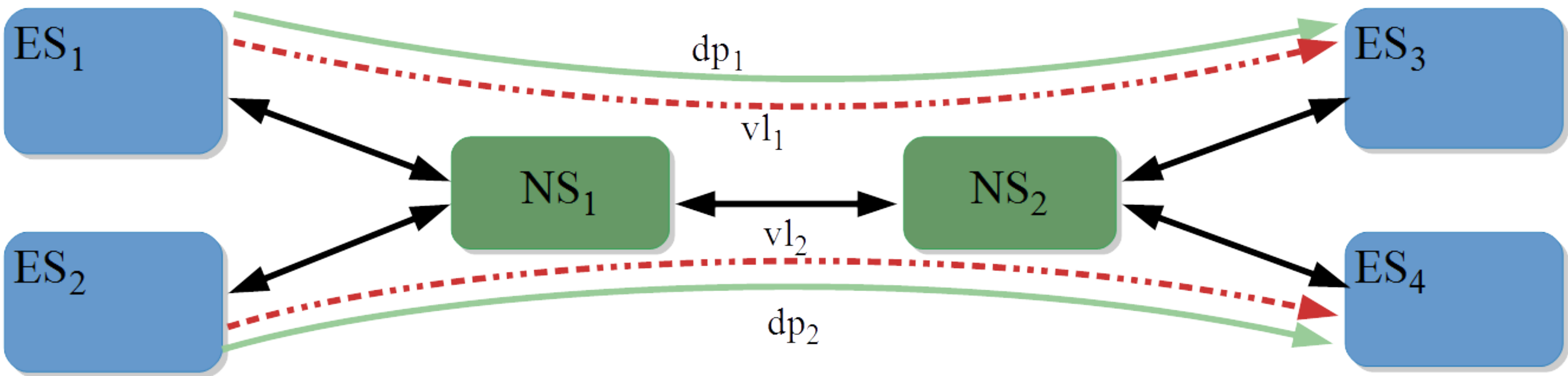
- Objectives
  - Implement TSN features
    - Protected windows
    - Credit-based shaper
    - Preemption
  - Use input files to generate
    - Messages
    - Network
    - Virtual links (routes)
    - GCLs
  - Produce output data for further analysis
- Event-based
- Randomness for non-time-critical data

Simulator – Implementation

- Java was chosen to create classes for entities (queues, ports, etc.)
  - Based on given simulator
- 5 different main parts
  - Scheduler
  - Port
  - Network
  - Virtual Network
  - Message
- Possibility to run until a stable state
- Possibility to define
  - Input parameters
  - Maximum difference for stable state
  - Credit-based shaper bandwidth reservation

