Ingress Filtering and Policing

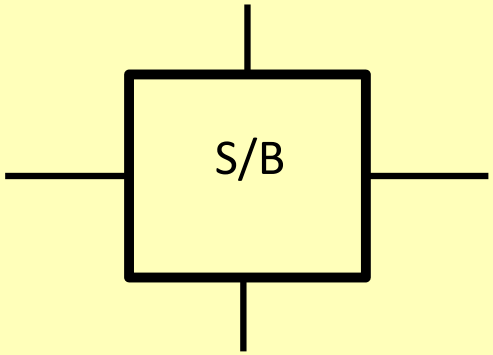
IEEE Std 802.1Qci-2017

802.1Qci: Ingress policing

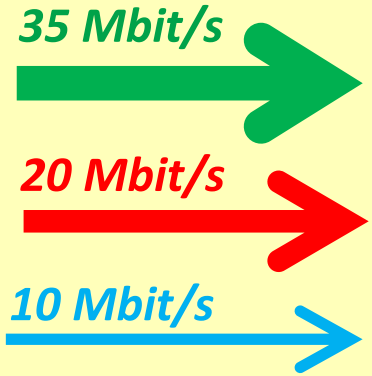
- Detect whether end stations violate timing contract
  - Sending more than has been reserved in the network
  - Sending outside their allowed time windows
  - Sending more than the maximum frame payload agreed upon
- Upon detection of error, isolate error from the network

Symbols and Abbreviations

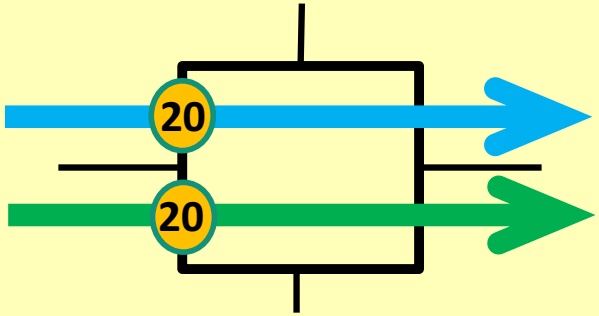
4 Port Switch/Bridge



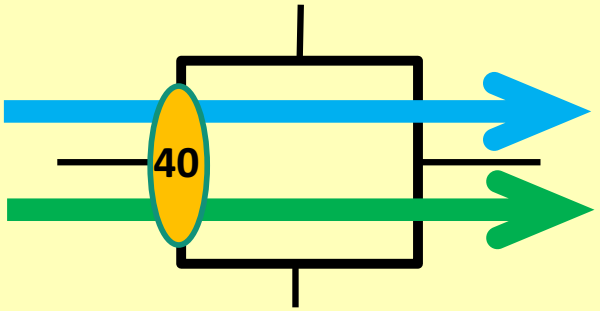
Class A Streams



Ingress Policing Filter

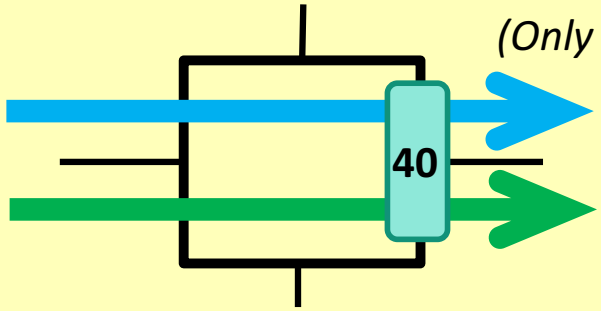


Two 20 Mbit/s Per Stream IPFs



40 Mbit/s Per Class IPF

Credit Based Shaper



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

- 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

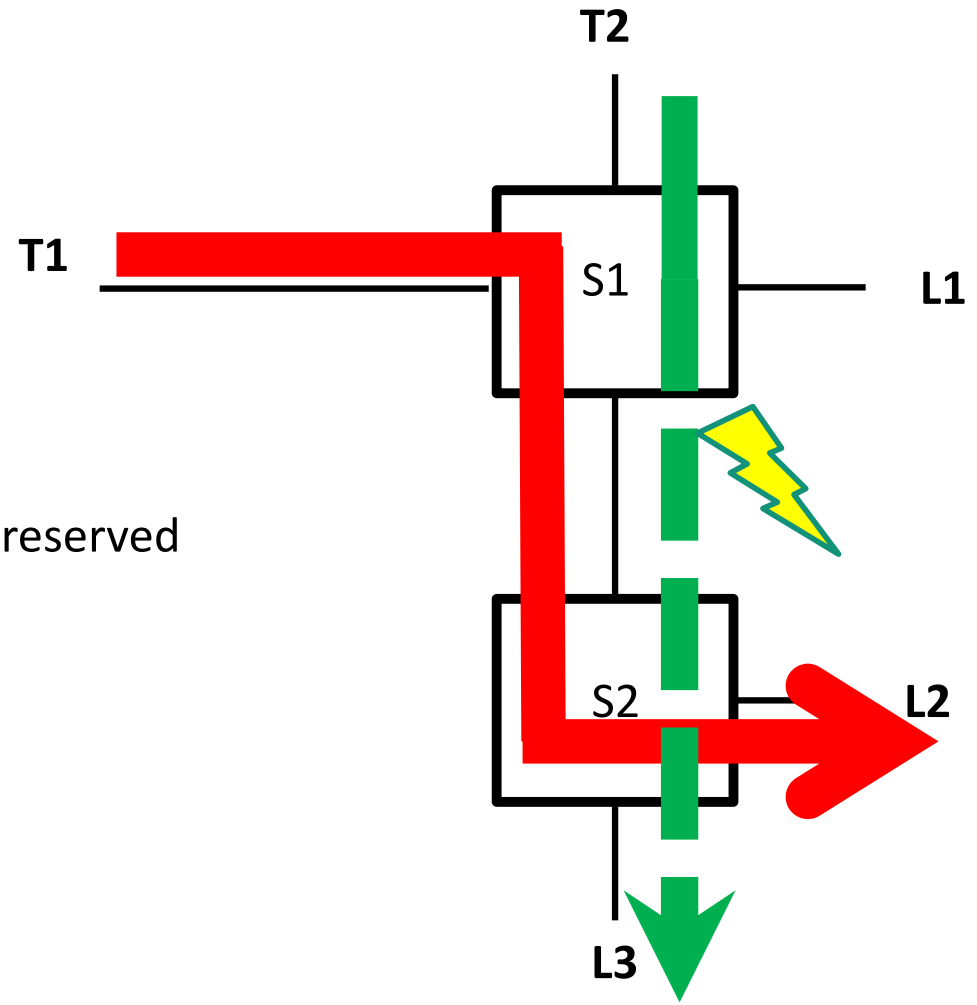
Example:

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

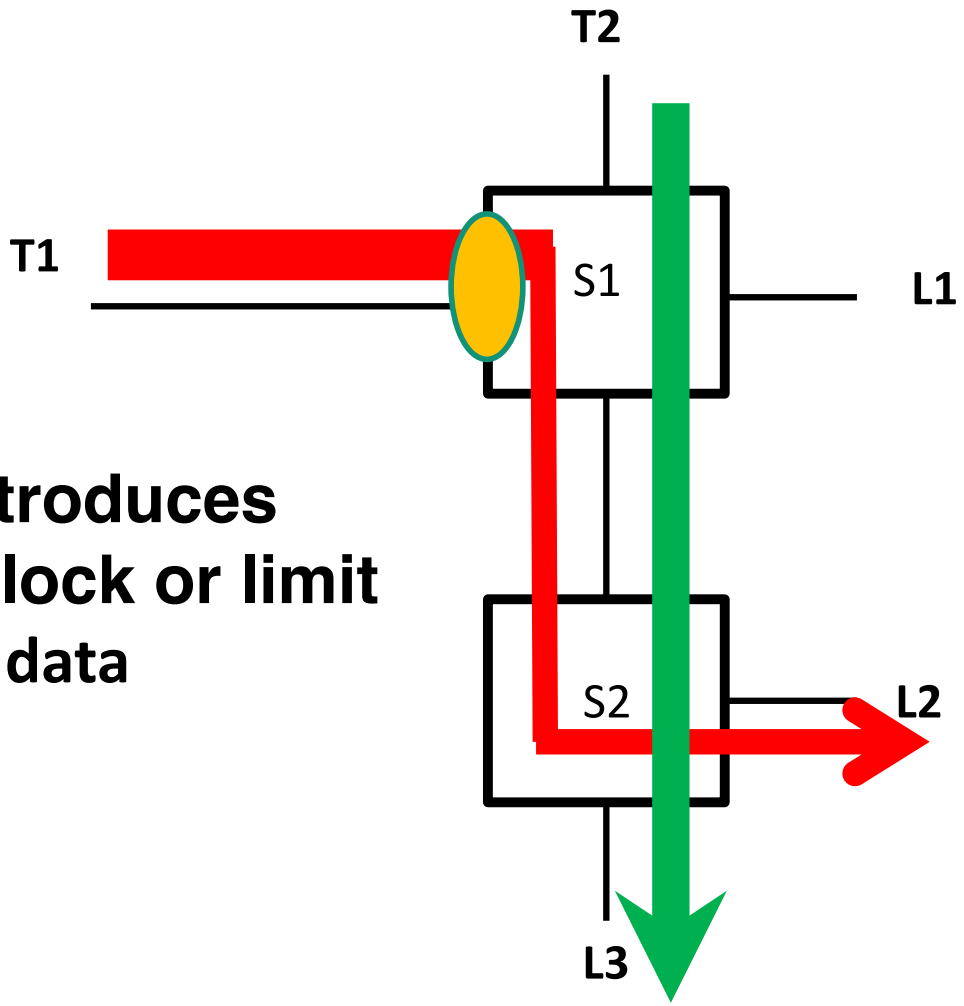
Red streams steals bandwidth that was reserved for the Green stream

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

But:
Green stream is faulty.



