PhD course:
Real-Time and Dependable Ethernet Communication with IEEE TSN Standards

Soheil Samii – course leader and examiner

6 hp – HT2021

LINKÖPING UNIVERSITY

Course topics

- Introduction to distributed embedded real-time systems
- Ethernet, AVB, and TSN
- Clock synchronization
- Real-time packet scheduling
- Ethernet frame preemption
- Fault tolerance
- Security protocols

Examination

- Written term paper (5-6 pages)
- Presentation of term paper at final seminars
- Read term papers of other course participants prior to final seminars

Topic selection for term paper

- You should select a topic for your term paper by end of the lectures
- List of suggested topics will be provided
- Office hours reserved for each course participant to discuss topics

Final seminar

- Presentation of term papers
- 20 minute per presentation (including Q&A)
- Group 1: February 4, 2022: 15:15-17:00
- Group 2: February 7, 2022: 14:45-17:00

- Key dates and details here: https://www.ida.liu.se/~sohsa65/courses/tsn-course-2021/final-seminar-ht21.pdf

Course folder

- https://www.ida.liu.se/~sohsa65/courses/tsn-course-2021/
- Includes course schedule, lecture notes, and topic/literature suggestions for term paper

Embedded systems introduction

Soheil Samii

LINKÖPING UNIVERSITY

Embedded systems

- There are many different definitions!
  - “A special-purpose computer system that is used for a particular task.”
  - “A computer based systems embedded in real life machines. Though computer based, it dose not have the usual key-board and monitors.”
- Some highlights what it is (not) used for:
  - “Any device which includes a programmable component but itself is not intended to be a general purpose computer.”
- Some focus on what it is built from:
  - “A collection of programmable parts surrounded by ASICs and other standard components, that interact continuously with an environment through sensors and actuators.”

Real-Time System

- “An information processing system that has to respond to input stimuli within a finite and specified time.”
  - The correctness depends not only on the logical result but also the time it was delivered.
  - Failure to respond in time is as bad as the wrong response!

Examples of embedded real-time systems

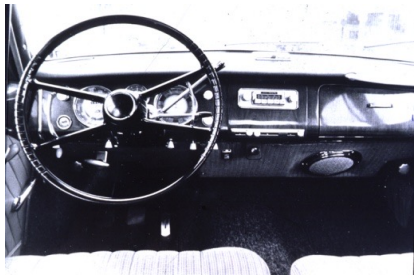
Characteristics of embedded systems

- Dedicated computers (not general purpose).
  - One or several applications known at design-time.
- Contain a programmable component.
  - But usually not programmable by the end-user.
- Interact (continuously) with the environment:
  - Real-time behavior (faster ≠ better).
  - Predictable, safe and reliable.
- Usually very cost sensitive:
  - Products in competitive markets, demanding low cost.
- Low power/energy is often preferred.
  - Battery life: High energy consumption ⇒ short battery life time.
  - Cost issue: High power consumption ⇒ strong power supply and expensive cooling mechanism.

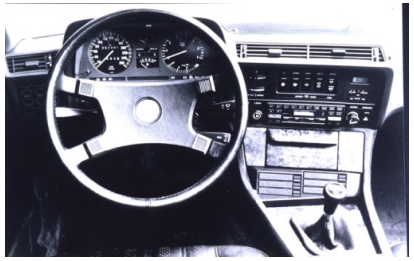
Many nonfunctional requirements

- Cost
- Real-time
- Dependability (fault tolerance, security)
- Maintainability (e.g., diagnostics and service)
- Extensibility (for future features)
- Scalability (across multiple product lines)
- Connectivity (WiFi, cellular, Internet)
- Low footprint

Innovation in automotive



Electronic Injection
Check engine control
Cruise control
Central locking
…



Gearbox control
Climate control
ASC Anti Slip Control
ABS Anti -lock Brake Sys.
Telephone
Seat heating control
Automatic mirrors

Navigation system
CD-changer
Active Cruise Control
Airbags
Dynamic Stability Control
Roll stabilization
Xenon lighting
Vehicle Assist
Voice input
Emergency call

ACC Stop&Go
Lane departure warning
Blind spot warning
Traffic sign recognition
Night vision
Active headlight system
Parking automation
Efficient dynamics
Hybrid engines
Internet access
Telematics
Online Services
Bluetooth integration
Local Hazard Warning
Personalization
SW Update
Smart Phone Apps

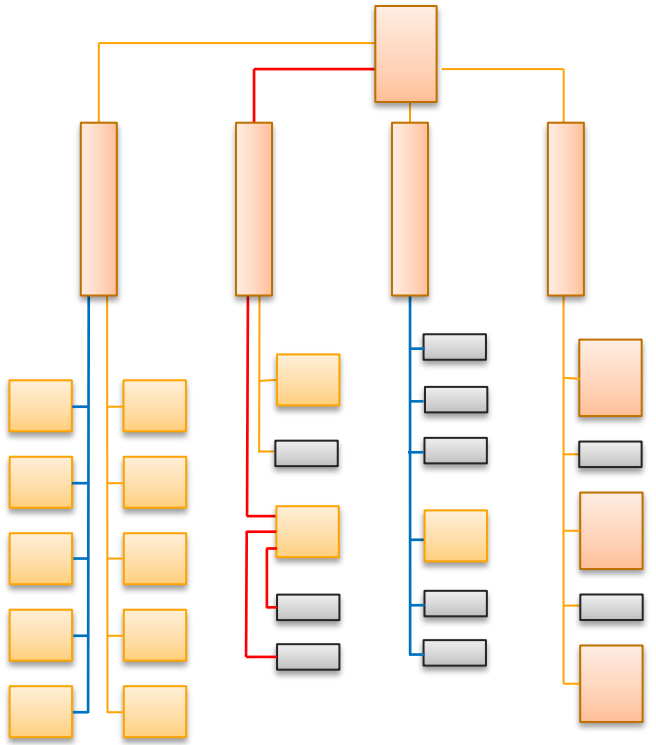
1970 1980 1990 > 2010

Automotive: Harsh environmental conditions

- Body & cabin: -40°C to 85°C
- Chassis & powertrain: -40°C to 125°C or even 150°C
- Mechanical accelerations
- Not a data center: Dirt, water, salt, dust, ice, snow, mud, oil, grease, transmission fluid, brake fluid, engine coolant, …
- Qualification of hardware is required (tight EMC requirements)
- Functional Safety/ASIL (ISO 26262) compliance

Current E/E architectures

- Domain based
- Special-purpose ECUs with specialized I/O for the application
- Low-speed networks (1 Mbps is rare)
- Hardware and software tied together with direct I/O for the particular application
  - Brake control
  - Electric Power Steering
  - Body control electronics
- Leads to a lot of modules, a lot of wiring, and a lot of silos



Automotive wiring

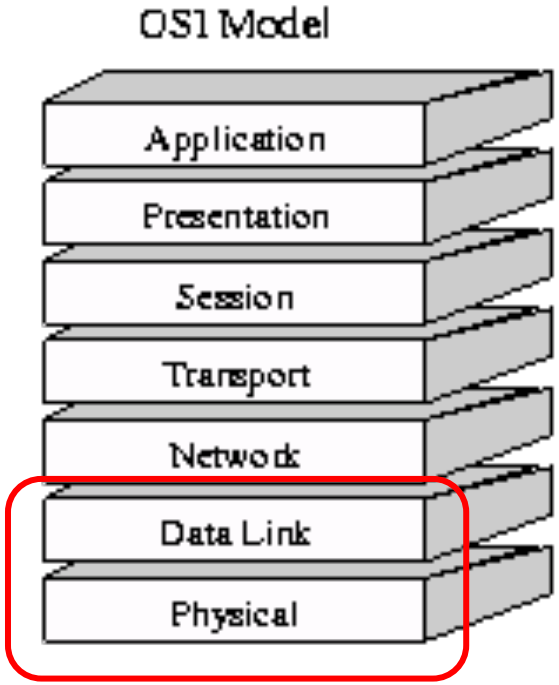
- Cost and mass top 3
- Huge warranty cost
- Kilometers of wiring

Why automotive Ethernet?

- Growing bandwidth requirements to push data around in and out of the vehicle
- Integration and fusion of sensor data
- New sensors for active safety & autonomous:
  - Camera, radar, lidar, HD maps
- Infotainment (audio, video, navigation, speech recognition, displays)
- V2V, V2I, and Internet connectivity
- Cloud computing and crowd-sourcing
- Up-integration of Electronic Control Units

Recent IEEE 802 standards for Ethernet

Real-time / dependability
- AVB and TSN standardized in IEEE 802.1
- Packet scheduling, traffic shaping
- Enhanced error detection and redundancy
- Security



Bandwidth
- PHYs for various data rates standardized in IEEE 802.3
- Single unshielded twisted pair copper cable
- 10MBit/s (multi-drop, bus)
- 100 Mbit/s, 1 Gbit/s, ... 10Gbit/s (full-duplex)

Relevant IEEE 802 standards

- IEEE Std 802.1Q: Defines behavior of switches and end stations in an Ethernet network
- IEEE Std 802.1BA: Audio/Video Bridging (AVB)
- IEEE Std 802.1AS: Clock synchronization
- IEEE Std 802.1Qbv-2015: Scheduled traffic (time-triggered). Also called TAS (Time Aware Shaper)
- IEEE Std 802.1Qbu-2016: Frame preemption. Allocation of queues to express vs. preemptable MACs
- IEEE Std 802.3br-2016: Companion standard 802.1Qbu. Describes how preemption is done on Ethernet

Relevant IEEE 802 standards

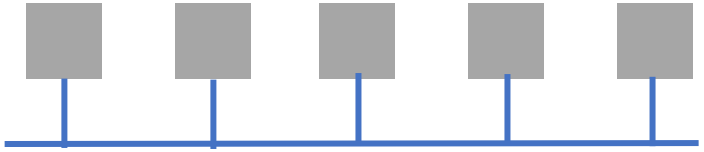
- IEEE Std 802.1Qcr-2020: Asynchronous traffic shaping
- IEEE Std 802.1Qci-2017: Ingress filtering and policing. Used to detect and isolate errors.
- IEEE Std 802.1CB-2017: Redundancy. Replication and elimination of packets
- IEEE Std 802.1AE-2006 (+ various amendments): MACsec. Integrity and privacy on switched Ethernet with symmetric-key crypto
- IEEE Std 802.1X: Port authentication and key agreement protocol
- IEEE Std 802.1AR: Secure device identity with public-key crypto (elliptic curves)

Ethernet and TSN introduction

Soheil Samii

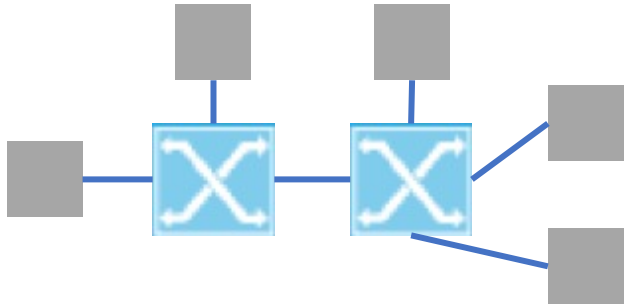
LINKÖPING UNIVERSITY

Ethernet CSMA/CD



- Created in the mid 70s
- Ethernet bus: shared media access
- 46 to 1500 byte payloads; 10 Mbps to 10 Gbps
- Node transmits as soon as medium is free
- Collisions can occur during the interval of one slot after start of transmission (512 bits)
- Jamming signal (32 bits) is sent in case of collisions
- Retransmission after random amount of time
- Outdated
- Not appropriate for real-time systems

Switched Ethernet



- IEEE 802.1D and 802.1p. Now all in 802.1Q with Virtual LANs (VLANs) and priority queues
- Better but still gaps for real-time and safety-critical systems:
  - Priority inversions in FIFO queues
  - Interference through shared memory
  - Additional forwarding delay with address learning and flooding
  - Delays vary with switch technology
  - “Plug and Play” has been the mindset: lots of protocols that were not primarily designed for real-time applications (long delays and nondeterminism)

Ethernet frame format

EtherType examples
0x0800 (IPv4), 0x86DD (IPv6), 0x88F7 (PTP), 0x88E5 (MACsec)

802.3 Ethernet packet and frame structure

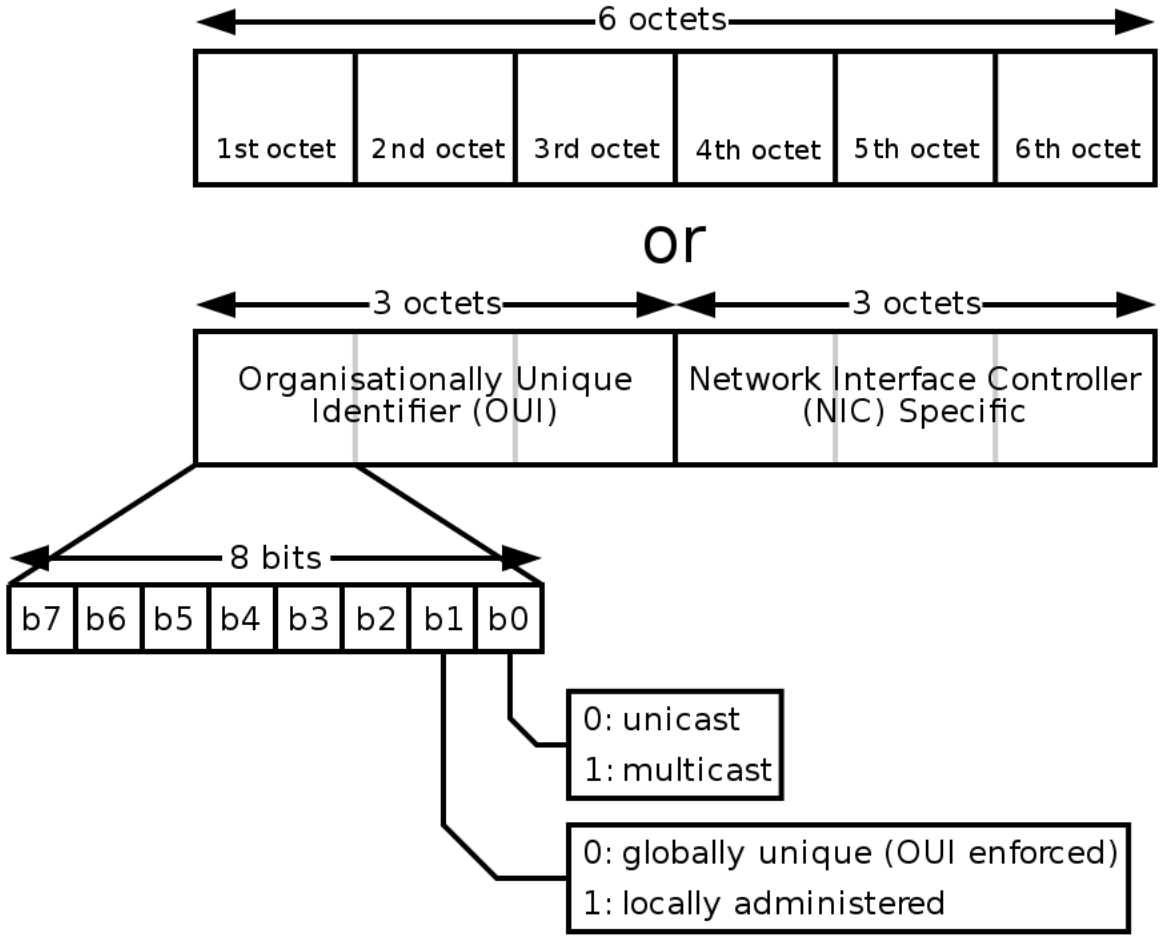
Layer	Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46-1500 octets	4 octets	12 octets
Layer 2 Ethernet frame			← 64–1522 octets →						
Layer 1 Ethernet packet & IPG	← 72–1530 octets →								← 12 octets →

802.1Q tag format

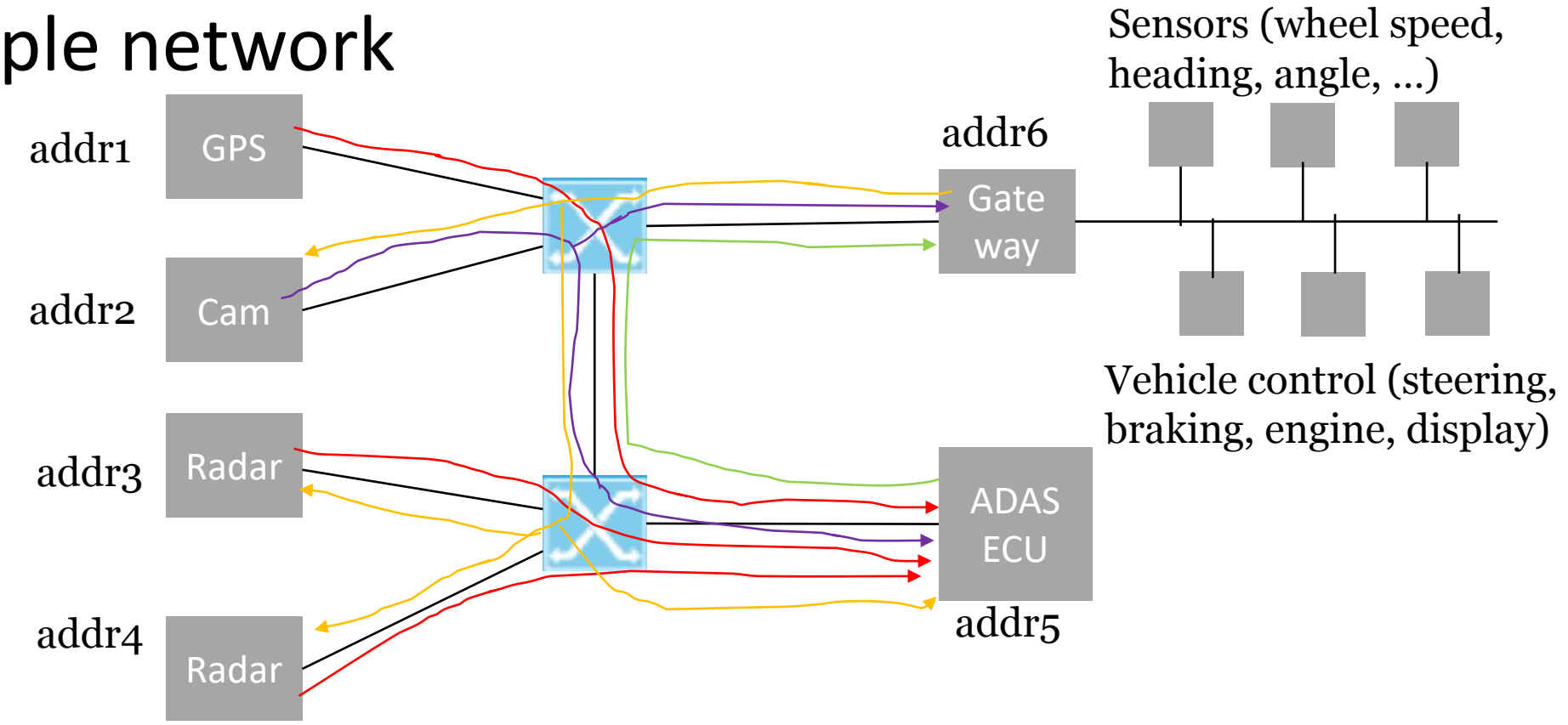
16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	DEI	VID

- Unicast frames (Direct addressing of the (only) receiver)
- Multicast frames (when there are multiple receivers)
  - Some reserved for Layer 2 protocols (e.g., Spanning Tree Protocol and Precision Time Protocol)
  - AVB and TSN streams are typically multicast
- Broadcast frames (FF:FF:FF:FF:FF:FF)
  - Flood frame on all ports

MAC address


6 octets
1st octet
2nd octet
3rd octet
4th octet
5th octet
6th octet
or
3 octets
3 octets
Organisationally Unique Identifier (OUI)
Network Interface Controller (NIC) Specific
8 bits
b7
b6
b5
b4
b3
b2
b1
b0
0: unicast
1: multicast
0: globally unique (OUI enforced)
1: locally administered

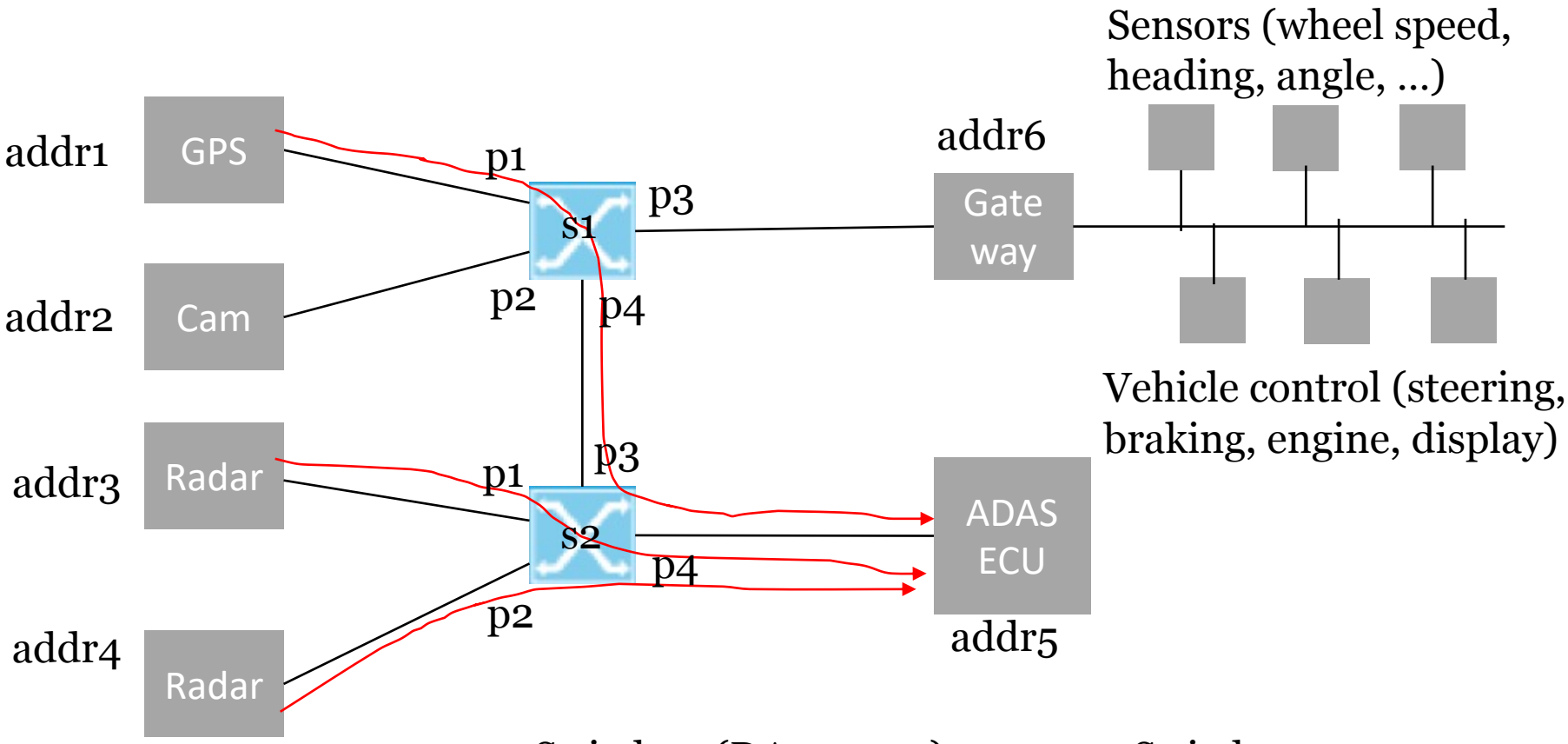
Example network



Streams:

- **GPS and Radars to ADAS ECU**
- **Cam to ADAS ECU and Vehicle domain**
- **Vehicle sensors to Cam, Radars, and ADAS ECU**
- **ADAS ECU to Vehicle domain (actuation and corrected GPS)**

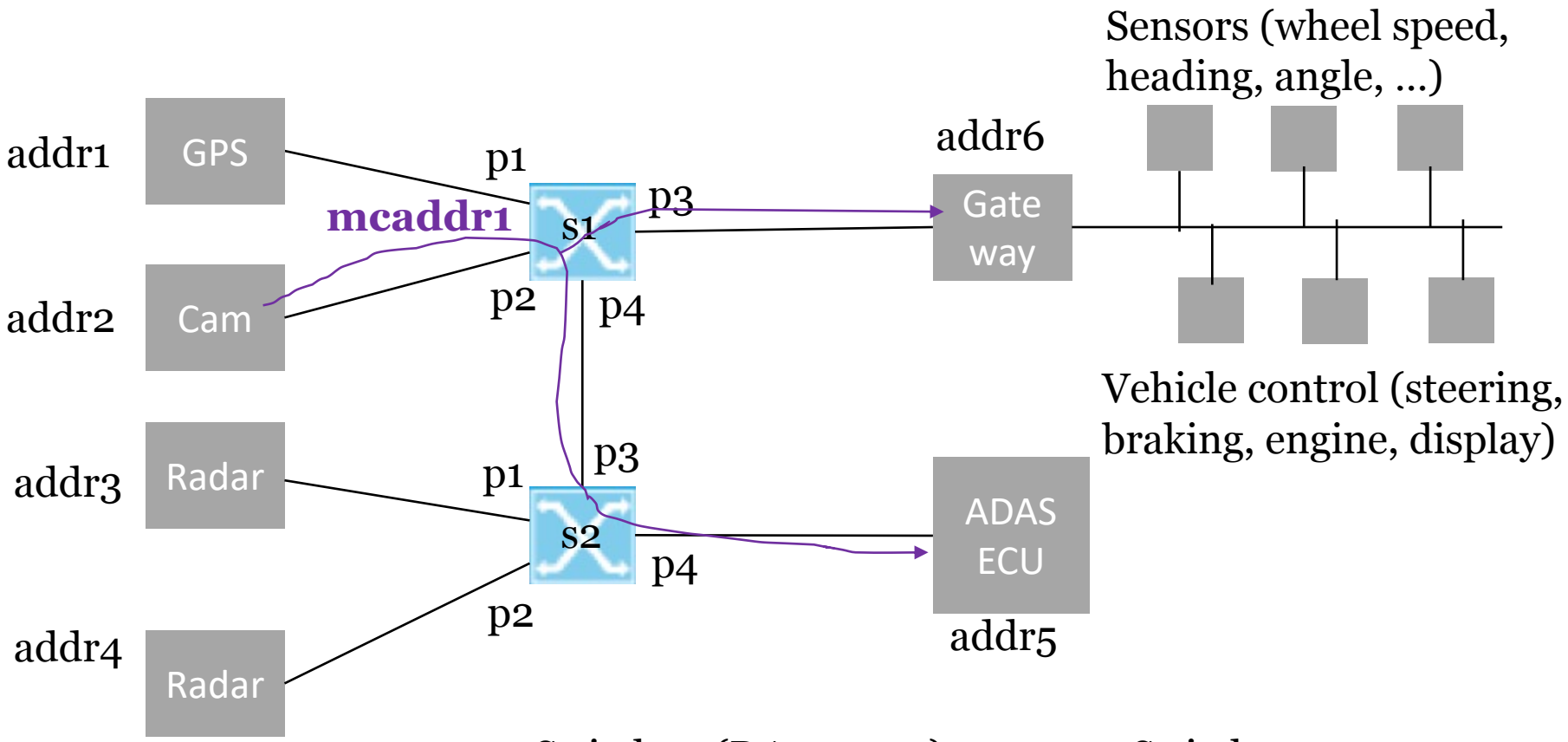
Forwarding tables



Switch s1 (DA -> port):
addr5 -> p4

Switch s2:
addr5 -> p4

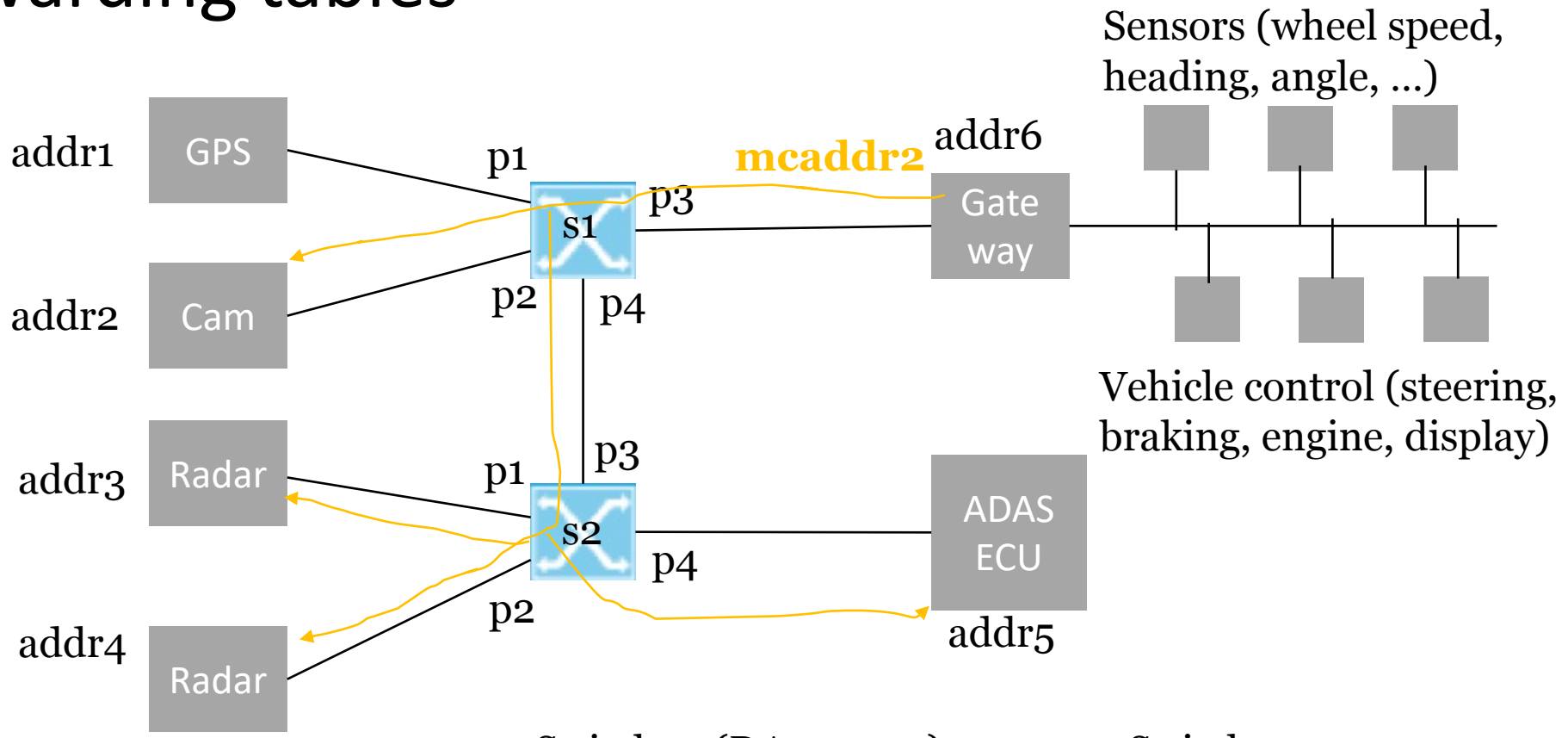
Forwarding tables



Switch s1 (DA -> port):
addr5 -> p4
mcaddr1 -> p3, p4

Switch s2:
addr5 -> p4
mcaddr1 -> p4

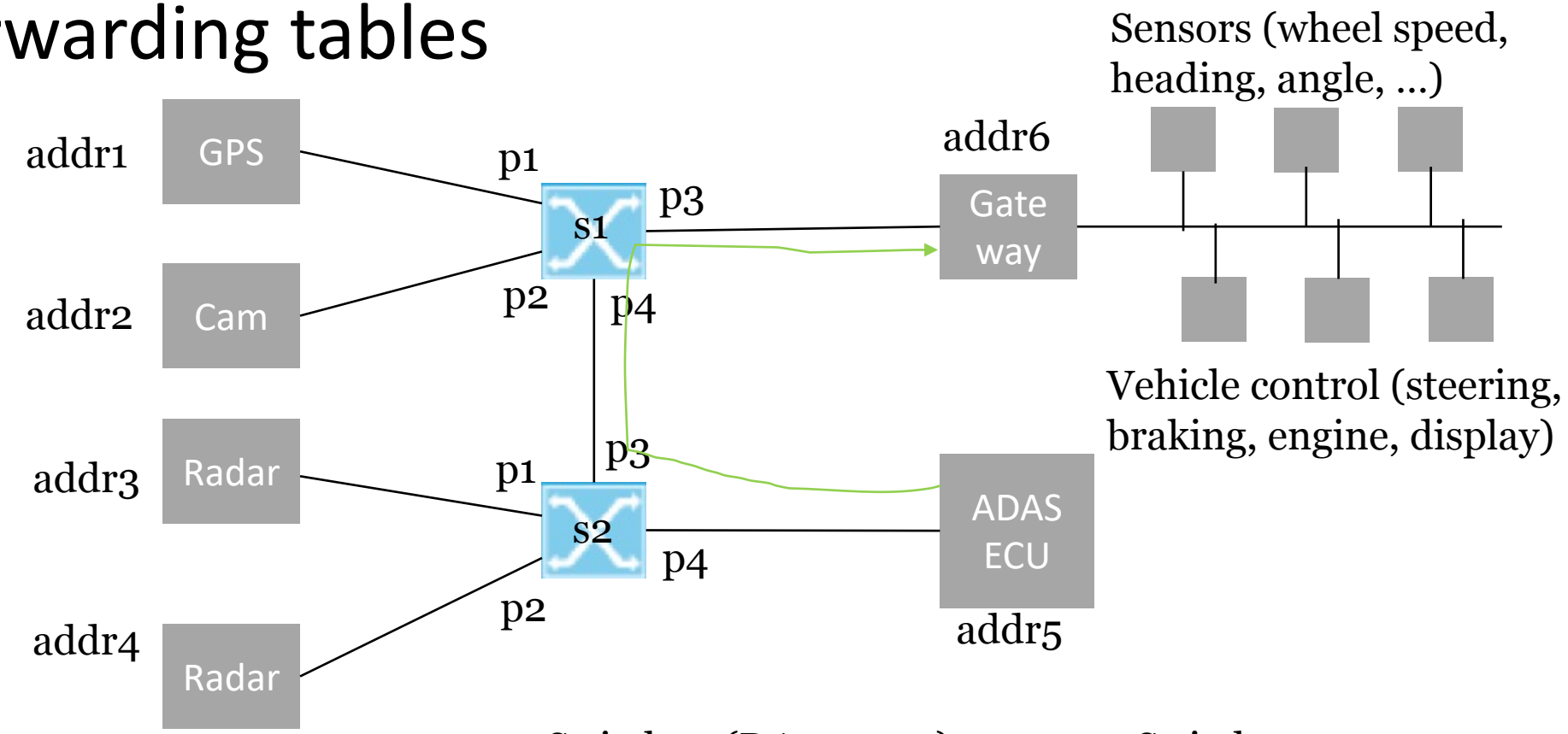
Forwarding tables



Switch s1 (DA -> port):
addr5 -> p4
mcaddr1 -> p3, p4
mcaddr2 -> p2, p4

Switch s2:
addr5 -> p4
mcaddr1 -> p4
mcaddr2 -> p1, p2, p4

Forwarding tables



Switch s1 (DA -> port):
addr5 -> p4
mcaddr1 -> p3, p4
mcaddr2 -> p2, p4
addr6 -> p3

