Ingress Policing

IEEE 802.1 TSN Plenary

November 10-15, 2013 – Dallas, USA

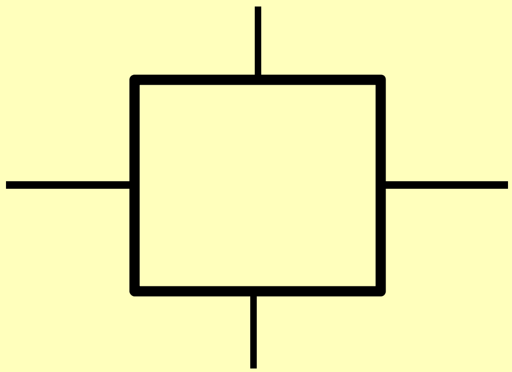
Markus Jochim

General Motors Research & Development

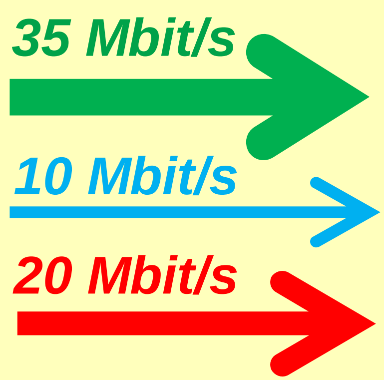
Symbols and Abbreviations

- IPF = Ingress Policing Filter
- Talker: T1, T2,...   Listener: L1, L2,...

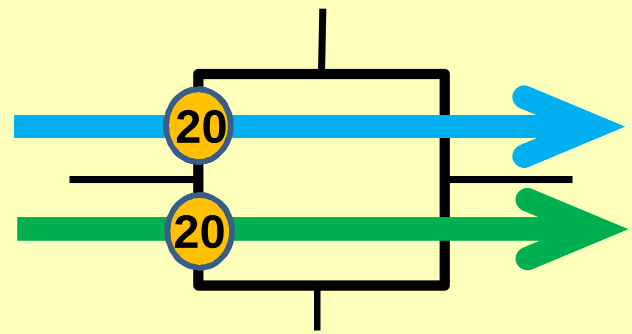
4 Port Bridge



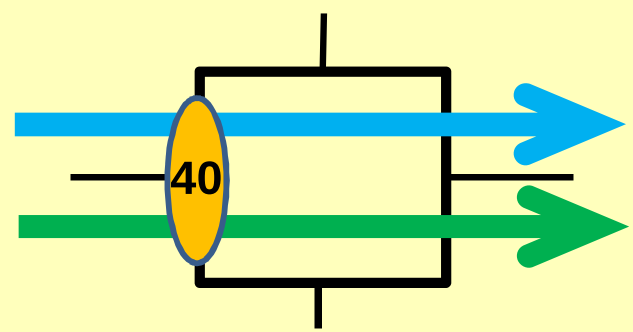
Class A Streams



Ingress Policing Filter

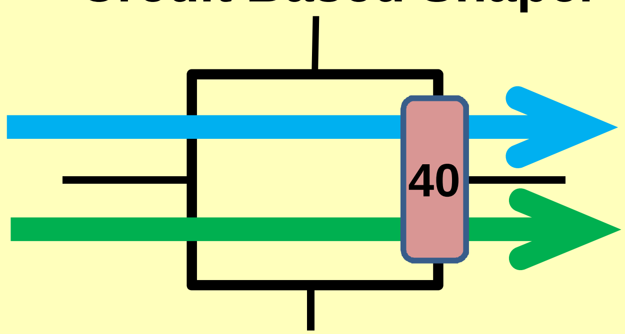


Two 20 Mbit/s Per Stream IPFs



40 Mbit/s Per Class IPF

Credit Based Shaper



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

Babbling Idiot Problem

- A faulty talker or switch (= Babbling Idiot)
  - sends too much traffic or
  - sends at the “wrong time”

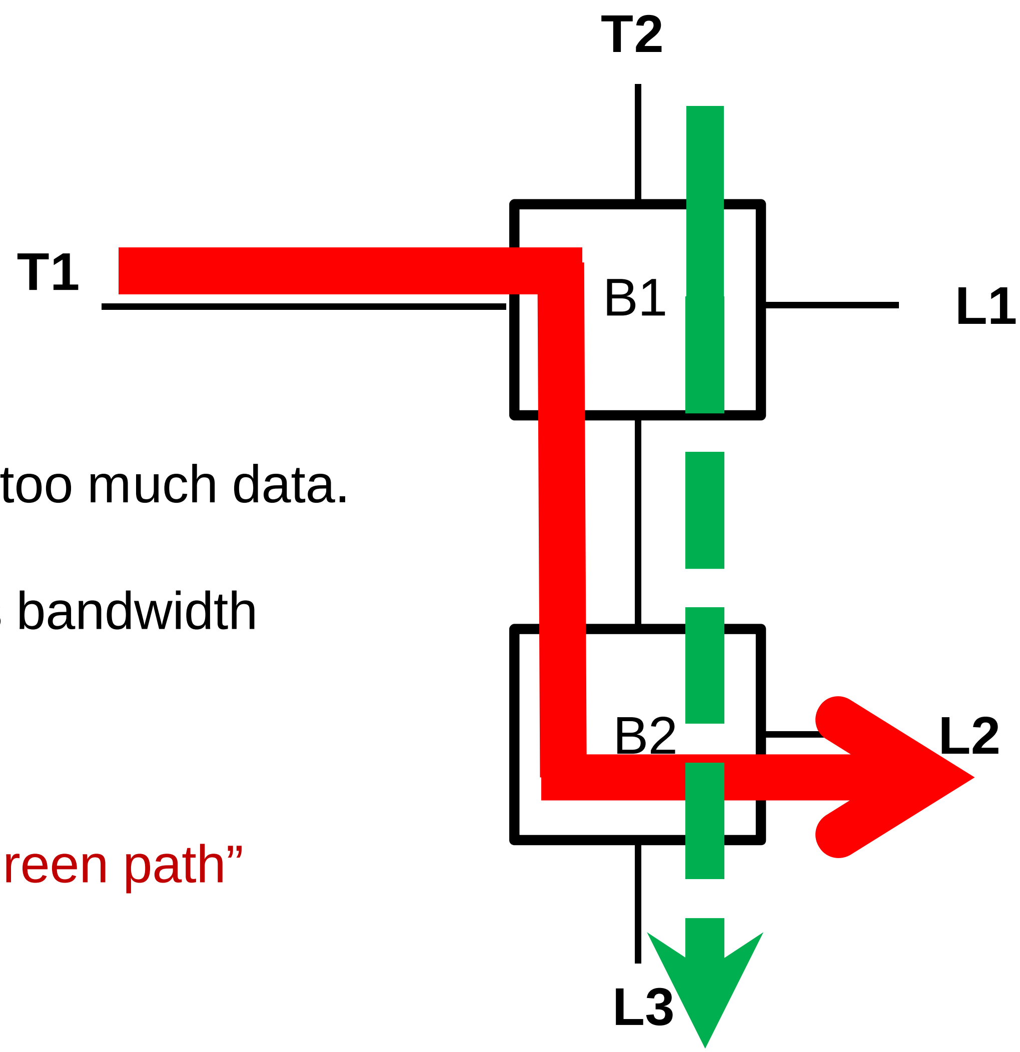
  and takes away bandwidth from other streams.