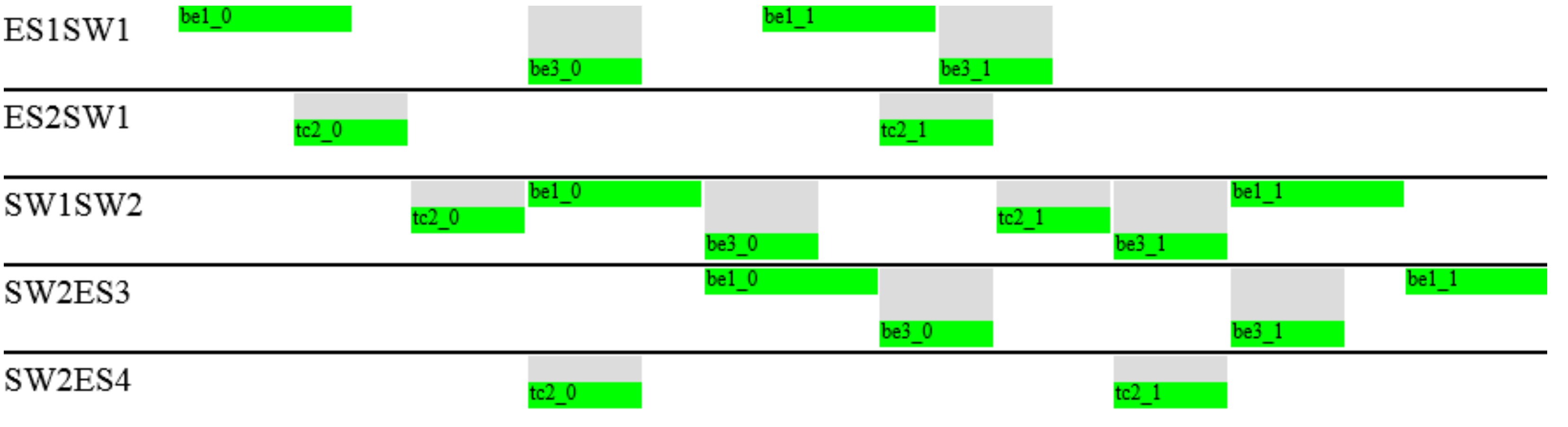
Simulator – Implementation

- Network is split to the whole network and virtual links
  - Virtual link is a subset of the network



Evaluation – Validation

- A test run where
  - be1 – low priority BE frame
  - tc2 – medium priority time-critical frame
  - be3 – high priority BE frame
  - Not a very likely situation but shows that protected windows work!



Evaluation – Test cases

- Three test cases
- Each test case has three variations
  - Original
  - Some BE-traffic added
  - Some more BE-traffic added
  - BE-traffic added with characteristics in real network
    - A lot of small (under 100 bytes) and large frames (over 1400 bytes), rest being somewhere between
- Test case 1
  - 20 TT-frames, 26 AV-frames
  - 12 ESs, 4 NSs
- Test case 2
  - 58 TT-frames, 51 AV-frames
  - 10 ESs, 5 NSs
- Test case 3
  - 92 TT-frames, 81 AV-frames
  - 35 ESs, 8 NSs

Evaluation – Results: WCD differences for AV-frames

- Qbv = protected windows
- Qbu = preemption
- Qav = credit-based shaper

Test case	Qbv	Qbv+Qbu	Qbv+Qav	Qbv+Qbu+Qav
1	0 %	−7.1 %	7.0 %	−1.1 %
1b	0 %	−15.2 %	8.9 %	−9.3 %
1c	0 %	−19.8 %	24.0 %	−14.2 %
2	0 %	−4.6 %	9.1 %	3.6 %
2b	0 %	−5.2 %	17.1 %	3.3 %
2c	0 %	−6.1 %	20.5 %	3.2 %
3	0 %	−5.0 %	3.6 %	−0.4 %
3b	0 %	−5.0 %	10.9 %	−1.4 %
3c	0 %	−5.8 %	24.3 %	−2.3 %

Evaluation – Results: ACD for BE-frames

Test case	Qbv	Qbv+Qbu	Qbv+Qav	Qbv+Qbu+Qav
1b	305	304	304	303
1c	413	393	411	393
2b	482	510	388	475
2c	494	518	407	494
3b	436	460	385	450
3c	593	579	506	575

Evaluation – Results: protected windows' effects

- Average WCD of TT-frames

Test case	Qbv	None	Difference
1	321	441	37%
3	674	829	23%
3c	674	916	36%

- Average WCD of AV-frames

Test case	Qbv	None	Difference
1	588	492	16%
3	133	125	6%
3c	135	128	5%

Evaluation – Results: findings

- Time-critical frames flow through the network as planned
- Preemption favors AV-frames
- Credit-based shaper favors BE-frames
- Preemption + credit-based tries to favor both AV and BE-frames

Evaluation – Limitations

- Stable state after three consecutive WCD differences stay within limits
  - Room for errors
- The simulator allows AV-frame to preempt BE-frame at any point, when it has enough credits
- Hardware delays are not considered
- Only one frame per message
- The network is homogeneous
- Time accuracy is 80 ns

Conclusion

- There are still things to determine
  - When frames can be preempted?
  - How CBS, protected windows and preemption works together?
- The simulator allows studying feature set effects on the network
- Protected windows provide timing guarantees
- The simulator is fast enough to study small to medium sized networks
  - Still room to improve its performance

Thank you!