Switch s2:
addr5 -> p4
mcaddr1 -> p4
mcaddr2 -> p1, p2, p4
addr6 -> p3

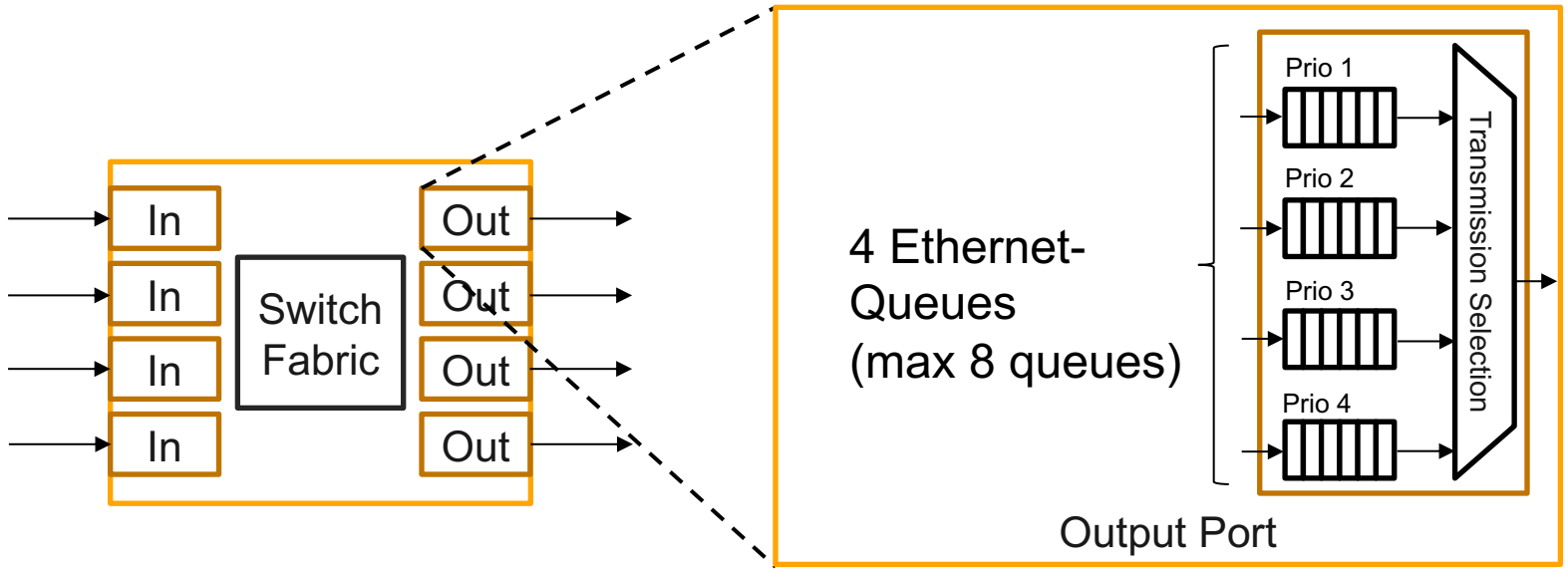
Table can also be a mapping from DA and VLAN ID to ports

Configuration of forwarding tables

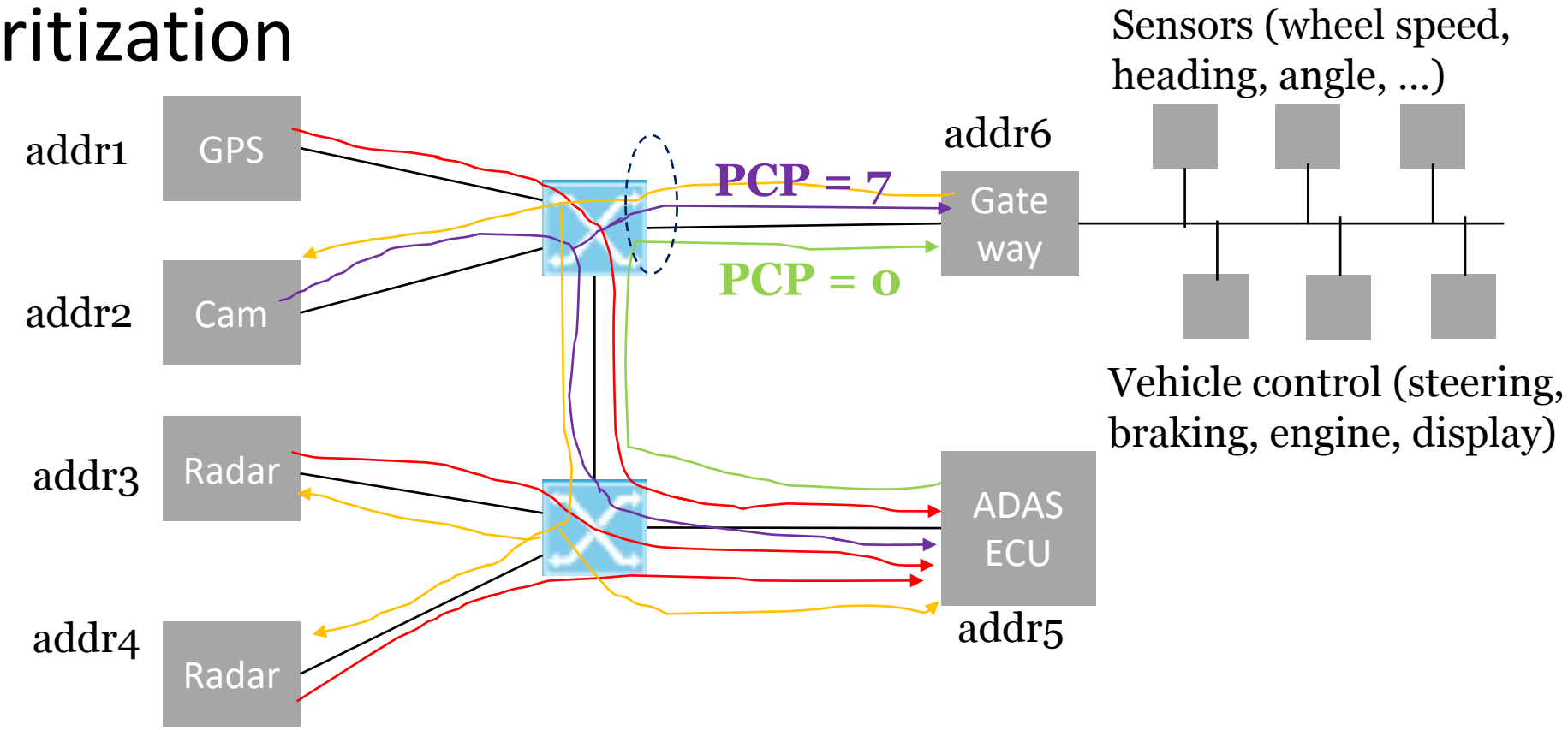
- For plug & play networks:
  - Flooding
  - Learning based on mapping ingress ports to source addresses (unicast)
  - Multiple stream registration protocol (multicast)
- Engineered networks (e.g., automotive, avionics)
  - Static configuration of forwarding tables
  - Protocols for learning are switched off (also helps to reduce risks for cybersecurity attacks)

Ethernet switch (“bridge” in IEEE 802.1)

- Frame carries priority
- Frames are assigned to egress ports (forwarding)
- Priorities are mapped to egress queues
- FIFO queues



Prioritization

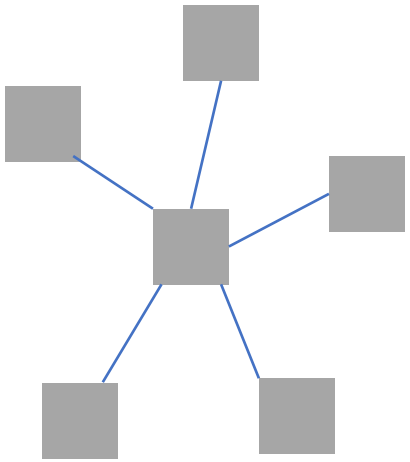


Streams:

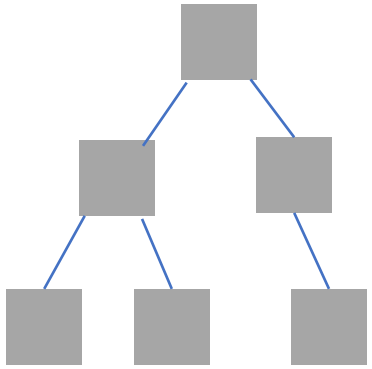
- **GPS and Radars to ADAS ECU**
- **Cam to ADAS ECU and Vehicle domain**
- **Vehicle sensors to Cam, Radars, and ADAS ECU**
- **ADAS ECU to Vehicle domain (actuation and corrected GPS)**

Different topologies

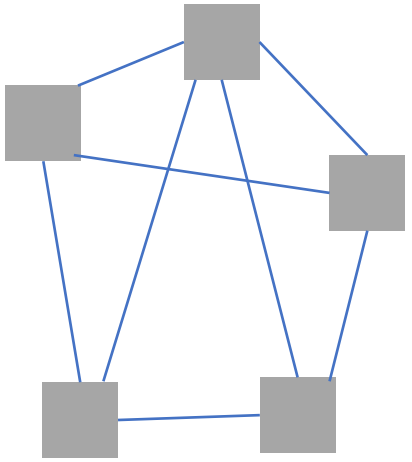
Star



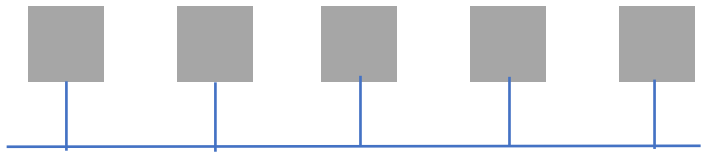
Tree



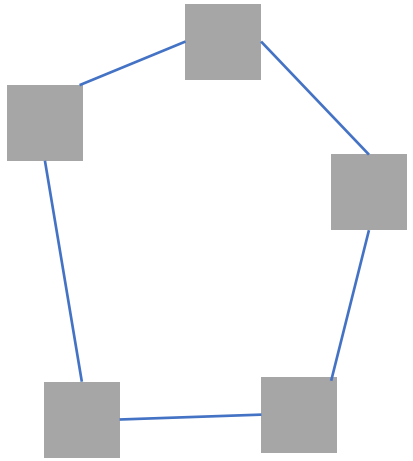
Mesh



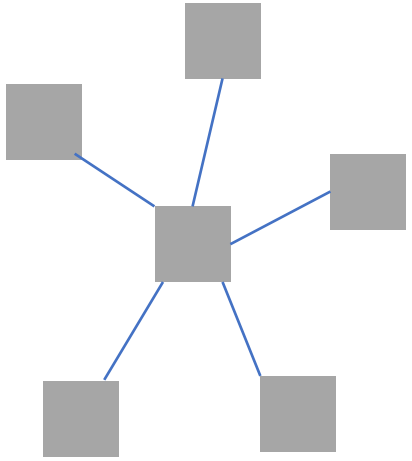
Bus

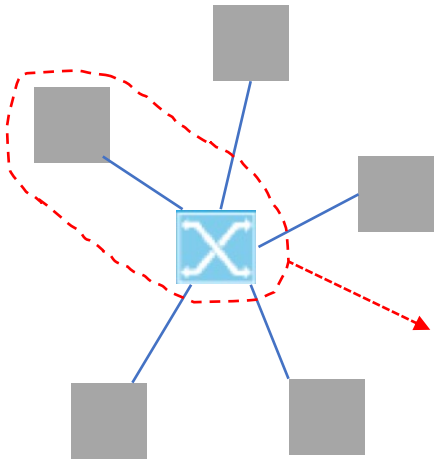


Ring



Star topology



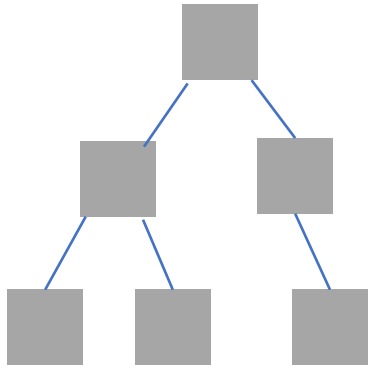


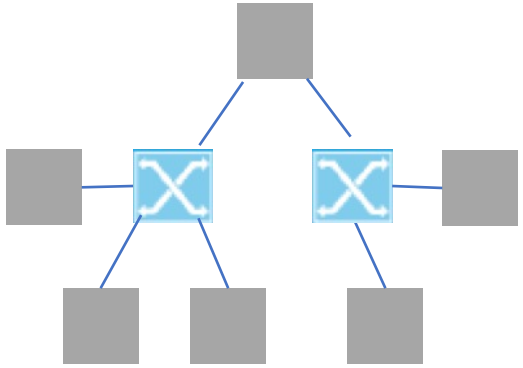
Some observations:

- One hop communication to all (low latency)
- Low cost
- Switch failure disconnects all communication
- Long wires, depending on physical location of endpoints

- Can be integrated in one module with MII, no PHY
- Integration is product and topology dependent

Tree topology

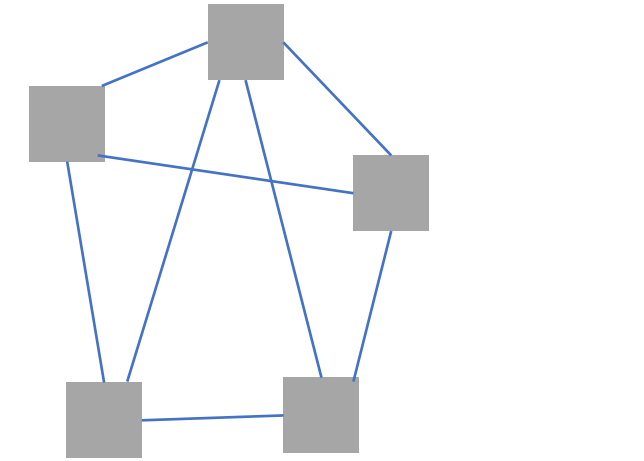


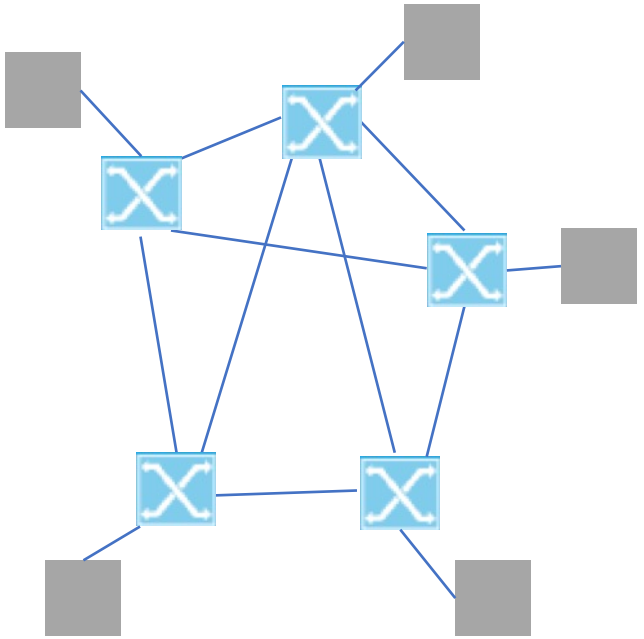


Some observations:

- Multiple hops between some nodes
- Switch failure disconnects several nodes
- Good for networks where communication mainly goes in one direction
- Segmented left and right side

Mesh topology

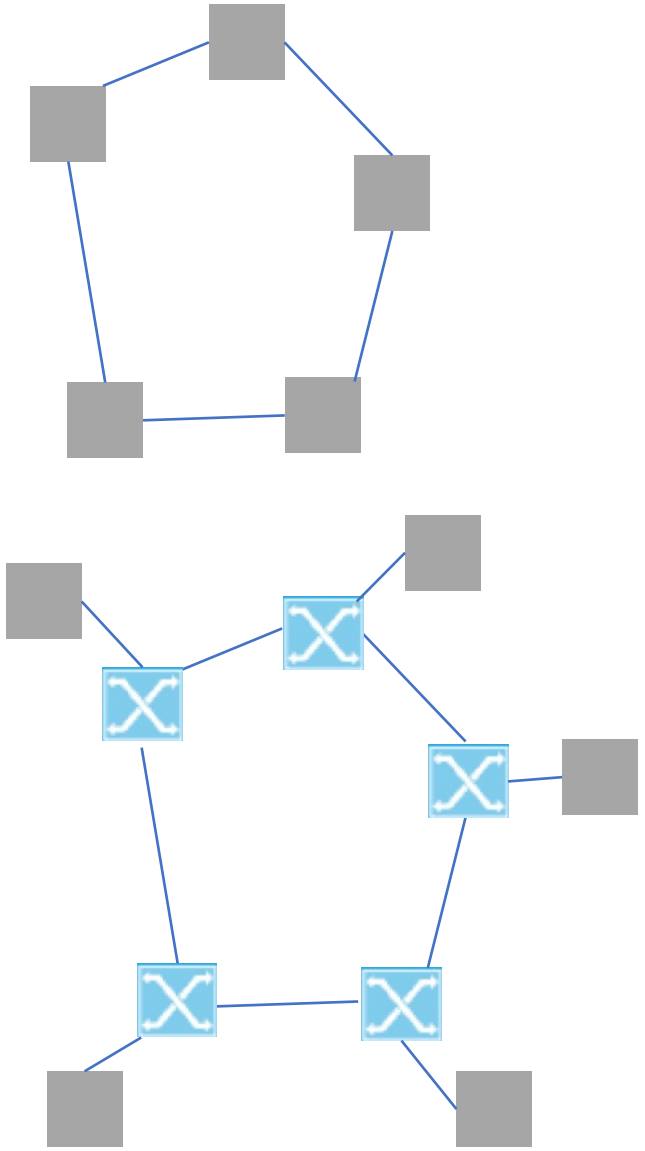




Some observations:

- Multiple hops between some nodes
- Reduced impact of one switch failure
- Multiple communication paths (easier to meet timing and redundancy needs)
- High cost due to many switches

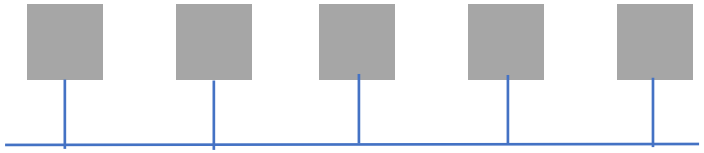
Ring topology



Some observations:

- Latency does not scale well with number of nodes
- Redundancy at lower cost than mesh solution
- Still many switches, but of lower port count
- Switch failure “only” disconnects one node

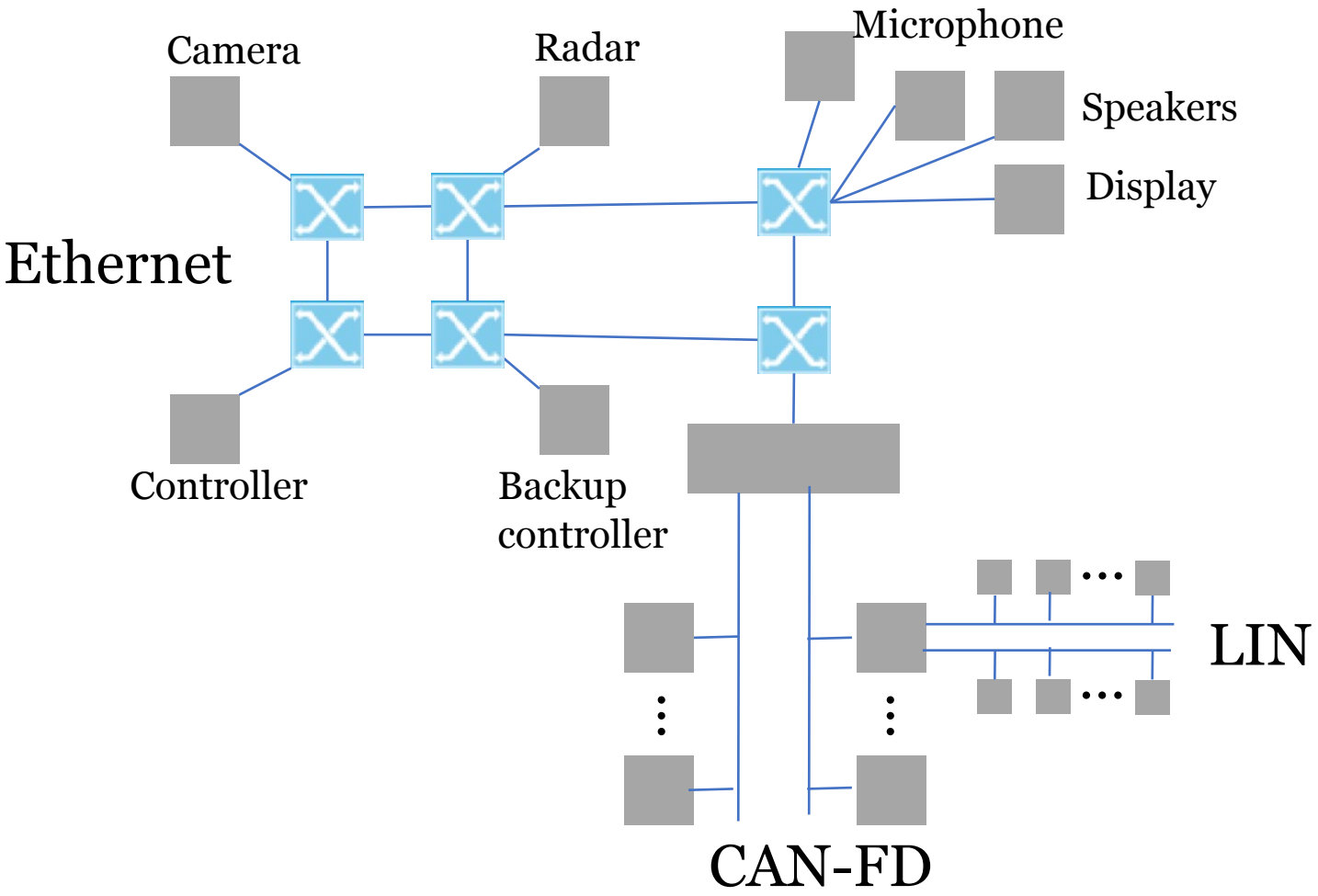
Bus topology



Some observations:

- 10BASE-T1S, IEEE Std 802.3cg
- Low cost (no switches)
- Single broadcast domain
- Shared media access
- 10BASE-T1S, IEEE Std 802.3cg, specifies PLCA (Physical Layer Collision Avoidance)
  - Master node sends beacon indicating new cycle
  - Nodes are allocated bandwidth during cycle in a round robin manner

Network is usually a mix of topologies and technologies



Adding bandwidth is not enough

- Support for real-time and safety-critical applications
- Several industries with similar communication requirements at OSI Layer 2 and above
  - Professional audio/video
  - Industrial automation
  - Automotive
  - Telecom
- Standardization development in IEEE 802.1
  - First, AVB. Now, TSN
  - Multi-hop, switched topologies make certain problems more complicated than bus-based real-time communication

Audio/Video Bridging (AVB)

- Main drivers:
  - synchronized audio and video applications on Ethernet
  - Plug and play
- IEEE 802.1AS: Plug and play Clock synchronization
- Amendments to IEEE 802.1Q:
  - Stream reservation protocol (admission control)
  - Credit-based traffic shaping (guaranteed bandwidth across priority levels; no starvation; zero congestion loss / no dropped packets)
- IEEE 802.1BA: Umbrella document

What was still missing?

- Guaranteed lower latencies; searching for the lowest possible latency
- Error detection and isolation
- Redundant communication
- Redundant clock synchronization

New TSN standards

- TSN: Group renamed itself from “Audio/Video Bridging” to “Time Sensitive Networking”

- Guaranteed lower latencies; searching for the lowest possible latency
  - 802.1Qbv: A priori defined, pre-scheduled communication
  - 802.1Qbu: Frame preemption
  - 802.3br: Preemption required changes in MAC
  - P802.1Qcr: Asynchronous traffic shaping

New TSN standards

- Error detection and containment
  - 802.Qci-2017: Monitoring of pre-defined “contract” between a data flow and the network
  - Meaning of “Contract” depends on the scheduling policy
- Redundant communication
  - 802.1CB-2017: Frame replication and elimination
- Redundant clock synchronization
  - 802.1AS-2020: updates to the clock synchronization standard to support backup masters and multiple time domains

Security standards

- 802.1 has developed three main standards for secure Ethernet communication

- 802.1AE (“MACsec”): integrity and, optionally, privacy, is ensured by symmetric key crypto (128 and 256 bit keys are supported)

- 802.1X:
  - Port authentication
  - Key agreement protocol

Security standards

- 802.1AR (Secure Device Identity):
  - Device identity based on public key crypto (elliptic curves)
  - Public Key Infrastructure and certificates

- These standards, in particular MACsec and Secure Device Identity, require parallel crypto logic on chip
  - Challenging in terms of cost, power, and size
  - Automotive industry has started to look at how to secure Ethernet communication and which portions of the 802.1 security standards apply

Many nonstandard solutions before TSN (still in use)

- Ethernet POWERLINK (Master/Slave protocol; open source)
- EtherCAT (2-port switch in each node; cut-through forwarding
  - Beckhoff automation (IEC 61158)
- PROFINET (TDMA)
  - Siemens
- AFDX (static configuration, traffic shaping) (Airbus)
- TTEthernet by TTTech (TDMA at message level, and rate-constrained traffic class).

Summary

- Ethernet is a point-to-point network
- Direct addressing with unicast and multicast
- Switches forward frames towards the destination(s)
- Ports have multiple queues assigned to different priority levels
- Many different Ethernet topologies
- AVB and TSN standards added real-time and dependability capabilities

Packet scheduling - AVB

Soheil Samii

Motivation for traffic shaping

- A basic best-effort Ethernet network provides variable service (varying packet delivery latency and sometimes packet get dropped)
- Why does it happen?
  - Overbooking / Overloading. Sources are pushing data that is greater than the capacity of the network elements (links, buffers)
  - Traffic jams due to bursts of traffic from sources
  - Traffic piles up at network “intersections” (i.e., switches/bridges) to go out on the same port

Prioritization

- Prioritize Traffic: Handle packets carrying deterministic traffic in high priority output queues
  - Implement a range of output queues from low- to high-priority on each output port.
  - Prioritized packets are sent even when lower-priority packets are waiting.
  - A burst of high-priority traffic interrupts lower-priority traffic.
  - Good for high-priority traffic. ***Bad/unfair*** to lower-priority traffic.

Contracts and reservations

- Use the privilege of prioritization according to an agreed “contract”
  - Each prioritized source agrees to a contract (aka, a “reservation”) with the network that provides service acceptable to that source. This agreement could be made dynamically or be “engineered” into the network ahead of time.
  - Reservation ensures that prioritized traffic won’t overwhelm best-effort (low-priority) traffic.
  - Traffic shaping is at the heart of a reservation

Shaping approaches

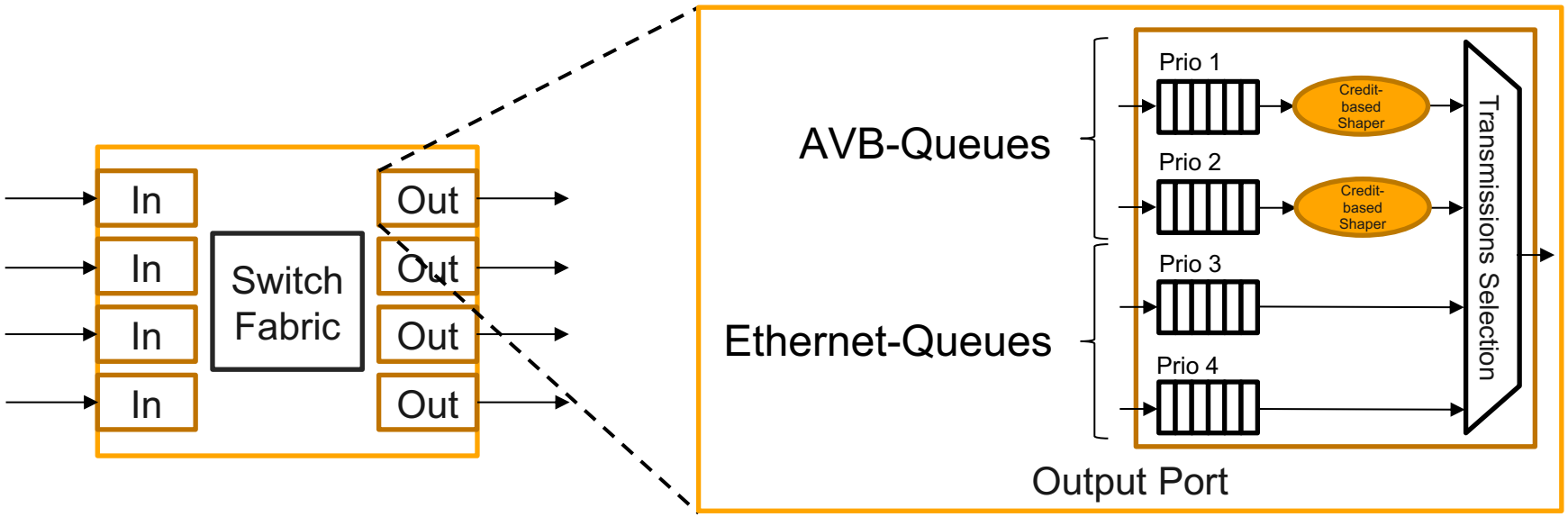
- Credit Based Shaper (CBS): Provides a smooth stream of packets within a maximum data rate
  - Also called FQTSS (Forwarding and Queuing for Time Sensitive Streams) in 802.1Qav, now part of 802.1Q-2014
- 802.1Qbv: Time Aware Shaper (TAS): Provides access to queue at specified times
- 802.1Qcr: Asynchronous Traffic Shaper (ATS): Provides immediately delivery of packets, up to a specified burst size, and within a maximum data rate

First approach: AVB

- 2 ms maximum delay
  - the maximum delay between a musician doing “something” and hearing that same “something” is 10 ms
  - the transit time of sound from monitor speakers to the musician, plus digital signal processing (DSP) delays, plus mixer delays, plus more DSP delays uses up 8 ms
  - network gets 2 ms
- maximum synchronization error less than 10 microseconds

Quality of Service

- Credit based shaper delays messages to avoid bursts
  - To avoid buffer overflow and message loss
  - To guarantee some bandwidth to lower priority traffic



AVB credit-based shaper

- Space out the high priority stream frames as far as possible
- The spaced-out traffic prevents the formation of long bursts of high priority traffic, which typically arise in traffic environments with high bandwidth streams
- Bursts are responsible for significant QoS reductions of lower priority traffic classes
  - Can completely block the transmission of the lower priority traffic for the transmission time of the high priority burst
  - Increases maximum latency of this traffic and thereby also the memory demands in the bridges and end stations.