- Bandwidth and latency guarantees of these “other streams” can no longer be guaranteed.

  $\Rightarrow$ They become faulty !

- The babbling idiot can affect many streams in a network! The fault effect propagates through the network.

Babbling Idiot Problem



Example:

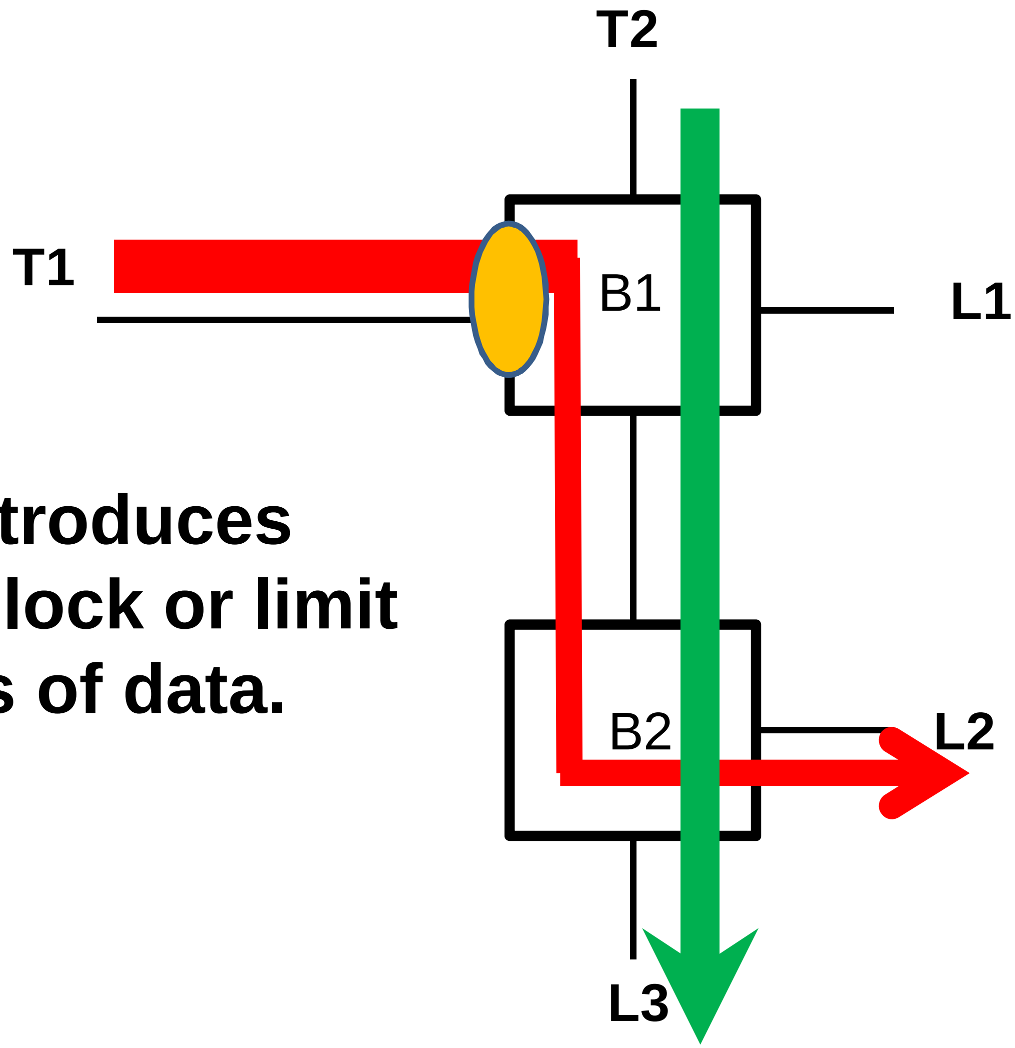
Babbling Idiot: T1
Faulty red stream sends too much data.

Green stream violates its bandwidth and latency guarantees.

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

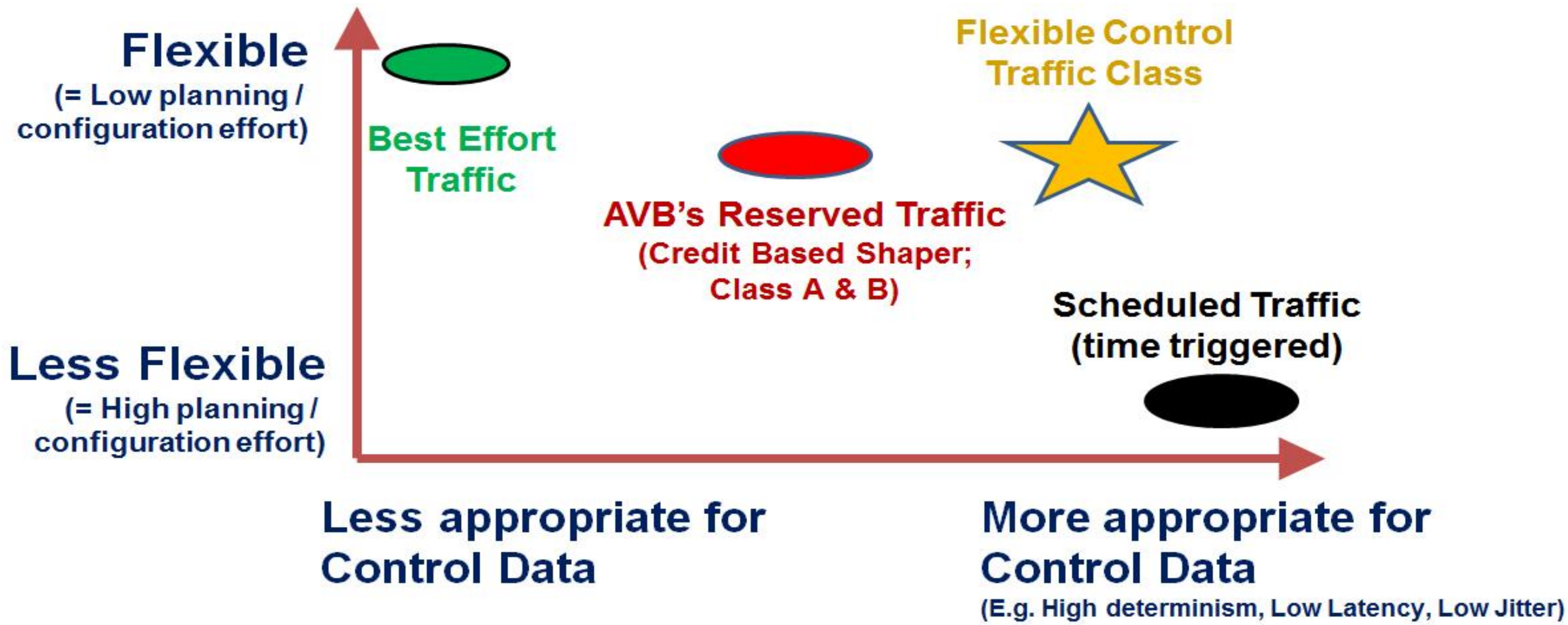
But:
Green stream is faulty.