Ingress policing in a nutshell



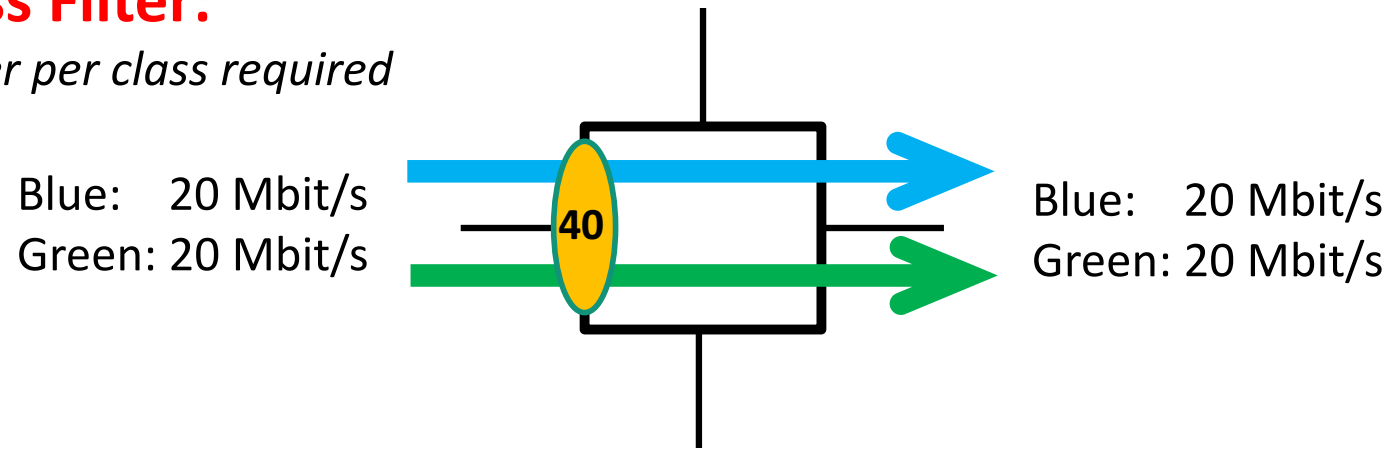
Ingress Policing introduces filters that will block or limit excessive amounts of data

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?”

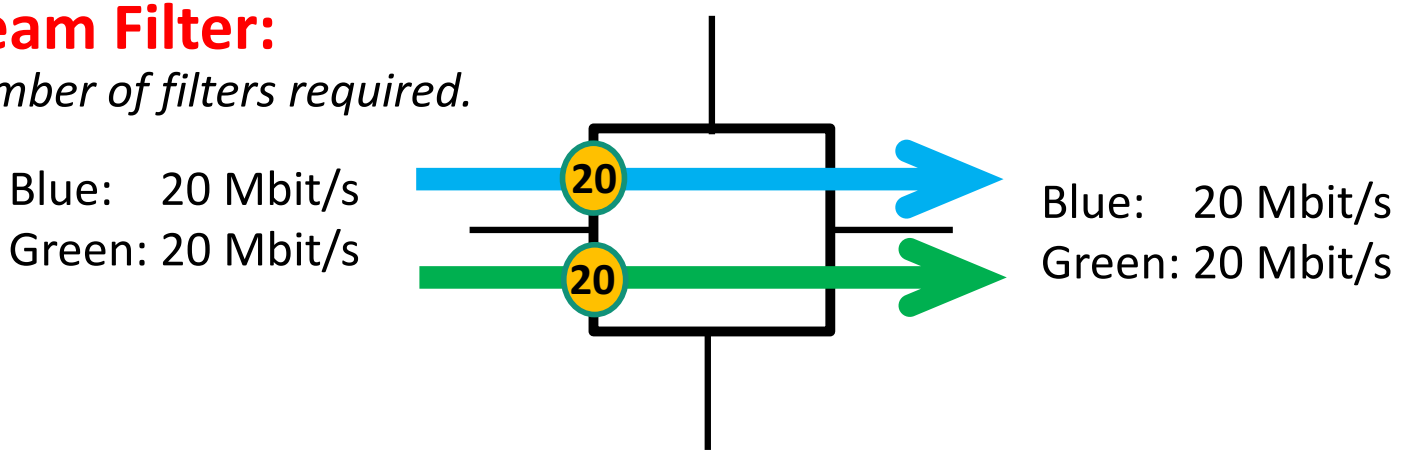
Per Class Filter:

Only 1 filter per class required



Per Stream Filter:

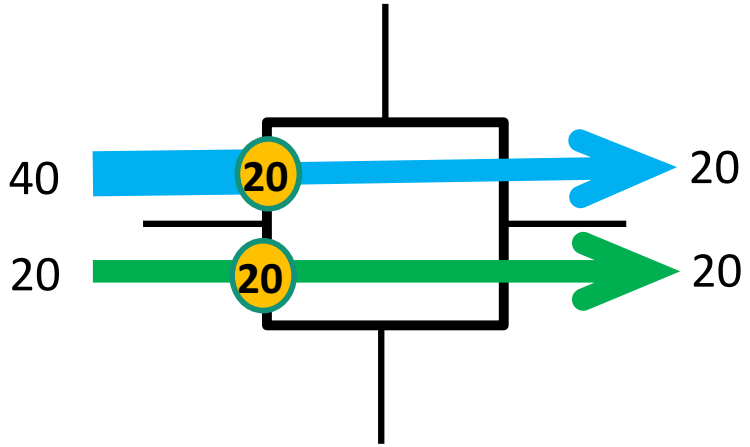
Higher number of filters required.



Ingress policing options

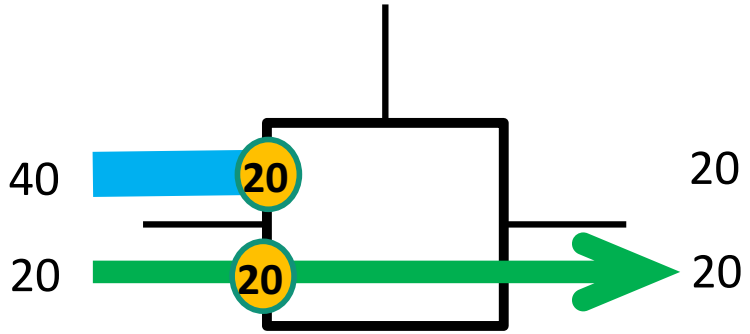
Criteria 2: What is the response if the threshold is exceeded?

Threshold Enforcing IPF:



Blocking IPF:

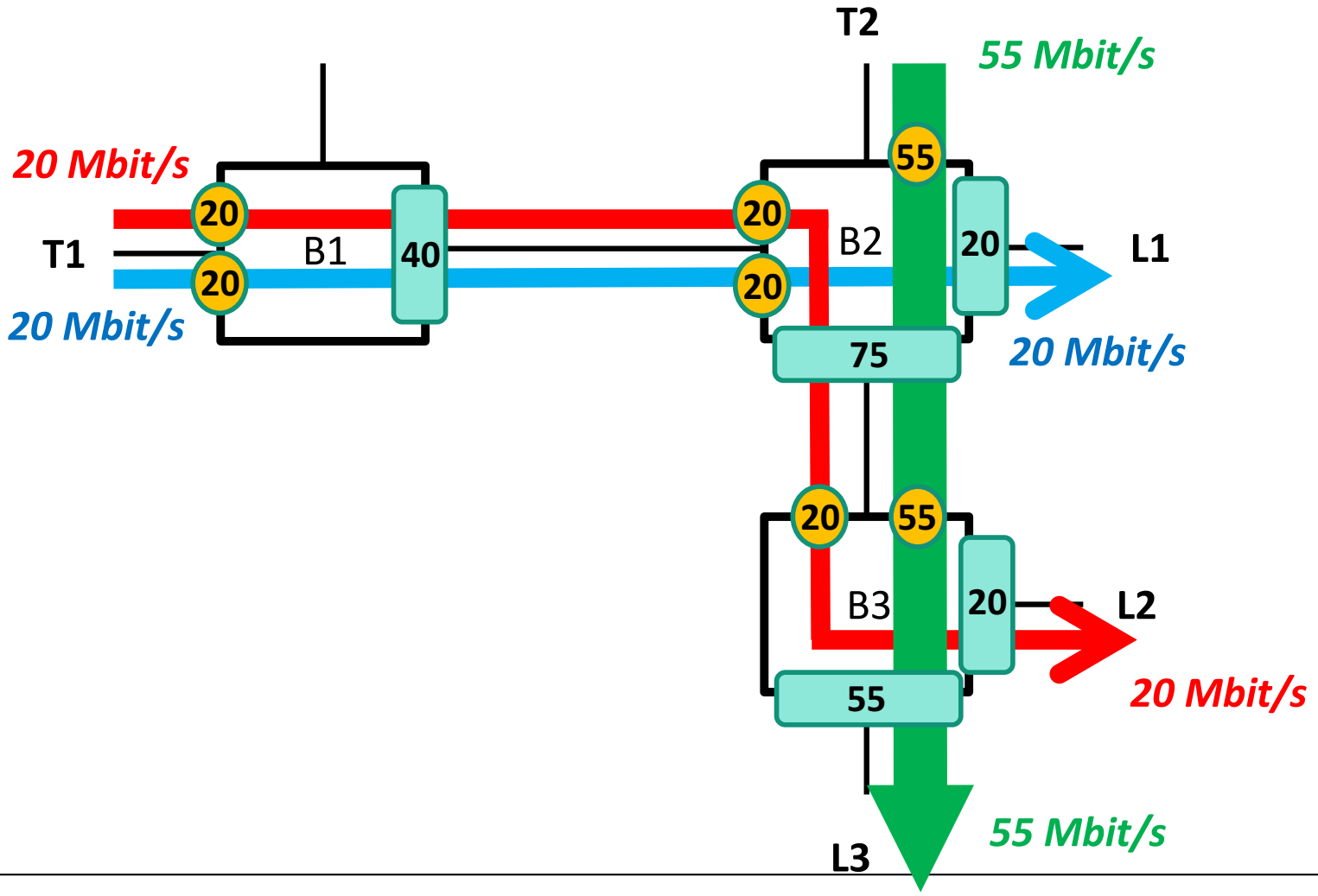
- Blocking is **permanent**
- Resetting the filter requires host interaction.