AVB credit-based shaper

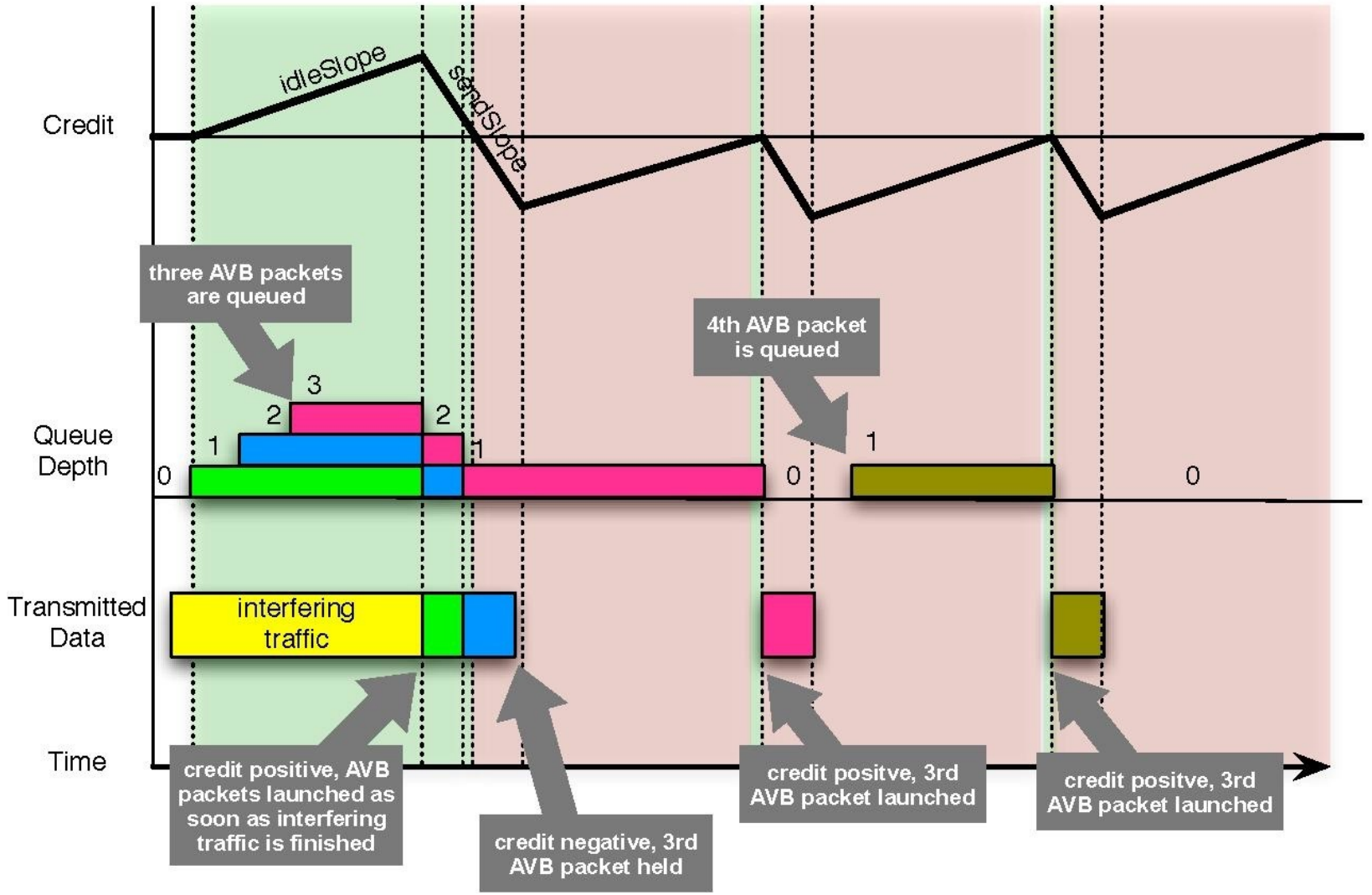
- Long bursts increase the interference time between high priority stream frames from different streams (which arrive from different ports) inside a bridge.
  - This increases the maximum latency of high priority stream frames and again the memory requirements in bridges
- Another task of the shaper is to enforce the bandwidth reservations. This enforces, on the one hand, that every AVB stream is limited to its reserved bandwidth in the talker, and, on the other hand, that the overall AVB stream bandwidth of each port (in talker and bridges) is limited to the reserved amount

Operation of credit-based shaper

$$\text{idleSlope} = \frac{\text{reservedBytes}}{\text{classMeasurementInterval}}$$

$$= \text{reservedBandwidth}.$$

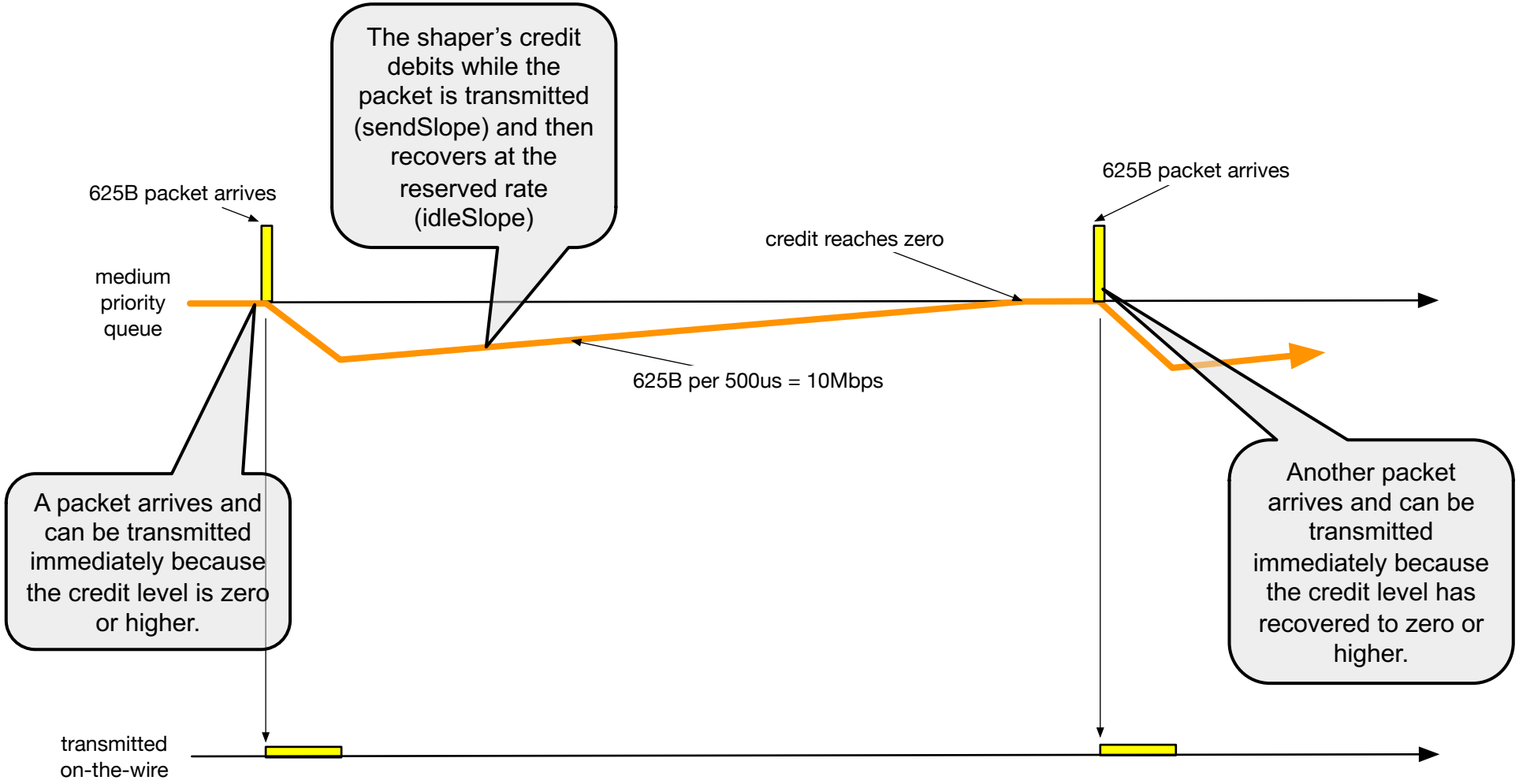
$$\text{sendSlope} = \text{idleSlope} - \text{portTransmitRate}$$



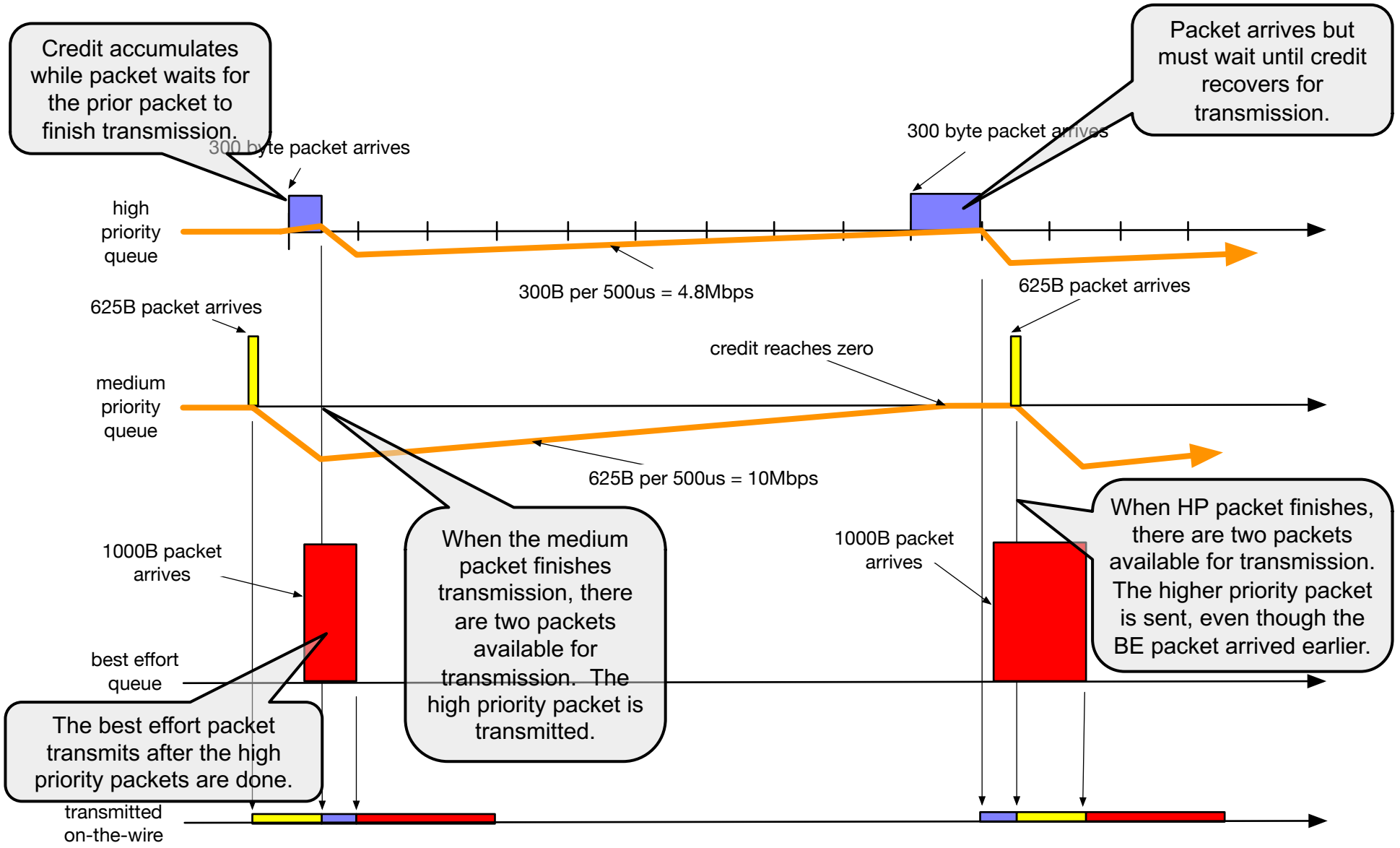
Credit calculation rules

- If there is positive credit but no frame to transmit, the credit is set to zero
- During the transmission of a frame, the credit is reduced with the send slope.
- If the credit is negative and no frame is in transmission, the credit is accumulated with the idle slope until zero credit is reached.
- If there is a frame in the queue that cannot be transmitted because another frame is in transmission, the credit is accumulated with the idle slope. In this case, the credit is not limited to zero.

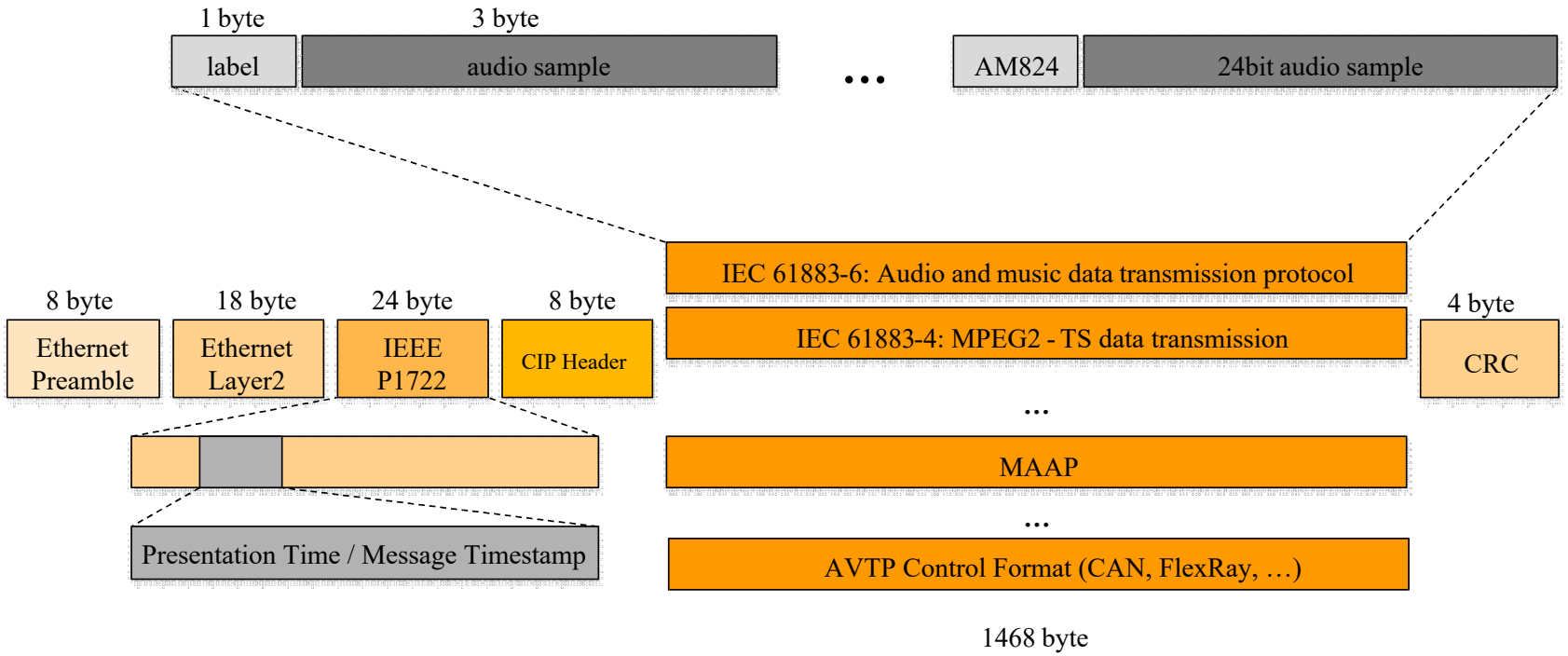
Credit-based shaping (802.1Q-2014 §34)



CBS with multiple queues



IEEE 1722: AVTP (AVB Transport Protocol)

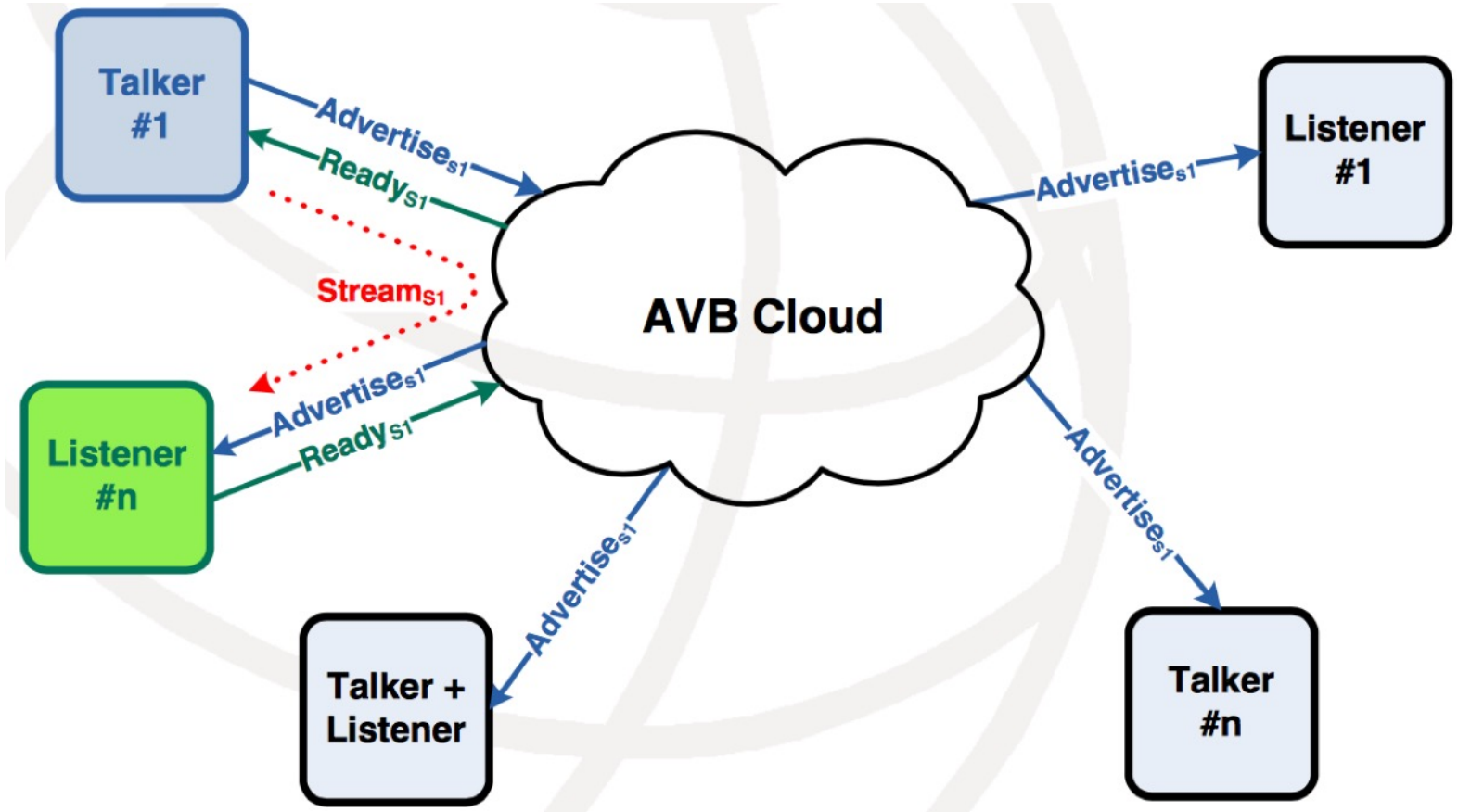


- AVTP control format added in 2016
- EtherType: 0x22F0

Stream reservation protocols

- 802.1Qat (now rolled into 802.1Q-2014 and later revisions)
- One of the core protocols of AVB
- Allows sources (talkers) to advertise streams to sinks/users (listeners) through the network
- Also allows to withdraw
- Gives end stations the tool to automatically configure the network to deliver content to the right users
- Multiple Stream Registration Protocol (MSRP)
- Multiple VLAN Registration Protocol (MVRP)
- MSRP and MVRP are in turn based on the Multiple Registration Protocol (MRP)

SRP Advertise and Ready frames



Talker advertise message format

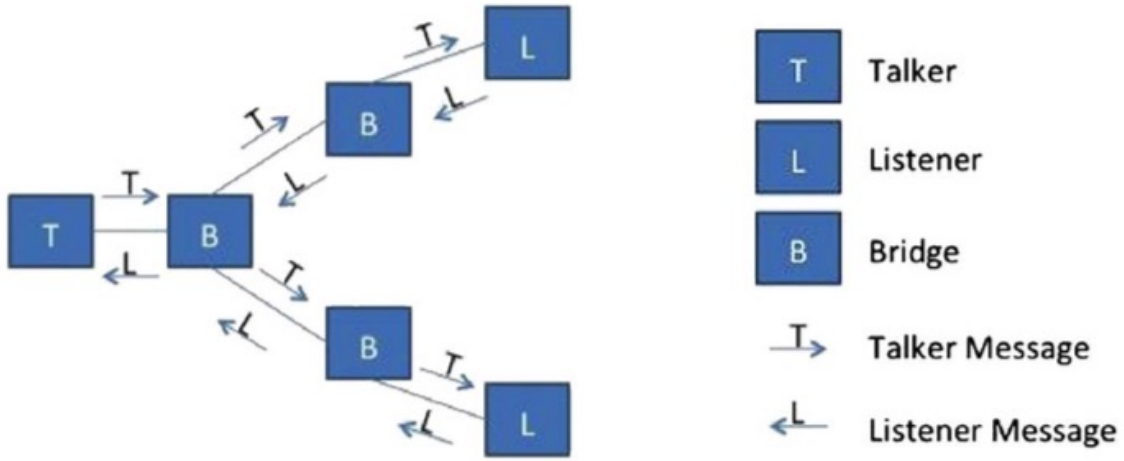
- stream ID (MAC address associated with the talker plus a 16 bit ID)
- stream DA
- VLAN ID
- priority (determines traffic class)
- rank (emergency or nonemergency)
- traffic specification (TSpec): max frame size; maximum number of frames per class interval
- accumulated latency

Forwarding of stream announce

- Talker send advertise message
- Each switch/bridge evaluates whether reservation can be made
  - whether sufficient bandwidth is available on each port
  - whether sufficient memory is available to guarantee no packet loss
  - reservation is not made; only when receiving listener message
  - forwards the talker message, after updating the accumulated the hop count

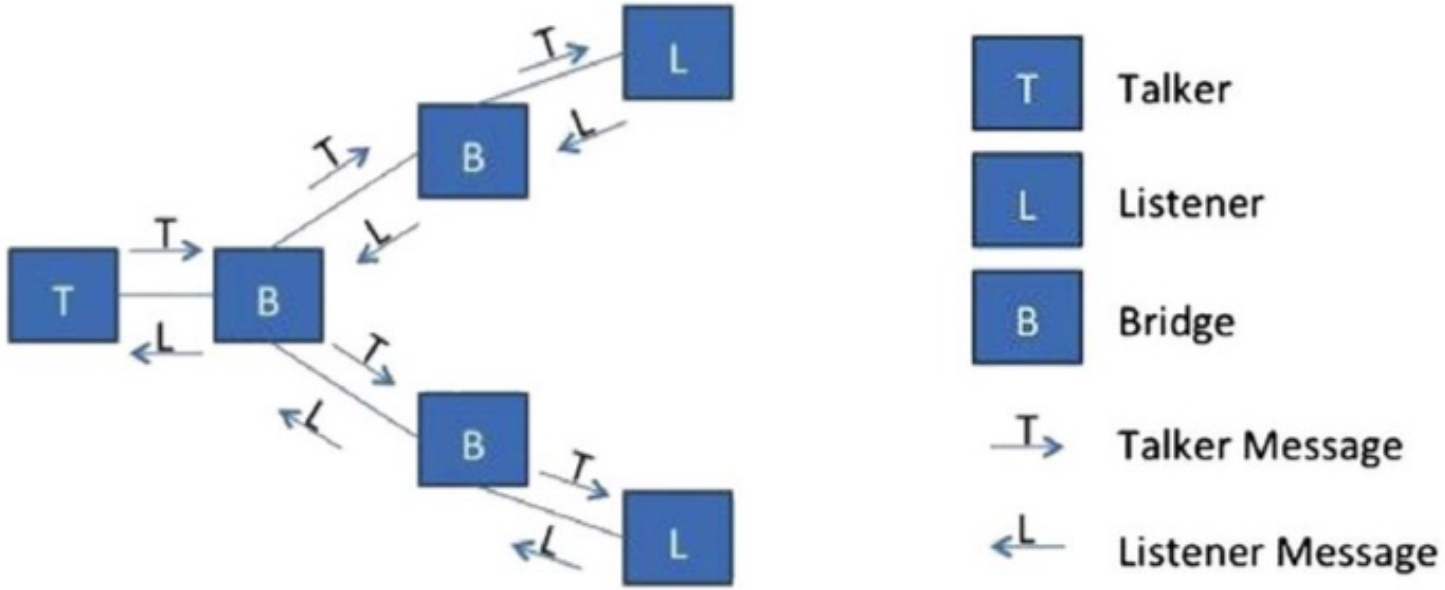
Reservation failures

- If any device on the path from talker to listener determines that the stream cannot be supported, it changes the type of the message from talker advertise to talker failed
- Then adds additional information to the message
  - bridge ID where the failure occurred;
  - reservation failure code to identify the reason for the failure.
  - Allows network engineer to pinpoint the location of the issue



Listeners

- Listeners send listener message if they want the stream
- Listener communicates the status of the stream by sending either a listener ready if it received a talker advertise or a listener asking failed if it received a talker failed



Reservations made

- When bridges receive a listener ready (or ready failed) message for a valid stream on a given port, they make a reservation on that port
  - update the bandwidth on the traffic shaper for the queue

$$\text{idleSlope} = \frac{\text{reservedBytes}}{\text{classMeasurementInterval}} = \text{reservedBandwidth}.$$

  - update available bandwidth for the given port
  - adding the port to the forwarding entry for the stream DA

Listener messages propagated back

- Listener message propagated back toward the talker
- Talker receives a listener ready message, it may begin transmitting
- If talker receives ready failed, it knows that at least one listener has requested the stream but the corresponding reservation could not be created

For engineered (static) networks

- Use SRP to establish data paths and bandwidth reservations once
  - Then program components with the resulting configuration
- “Manual” static configuration or network design tool

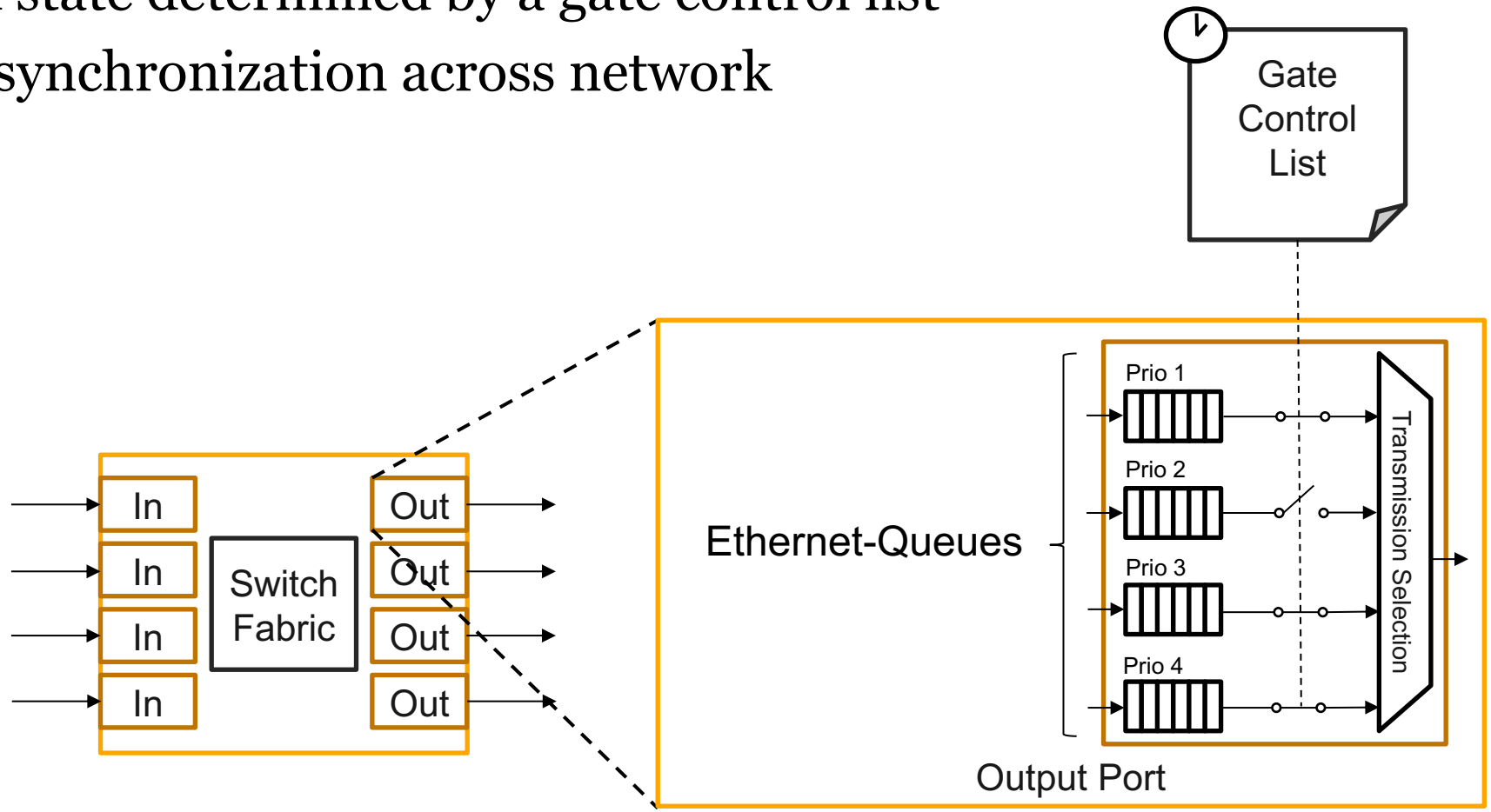
Packet scheduling - TSN

Soheil Samii

LINKÖPING UNIVERSITY

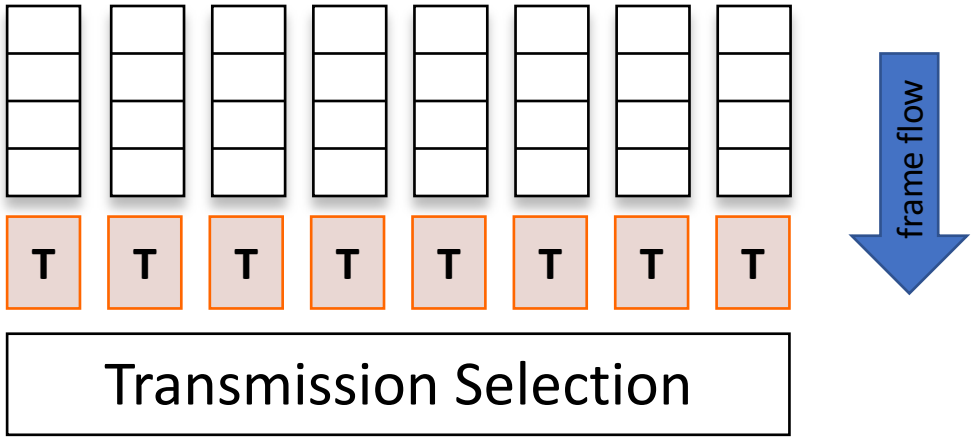
IEEE 802.1Qbv: Time-Aware Shaper (TAS)

- Time gate on queue
- Open or closed state determined by a gate control list
- Requires time synchronization across network

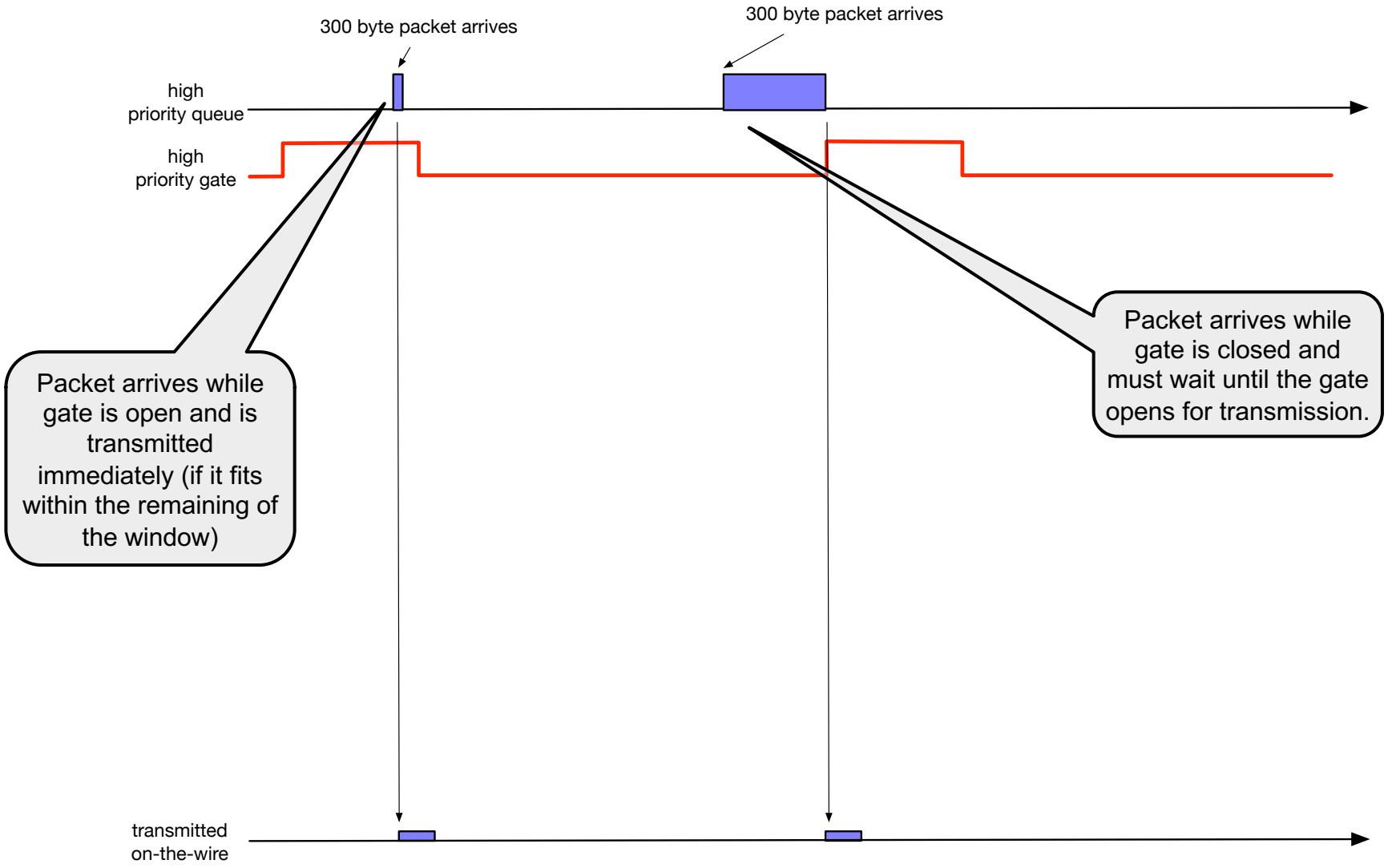


Scheduled traffic

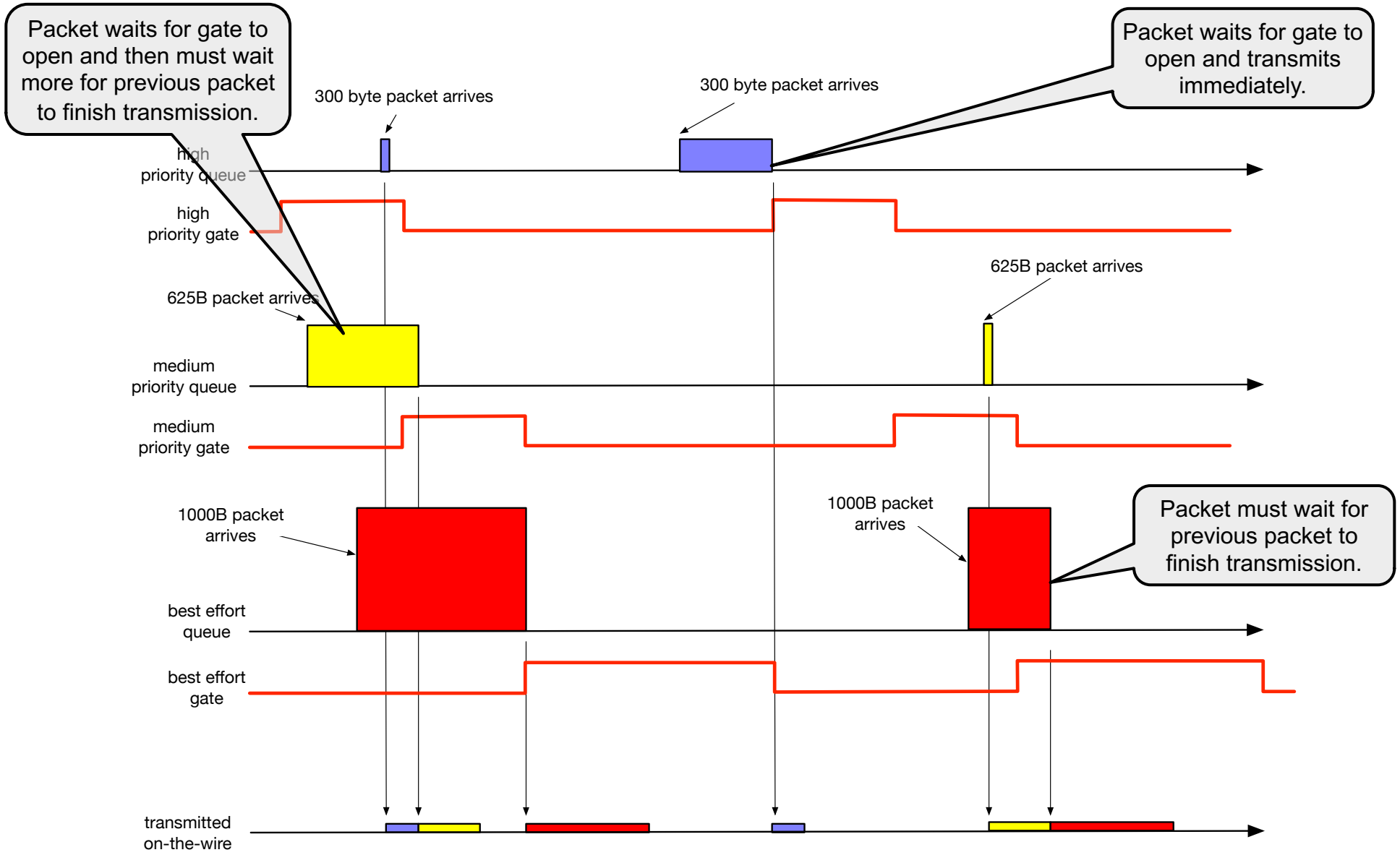
- Reduces latency variation for Constant Bit Rate (CBR) streams, which are periodic with known timing
- Time-based control/programming of the up to 8 bridge queues (802.1Qbv)
- Time-gated queues
- Gate: Open or Closed
- Periodically repeated time-schedule
- Time synchronization is needed