Ingress Policing in a Nutshell



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

Ingress Policing for which Traffic Class ?



- For Reserved Traffic
- For the new Flexible Control Traffic Class (*1)
- For Scheduled Traffic (*2)
- Not for Best Effort (*3)

The examples given in this presentation are focused on Reserved Traffic. This does not imply that the concepts are limited to this traffic class.

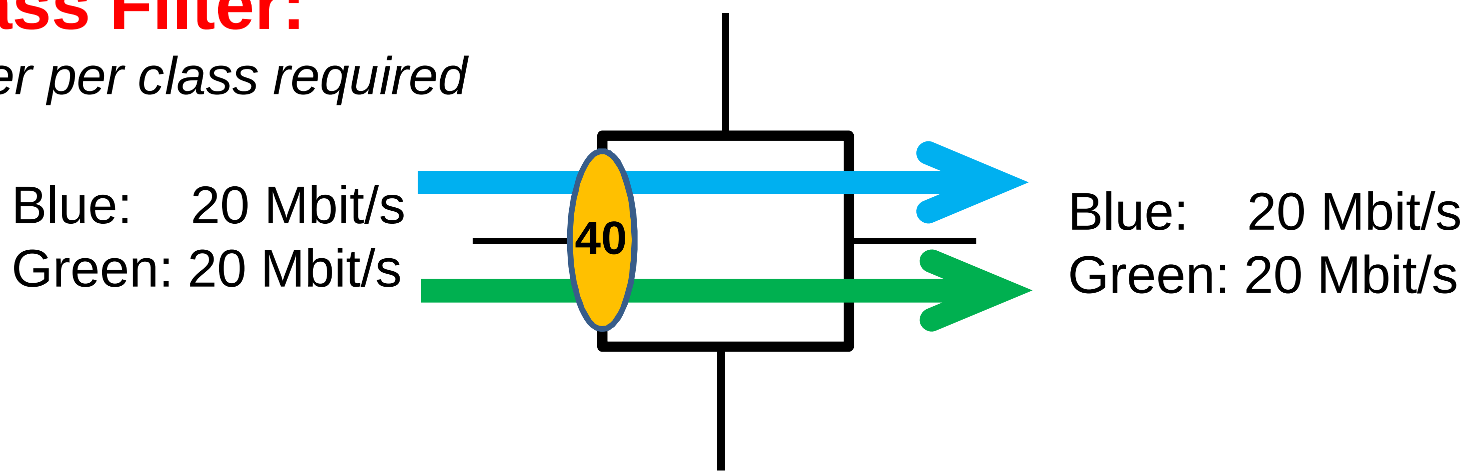
(*1): Peristaltic Shaper or Urgency Based Scheduler or Burst Limiting Shaper or whatever else the 802.1TSN group may decide to pick. (An ongoing discussion...)
(*2): Concepts presented in this slide deck are not primarily focused on Scheduled Traffic and may need to be adjusted for this traffic class.
(*3): There is no traffic contract for BE and we don't know how much traffic is legitimate. Excessive BE traffic will drop on Egress and will not have an impact on other traffic classes.

Ingress Policing: Alternative Options

- **Criteria 1:**
  What is a filter “observing” or “counting”

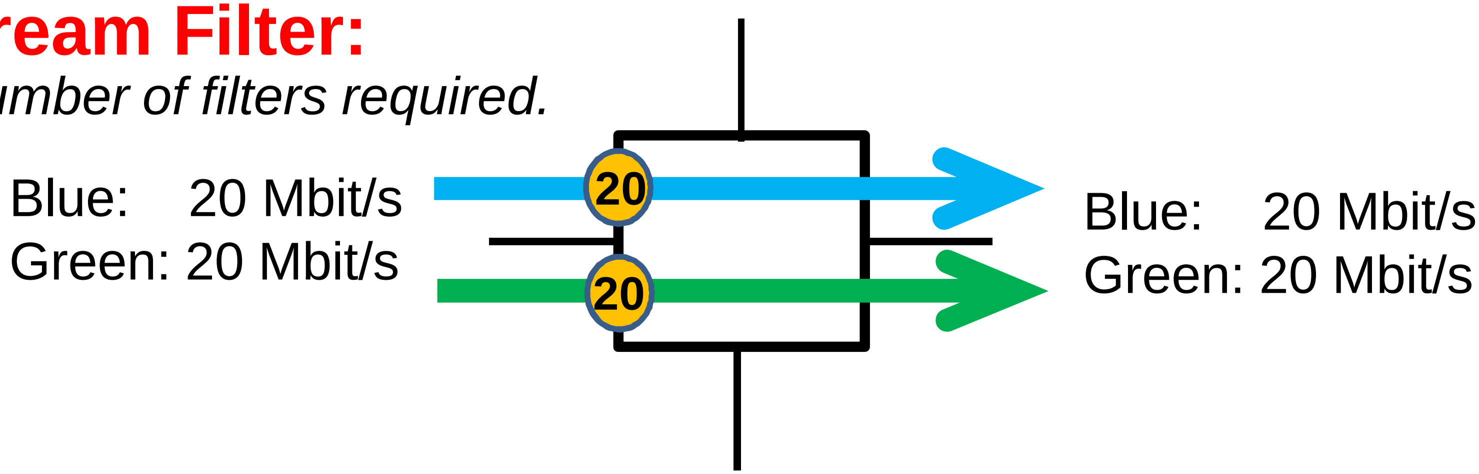
Per Class Filter:

Only 1 filter per class required



Per Stream Filter:

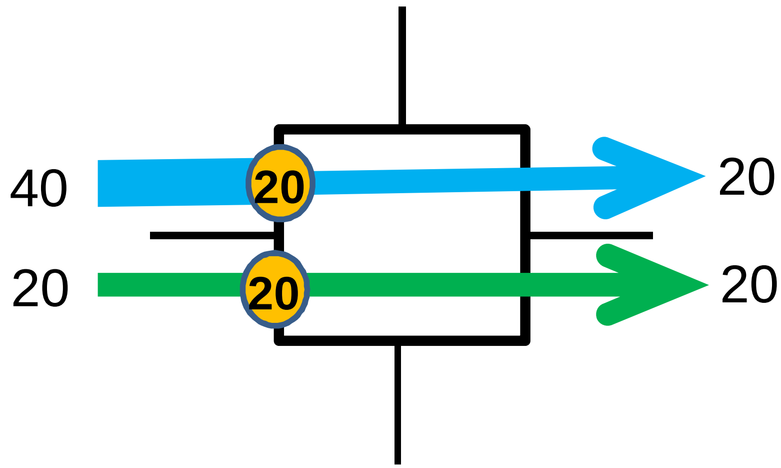
Higher number of filters required.