These diagrams show threshold enforcing on a per stream basis

Example: Fault-free case

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



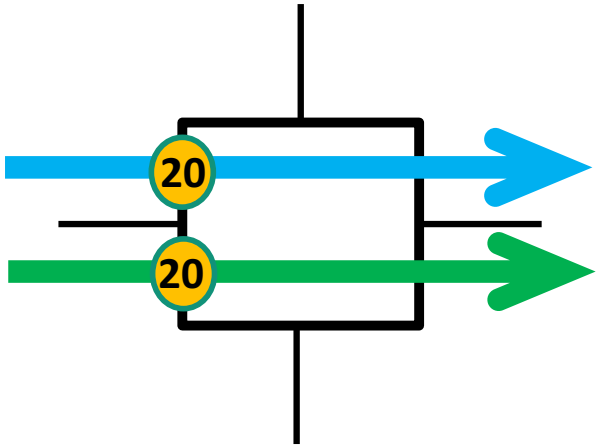
Four combinations

	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per-stream + Threshold enforcing

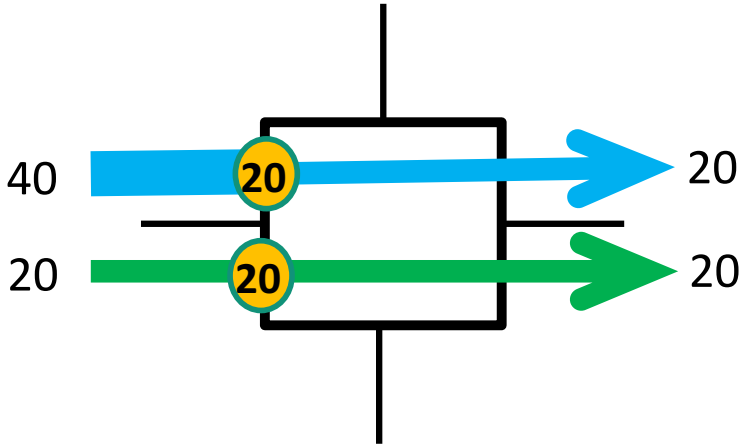
Example :

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



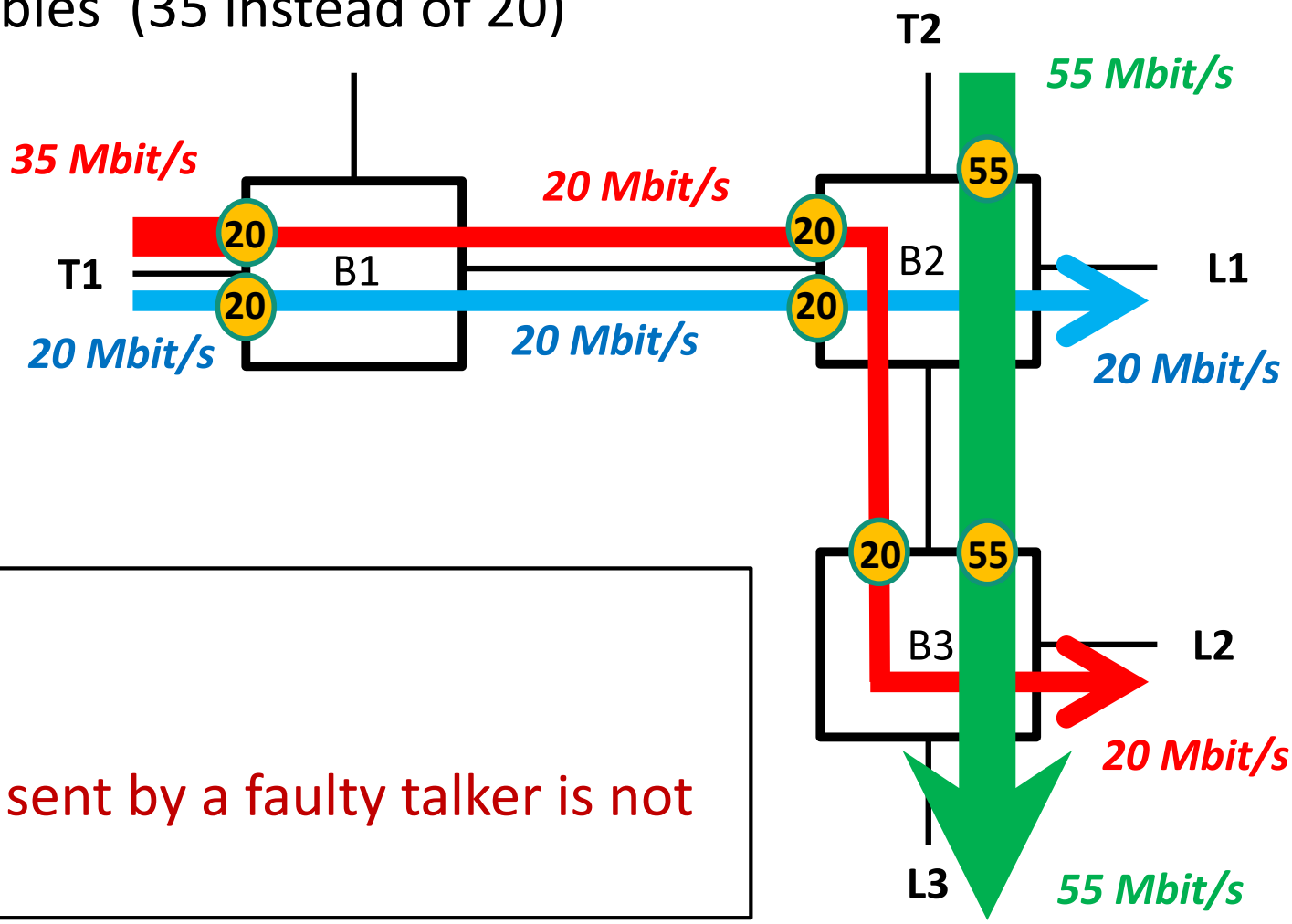
Blue: 20 Mbit/s
Green: 20 Mbit/s

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



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".

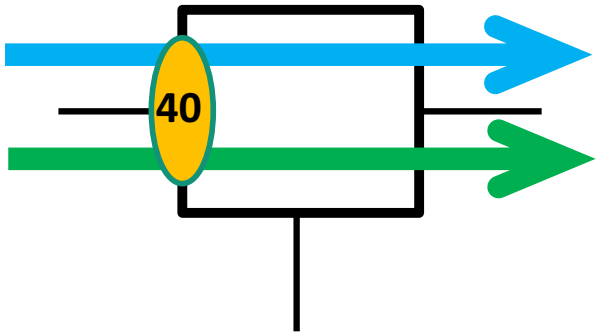
Four combinations

	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per class + Threshold Enforcing

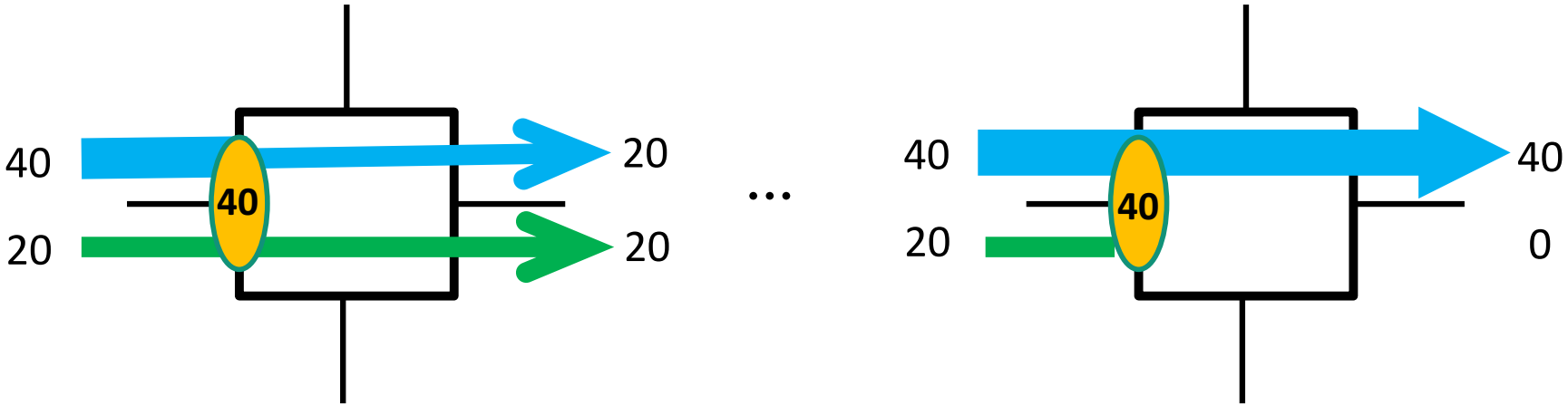
Example :

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)