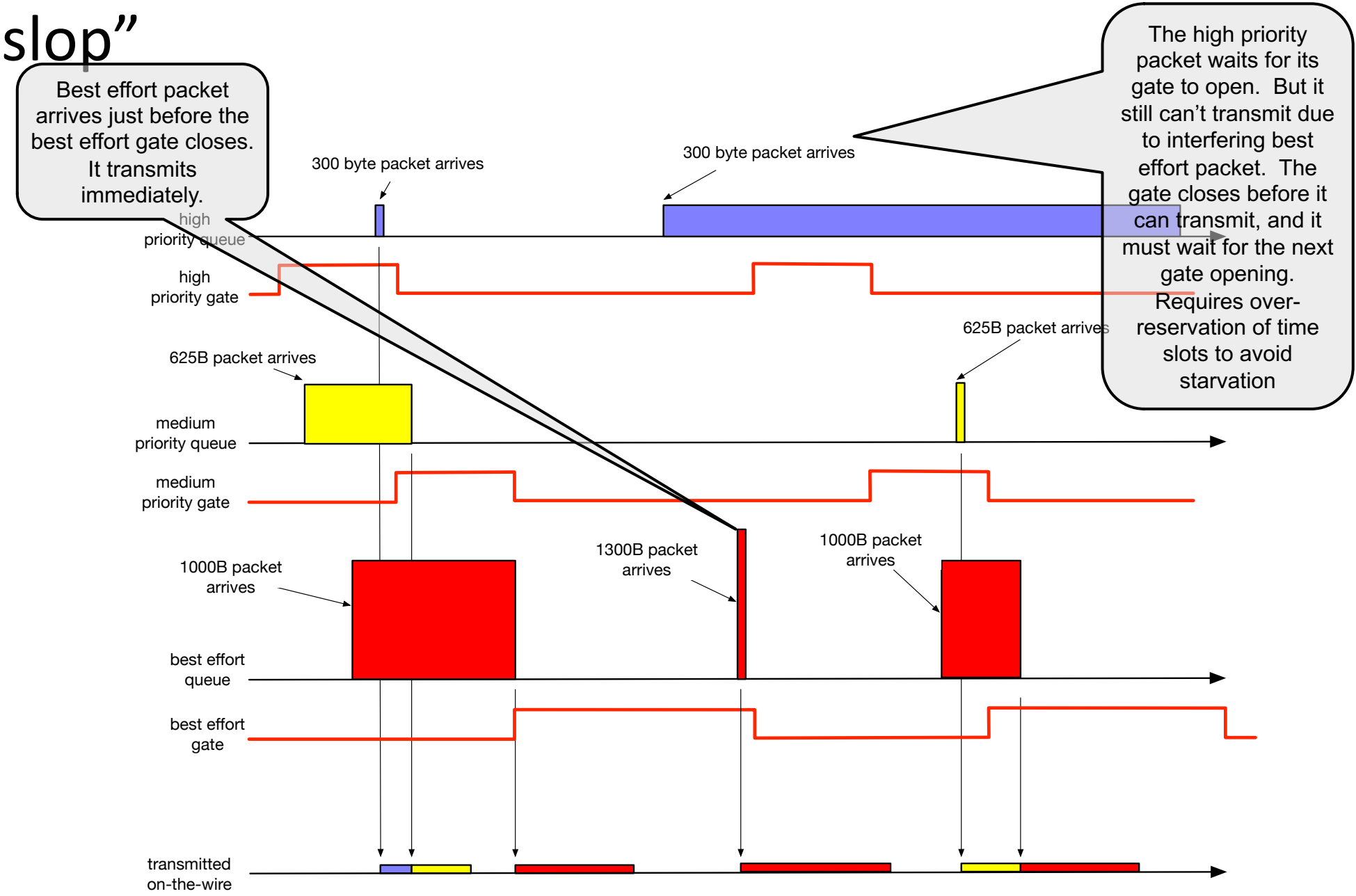
TAS – one queue



TAS – multiple queues



TAS - ”slot slop”



802.1Qbv TAS observations

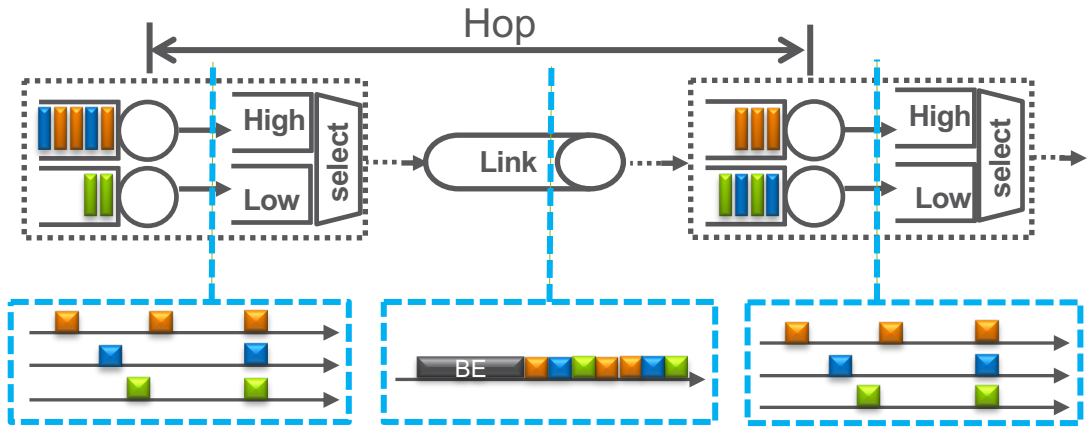
- Pre-defined time access to queues
- Suitable for highly engineered networks
- Suitable for carrying streams with common and regular structure (e.g., sensors that send small packets at very regular periodicity)

802.1Qbv TAS observations

- Engineering the network can be difficult: depending on stream makeup, queue scheduling can be difficult to optimize or create
- Careful engineering to maximize efficiency and avoid “slot slop”
  - Add “guard band” to time slots
  - Eliminate non-engineered traffic
  - Use MACs with frame preemption capability (802.1Qbu-2016 / 802.3br)
- Likely need to synchronize software execution on nodes to avoid missing open Qbv windows

802.1Qcr-2020: Asynchronous traffic shaping (ATS)

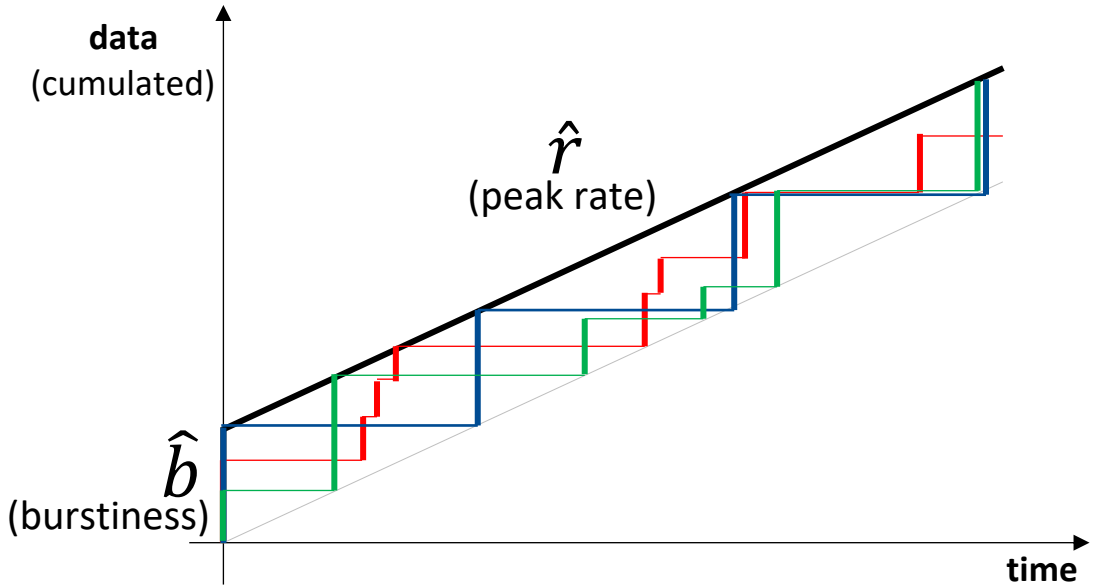
- Zero congestion loss without time synchronization
- Asynchronous Traffic Shaping (P802.1Qcr ATS)
  - Smoothen traffic patterns by re-shaping per hop
  - Prioritize urgent traffic over relaxed traffic



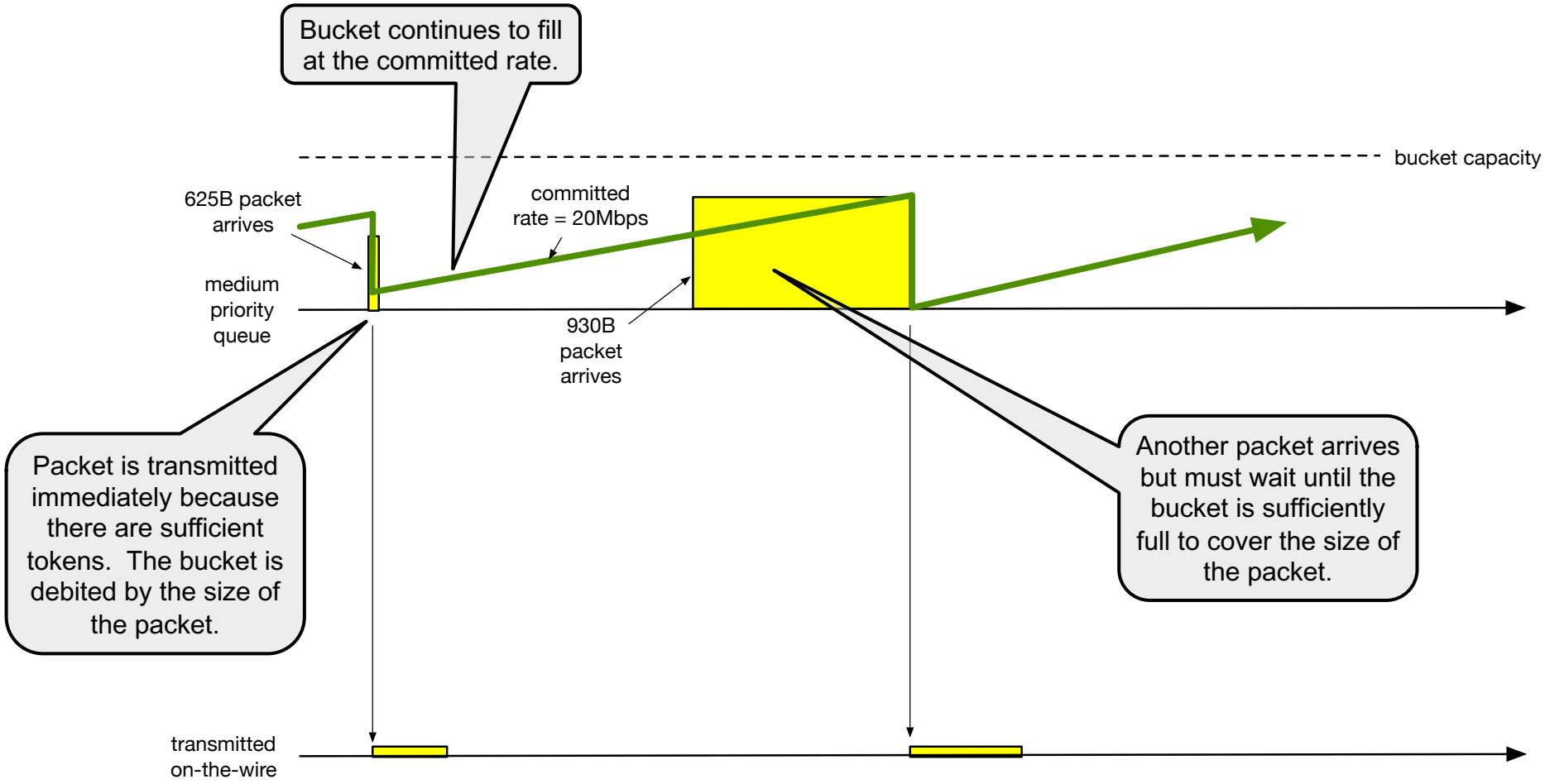
ATS

- ATS is based on the token bucket algorithm
- Shaper has a token bucket that fills with tokens at a committed bit rate, until it reaches the maximum capacity
- Packets arrive at random times and with random size
- Shaper releases packets for transmission scheduling when the bucket holds tokens greater than or equal to the size of the packet, followed by incrementing the bucket size
- If there is an insufficient number of tokens to release a packet for transmission scheduling after a maximum residence time has elapsed, the shaper discards the packet

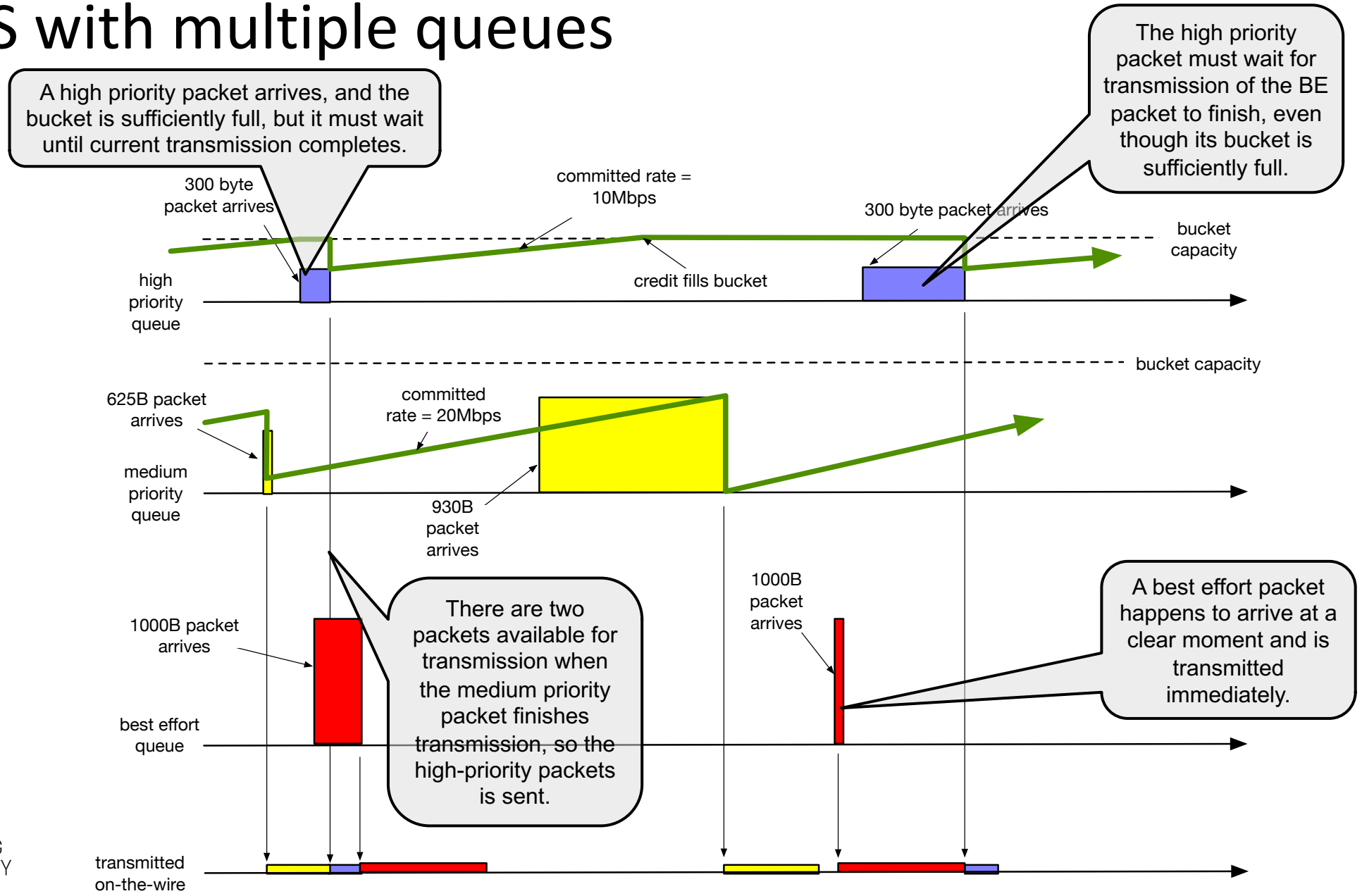
Traffic model



ATS operation



ATS with multiple queues



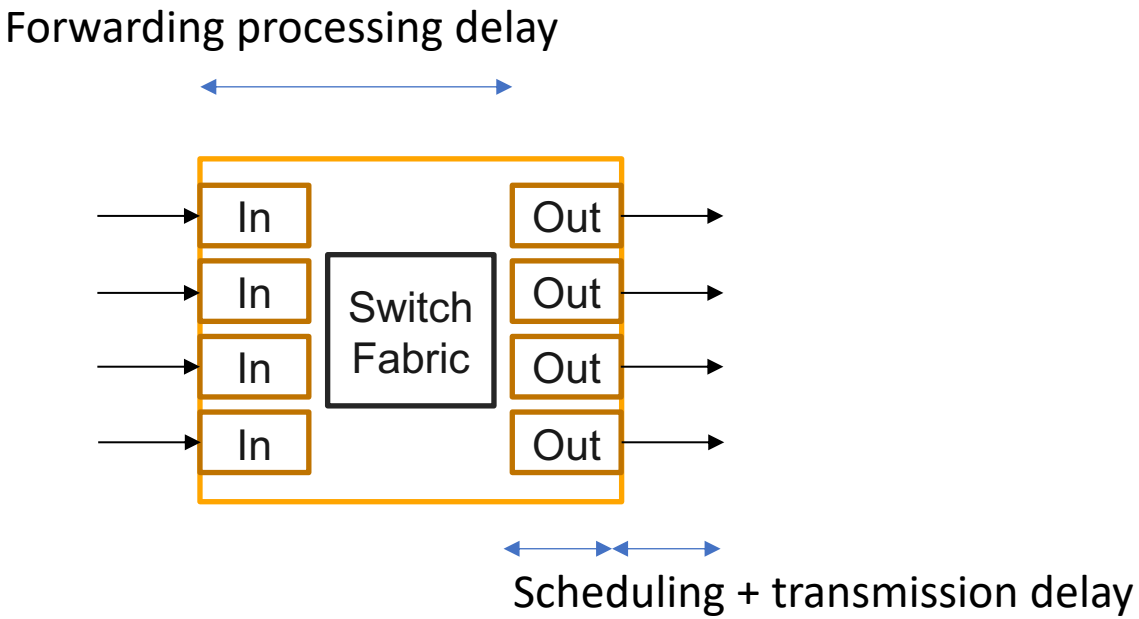
A few words on forwarding delay within a switch

Soheil Samii

LINKÖPING UNIVERSITY

Forwarding processing delay ("store & forward" delay)

- Delay between when the last CRC byte was received until the frame is put in an egress queue available to be selected for transmission (e.g., by CBS or TAS)
- IEEE 802.1 does not say anything about this delay (no definition and no maximum value for compliance)



Low port-count switches

- While MAC receives the Ethernet frame, it transmits a byte stream to the switch core
- Switch core buffers the bytes in global memory (not yet known whether it's a valid frame)
- CRC is calculated over the stream on the fly for future comparison with the last 4 bytes (the FCS)
- Once MAC has signaled IPG (inter-packet gap) to the switch core, the comparison is performed to determine whether or not the frame is valid
- Pointer to the buffer space (plus metadata extracted from header) is passed to the egress ports)
- Low tens of clock cycles for 1 Gbps switches (125 Mhz)

Not as simple for all switches

- Higher speeds (e.g., 10 Gbps)
  - Cannot bump up the clock frequency to 1.25 GHz due to power limitations
- High port-count switches
  - Pack two 5-port switches to create an 8 port switch (but it creates a huge bottleneck for the cascaded connection)
  - Connect multiple 3 or 4 port switches in an internal ring topology (complicated analysis of the arbitration scheme)
  - Use output port buffers in addition to global memory

Forwarding delay

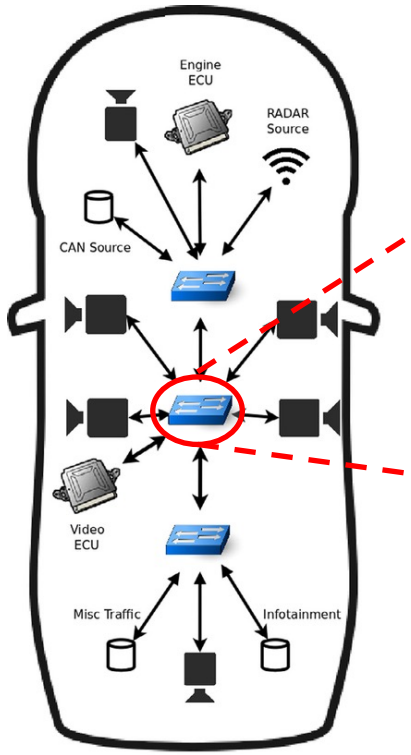
- It is complicated to calculate a worst-case S&F delay, especially in case of many ports and in case of multi-Gpbs networks
- It is highly switch and vendor specific
- 802.1 does not specify internal switch implementations
- Solutions?
  - Make pessimistic assumptions (anyway, this delay is typically orders of magnitude lower than the dynamics of the application)
  - Ask vendors for performance numbers
  - Use measurement equipment
  - Create application specific industry profile specifications to add nonfunctional requirements

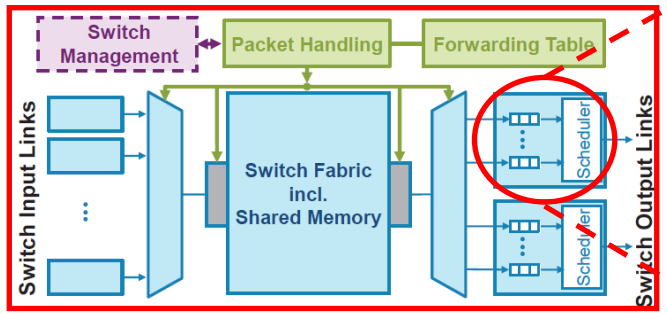
Frame preemption

IEEE Std 802.3br-2016
IEEE Std 802.1Qbu-2016

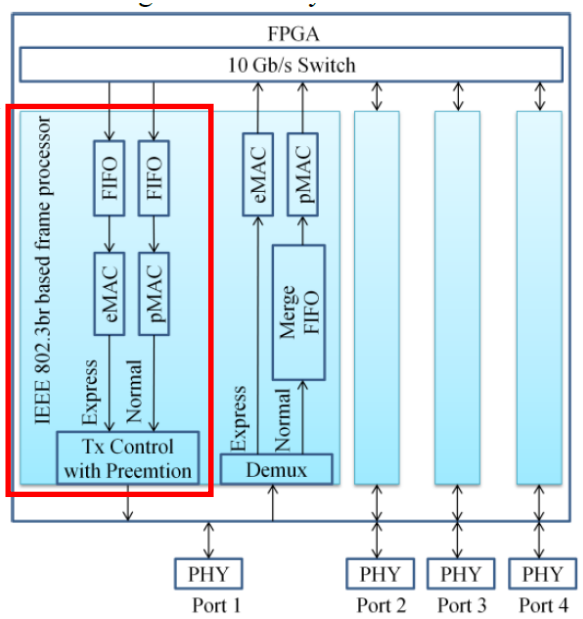
Soheil Samii

Switched Ethernet with preemption



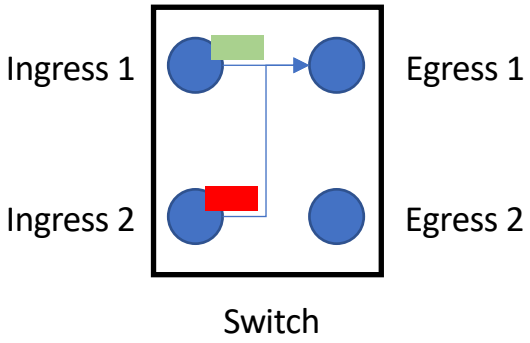


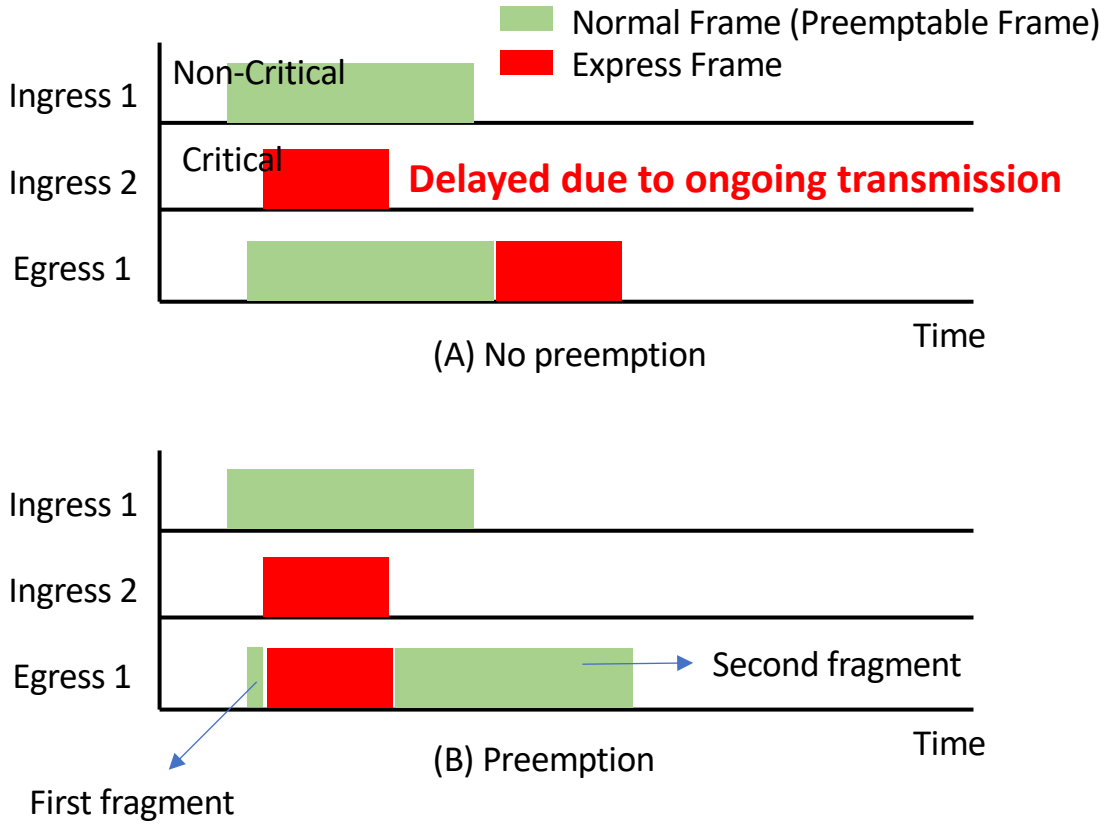
ingress egress



An egress port that supports preemption

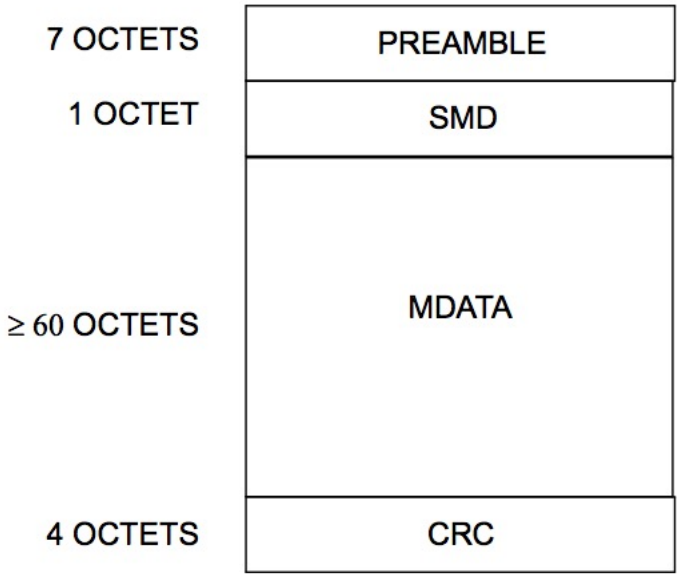
What is preemption?





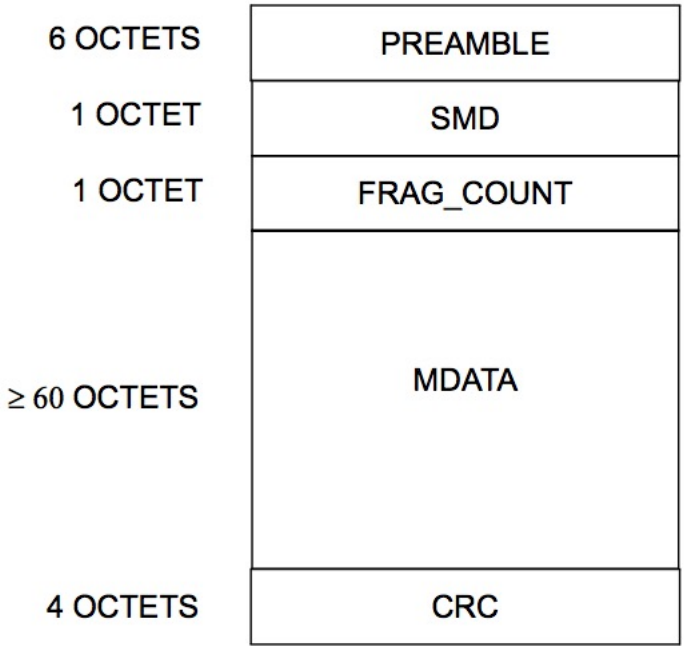
Packet formats (802.3br)

MAC Merge frame



mPacket containing an express packet, a complete preemptable packet or an initial fragment of a packet

(a)



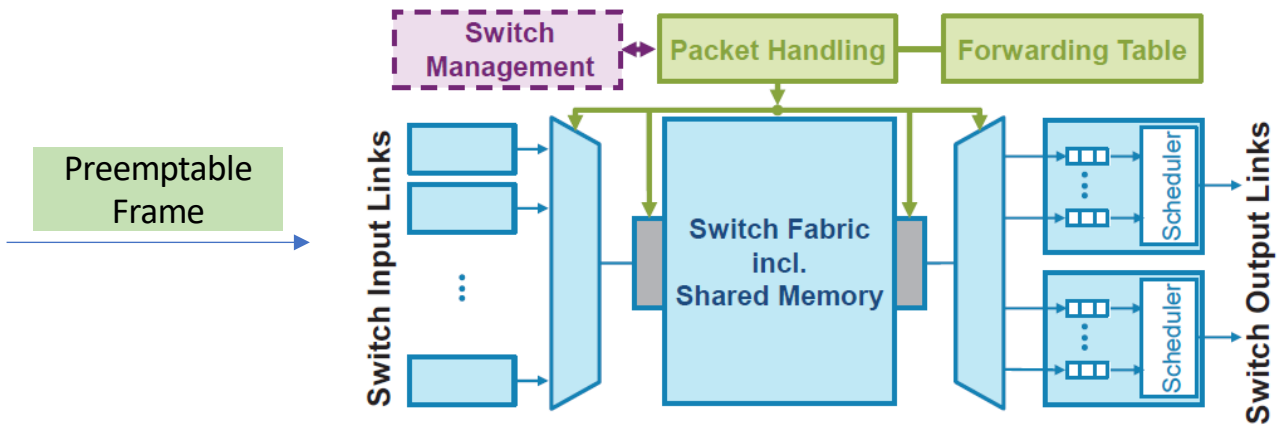
mPacket containing a continuation fragment of a packet

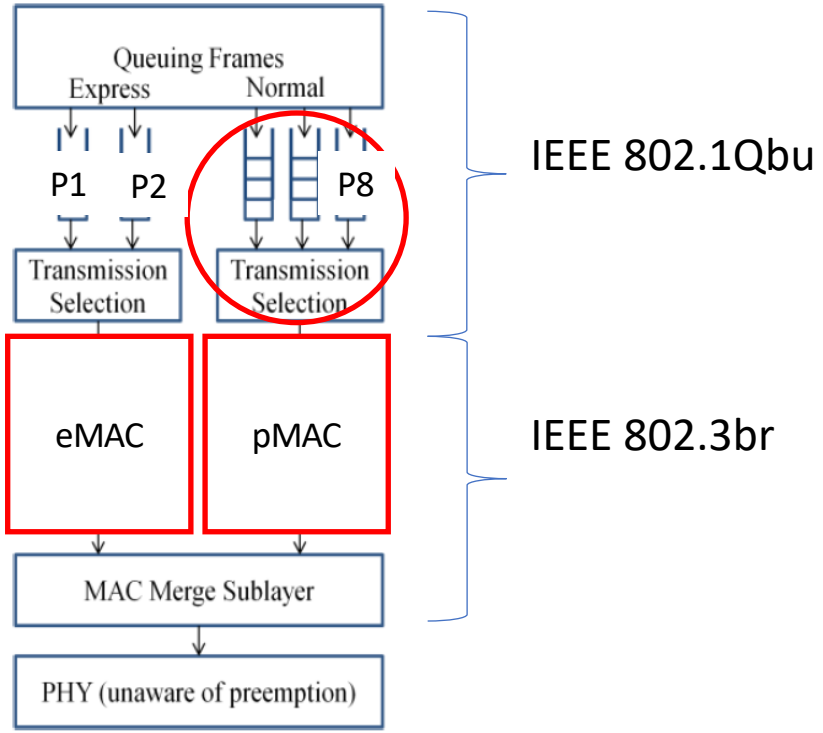
(b)

Figure 99–4—mPacket format

Last mPacket = FCS of original Ethernet frame

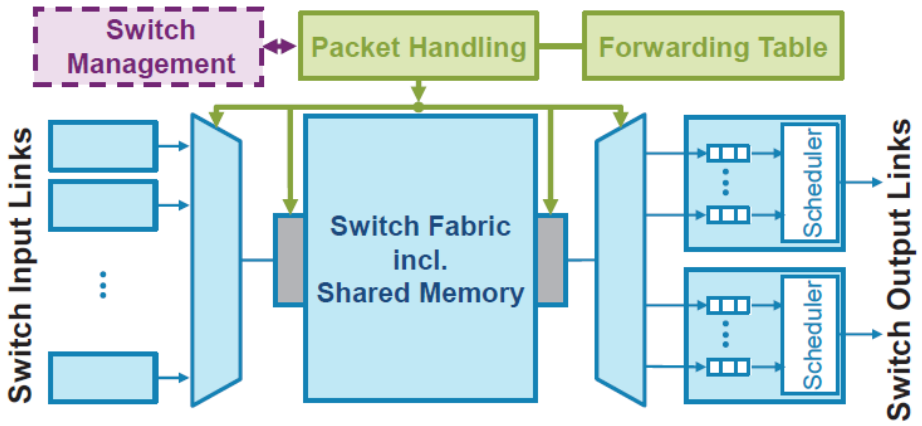
Preemption

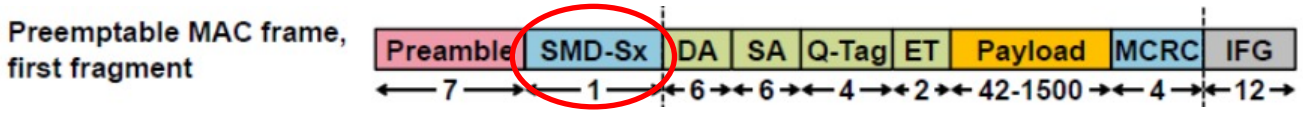




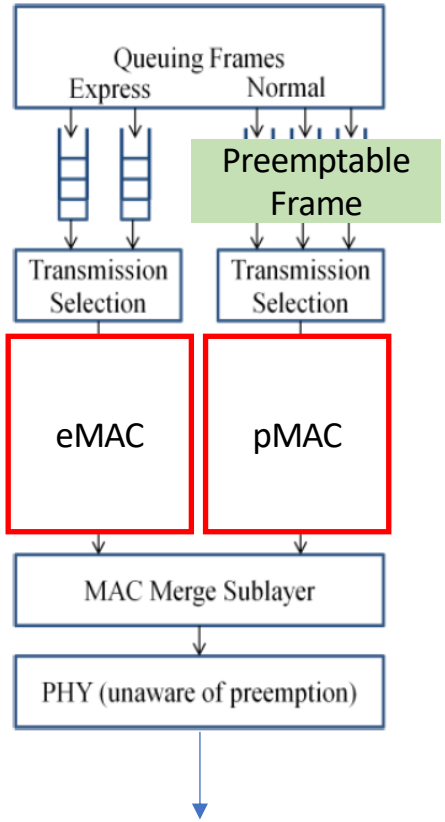
- A queue is chosen based on PCP value in a frame
- Each queue has own priority
- Express/Preemptable queue is statically assigned