Ingress Policing: Alternative Options

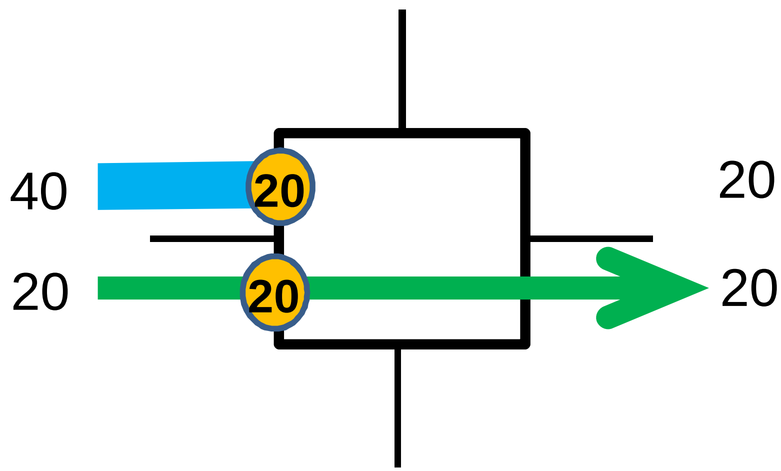
- **Criteria 2:**
  What is the response if the threshold is exceeded

Threshold Enforcing IPF:



Blocking IPF:

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



The diagrams on this slide show "Threshold enforcing" and "Blocking" on a per stream basis

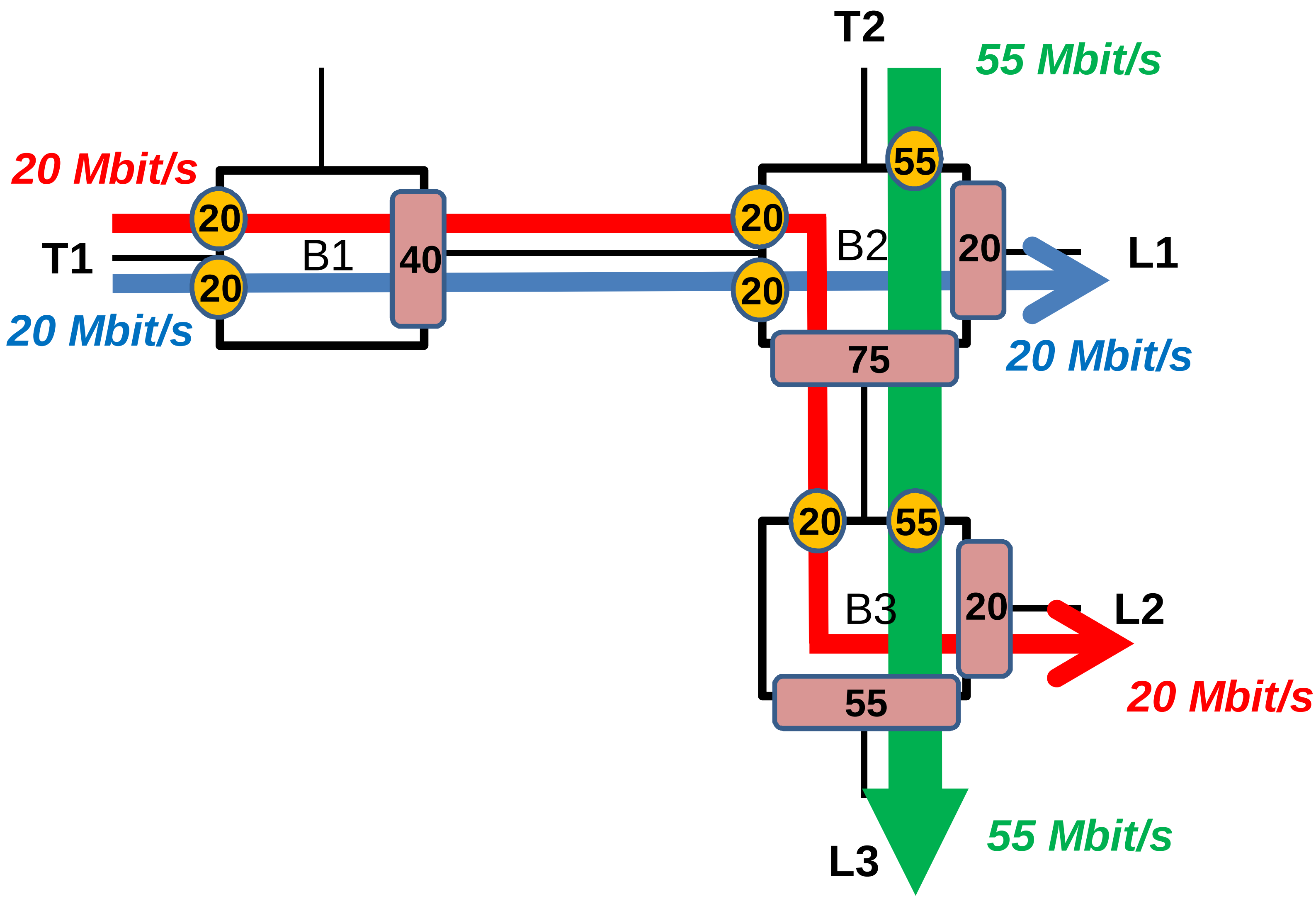
Four Combinations

	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

- Let’s go through some of the combinations to understand some of the trade-offs.
- Let’s then discuss which trade-offs are acceptable / unacceptable

Example: Fault free case

- Assumption: Only AVB Class A Traffic
- Streams **T1-red**, **T1-blue**, **T2-green**