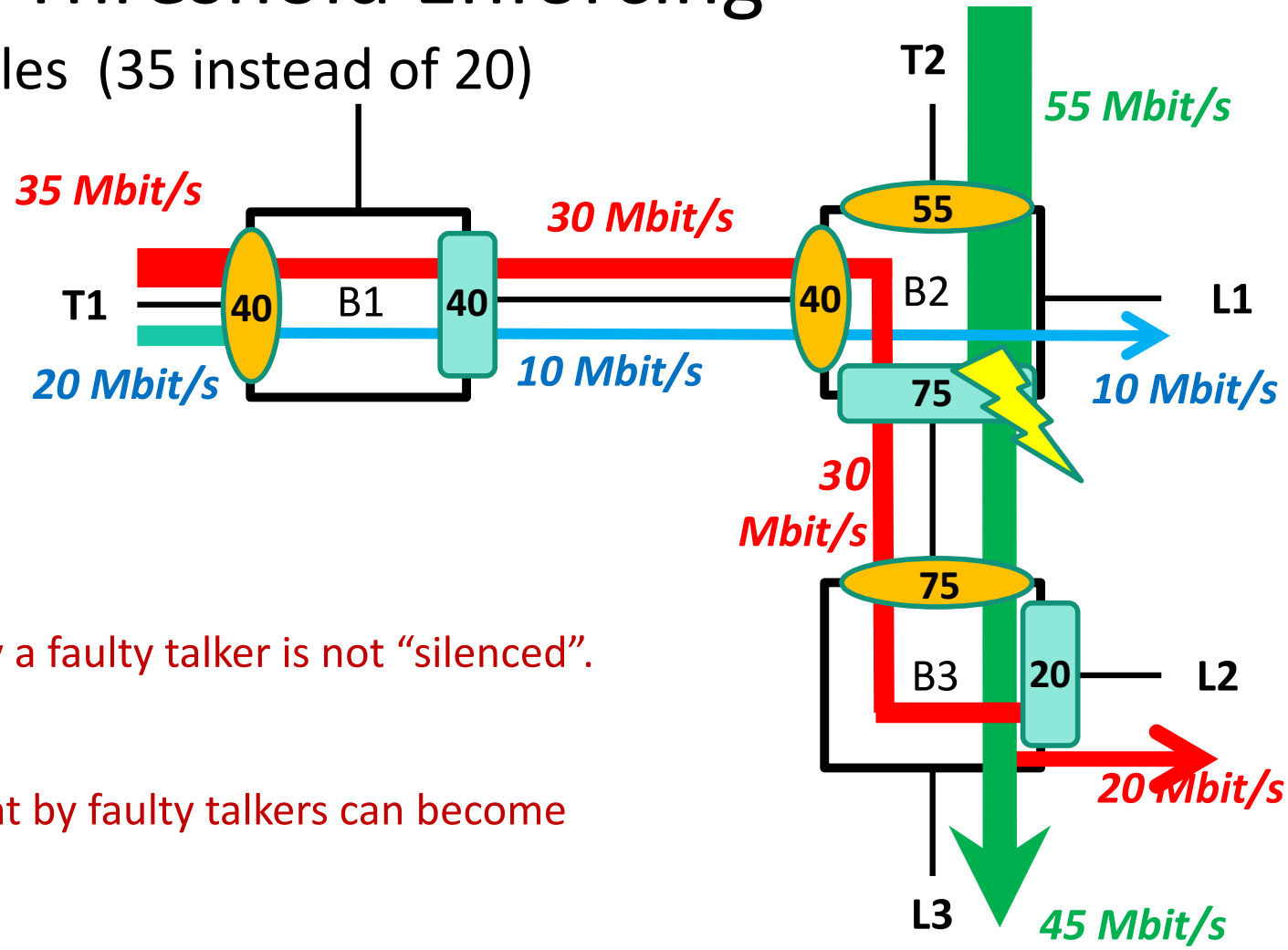


All kinds of behavior between the two above results are possible!

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)

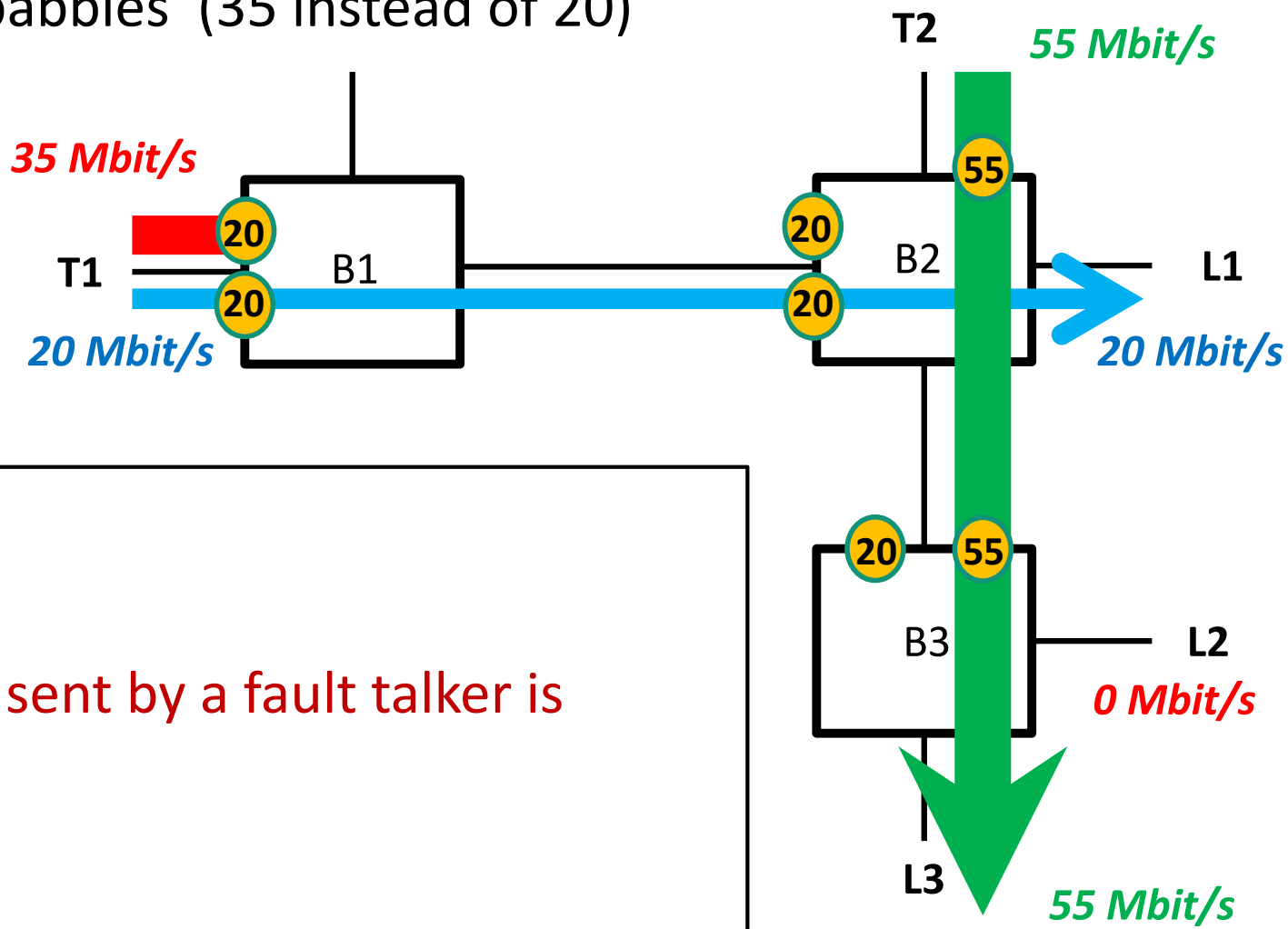
Note: This diagram shows one out of many different ways of how things could play out.

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:

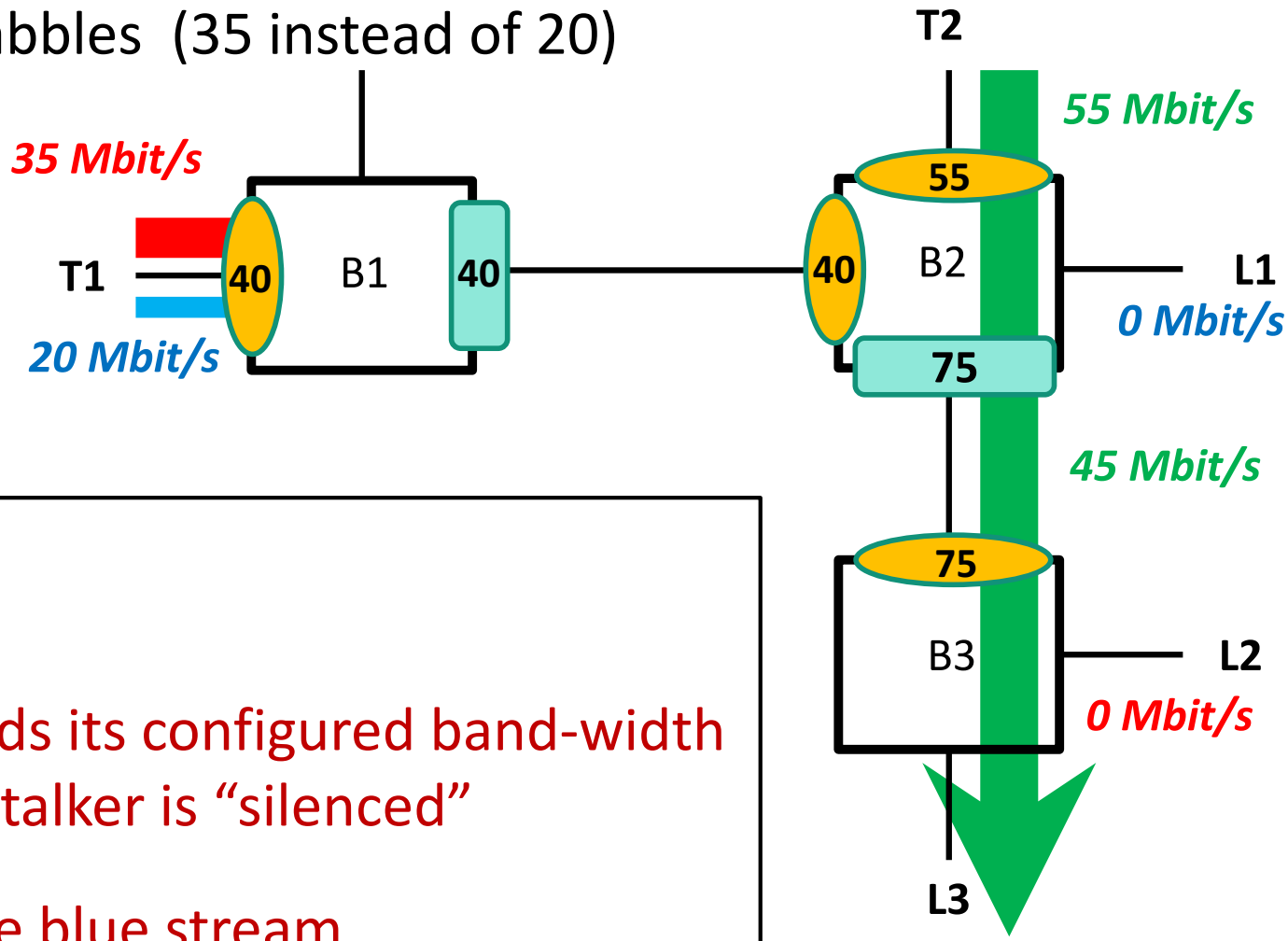
- T1-red:
  A faulty stream sent by a fault talker is silenced.
- T1-blue:
  Non-faulty streams sent by faulty talker are not necessarily silenced.