Preemption

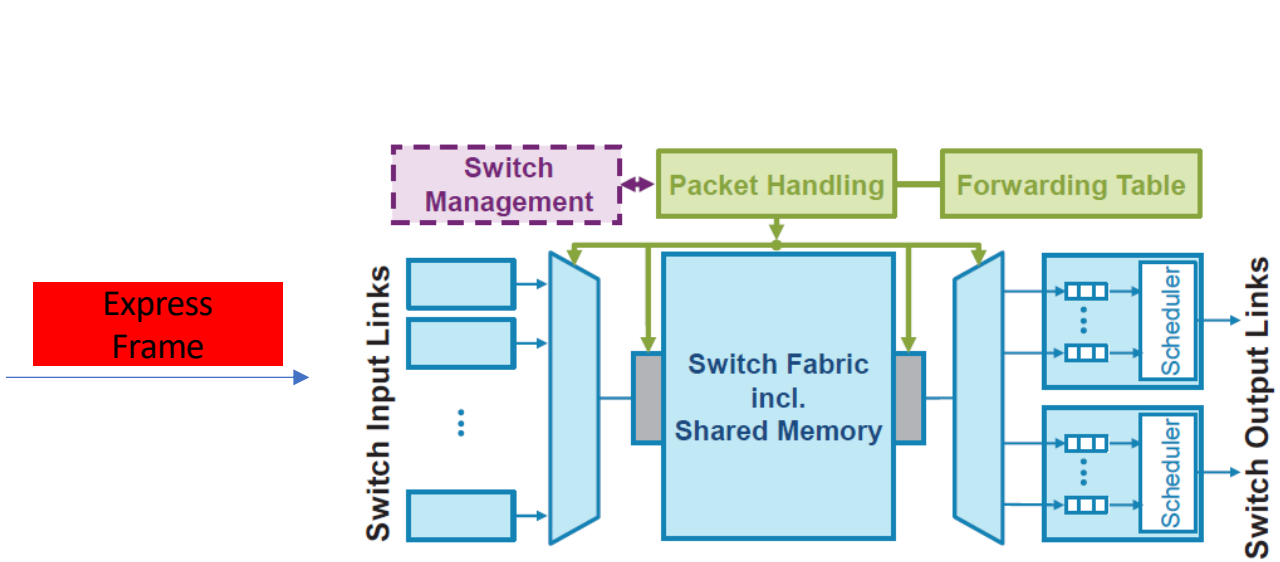


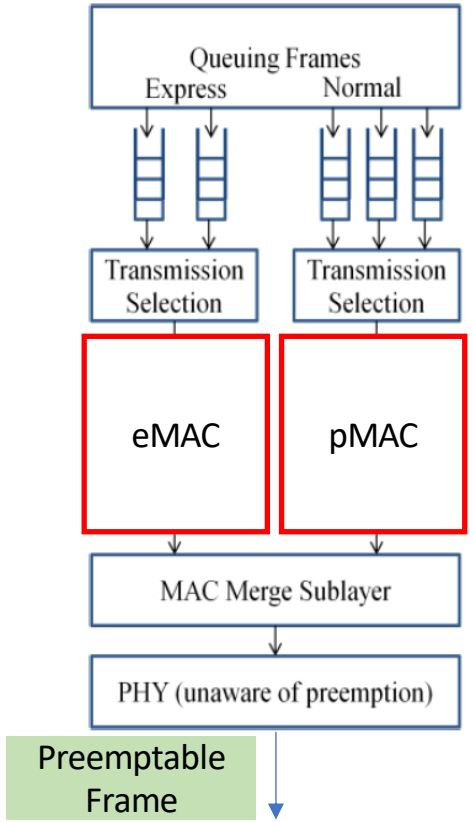


Indicates the type of packet

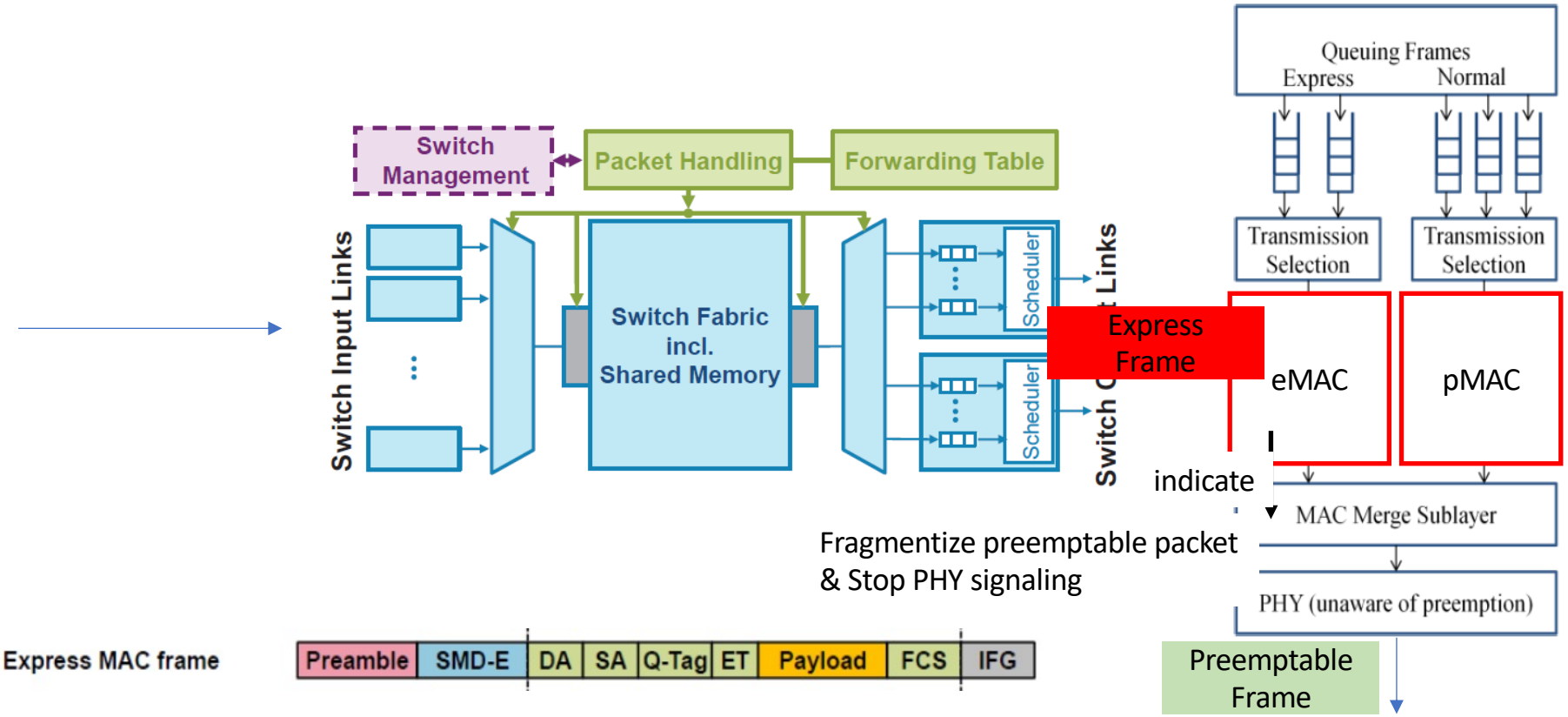


Preemption


Express Frame
Switch Management
Packet Handling
Forwarding Table
Switch Input Links
Switch Fabric incl. Shared Memory
Scheduler
Scheduler
Switch Output Links

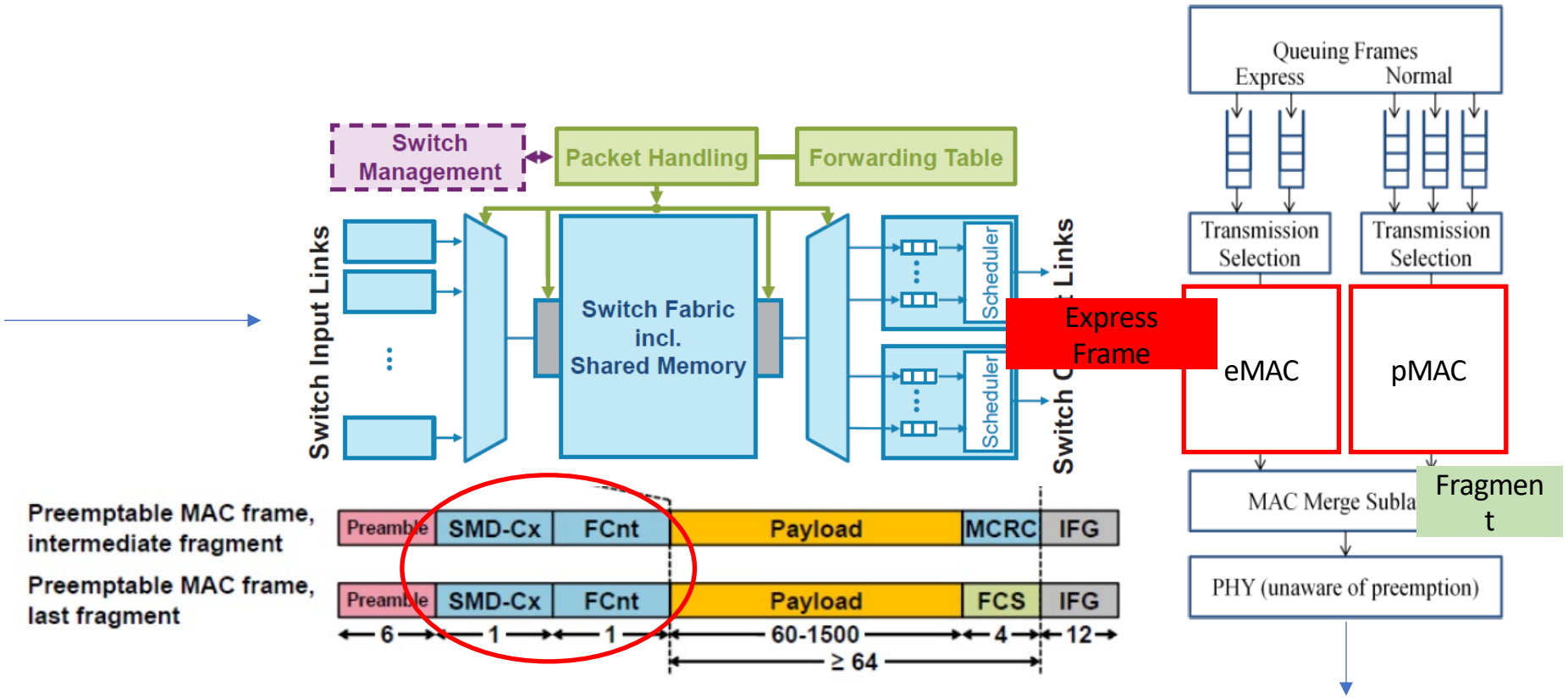

Queuing Frames
Express
Normal
Transmission Selection
Transmission Selection
eMAC
pMAC
MAC Merge Sublayer
PHY (unaware of preemption)
Preemptable Frame

Preemption


Switch Management
Packet Handling
Forwarding Table
Switch Input Links
Switch Fabric incl. Shared Memory
Scheduler
Scheduler
Switch Links
Express Frame
Queuing Frames
Express
Normal
Transmission Selection
Transmission Selection
eMAC
pMAC
indicate
MAC Merge Sublayer
PHY (unaware of preemption)
Fragmentize preemptable packet
& Stop PHY signaling
Preemptable Frame
Express MAC frame
Preamble
SMD-E
DA
SA
Q-Tag
ET
Payload
FCS
IFG

Preemption

Minimum fragment size is 64 bytes



Summary

- Preemption capability required a change in the MAC layer (802.3br)
  - Express MAC (eMAC) and Preemptable MAC (pMAC)
  - Ethernet Start-of-Frame delimiter was generalized to indicate whether a frame is express, preemptable, or continuation packet
  - Also include special packets for a port to verify preemption capability of the link partner; also to respond to a verification request
- Only one level of preemption
- Queues are allocated either to the eMAC or pMAC
  - Allocation can be unique per port

Time synchronization

Soheil Samii

LINKÖPING UNIVERSITY

Why do we need to synchronize clocks?

- Time stamp data and events
- Synchronous data acquisition
- Sensor fusion
- Synchronous actuation
- Data analytics without loss of temporal information
- Establish causal relationships that led to a failure
- Synchronized task execution
- Scheduling of packets on the network (e.g., TDMA)

IEEE 802.1AS

- First published in 2011. Revision published in 2020.

- gPTP – Generalized Precision Time Protocol
  - Grand Master (GM)
  - Time Slave
  - Time Relay (i.e., Ethernet switches)

- BMCA – Best Master Clock selection Algorithm
  - ”GM capable” components
  - Distributed algorithm to elect GM
  - Re-elects new GM if current GM disappears

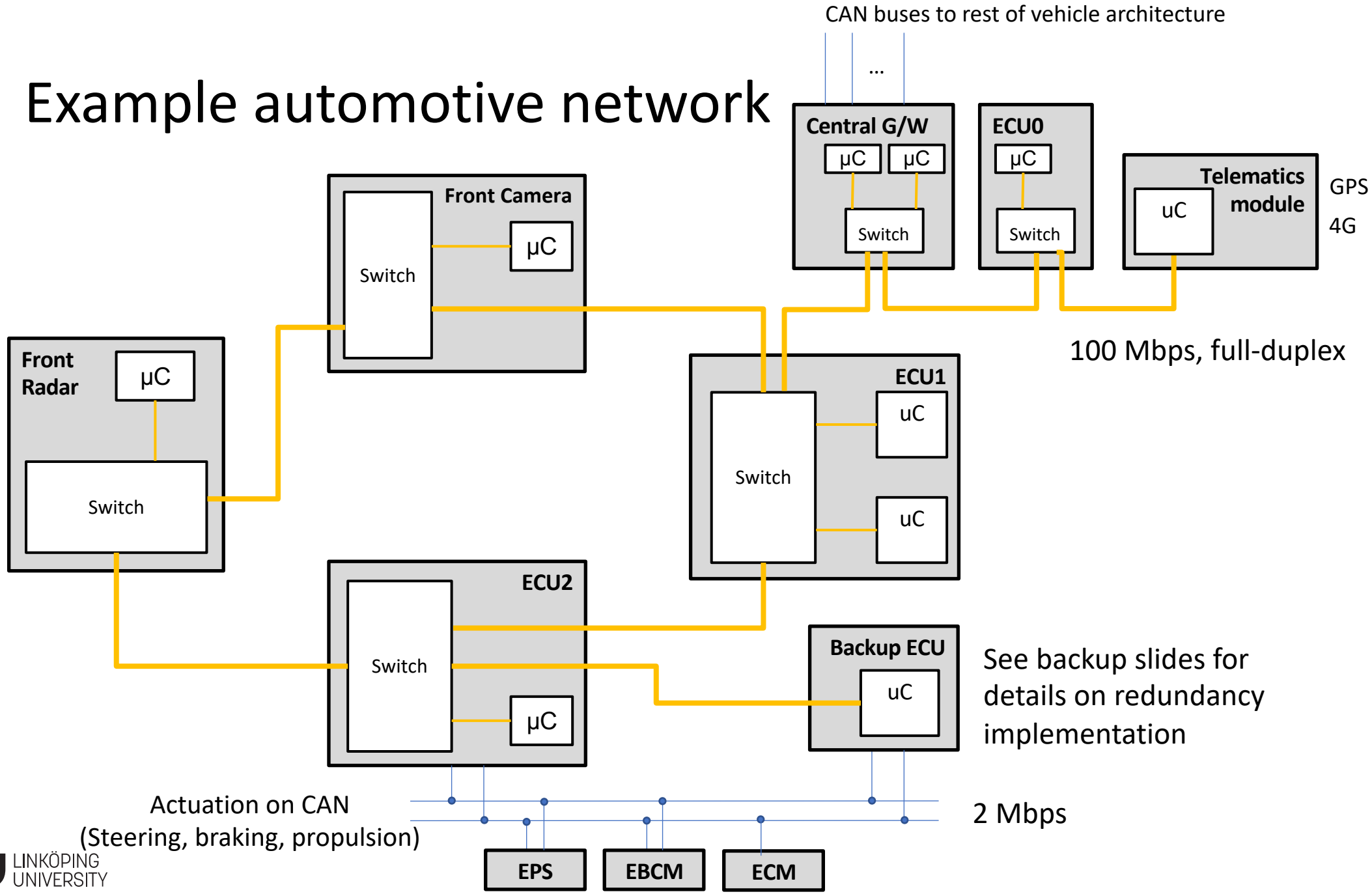
gPTP

- Precision time protocol; profile of IEEE 1588 for full-duplex Ethernet
- Time sync frames are transmitted by a Grand Master (GM)
- Time sync frames are time stamped on ingress and egress
- Time sync frames are modified with corrections based on time measurements and propagated through the network to time slaves
- Time slaves adjust their time based on received time sync frames
- Link delay is measured at each port
- Syntonization: Rate ratio is measured by each endpoint and switch (except GM), because clocks may have different frequencies

BMCA

- Best Master Clock selection Algorithm
- Election process based on Announce frames to select the Grand Master and configure port roles, thereby establishing a path to all time slaves
- Priorities in Announce frames are configurable

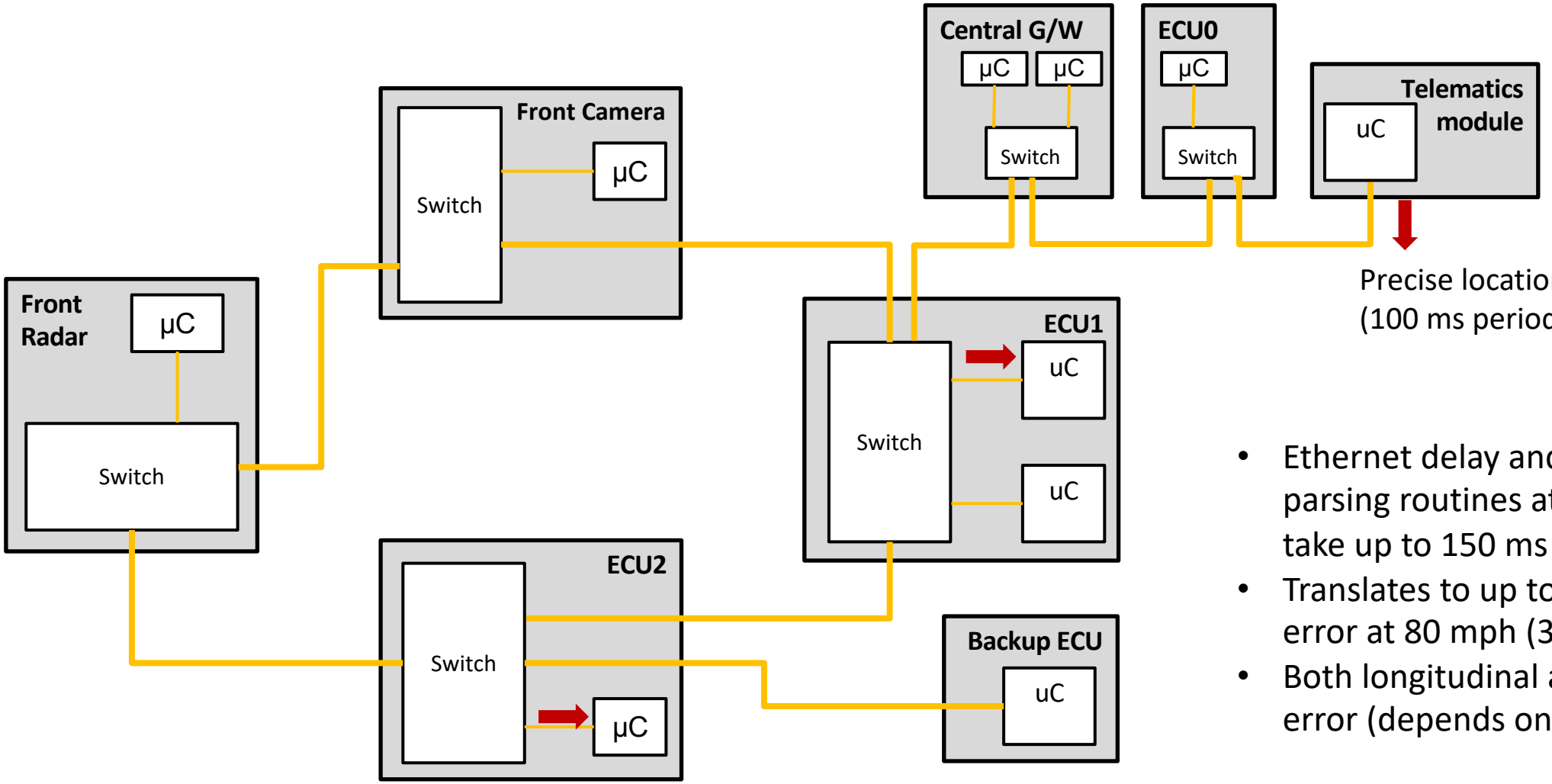
Example automotive network



Problem: Latency and jitter in the network cause control errors

- Software applications receive many inputs from different sensors across the network
- Each input experiences network latency, which is different from sample to sample
- Latency indirectly introduces estimation and prediction errors in the control system

Problem illustration: vehicle location



- Ethernet delay and frame parsing routines at receiver may take up to 150 ms
- Translates to up to around 5 m error at 80 mph (35.8 m/s)
- Both longitudinal and lateral error (depends on curvature)

Precise location from source to destination

- Common that systems implement GPS corrections to bring down GPS error from ~10 meters to sub-meter precision, sometimes even centimeter level precision
- Task scheduling and variable Ethernet communication latency can cause an end-to-end delay to consumer application of 100-200 ms
- At 100 km/h, this leads to multiple meters of error
- Solution: GPS application time stamps data before transmitting, the consumer application can make temporal corrections without loss of GPS accuracy

Approach

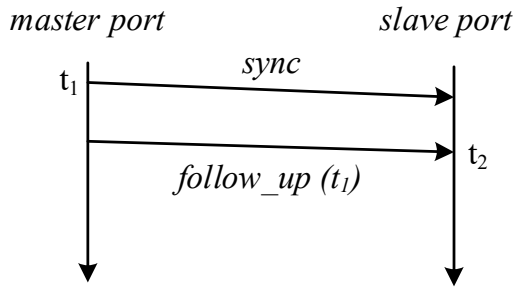
- Establish a common, precise time base through IEEE Std 802.1AS
- Timestamp data as close as possible to data acquisition
- Receiving application calculates time difference and implements corrections accordingly
- Note: Complementary technique to frame priorities, scheduling, and traffic shaping

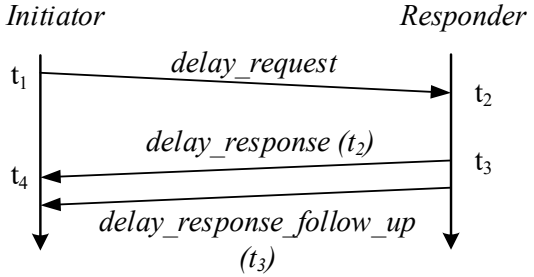
IEEE Std 802.1AS

IEEE 1588g – ongoing revision to add a suitable and inclusive terminology to the terms ”master” and “slave”

Generalized Precision Time Protocol

- Roles: Grand Master (GM), Time Slave, Time Relay
- Functions: Distribution of time throughout network, peer link delay measurement





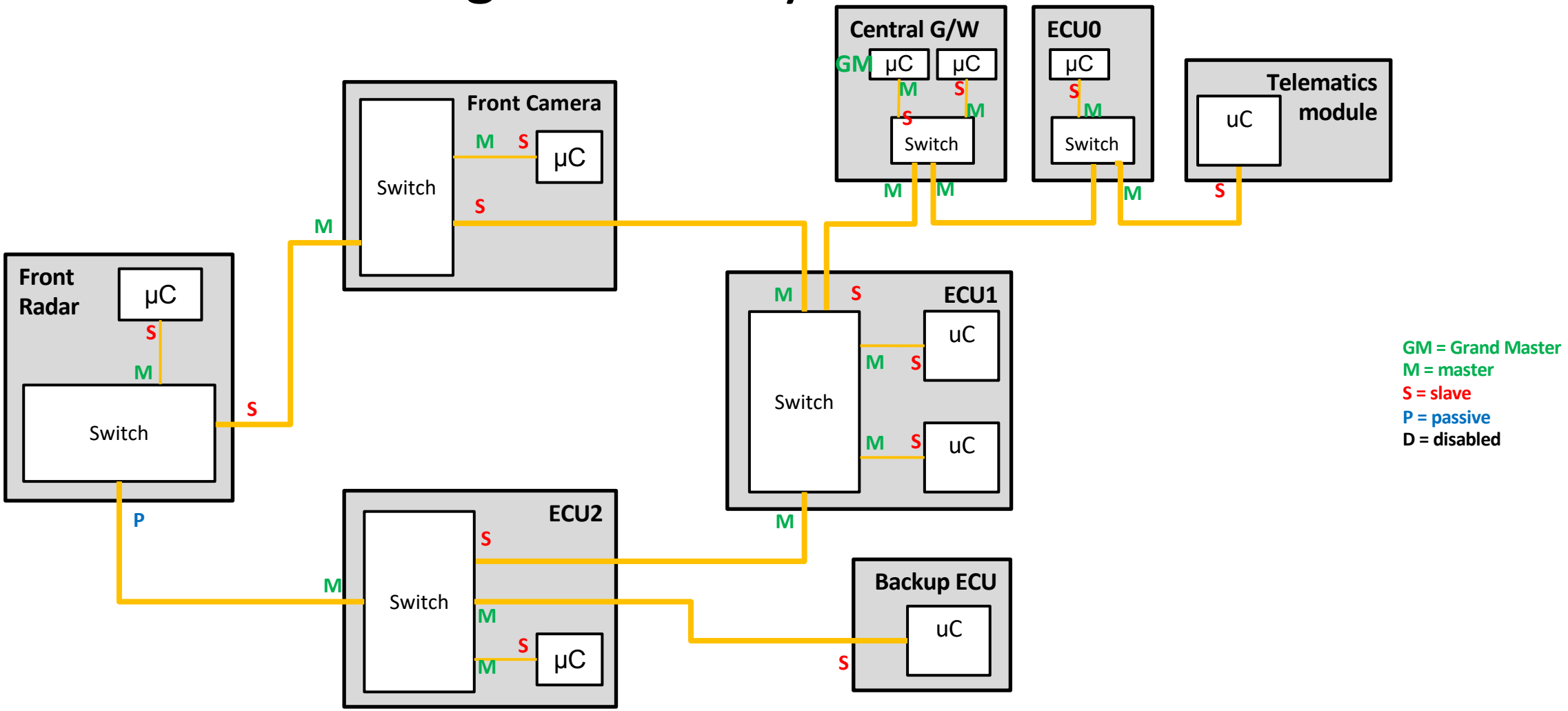
Best Master Clock Algorithm (BMCA)

- Distributed election of GM
- Re-elects new GM if current GM disappears
- Alternative: use static configuration

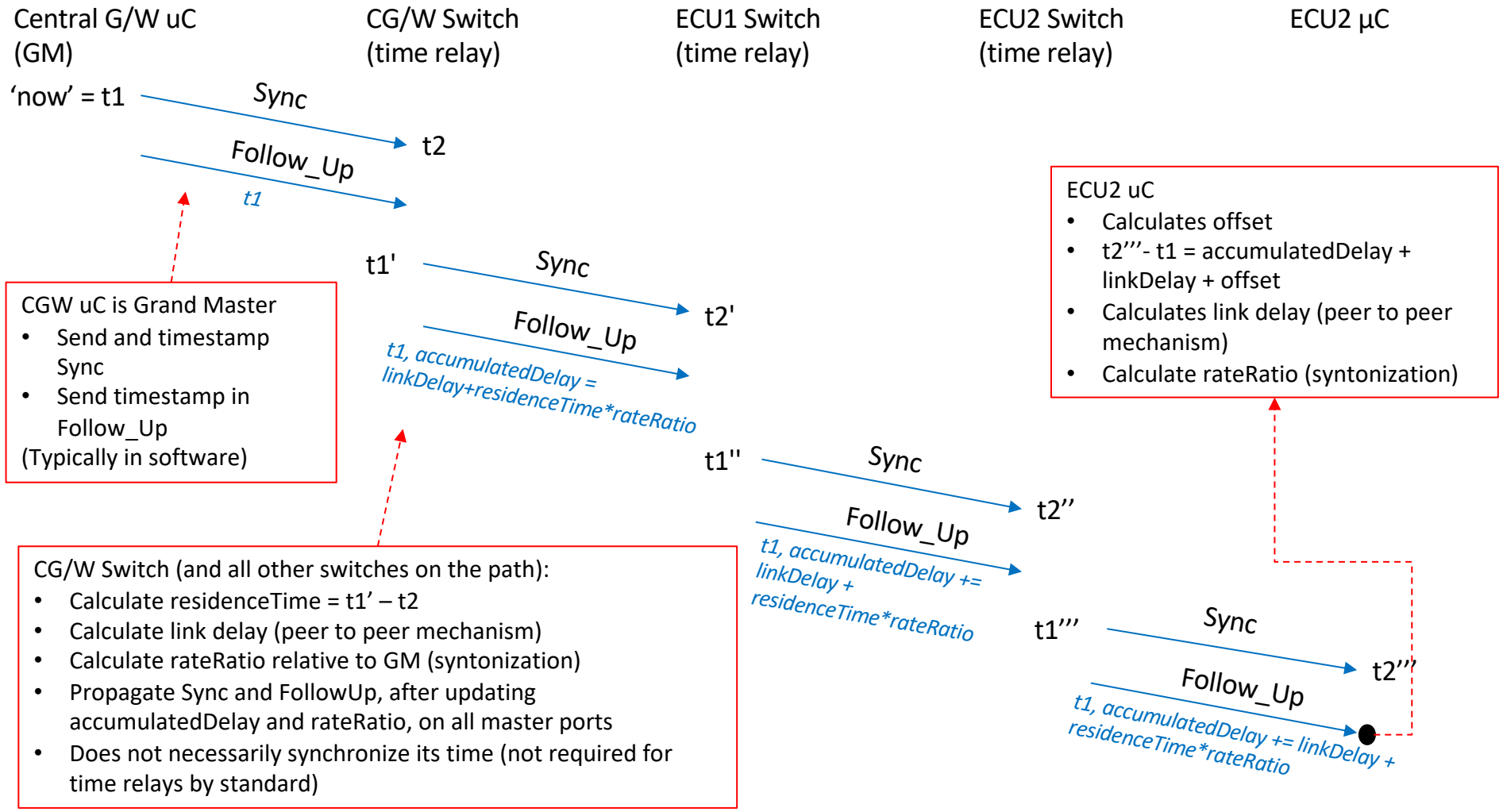
Typically:

- MAC/PHY hardware support for timestamping at ingress and egress
- Software implementation of gPTP and BMCA

Port role configuration – synchronization tree



One branch of the synchronization tree



IEEE 802.1AS frames and format

- Reserved multicast address: 01-80-C2-00-00-0E
- Reserved EtherType: 0x88F77
- Event frames are time stamped on ingress and egress

gPTP frame type	Function	Class	Value in header
Sync	Time sync	Event	0x0
Follow_Up	Time Sync	General	0x8
Pdelay_Req	Link Delay	Event	0x2
Pdelay_Resp	Link Delay	Event	0x3
Pdelay_Resp_Follow_Up	Link Delay	General	0xA
Announce	BMCA	General	0xB
Signaling	Power Saving (e.g., request reduced frequency in delay measurements)	General	0xC

Vehicle location timestamp – AUTOSAR StbM APIs

Telematics module (sender)

```
StbM_TimeStampType timeStamp;
StbM_UserDataType userData;
VehiclePositionType preciseLocation = GPS_fusion(); // not listing input parameters
StbM_getCurrentTime(0, &timeStamp, &userData);
RTE_Write_Outputs(preciseLocation, timeStamp);
```

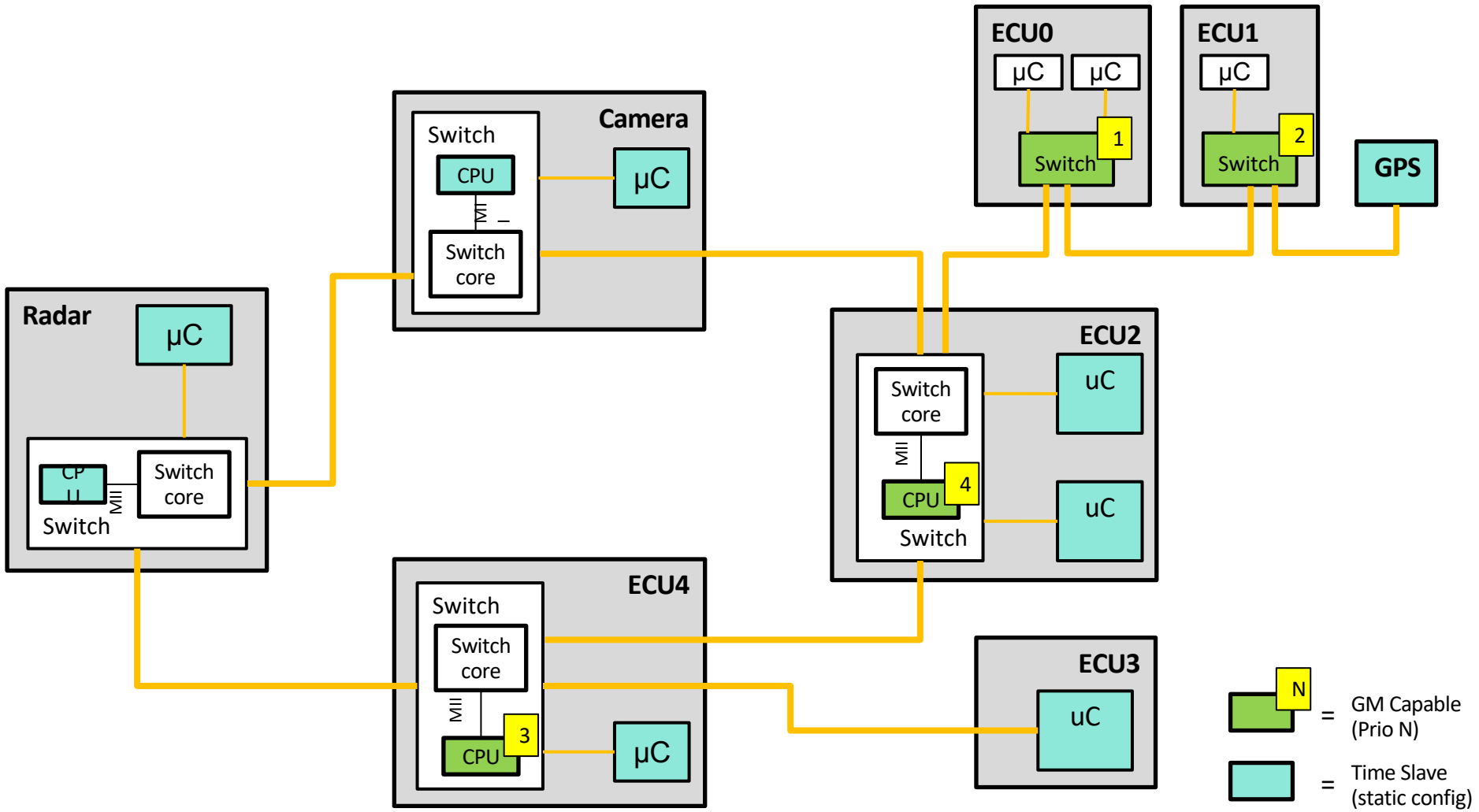
ECU2 (receiver)

```
StbM_TimeStampType locationTimeStamp;
RTE_Read_Inputs(location, &locationTimeStamp);
StbM_TimeStampType currentTime;
StbM_UserDataType userData;
StbM_getCurrentTime(0, &currentTime, &userData);
StbM_TimeType timeDiff = currentTime  – locationTimeStamp;
```