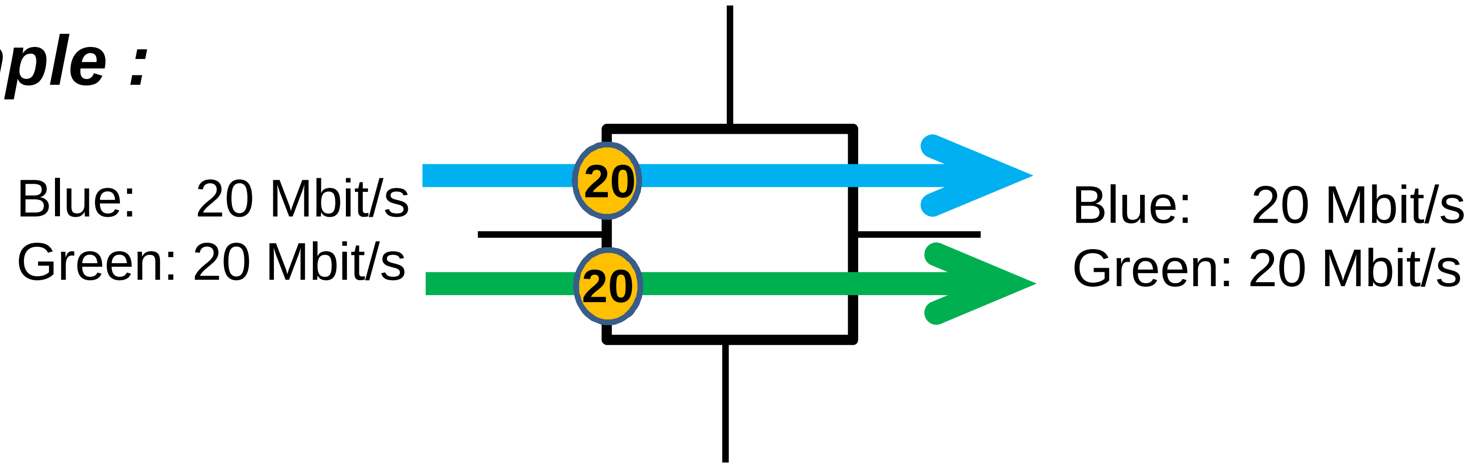


Four Combinations

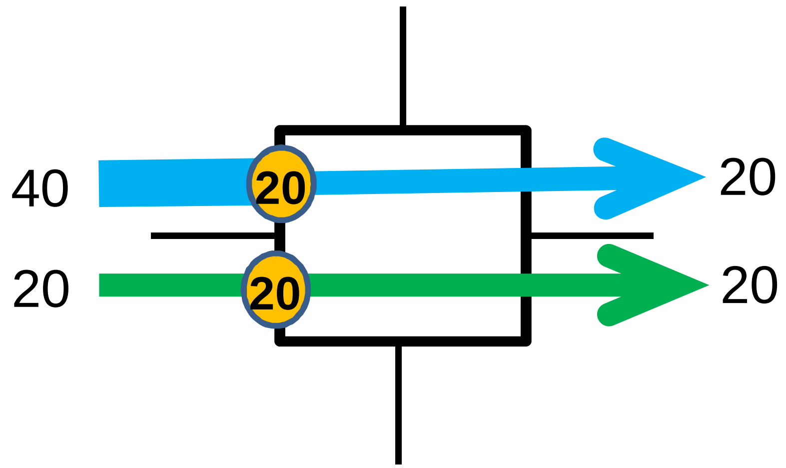
	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per-Stream X Threshold Enforcing Filter

Example :

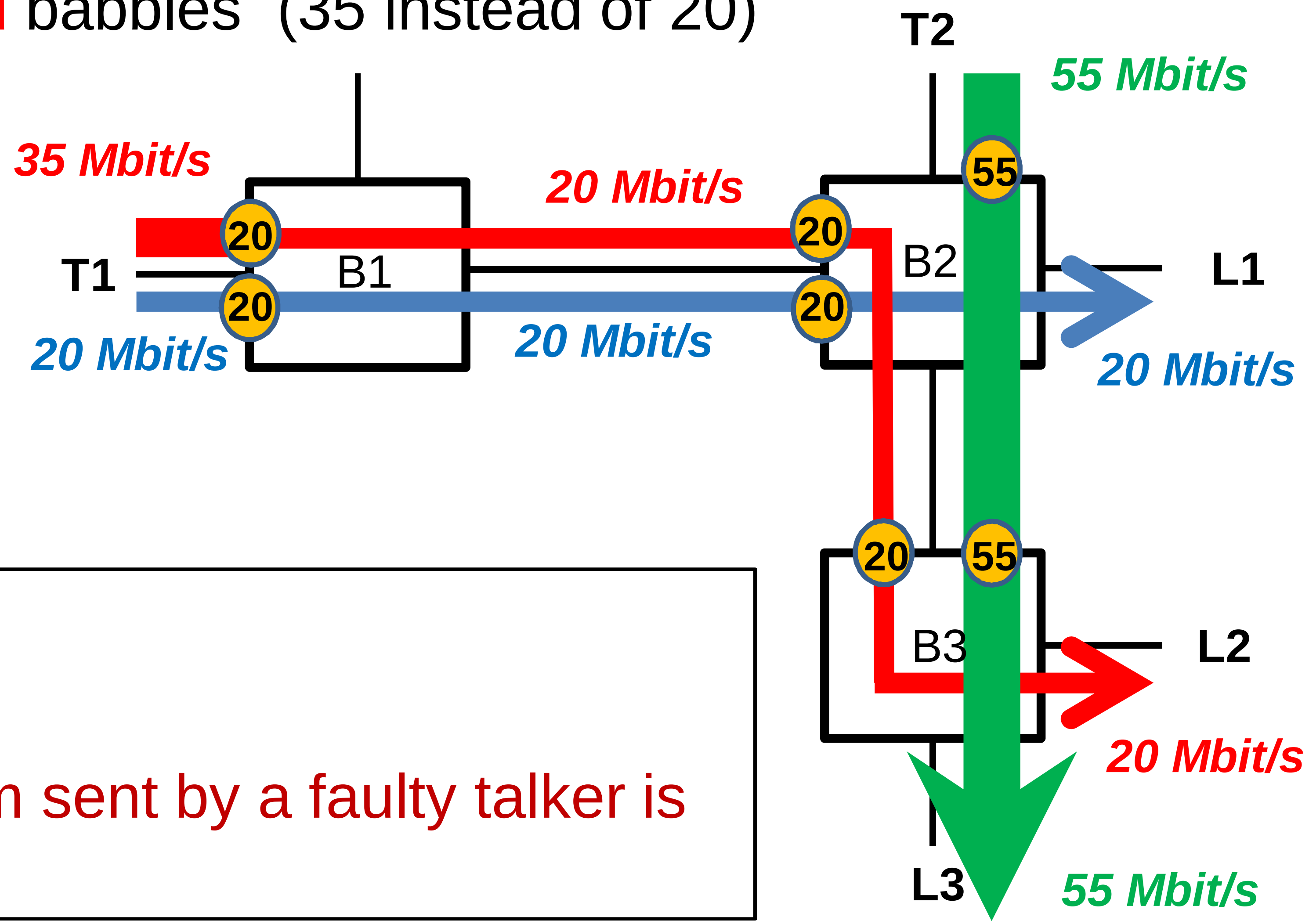


Fault: Blue stream babbles (40 Mbit/s instead of 20 Mbit/s)