Four combinations

	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per class + Blocking

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

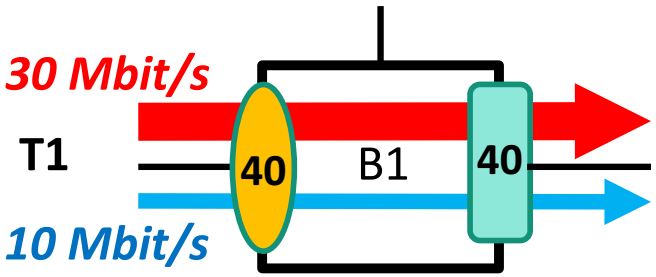


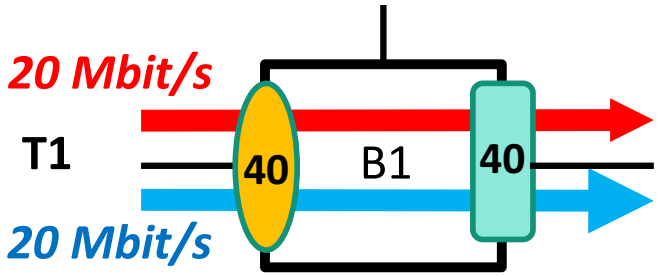
Observations:

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

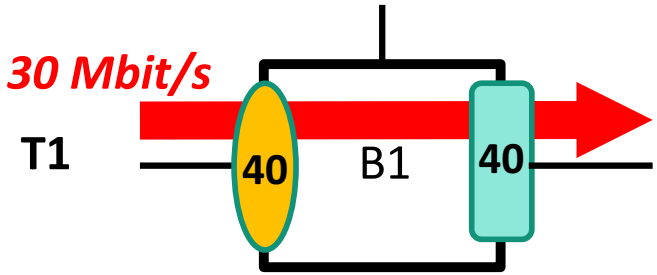
Per class + Blocking: “Moderate babbler”

- **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.





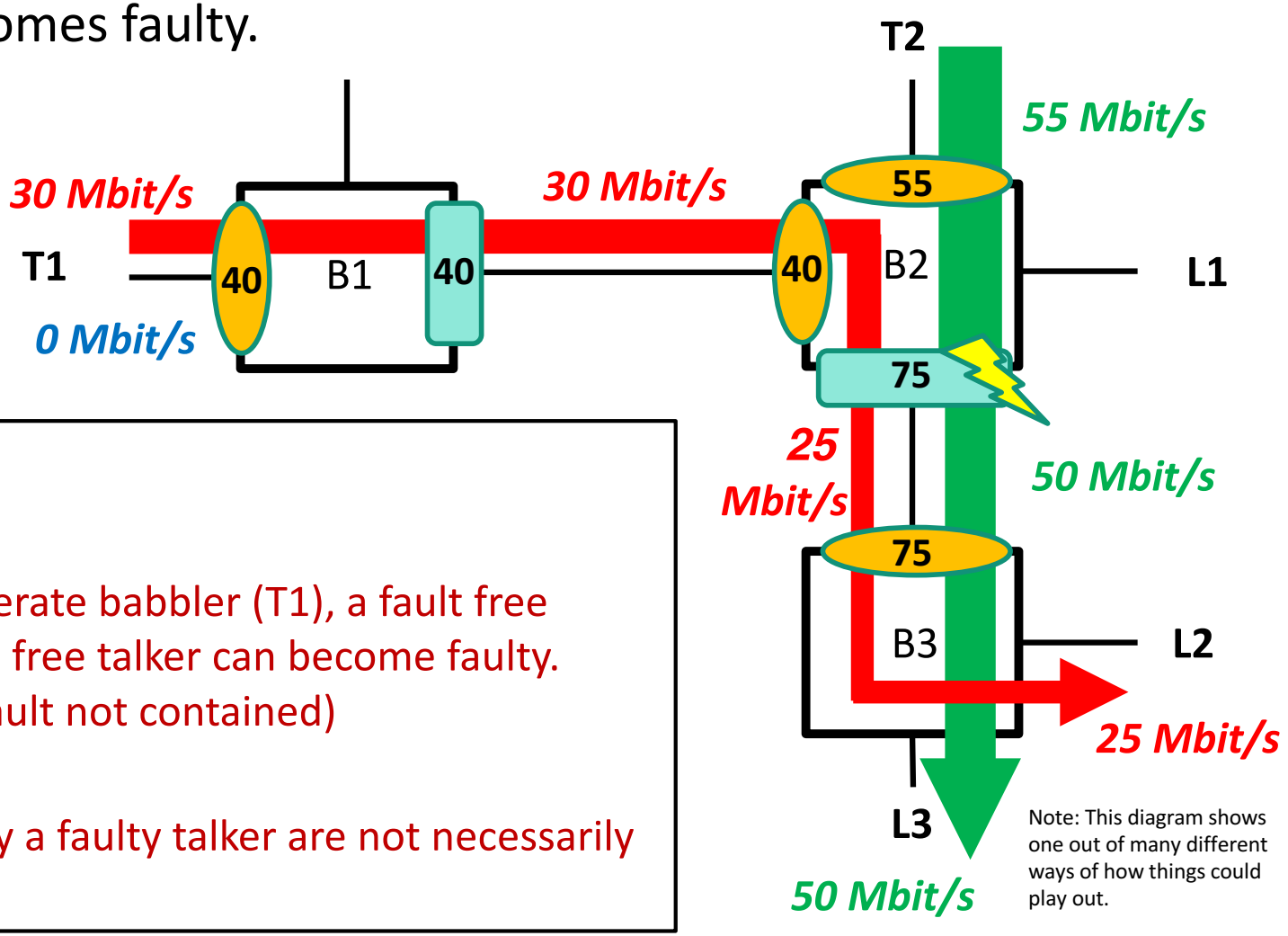
- **More realistic example of a Moderate Babbler:**



  - 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”