Other uses of network time synchronization

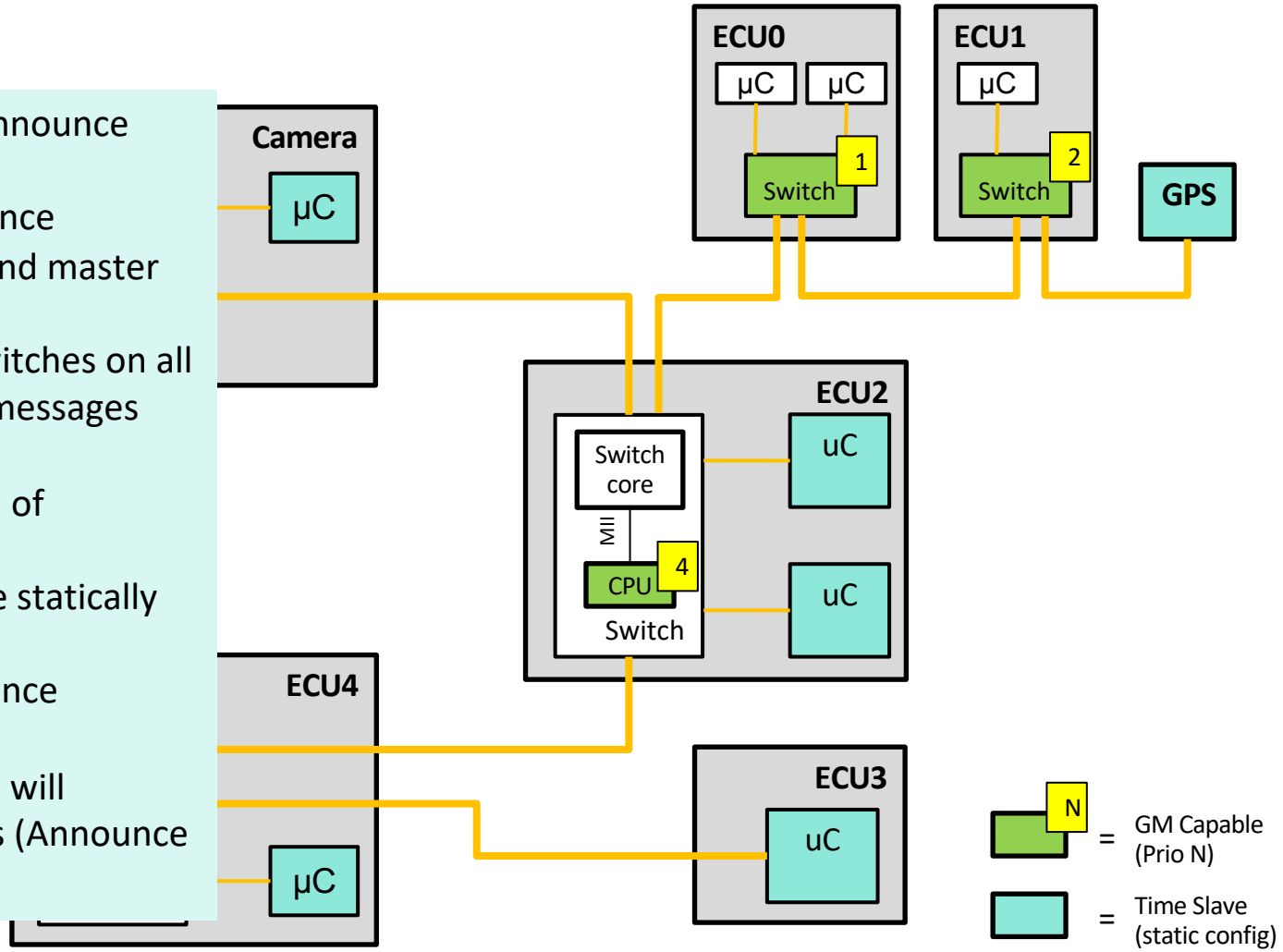
- Timestamp data at creation
- Sensor fusion, improved precision
- Synchronized sensor capture
- Synchronized execution
- Improved logging and troubleshooting through establishment of causal data relationships
- Scheduling of packets on network (e.g., Time-Aware Shaper 802.1 TSN)

BMCA for GM election and automatic port configuration



BMCA in operation

- ECU0, ECU1, ECU2, and ECU4 switches will send gPTP Announce messages with their statically defined priorities
- All other switches and microcontrollers will send Announce messages with priority 255 (i.e., they are slaves; not grand master capable)
- Highest priority Announce message is propagated by switches on all ports for which asCapable == TRUE (inferior Announce messages are discarded)
- All switches will configure port roles based on reception of Announce, making ECU0 switch the GM
- Microcontrollers can participate as slaves in BMCA or be statically configured to be slaves
- If a link fails, switches will reconfigure port roles (Announce messages are sent continuously in BMCA)
- If the current GM fails, the next best GM capable device will become GM and the switches will reconfigure port roles (Announce messages are sent continuously)



Recommended configurations

- Sync and FollowUp frequency: 8 times per second
- PDelay_Req frequency: 3 times in each direction per link per second
- Initial link delay parameters: calibrated (e.g., in automotive, constant wiring length, no plug and play)
- Frequency of Announce frames: 1 per second
- Static priorities and clock qualities defined for each Grand Master capable device.
- asCapable is set to true/false depending on the desired part of the network to participate in 802.1AS

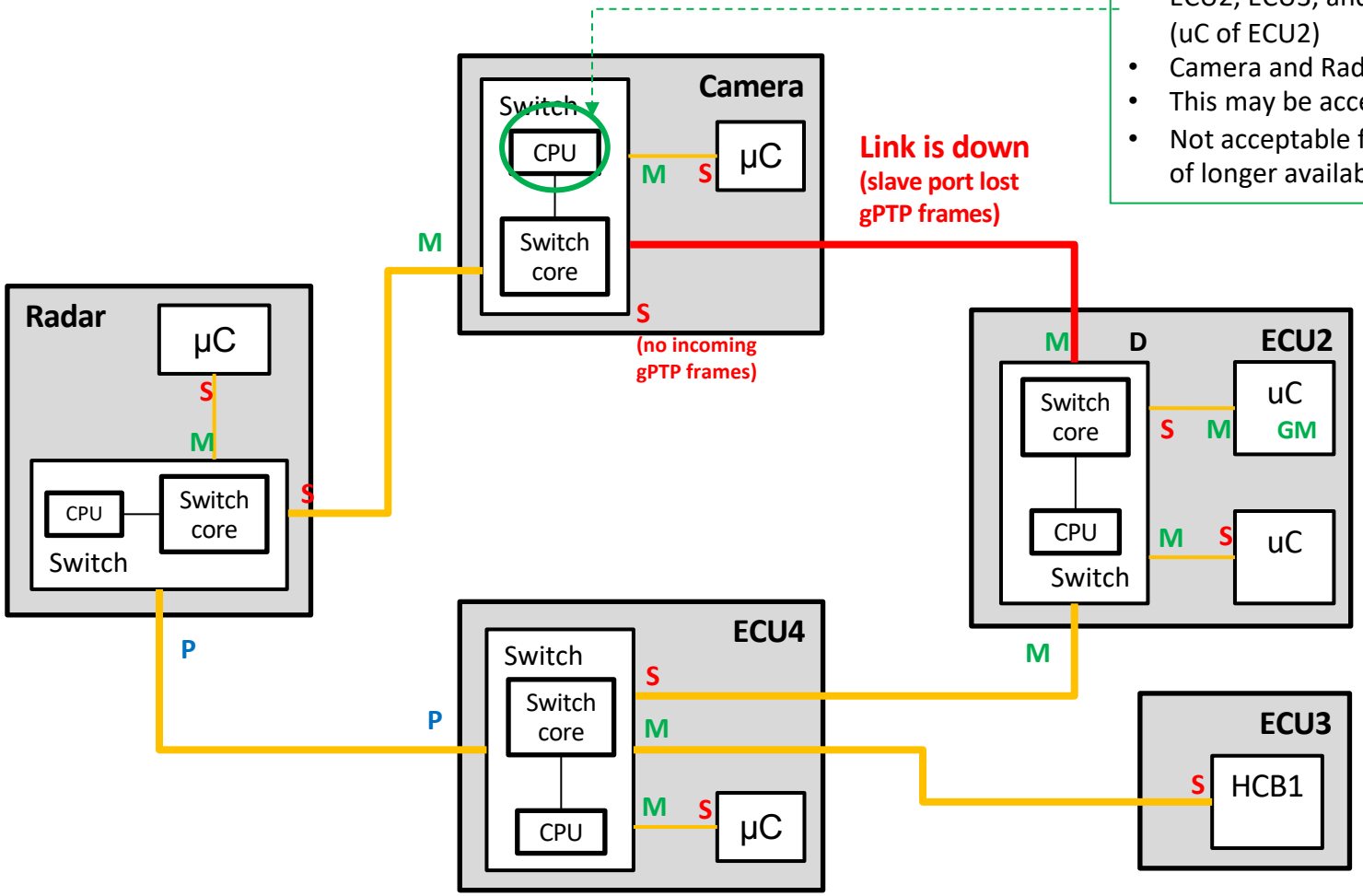
Failure modes

- Loss of time synchronization may be catastrophic
  - Autonomous driving systems: perception functions typically rely on time-stamped sensor data
  - Loss of data communication in case of time-triggered scheduling (like TAS)

- Approaches towards fault-tolerance:
  - Proxy mode implemented in switches (not in 802.1AS)
  - Use BMCA (multiple second recovery)
  - Use multiple clock domains as defined in IEEE Std 802.1AS-2020 with statically configured redundant synchronization domains (trees), and arbitrate among multiple time domains in software (fast recovery)

Proxy mode (link failures)

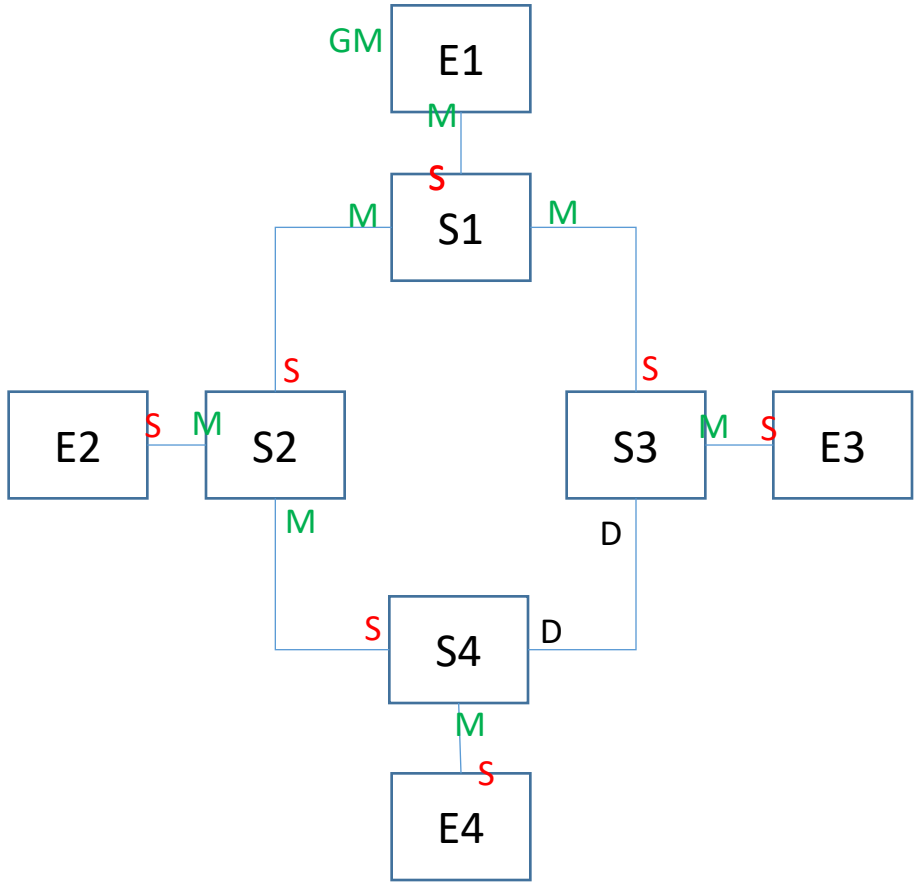
Switch acts as “proxy GM”
- Uses the last known accumulated delay from GM
- Camera and Radar will sync with Camera switch, which is “proxy GM”
- ECU2, ECU3, and ECU4 will sync with GM (uC of ECU2)
- Camera and Radar will drift from GM time
- This may be acceptable for short time
- Not acceptable for fail-operational systems of longer availability



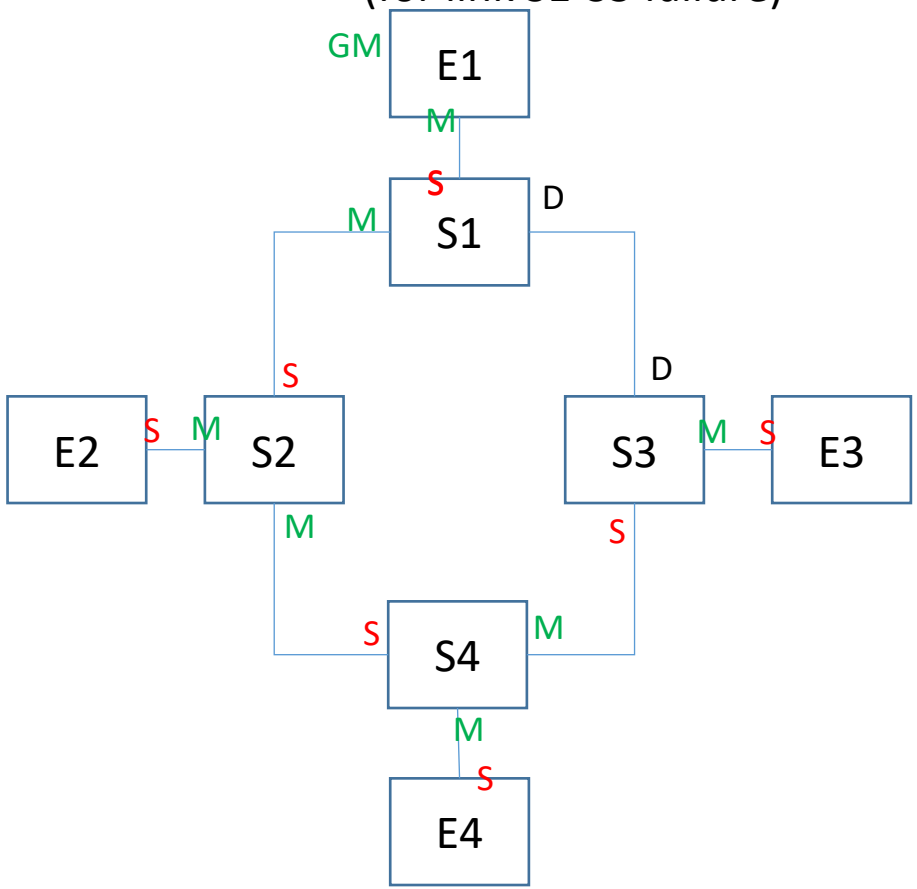
GM = Grand Master
M = master
S = slave
P = passive
D = disabled

802.1AS-2020: multiple clock domains

Domain 0
(works in normal mode and link S3-S4 failure)



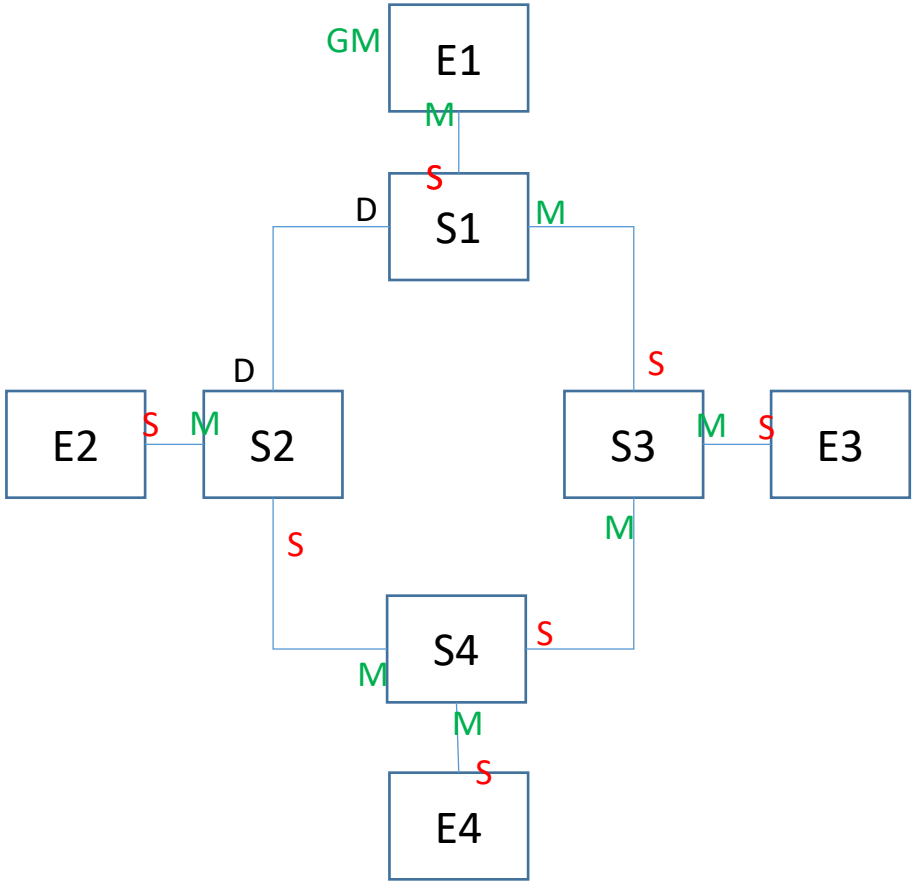
Domain 1
(for link S1-S3 failure)



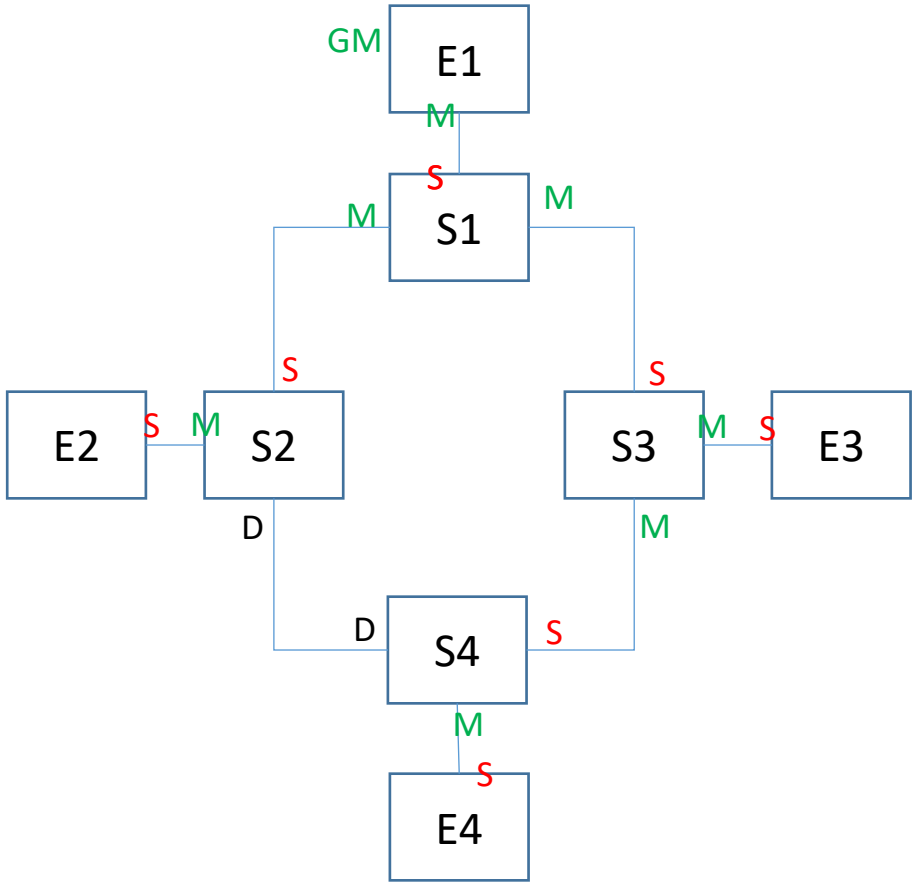
M – Master
S – Slave
D – Disabled (for time sync purposes only)

Fault-tolerant synchronization using multiple clock domains

Domain 2
(for link S1-S2 failure)



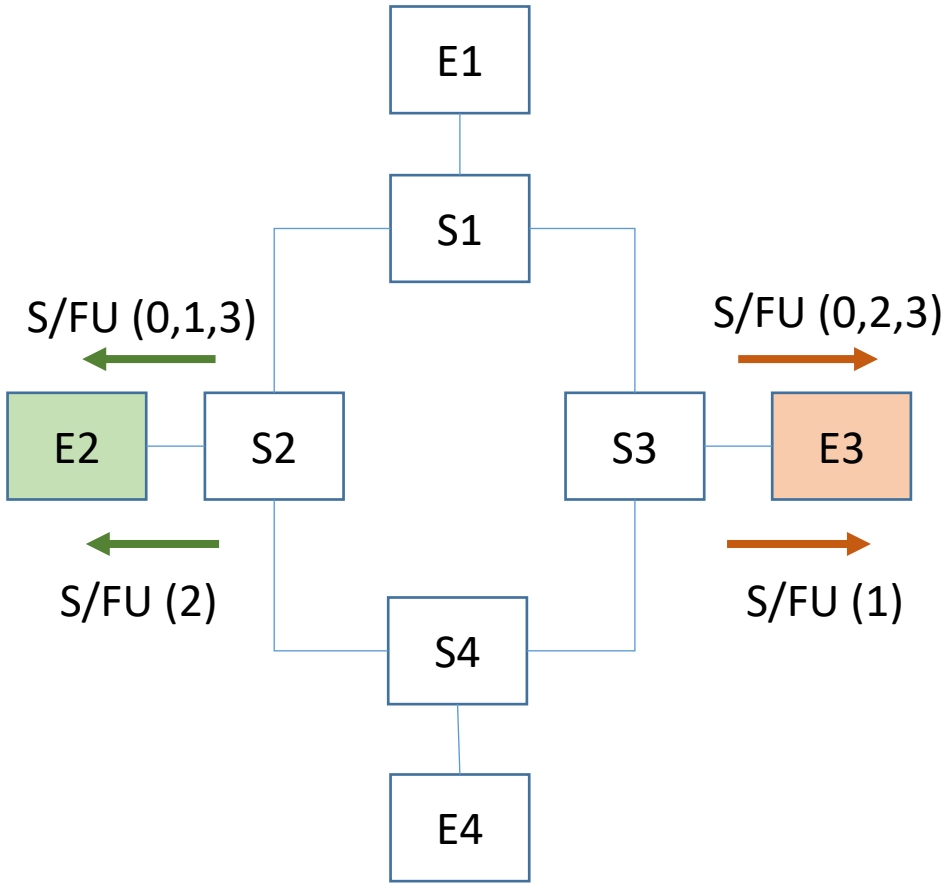
Domain 3
(for link S2-S4 failure)



Arbitration of clock domains

E2:

- Use domain 0 if link S1-S2 is up
- Otherwise use domain 2

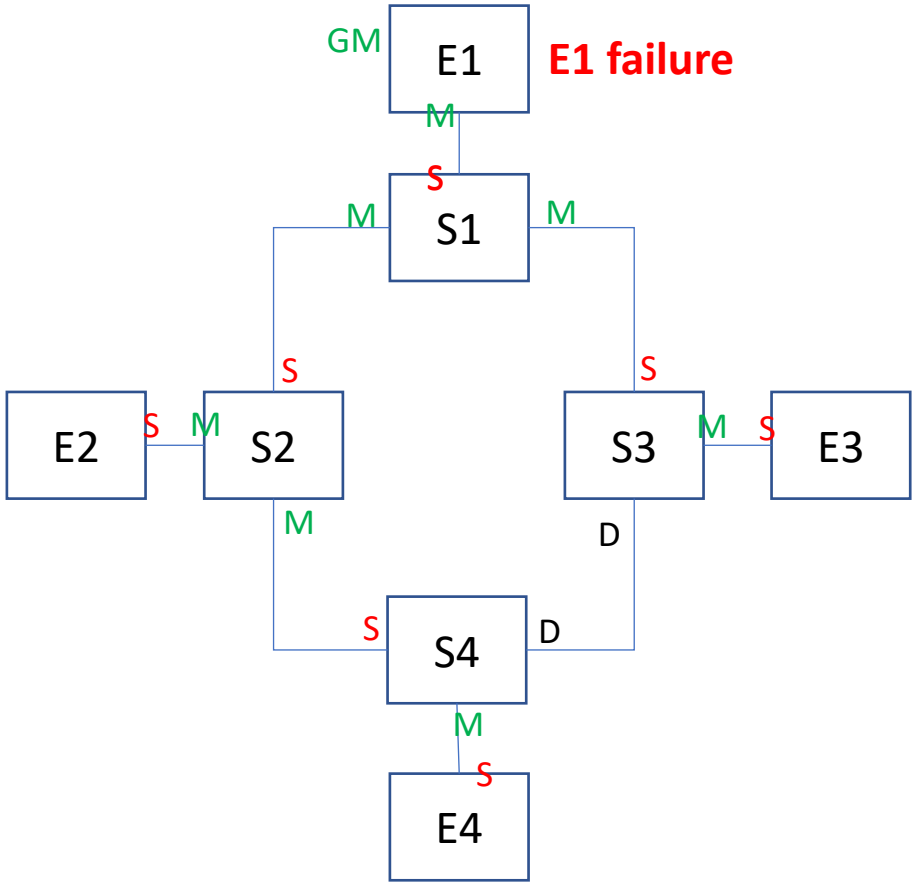


E3:

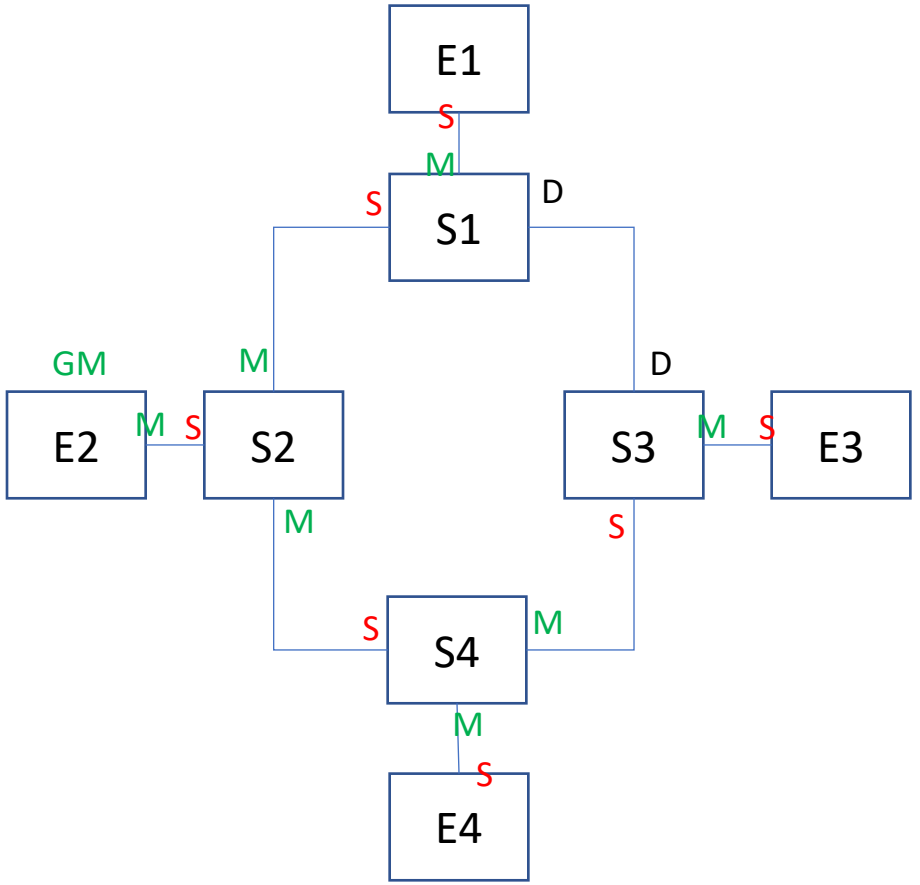
- Use domain 0 if link S1-S3 is up
- Otherwise use domain 1

Dealing with Grand Master failure

Domain 0



Domain 5



E3:

- Use domain 0 if link S1-S3 is up
- Use domain 1
- ***Otherwise use domain 5***

Summary

- Many applications need a global notion of time
- Precise network time synchronization is an important foundation
- IEEE Std 802.1AS has been proven in use and recently amended with redundancy (2020 edition)
  - Election of grand master and establish sync tree
  - Measure delays and rate differences
  - Propagate time frames through sync tree

Ingress filtering and policing

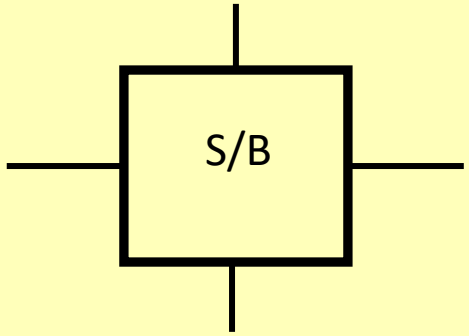
Soheil Samii

Ingress policing

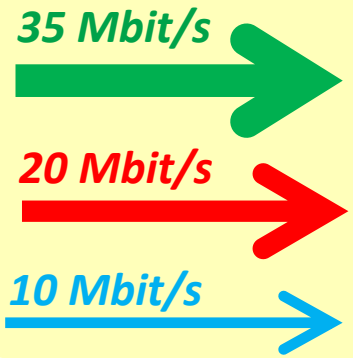
- Detect whether end stations violate timing contract
  - Sending more than has been reserved in the network
  - Sending outside their allowed time windows
  - Sending more than the maximum frame payload agreed upon
- Upon detection of error, isolate error from the network

Symbols and abbreviations

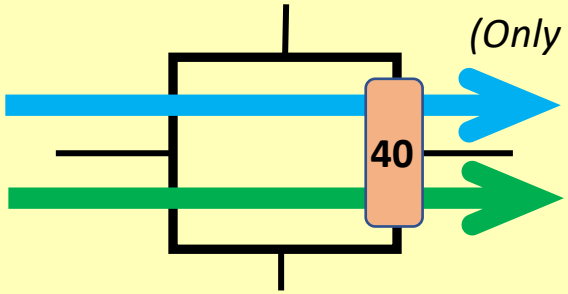
4 Port Switch/Bridge



Class A Streams

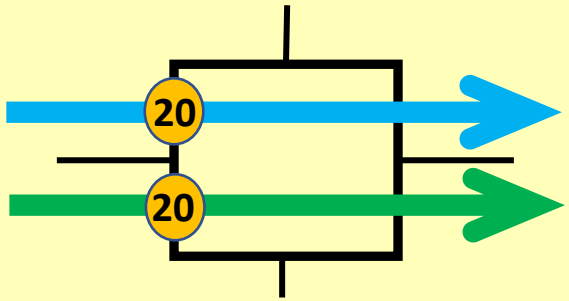


Credit Based Shaper

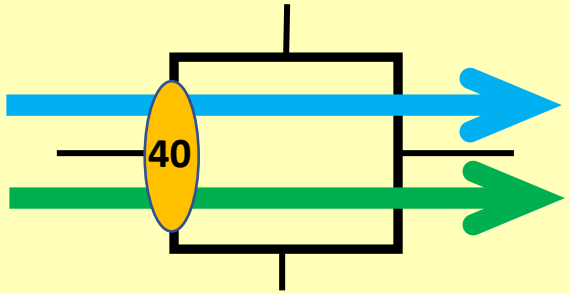


40 Mbit/s Class A Shaper.
(Only shown when essential to a diagram)

Ingress Policing Filter



Two 20 Mbit/s Per Stream IPFs



40 Mbit/s Per Class IPF

- IPF = Ingress Policing Filter
- Talker: T1, T2, … Listerner: L1, L2, …

“Babbling idiot” problem

- “Babbling idiot:” A faulty talker or switch
  - Sends too much data, or
  - Sends at the “wrong time”
  - Takes away bandwidth and other timing guarantees from other streams
- Bandwidth and latency guarantees of these “other streams” may no longer be valid
  - Error propagates through parts of the network and causes errors for other streams
- What can cause a babbling idiot error?
  - MAC or PHY issues, software, clocks, attack, intrusion

“Babbling idiot” problem

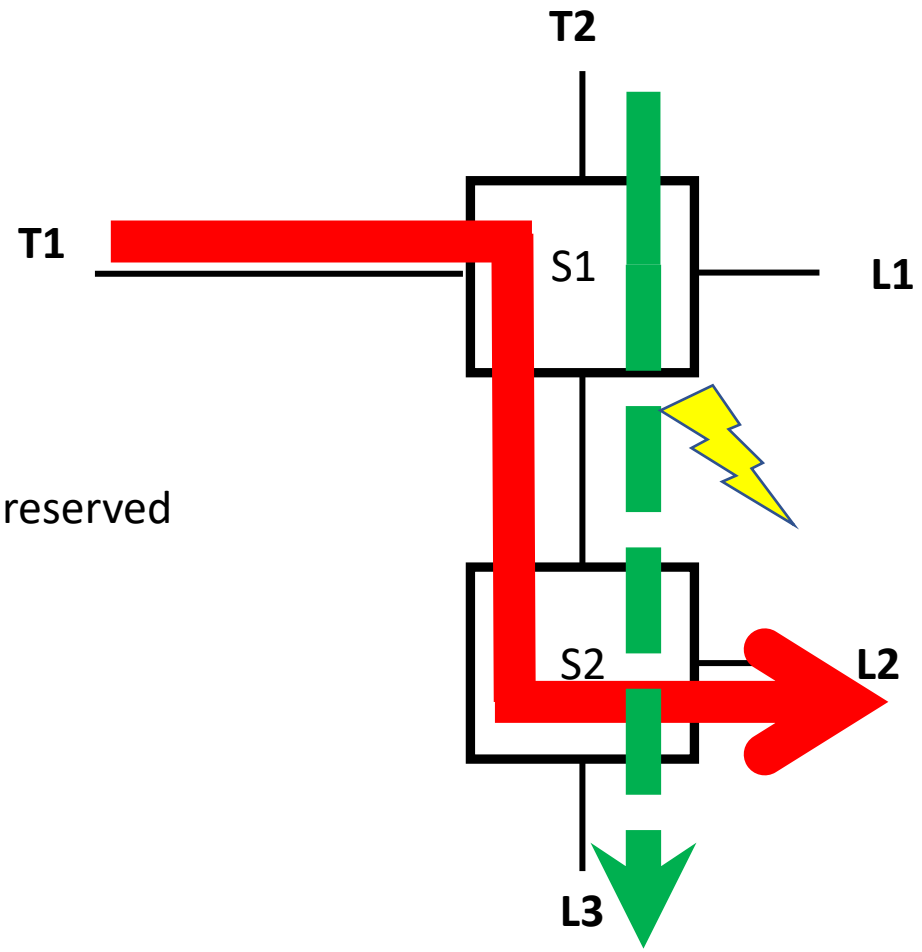
Example:

Babbling Idiot: T1
Faulty red stream sends too much data.