Per Stream X Threshold Enforcing

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



Observations:

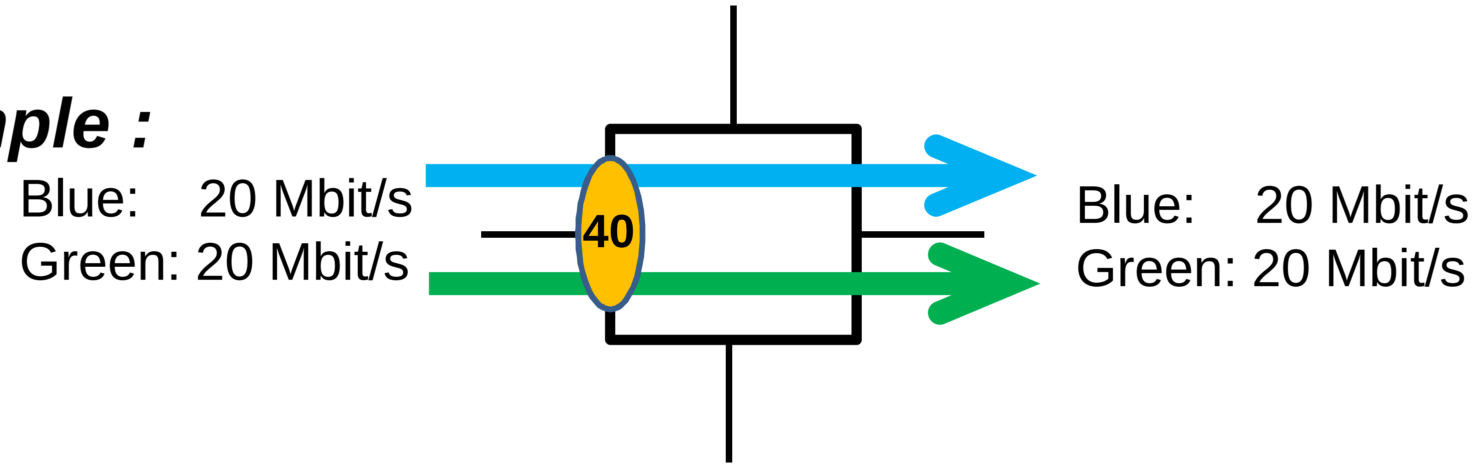
- T1-red:
  A faulty stream sent by a faulty talker is not "silenced".

Four Combinations

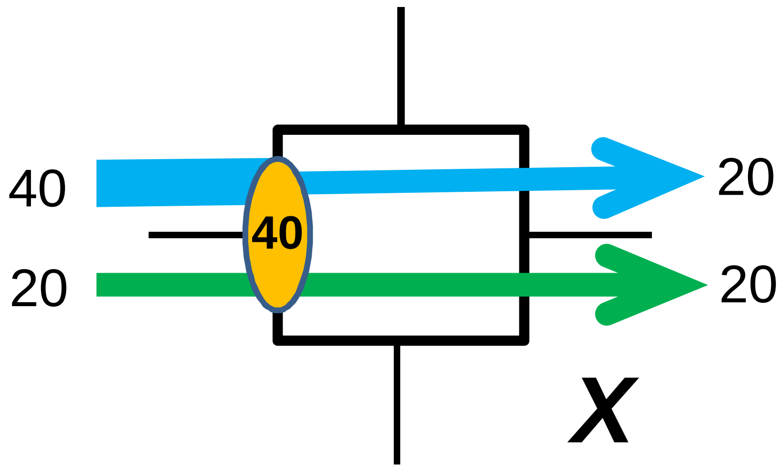
	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

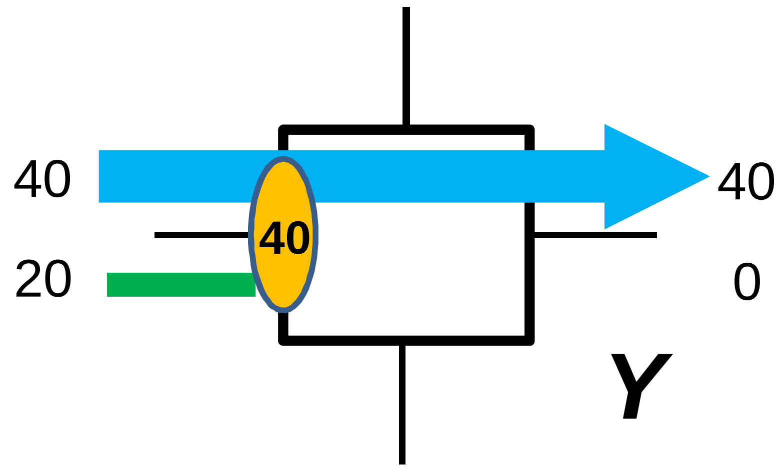
Per-Class X Threshold Enforcing Filter

Example :