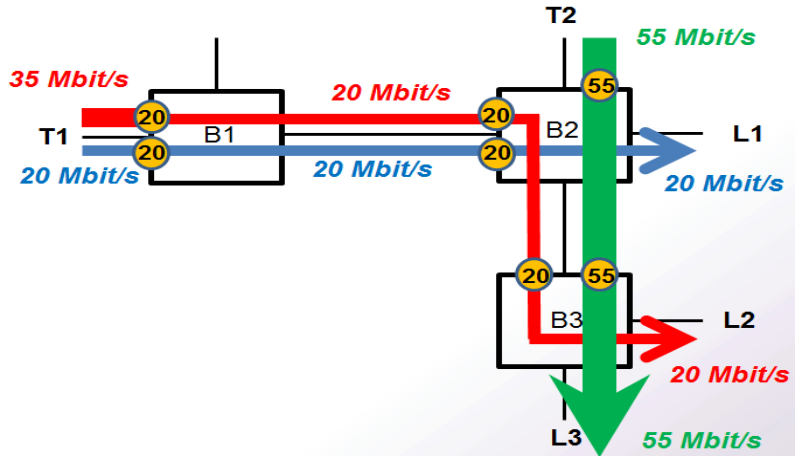
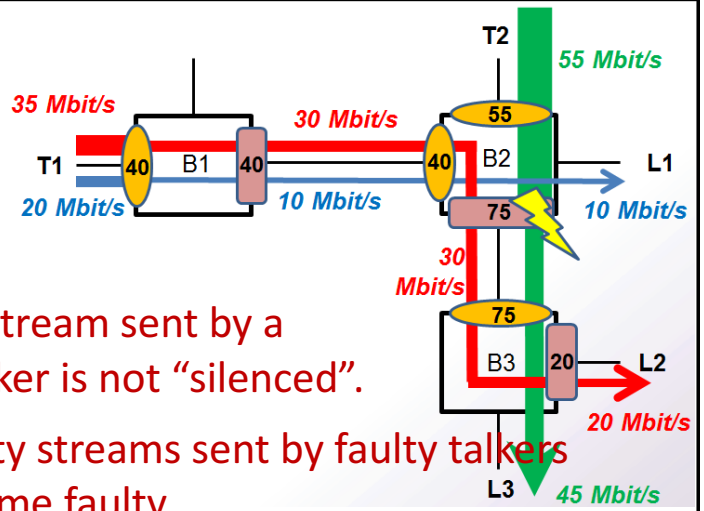
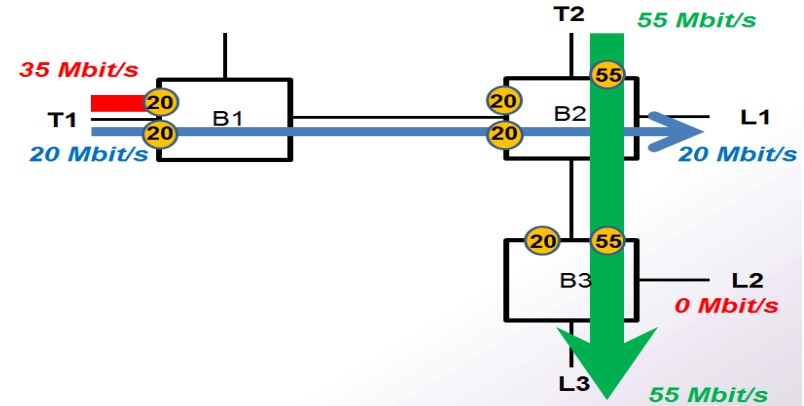
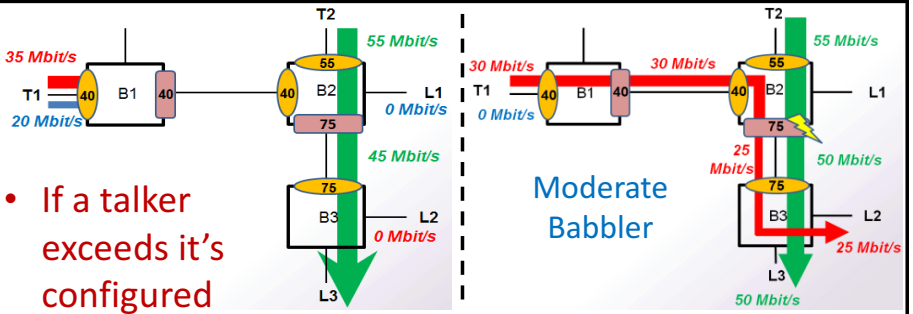
- Moderate Babbler T1: 30 + 0 instead of 20+20
- Shaper at B2 drops frames
  => T2-green becomes faulty.



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.

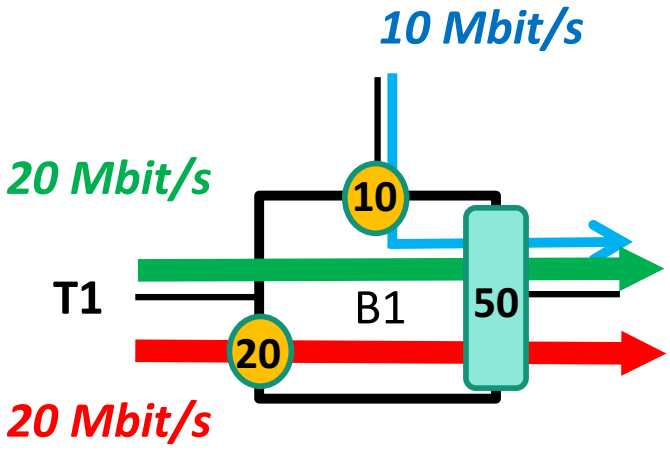
Comparison

	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.

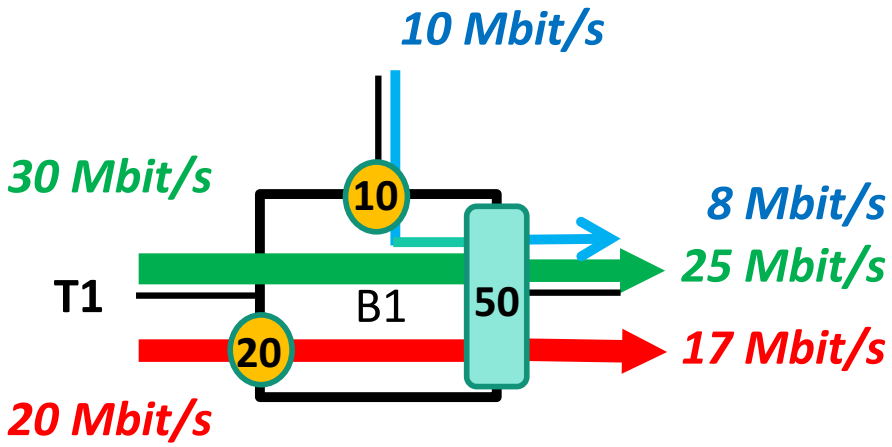
How many filters for per-stream policing?

- 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

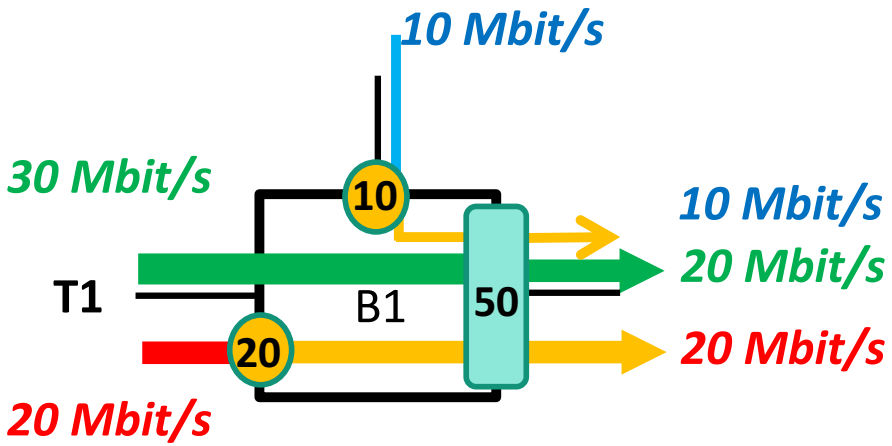
(T1-green sends 30 instead of 20)

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 send 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.

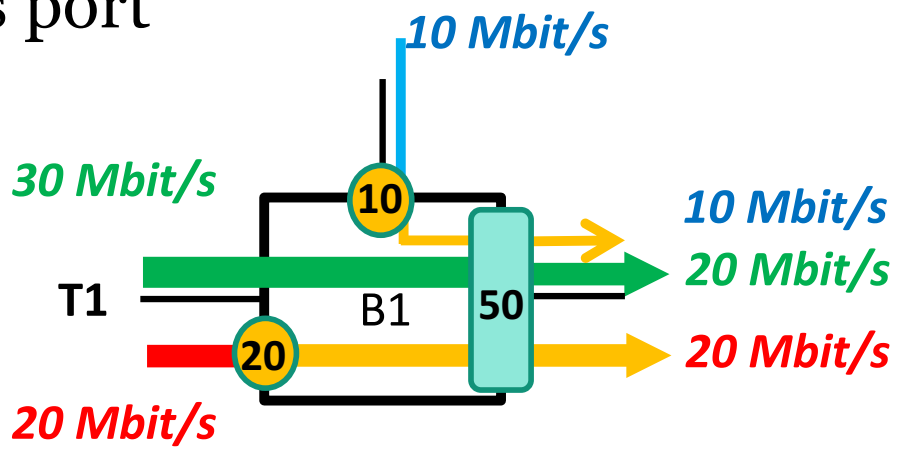


Faulty case

(T1-green sends 30 instead of 20)

Less than one IPF per stream

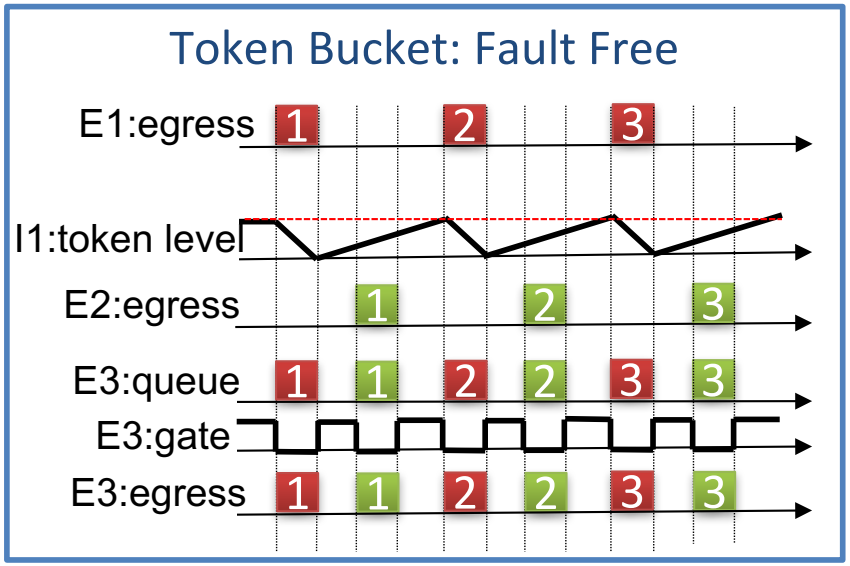
- We can thus reduce the number of IPFs to the anticipated maximum number of safety-critical streams per port
- … without imposing any limitation on total number of streams
- Requires changes in egress port