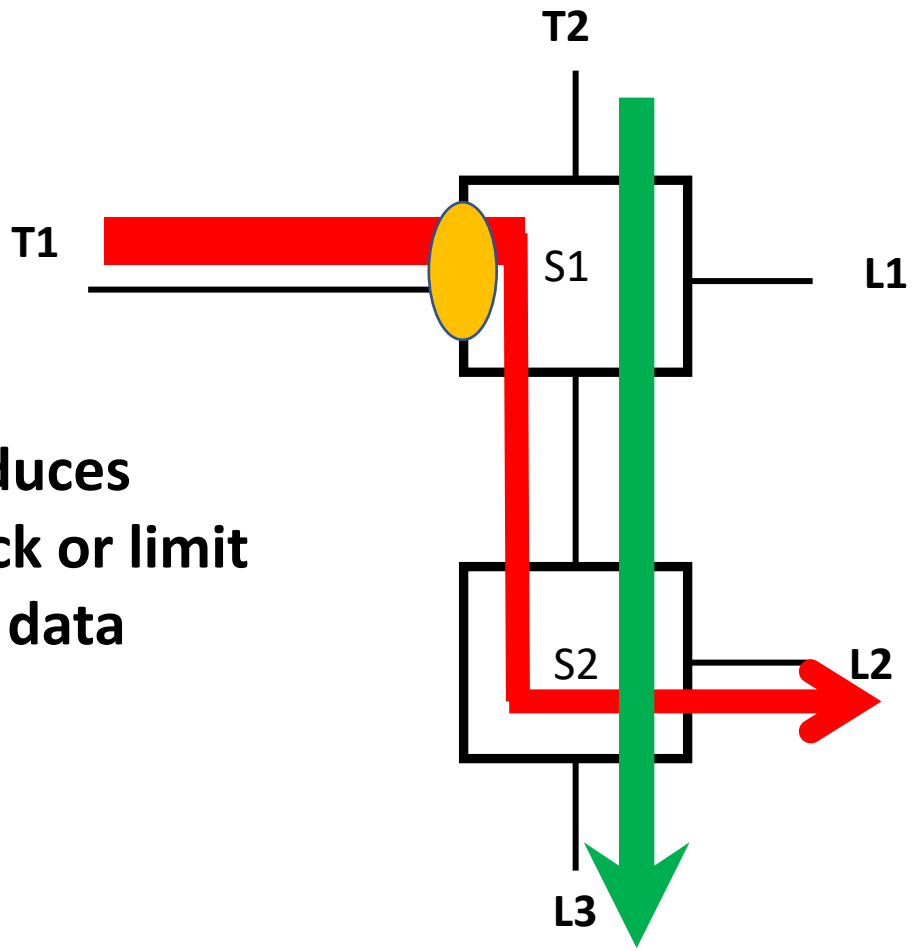
Red streams steals bandwidth that was reserved for the Green stream

Note:
All components on the “green path” are fault free.

But:
Green stream is faulty.



Ingress policing in a nutshell



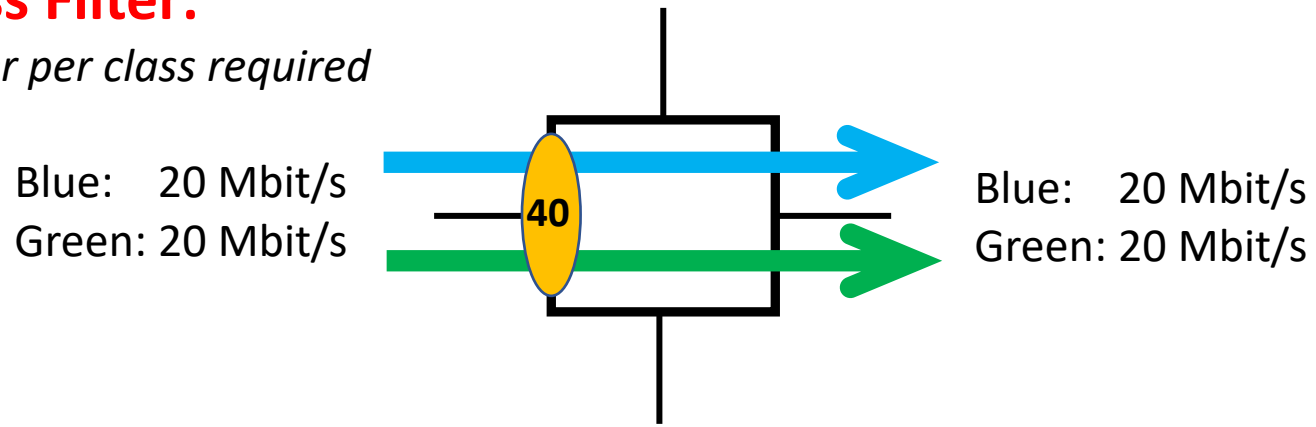
Ingress Policing introduces filters that will block or limit excessive amounts of data

Ingress policing options

Blue and Green are sent from the same queue
(i.e., same priority and same credit-based shaper)
Criteria 1: What is a filter "observing" or "counting?"

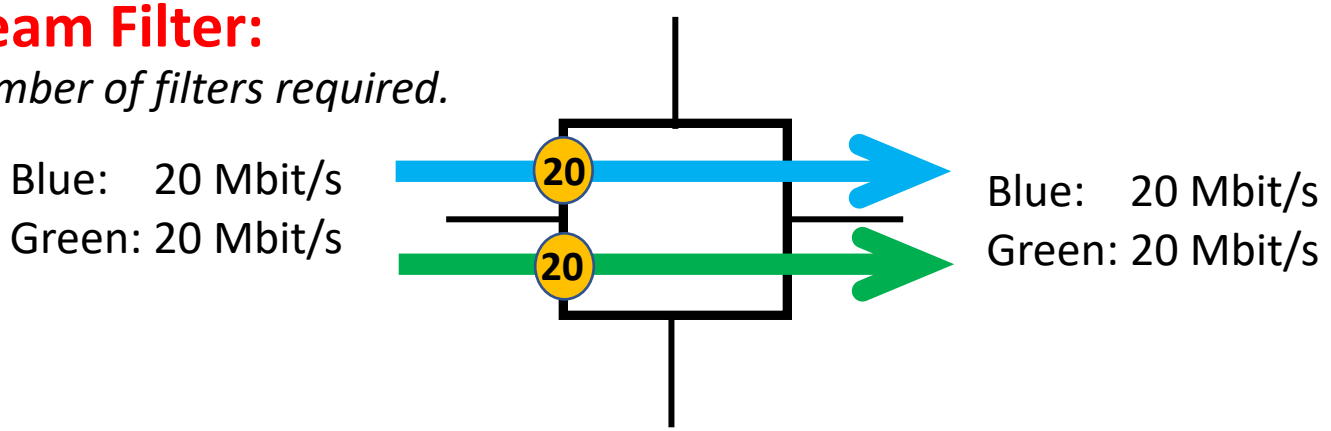
Per Class Filter:

Only 1 filter per class required



Per Stream Filter:

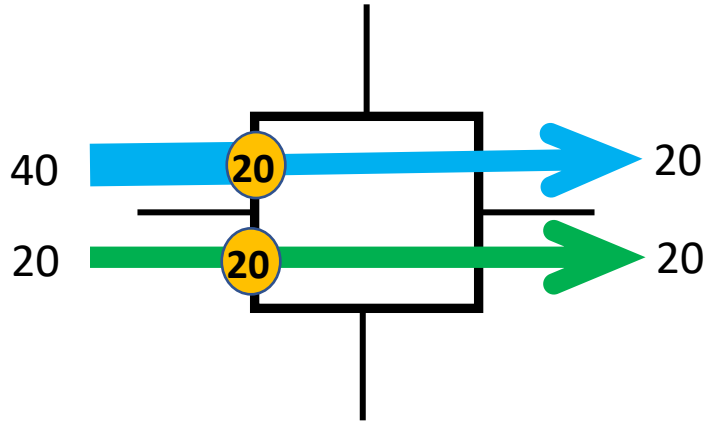
Higher number of filters required.



Ingress policing options

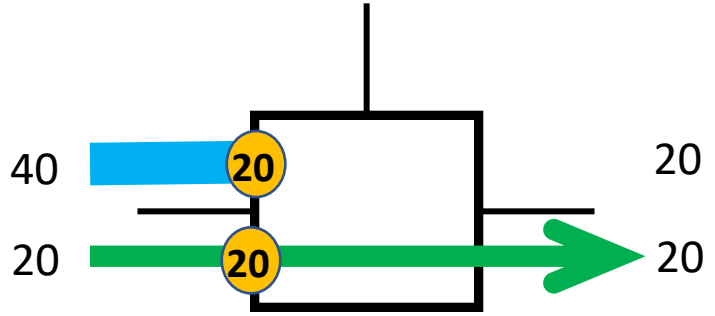
Criteria 2: What is the response if the threshold is exceeded?

Threshold Enforcing IPF:



Blocking IPF:

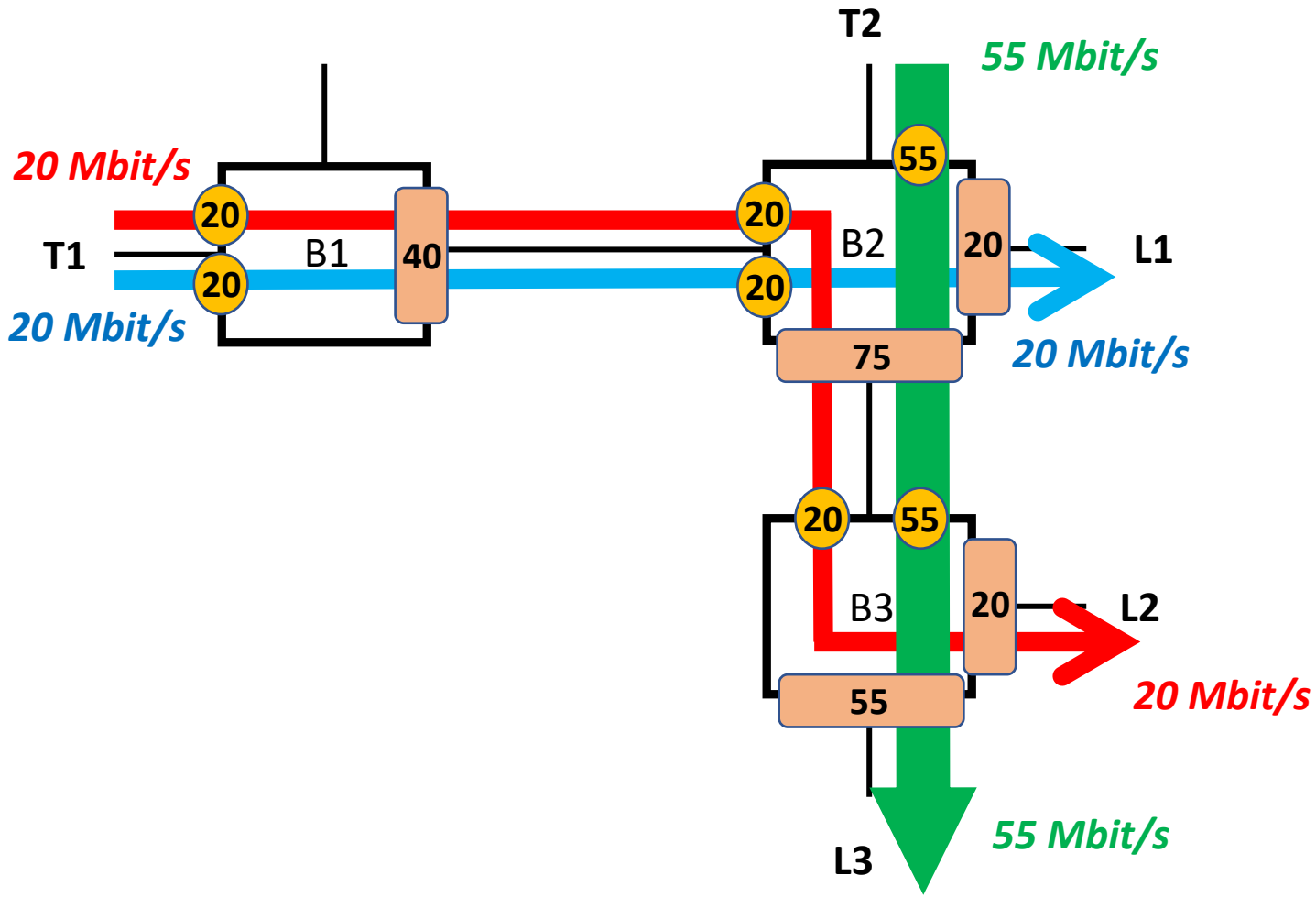
- Blocking is **permanent**
- Resetting the filter requires host interaction.



These diagrams show threshold enforcing on a per stream basis

Example: fault-free case

Streams **T1-red**, **T1-blue**, **T2-green**

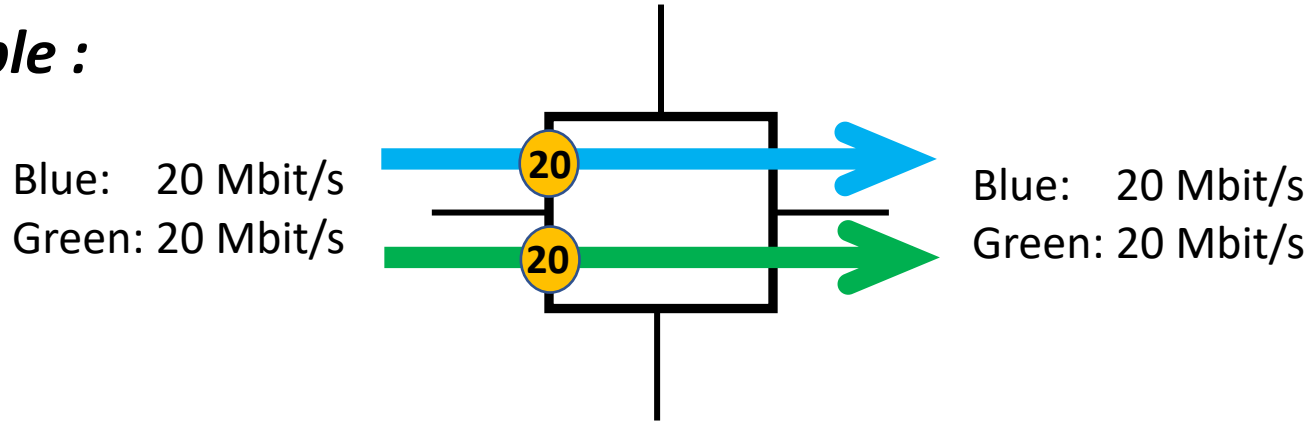


Four combinations

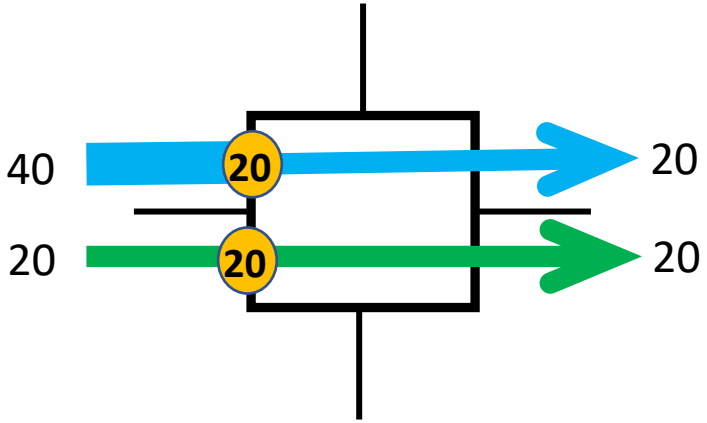
	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per-stream + threshold-enforcing

Example :

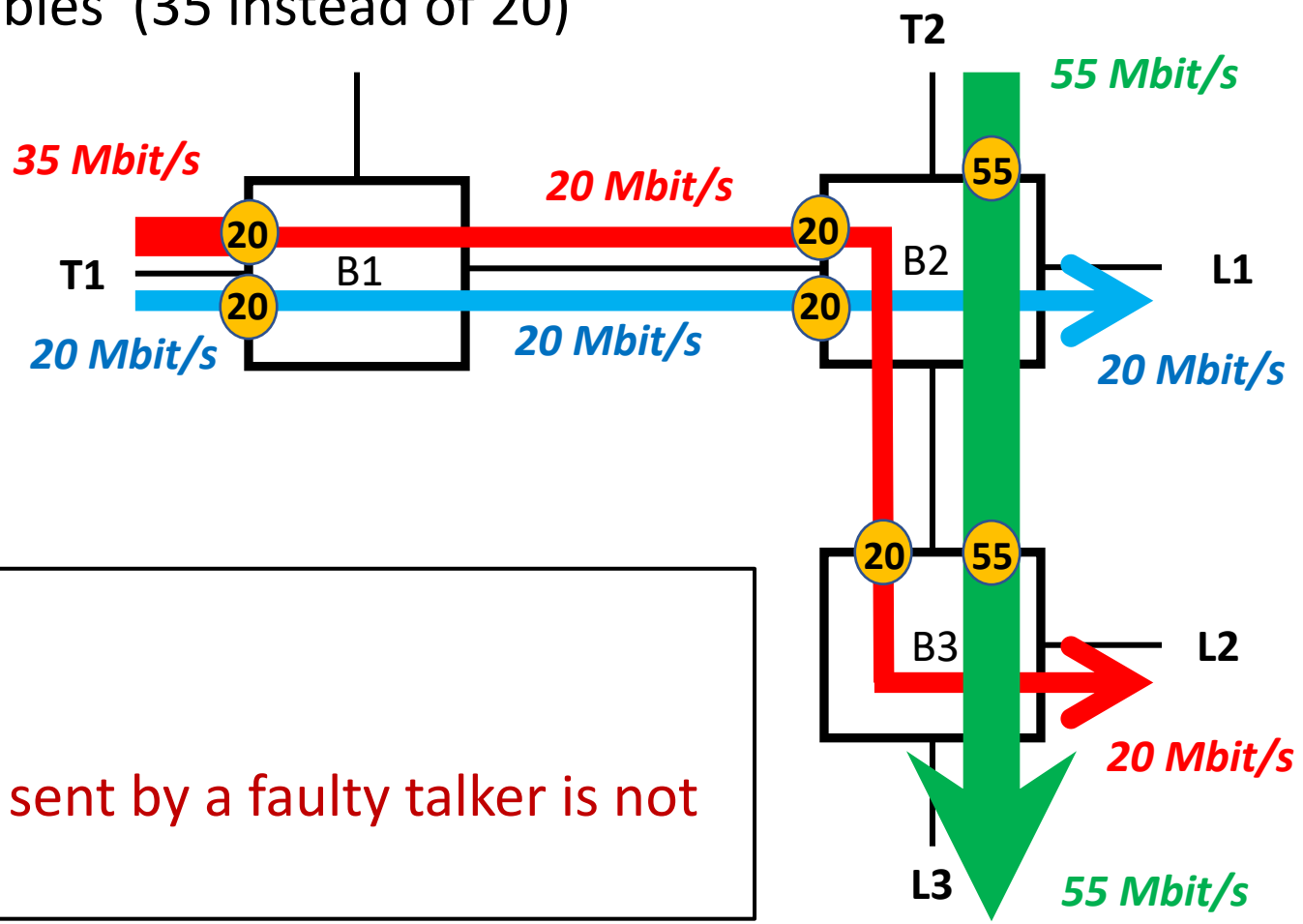


Fault: Blue stream babbles (40 Mbit/s instead of 20 Mbit/s)



Per-stream + threshold-enforcing

Fault: **T1-red** babbles (35 instead of 20)



Observations:

- T1-red:
  A faulty stream sent by a faulty talker is not "silenced".

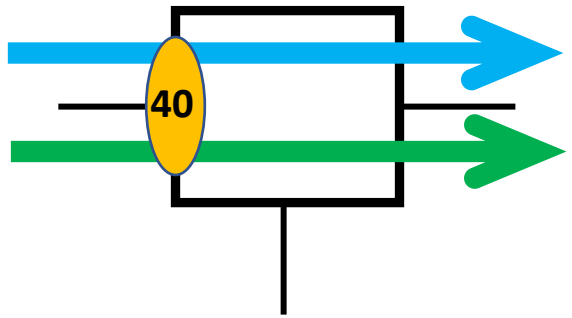
Four combinations

	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per-class + threshold-enforcing

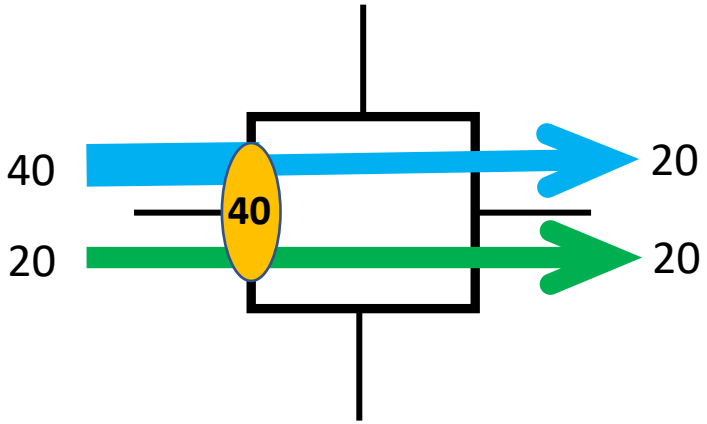
Example :

Blue: 20 Mbit/s
Green: 20 Mbit/s

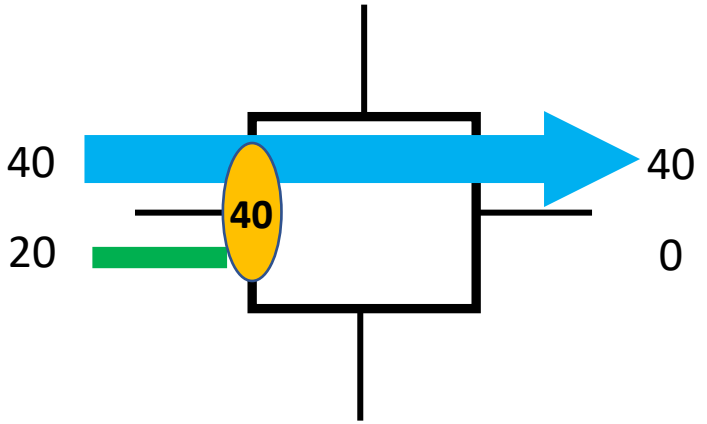


Blue: 20 Mbit/s
Green: 20 Mbit/s

Fault: Blue stream babbles (40 Mbit/s instead of 20 Mbit/s)



…

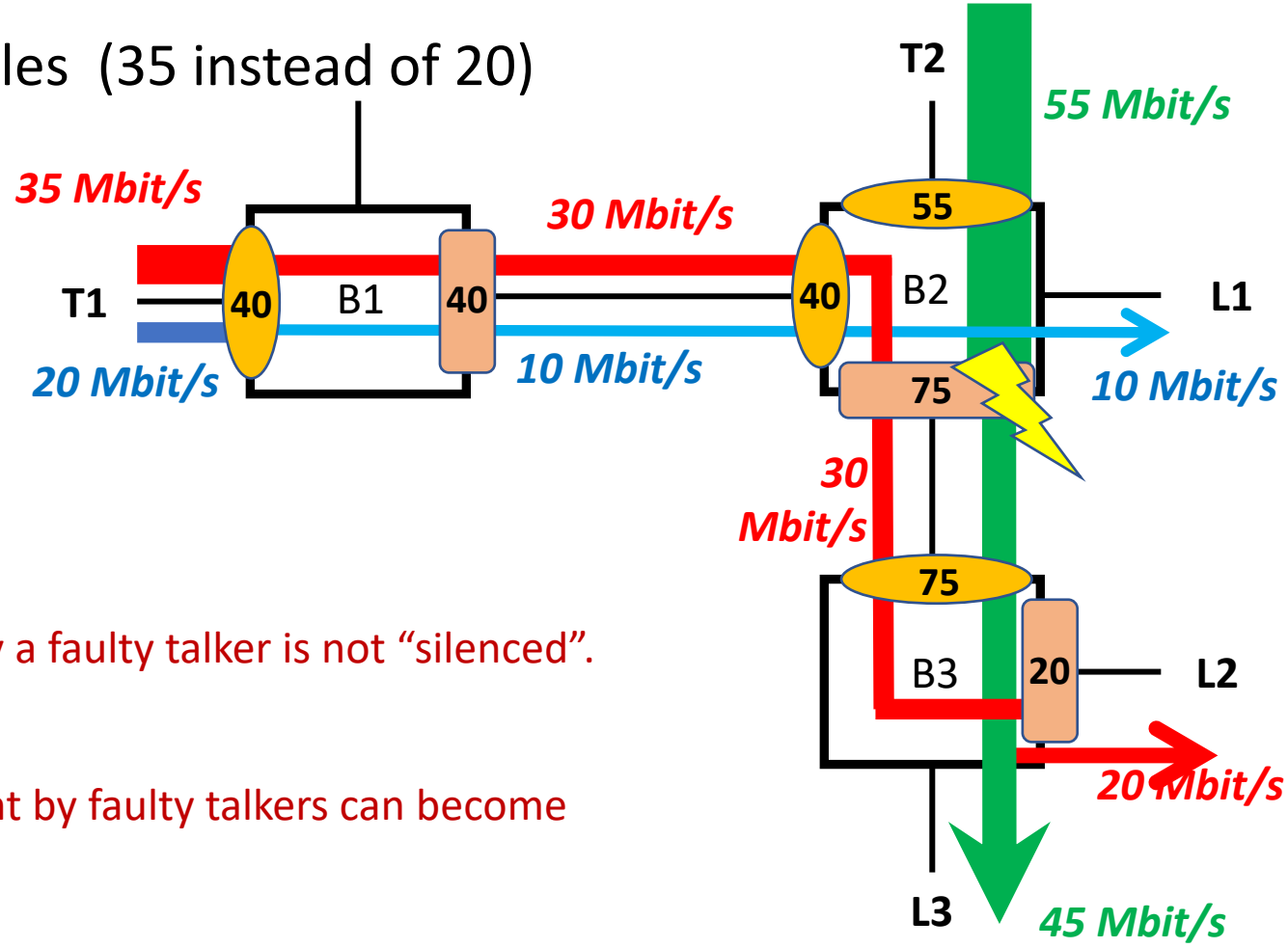


All kinds of behavior between the two above results are possible!

Since a per class ingress policing mechanism is not aware of any streams, it can only discard arbitrary class A frames once the established bandwidth threshold is exceeded. The discarded frames could be blue frames only, or green frames only, or any mix of blue and green frames we can think of.

Per-class + threshold-enforcing

Fault: **T1-red** babbles (35 instead of 20)



Observations:

- T1-red:
  A faulty stream sent by a faulty talker is not “silenced”.
- T1-blue:
  Non-faulty streams sent by faulty talkers can become faulty.
- T2-green:
  A fault free stream sent by a fault free talker becomes faulty. (Fault propagation. Fault not contained)

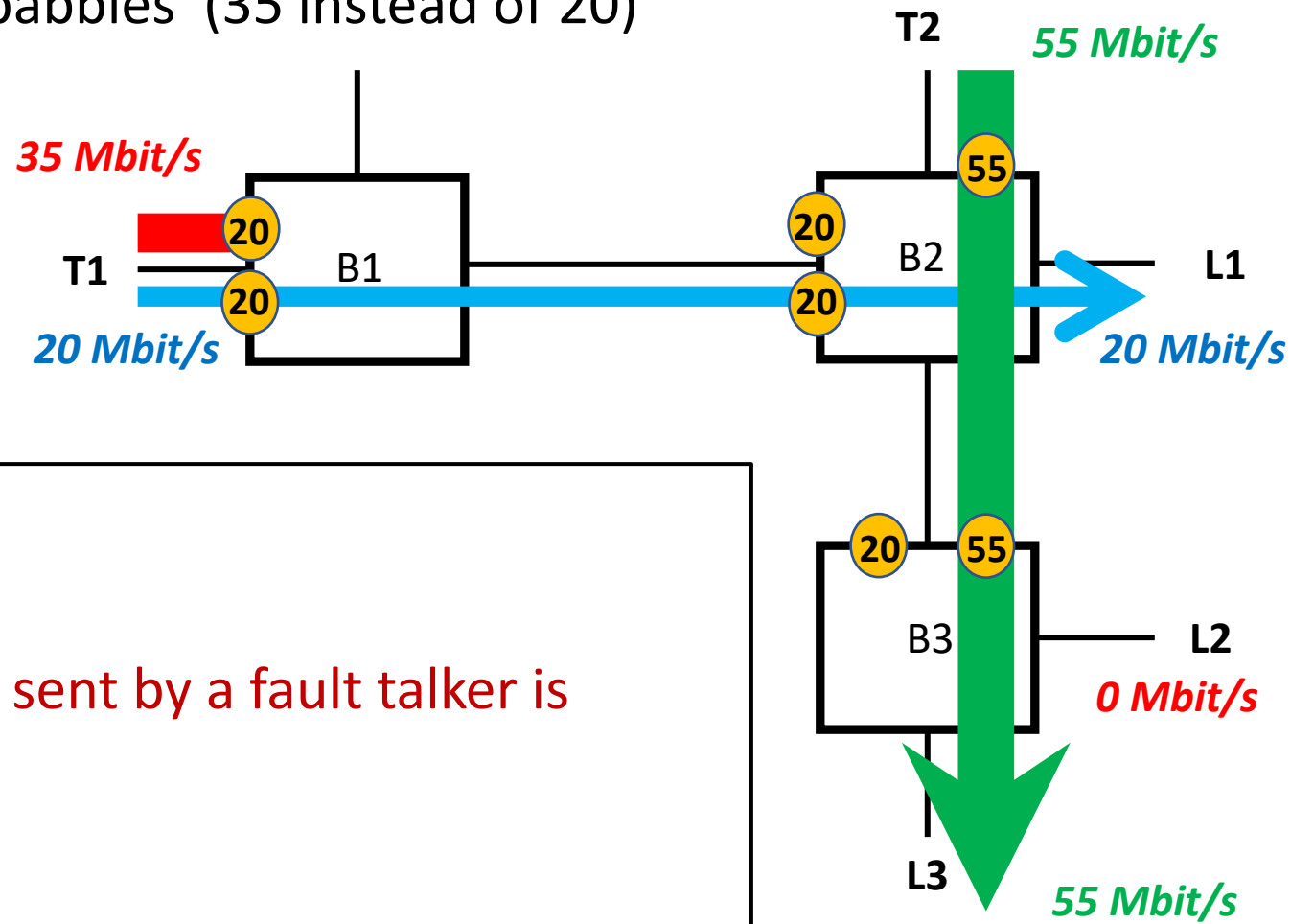
Note: This diagram shows one out of many different ways of how things could play out.

Four combinations

	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per-stream + blocking

- Fault: **T1-red** babbles (35 instead of 20)



Observations:

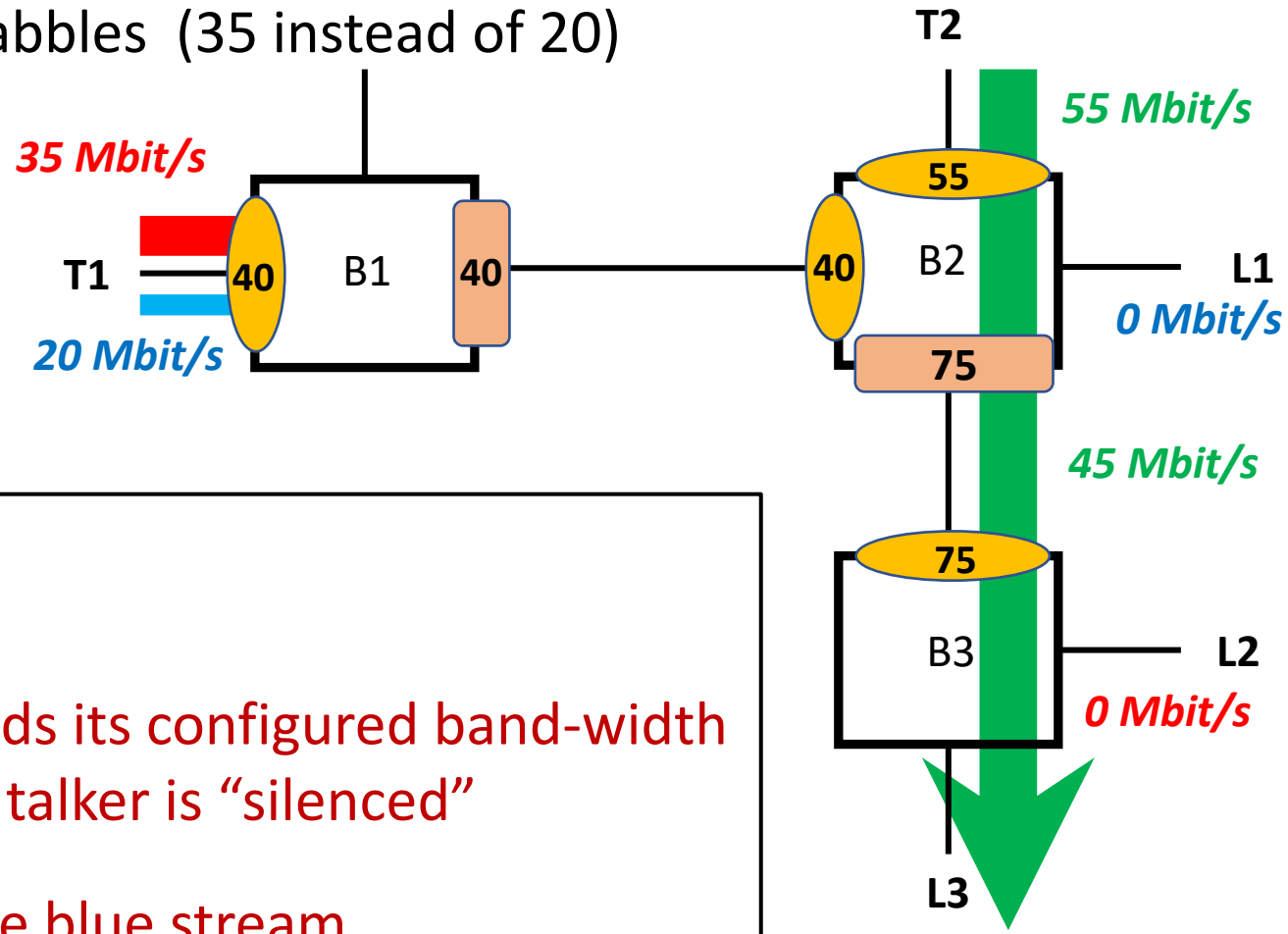
- T1-red:
  A faulty stream sent by a fault talker is silenced.
- T1-blue:
  Non-faulty streams sent by faulty talker are not necessarily silenced.

Four combinations

	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per-class + blocking

- Fault: **T1-red** babbles (35 instead of 20)

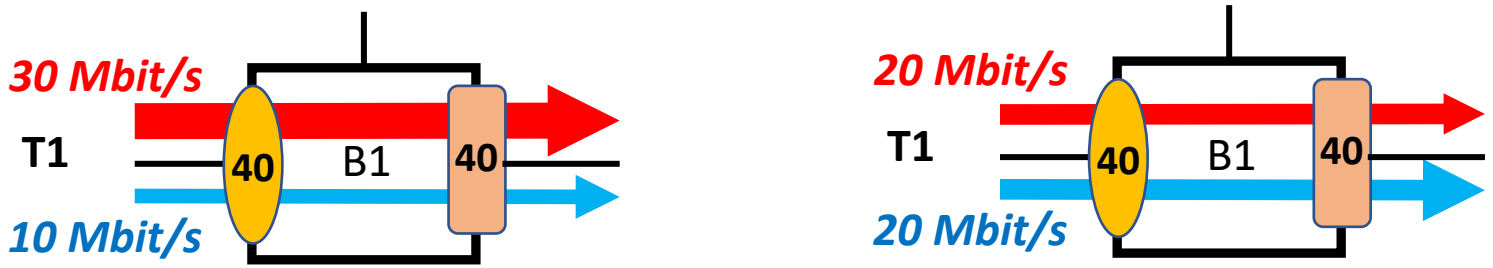


Observations:

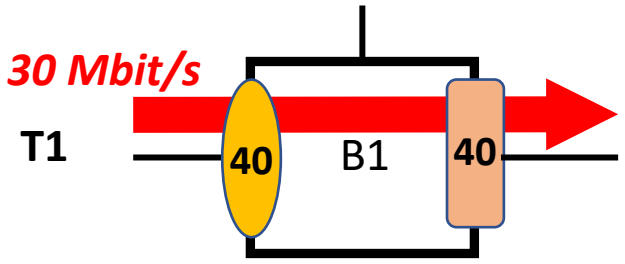
- T1:
  If a talker exceeds its configured band-width limit, the faulty talker is “silenced”
  - Including the blue stream

Per-class + blocking: “moderate” babbler

- **Moderate Babbler:**
  - Does not exceed the IPF bandwidth threshold.
  - Sends too much on one stream, but less on another.
  - Example: T1 sends 30+10 instead of 20+20.