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



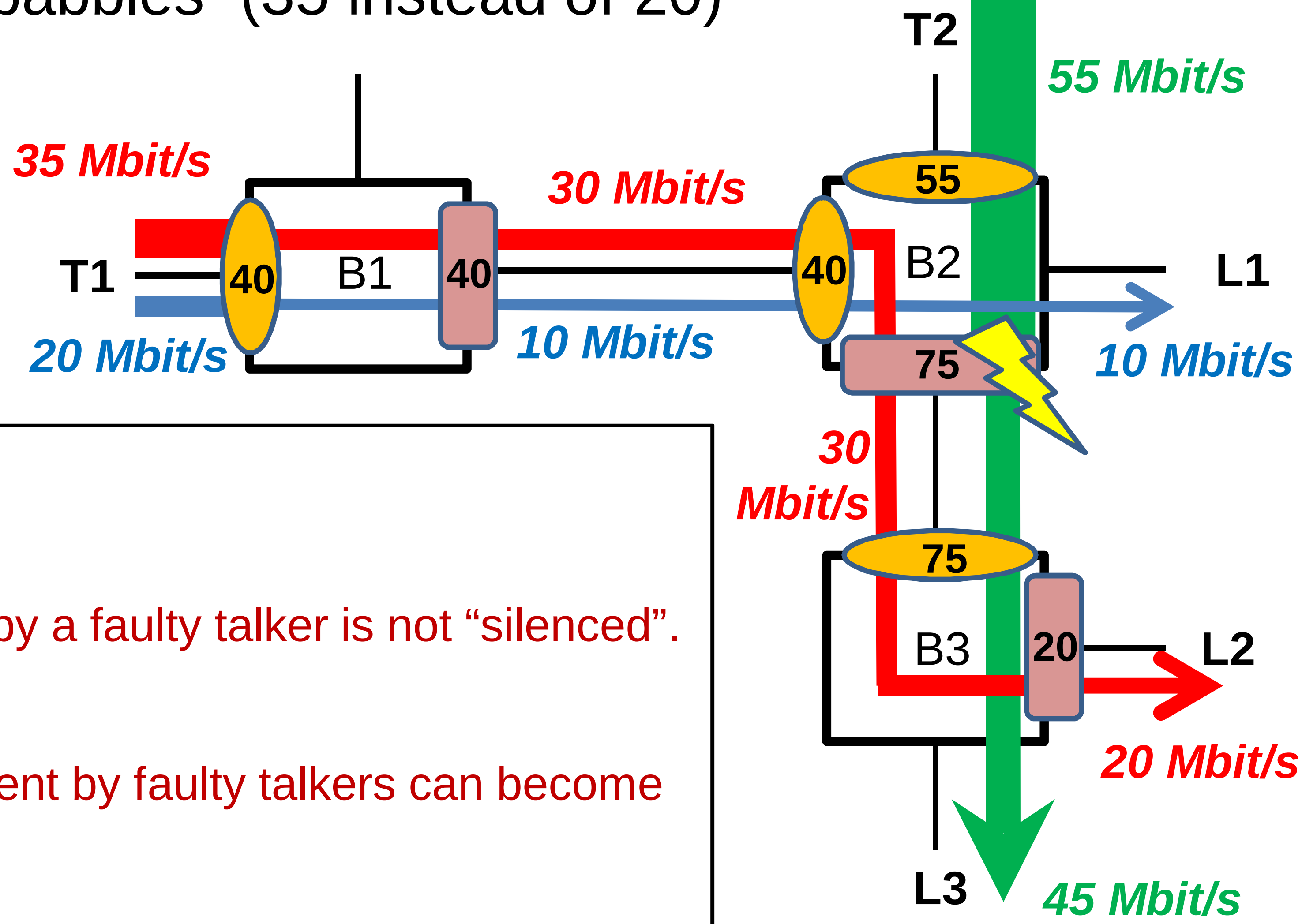


S1: All kinds of behavior (*X* or *Y* or anything in between) are possible!

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 X 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)

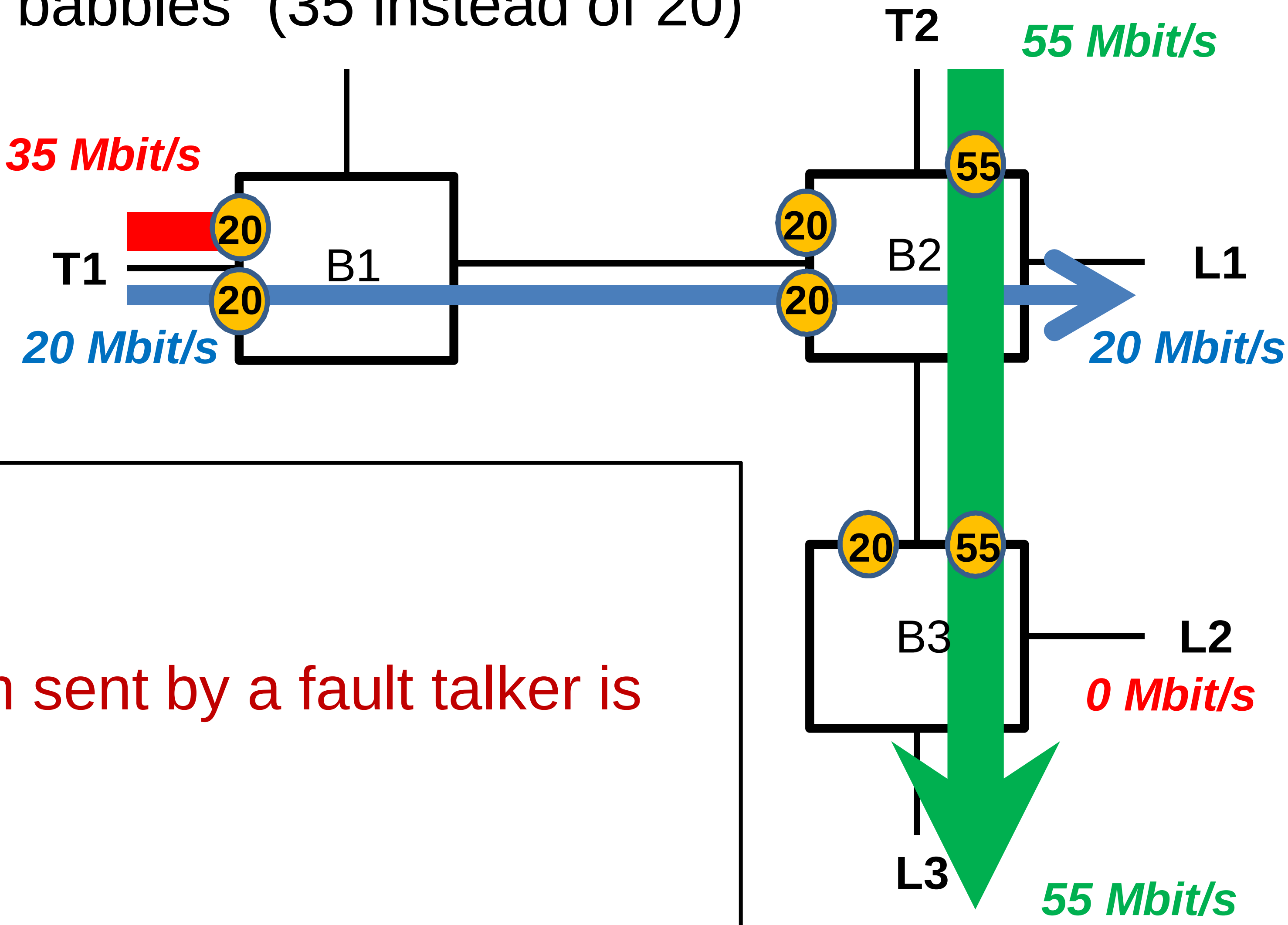
Note: This diagram shows one out of many different ways of how things could play out.(See statement S1 on previous slide)

Four Combinations

	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per Stream X Blocking

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



Observations:

- T1-red:
  A faulty stream sent by a fault talker is silenced.