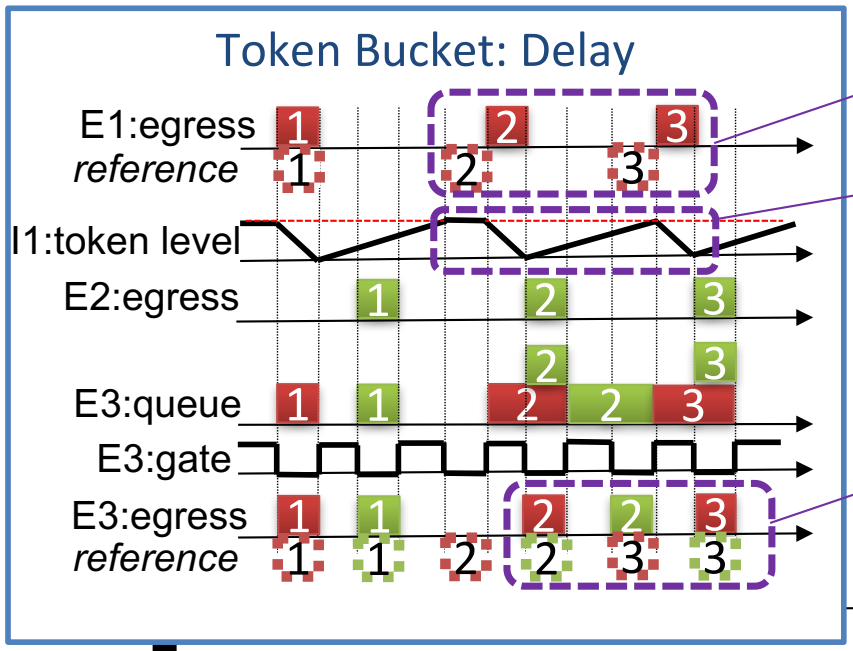


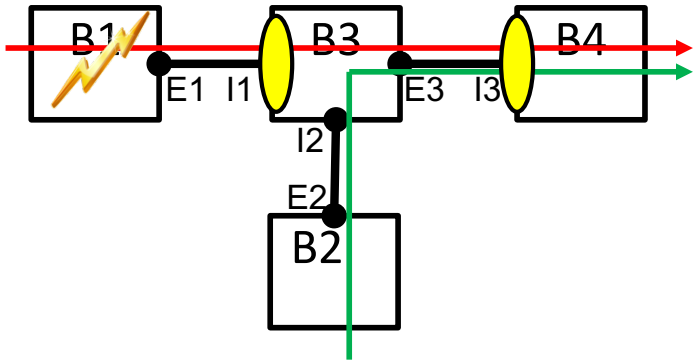
Faulty case

(T1-green sends 30 instead of 20)

Token bucket alone does not work for 802.1Qbv







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

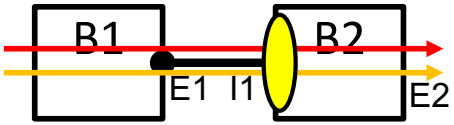
Ingress Windows

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

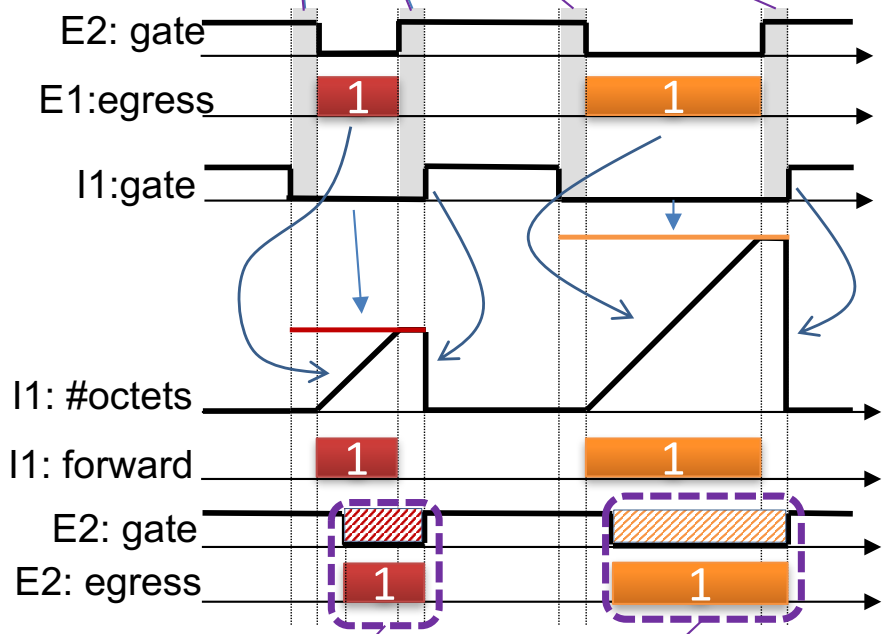
- Open: Accept consecutive started packets until next ingress close
- Close: Discard consecutive started packets entirely

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

Fault-free case

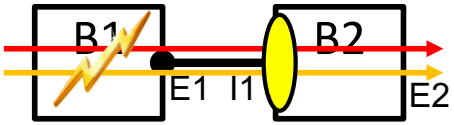


Variances (PTP, 802.1Qbv, …)



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

Faults covered by ingress window

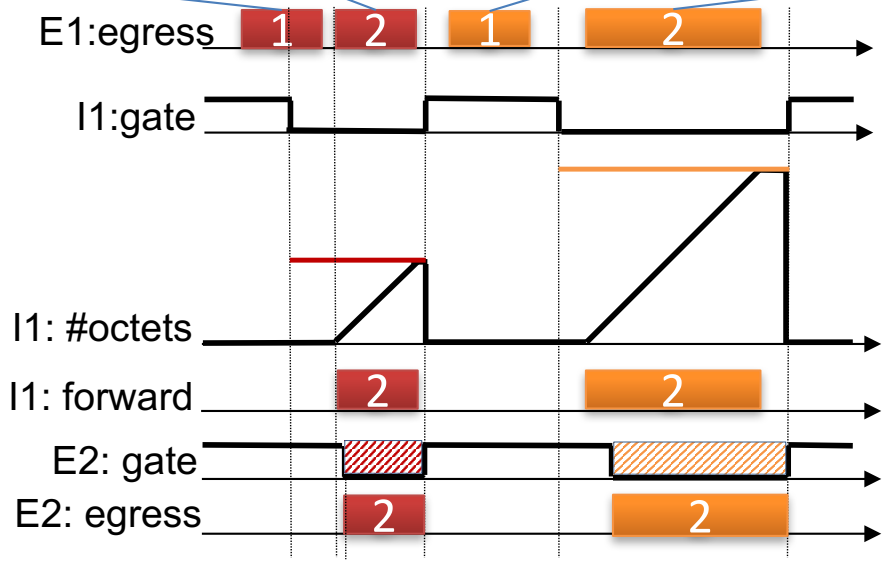


Starts before ingress window → Entirely discarded

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

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

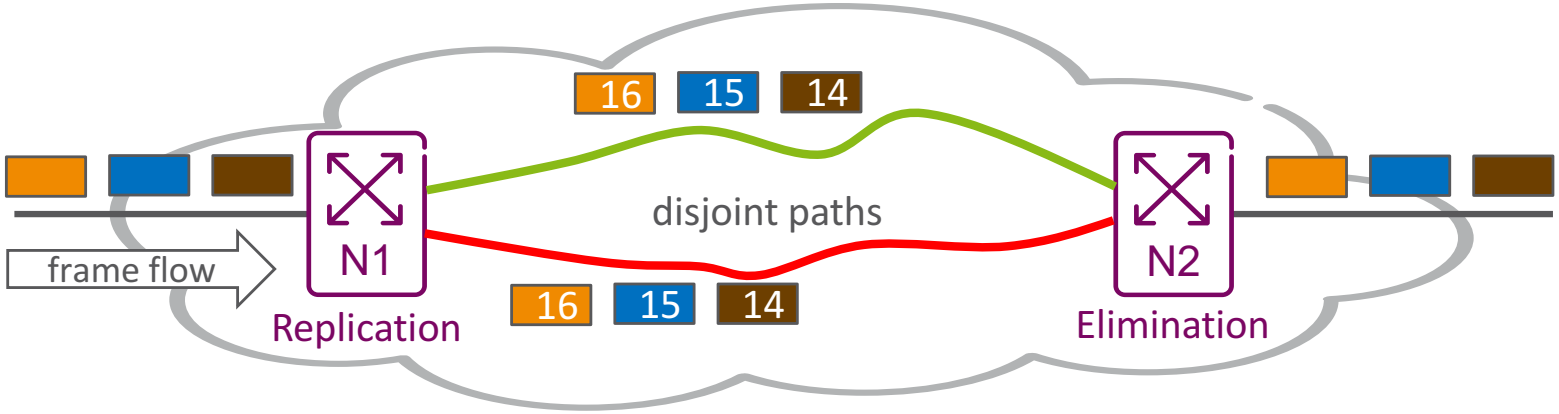
- Industrial automation systems already implemented redundancy on top (proprietary) Ethernet
  - PRP: Parallel Redundancy Protocol
  - HSR: High-availability Seamless Redundancy
- Need to standardize in 802.1
  - Industrial automation (converging towards IEEE 802 standardized Ethernet networks)
  - Professional 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