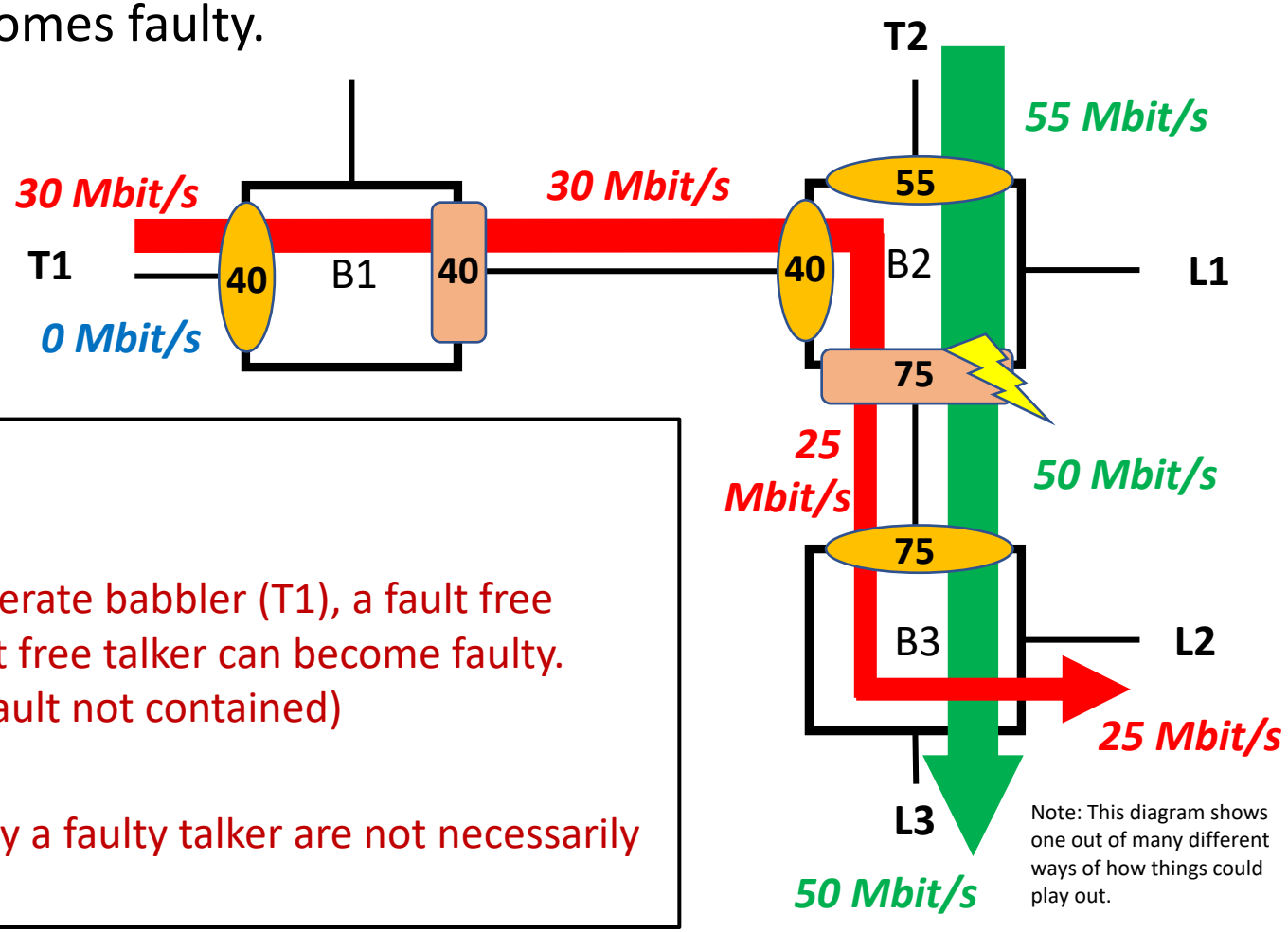
- **More realistic example of a Moderate Babbler:**



  - Streams do not necessarily permanently use their reserved bandwidth
  - Imagine:
    “Blue” has currently (temporarily) nothing to transmitting and “red” starts to babble.

Per-class + blocking: “moderate” babbler

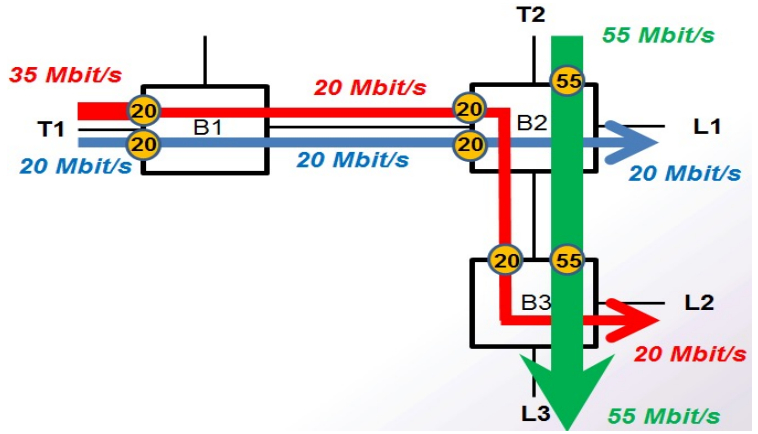
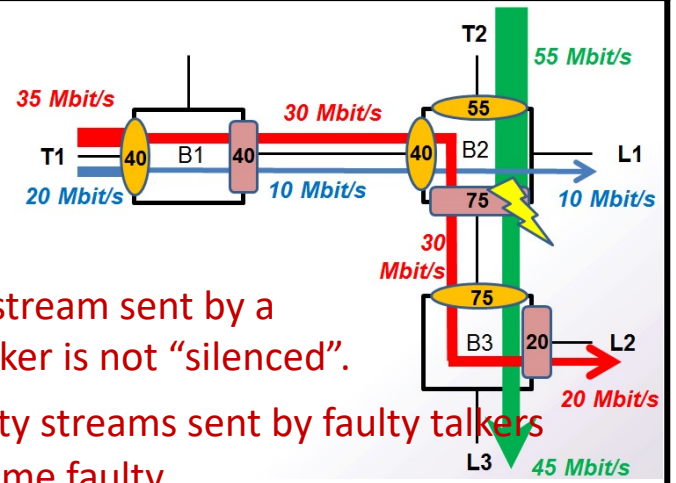
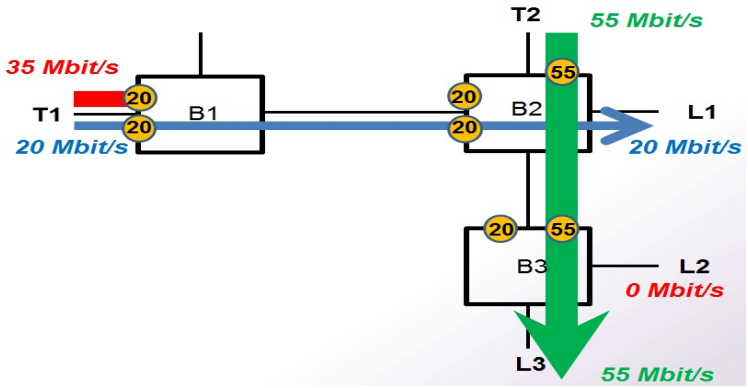
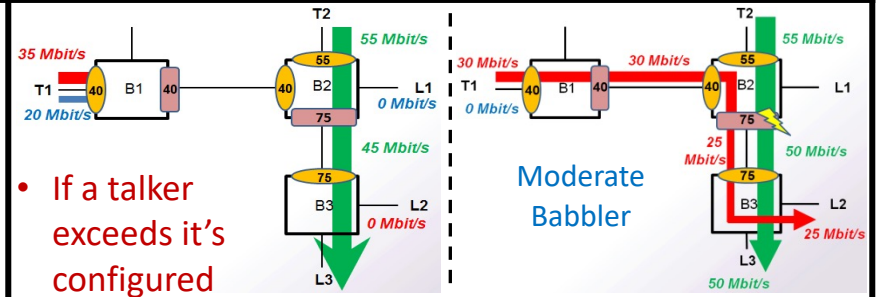
- Moderate Babbler T1: 30 + 0 instead of 20+20
- Shaper at B2 drops frames
  => T2-green becomes faulty.



Observations:

- T2-green:
  In presence of a moderate babbler (T1), a fault free stream sent by a fault free talker can become faulty. (Fault propagation. Fault not contained)
- T1-red:
  Faulty streams sent by a faulty talker are not necessarily silenced.

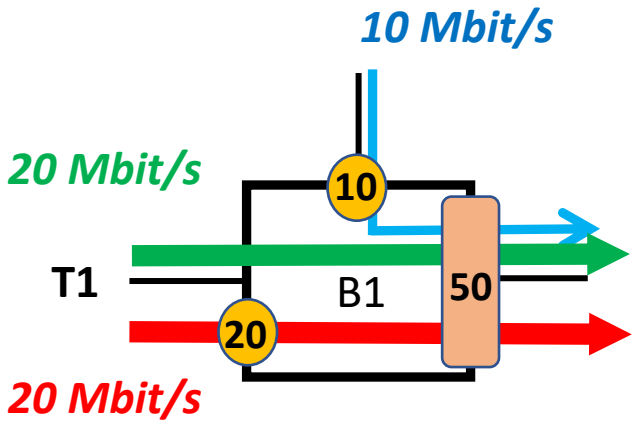
Comparison

	Per Stream *(= Potentially higher number of filters per port)*	Per Class *(= Small number of filters per port)*
Threshold Enforcing	 • A faulty stream sent by a faulty talker is not “silenced”. • Other streams from faulty / fault free talkers not affected.	 • A faulty stream sent by a faulty talker is not “silenced”. • Non-faulty streams sent by faulty talkers can become faulty. • A fault free stream sent by a fault free talker becomes faulty. (Fault propagation. Fault not contained)
Blocking	 • A faulty stream sent by faulty talker is “silenced”.	 • If a talker exceeds it’s configured bandwidth limit, the faulty talker is “silenced”. • In presence of a moderate babbler, a fault free stream sent by a fault free talker can become faulty. (Fault propagation. Fault not contained). • Faulty streams sent by a faulty talker are not necessarily silenced.

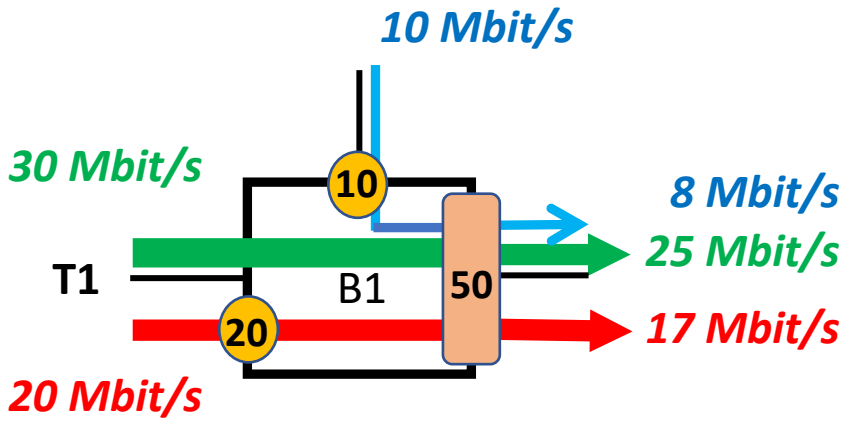
How many filters for per-stream policing?

- We need IPFs for safety-critical streams (to detect critical errors)
- But we've also seen that other streams need IPFs (to avoid error propagation)
- One IPF per stream at each port may lead to a waste of hardware resources
  - It is also costly and adds chip area
- Can we find a compromise?

Less than one IPF per stream



Fault free case



Faulty case

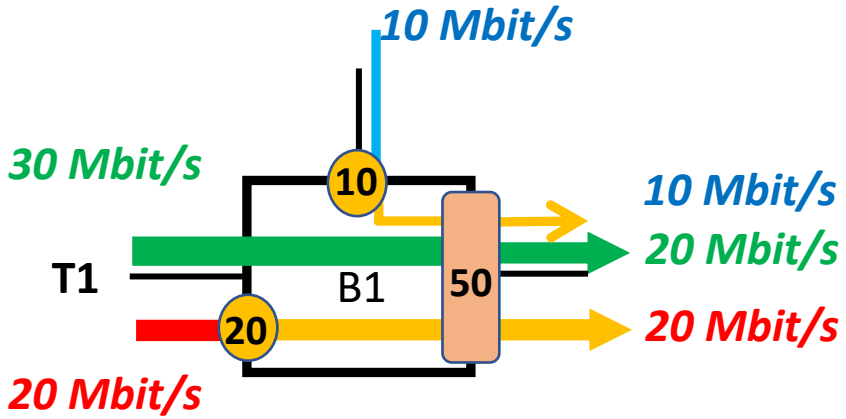
(T1-green sends 30 instead of 20)

Since there is no IPF for T1-green, the shaper will drop blue, green and red frames on egress!

Less than one IPF per stream

Now assume that:

- only some of the streams (red and blue) are safety critical.
- only safety critical streams will be sent through an IPF.
- streams that pass an IPF turn into golden streams.
- egress ports are configured to know which streams are golden.
- if an egress queue fills up too much, it will start to exclusively drop frames that are not golden.

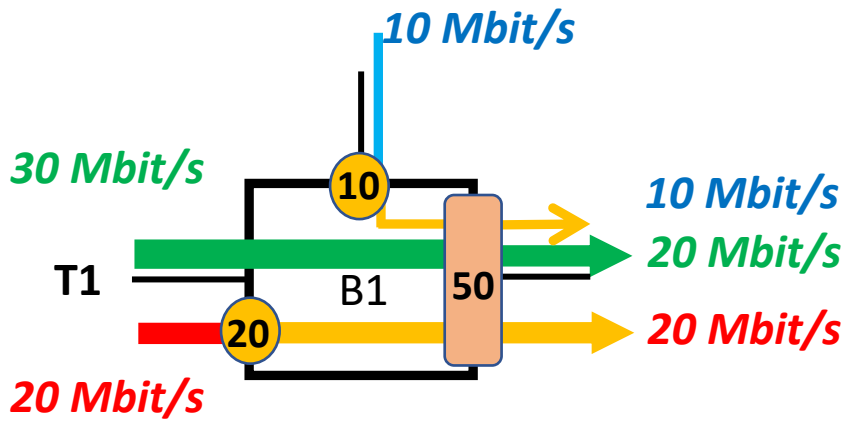


Faulty case

(T1-green sends 30 instead of 20)

Less than one IPF per stream

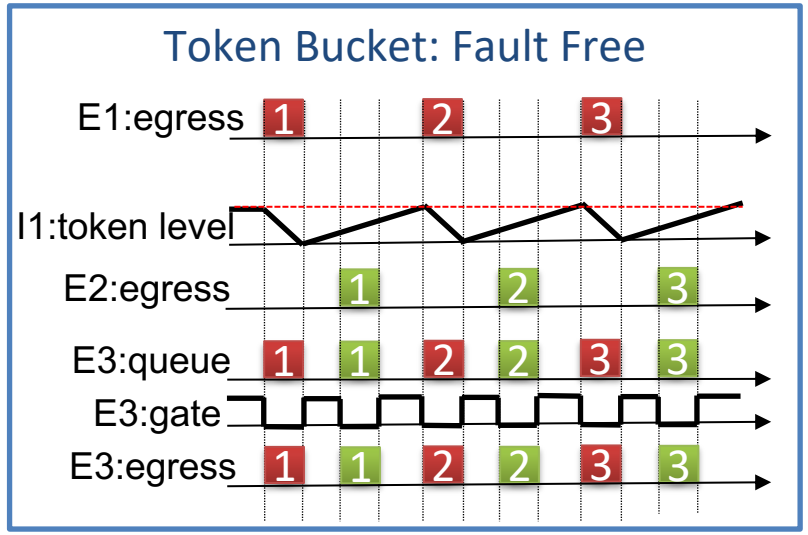
- We can thus reduce the number of IPFs to the anticipated maximum number of safety-critical streams per port …
- … without imposing any limitation on total number of streams
- Requires changes in egress port

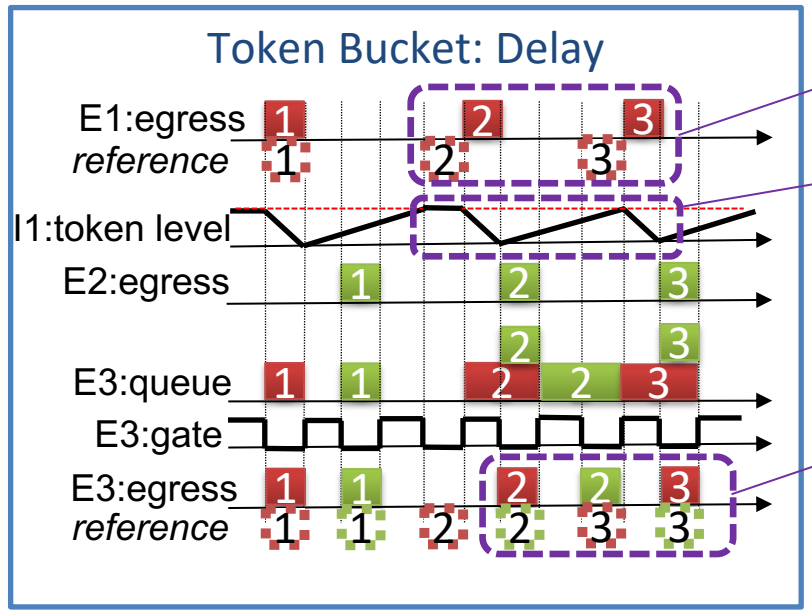


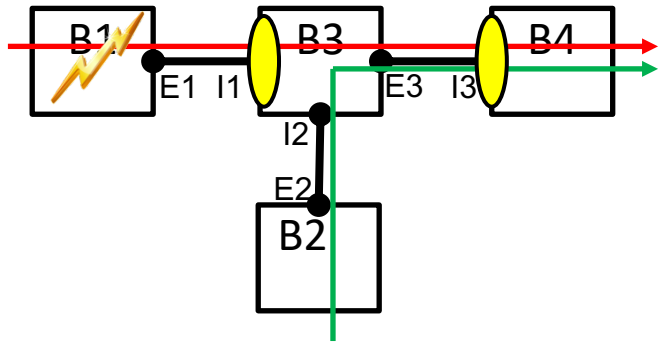
Faulty case

(T1-green sends 30 instead of 20)

Token bucket alone does not work for TAS







Delayed Packets

Token limit reached, but this does not affect delayed packet acceptance

Delayed packet 2 of B1 (faulty) congests the queue: Packets 2, 2 and 3 sent in wrong windows

LINKÖPING UNIVERSITY

”Reverse 802.1Qbv” gates defined in 802.1Qci

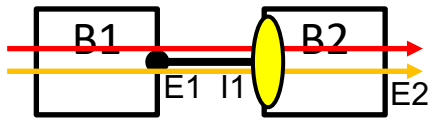
Ingress Windows

Extend the 802.1Qbv gate-states by an ingress open/close flag, i.e. ingress gate:

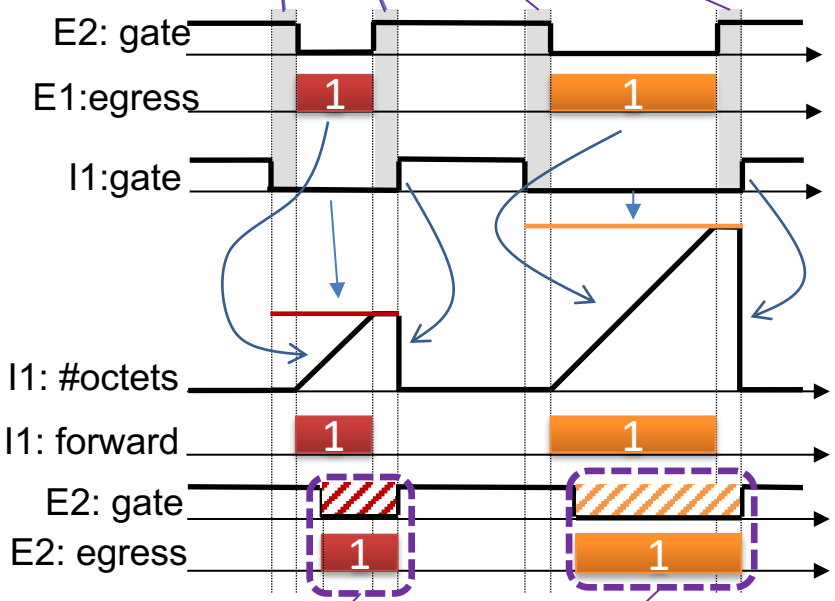
- Open: Accept consecutive started packets until next ingress close
- Close: Discard consecutive started packets entirely

Implication:
Common time for egress and ingress operation at the same port

Fault-free case

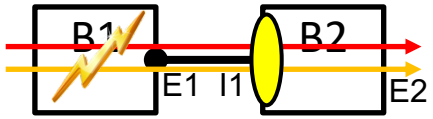


Variances (PTP, 802.1Qbv, …)



Scheduling: Egress windows aligned to the end of corresponding ingress windows (or later) prevents increasing window size (tolerance) along path

Faults covered by ingress windows

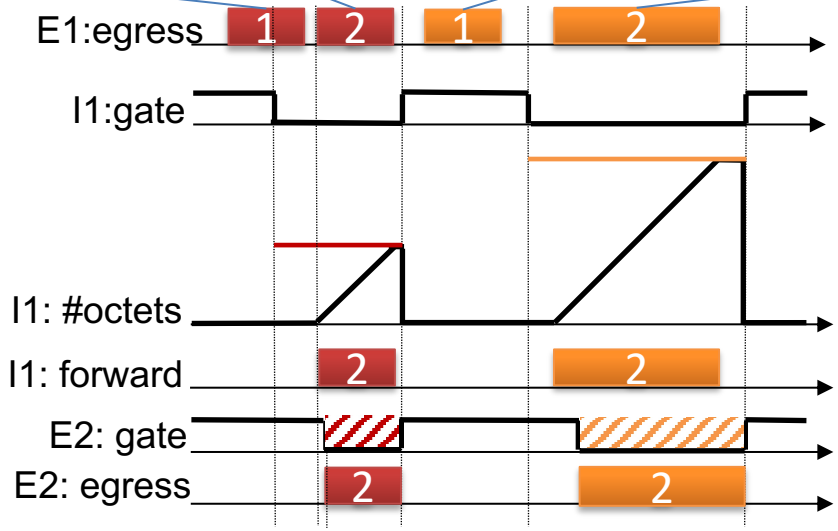


Starts before ingress window → Entirely discarded

Starts in ingress window → Ok

Starts out of ingress window → Entirely discarded

Expected → ok



Solutions in standard

- 802.1Qci

- Any of the mentioned filters can be applied to a stream identified by the following alternatives:
  - Source MAC address and VLAN identifier
  - Destination MAC address and VLAN identifier

Summary

- Ingress filtering and policing is required to properly detect and isolate temporal errors in the network
- Without it, errors can propagate and “steal” reservations from other streams are behaving (i.e., sending no more than the maximum amount of bandwidth/time that has been reserved)
- 802.1Qci defines ingress filtering gates that can monitor bandwidth (with a token bucket algorithm) and/or monitor that the system behaves according to the planned 802.1Qbv schedule
- Also, good as one layer of defense against some DoS attacks

Redundancy: frame replication and elimination

Soheil Samii

802.1CB

- Frame Replication and Elimination for Reliability (FRER)
- Specified protocols for bridges and end systems:
  - Replication of packets
  - Identification of duplicate packets
  - Redundant transmission
  - Merge points and elimination of redundant packets
  - Optional: Proxy mode of operation
  - Optional: Auto-configuration to establish redundant paths

802.1CB history

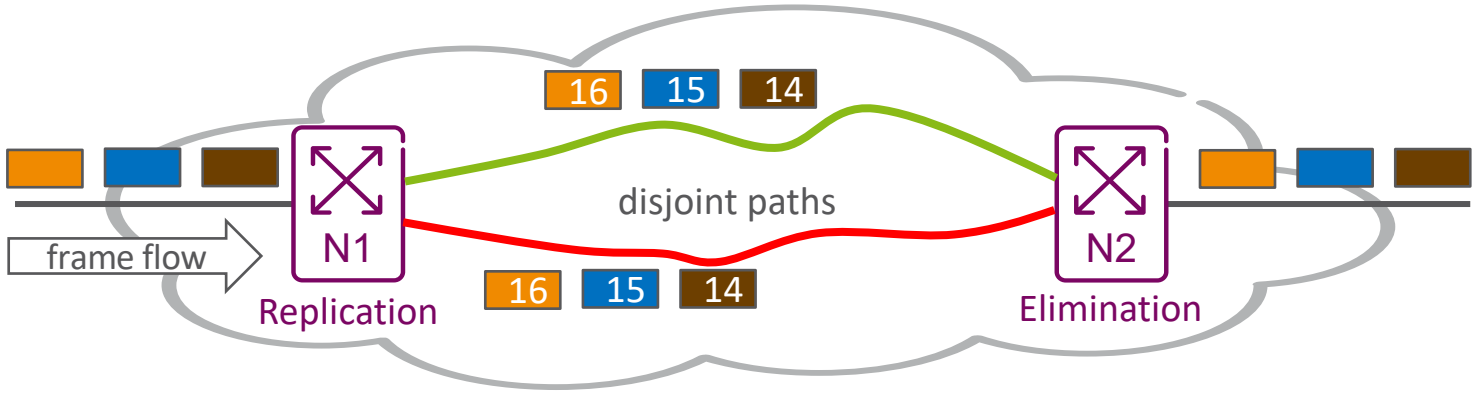
- Industrial automation systems already implemented redundancy on top (proprietary) Ethernet
  - PRP: Parallel Redundancy Protocol
  - HSR: High-availability Seamless Redundancy
- Need to standardize in 802.1
  - Industrial automation (converging towards IEEE 802 standardized Ethernet networks)
  - Professional audio/video needs redundancy for availability reasons
  - Automotive, and other safety critical application domains, have fail-operational requirements

802.1CB goals

- Increase probability that a given packet will be delivered on time
- Consider a range of failures in the communication path that could cause packet errors or packet drops:
  - Connector
  - Wire
  - Electrical components on PCB
  - PHY and MAC
  - Switch internal errors
  - Power
  - Software

Frame replication and elimination

- Add sequence numbers to frames
- Send on two maximally disjoint paths
- Then combine and delete extras



Redundancy without 802.1CB?

- The 802.1 Rapid Spanning Tree Protocol (RSTP) is a distributed agreement protocol used to disable loops in a given physical network topology
- In case of link or switch failures, RSTP will enable previously disabled links to re-establish connectivity
- But this takes time and there is no worst-case latency guarantee
  - Not acceptable for applications with stringent availability requirements (e.g., autonomous driving or “Superbowl” commercials)
  - Some applications need “instantaneous” response in failure modes

802.1CB

- Identification of streams
  - Identify and mark packets
- Replication
  - Create copies and forward on redundant paths
- Elimination
  - Eliminate duplicate packets
  - Recipient has an “acceptance window” for frame duplicates arriving out of order

NEW

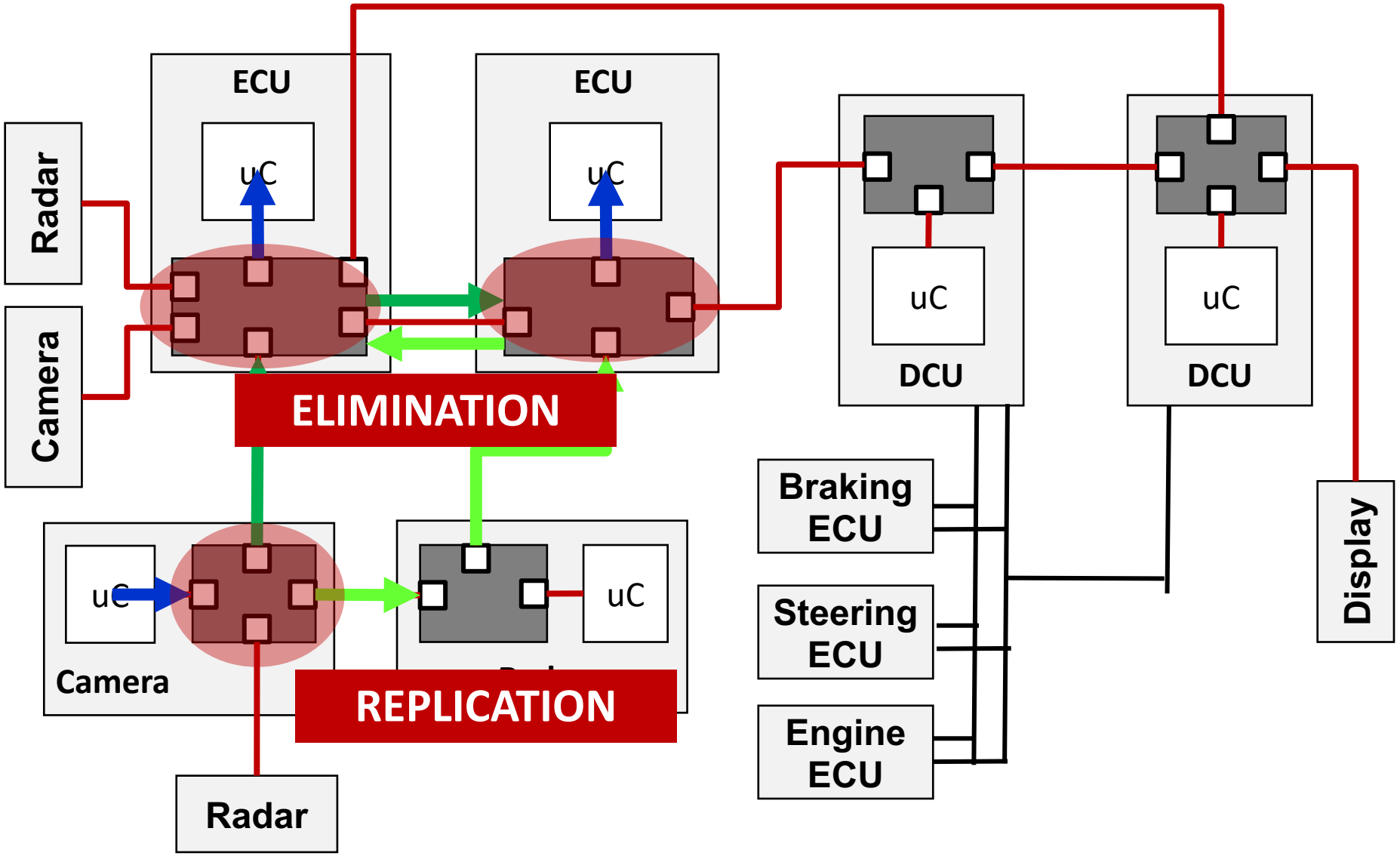
Field	Offset	Length
Destination MAC address	0	6
Source MAC address	6	6
C-tag EtherType	12	2
Priority, DE, VLAN ID	14	2
FRER Ethertype	16	2
sequence number	18	2
Payload Length/EtherType	20	2
data	22	*n*
Frame Check Sequence	22+*n*	4

Example Ethernet frame format with embedded R-Tag

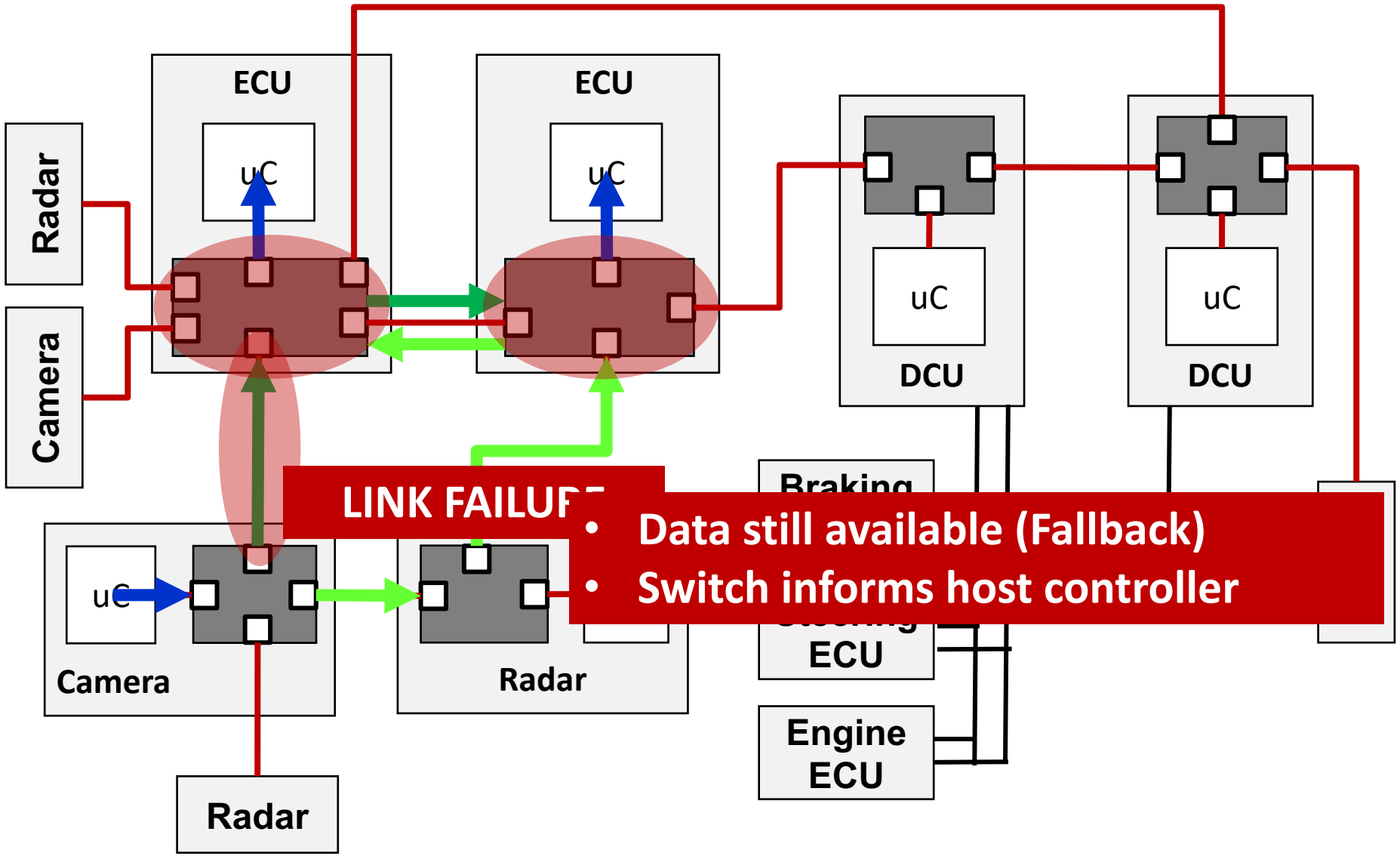
802.1CB operation

- Replicating packets at source or switch
- Send on separate paths
- Elimination of duplicates at sink or switch
- Proxy mode: all is handled by switches

Bridges with proxy mode



Link failure



- Data still available (Fallback)
- Switch informs host controller

The trade-offs

- Need rings in the network (additional links and switches)
- More bandwidth usage due to duplication
  - Need to pay attention to specific ports
  - Need to pay attention to latency increase caused to other flows in the network