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

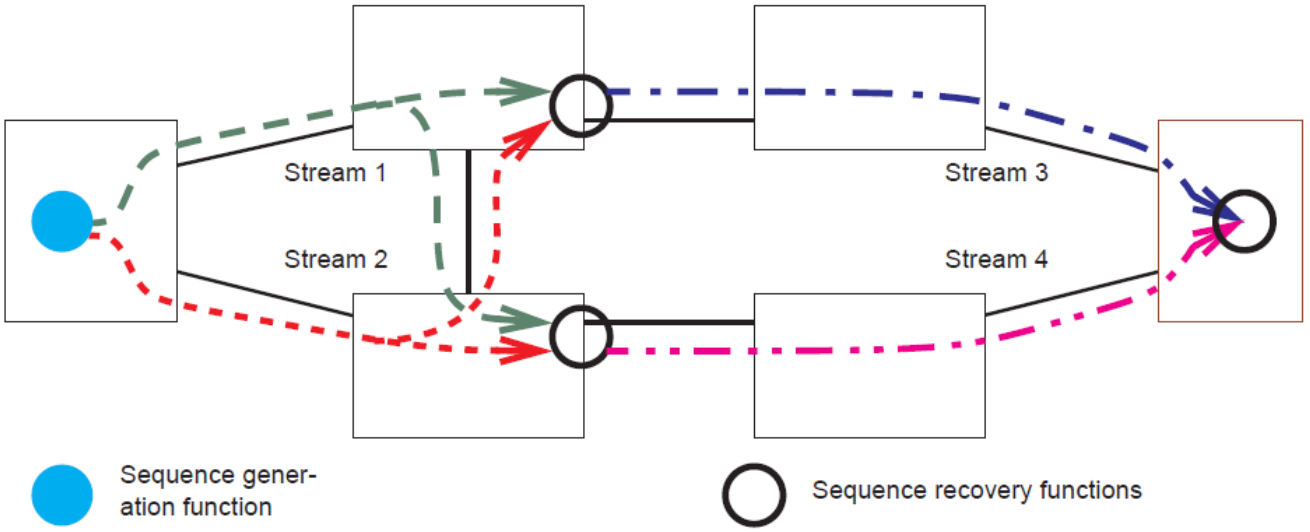
Field	Offset	Length
Destination MAC address	0	6
Source MAC address	6	6
C-tag EtherType	12	2
Priority, DE, VLAN ID	14	2
FRER Ethertype	16	2
sequence number	18	2
Payload Length/EtherType	20	2
data	22	*n*
Frame Check Sequence	22+*n*	4

Example Ethernet frame format with embedded R-Tag

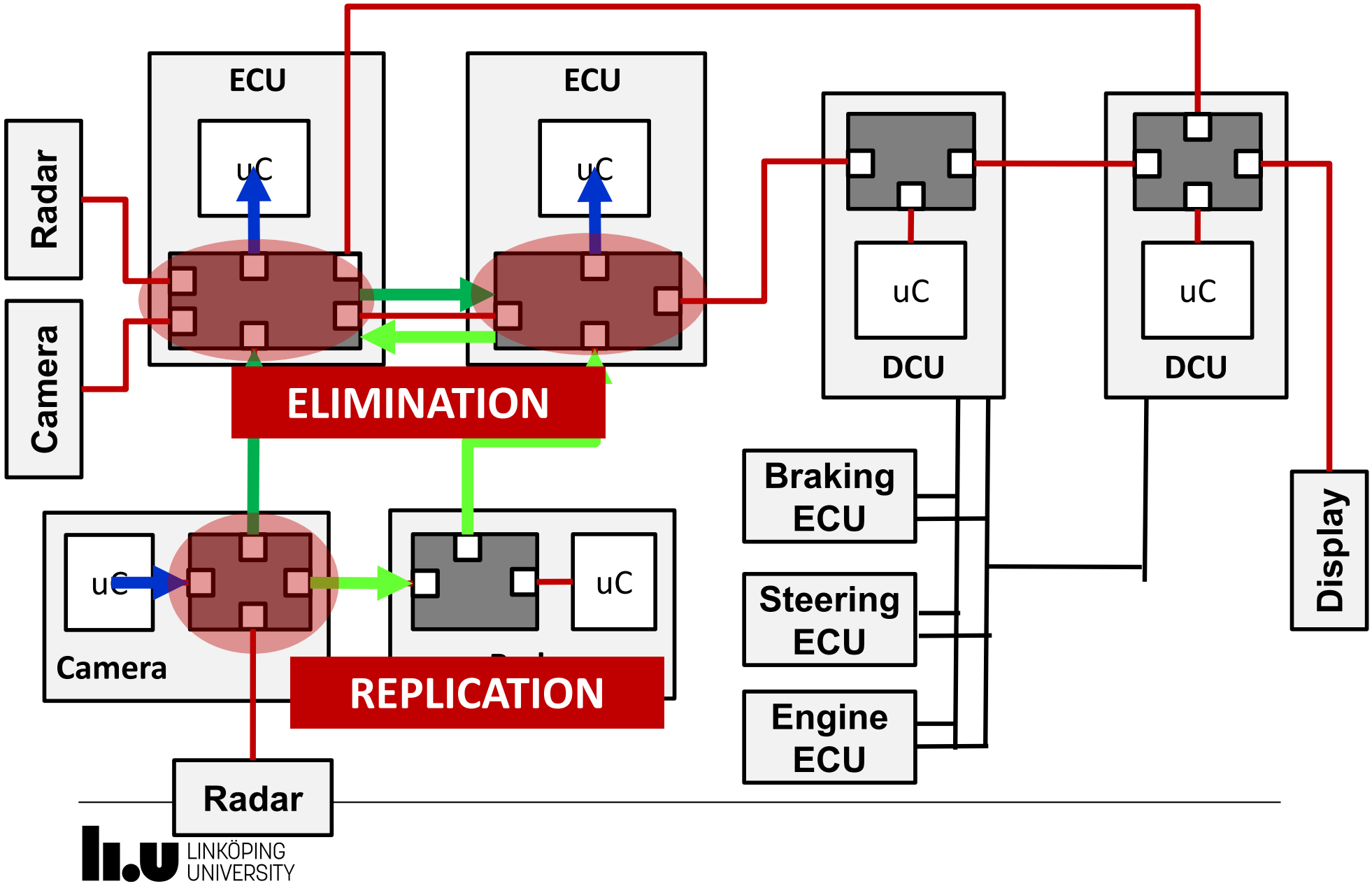
802.1 operation

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

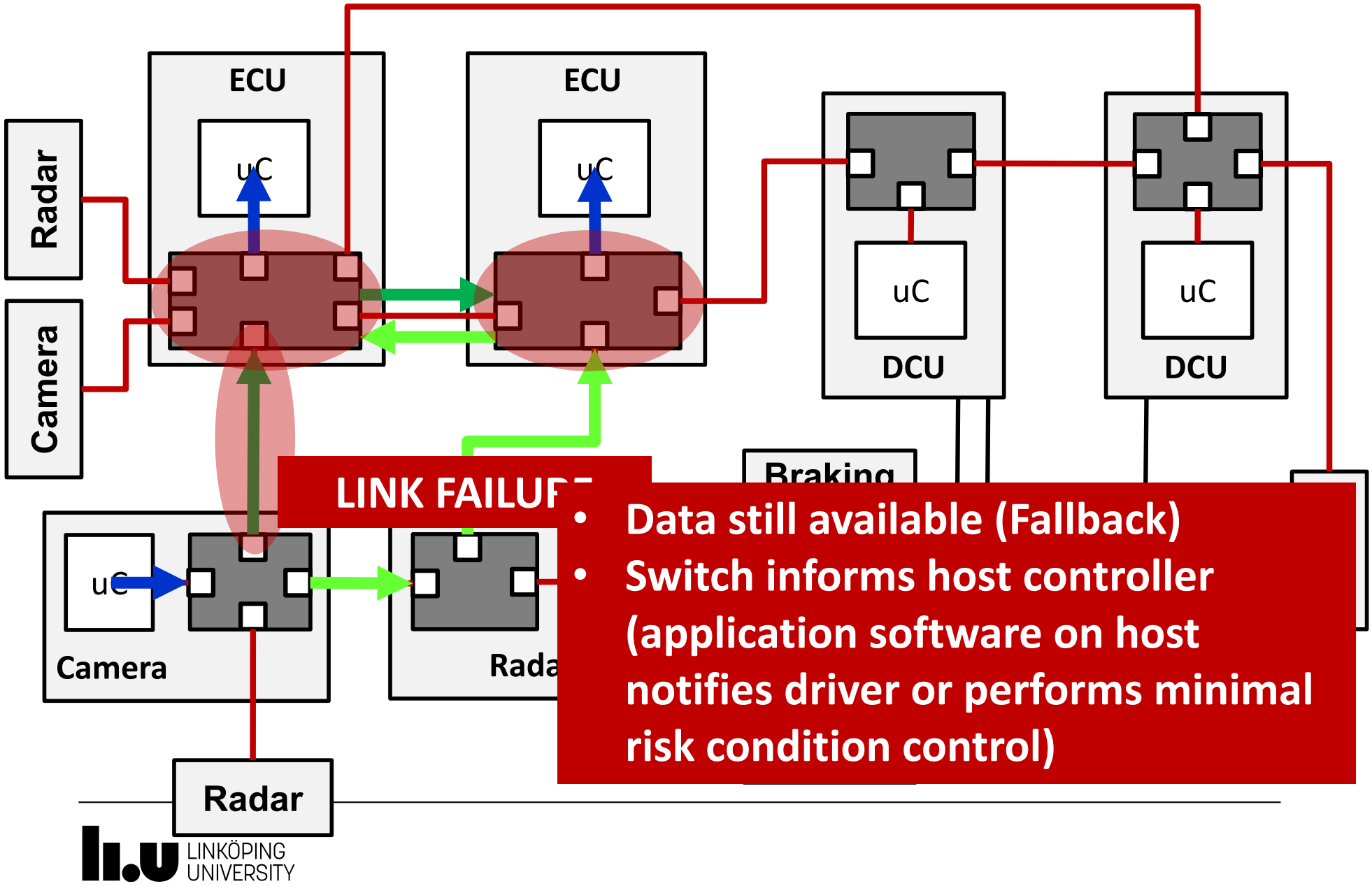
(This configuration protects against all 7 possible one-link failures, and against 16 of 21 possible two-link failures)



802.1CB with proxy mode



802.1CB and Failures



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