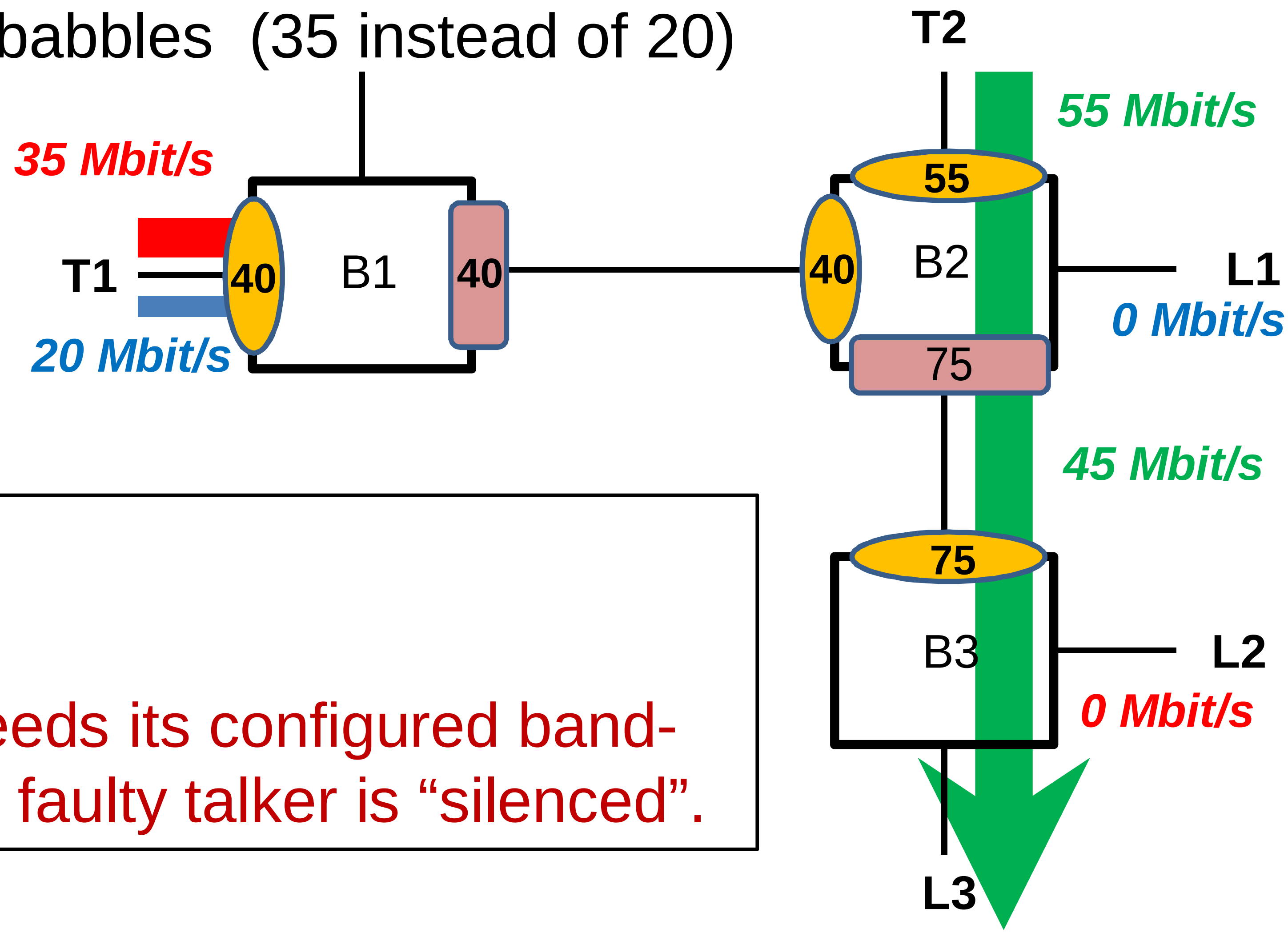
- T1-blue:
  Non-faulty streams sent by faulty talker are not necessarily silenced.

Four Combinations

	Per Stream	Per Class
Threshold Enforcing	1	2
Blocking	3	4

Per Class X Blocking

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



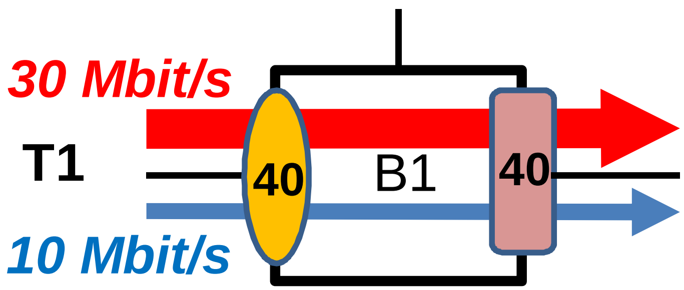
Observations:

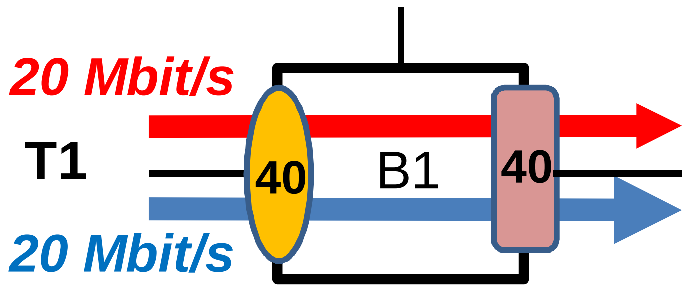
- T1:
  If a talker exceeds its configured band-width limit, the faulty talker is "silenced".

Note: If T1 starts babbling at time t0, the IPF filter shown in B1 will trigger at time t0+Δt_IPF, where Δt_IPF is the time it takes until the filter starts responding to the fault (= filter latency). At a first glance one might come to the conclusion that during [t0, t0+Δt_IPF] the IPF of B2 will also be exposed to 35 Mbit + 20 Mbit/s and will eventually trigger in parallel to the IPF of B1. Similarly one might assume that the IPF on B3 will also eventually trigger, since it will be exposed to 35 + 55 Mbit/s during [t0, t0+ Δt_IPF]. This, however, is a wrong conclusion, since the per class credit based shaper on the egress port of the fault free bridge B1 ("fault free" since we assume only one faulty component) will enforce a maximum of 20 + 20 Mbit/s to flow into the direction of B2 and B3.

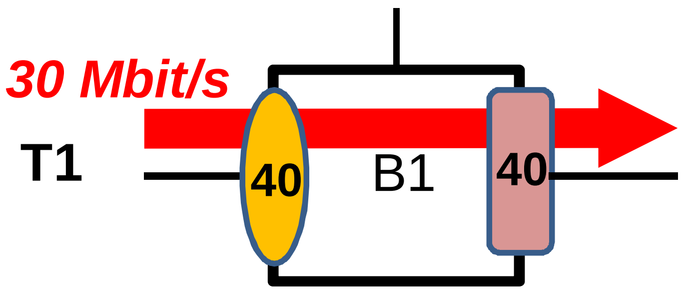
Per Class X Blocking: Moderate Babbler

- **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.





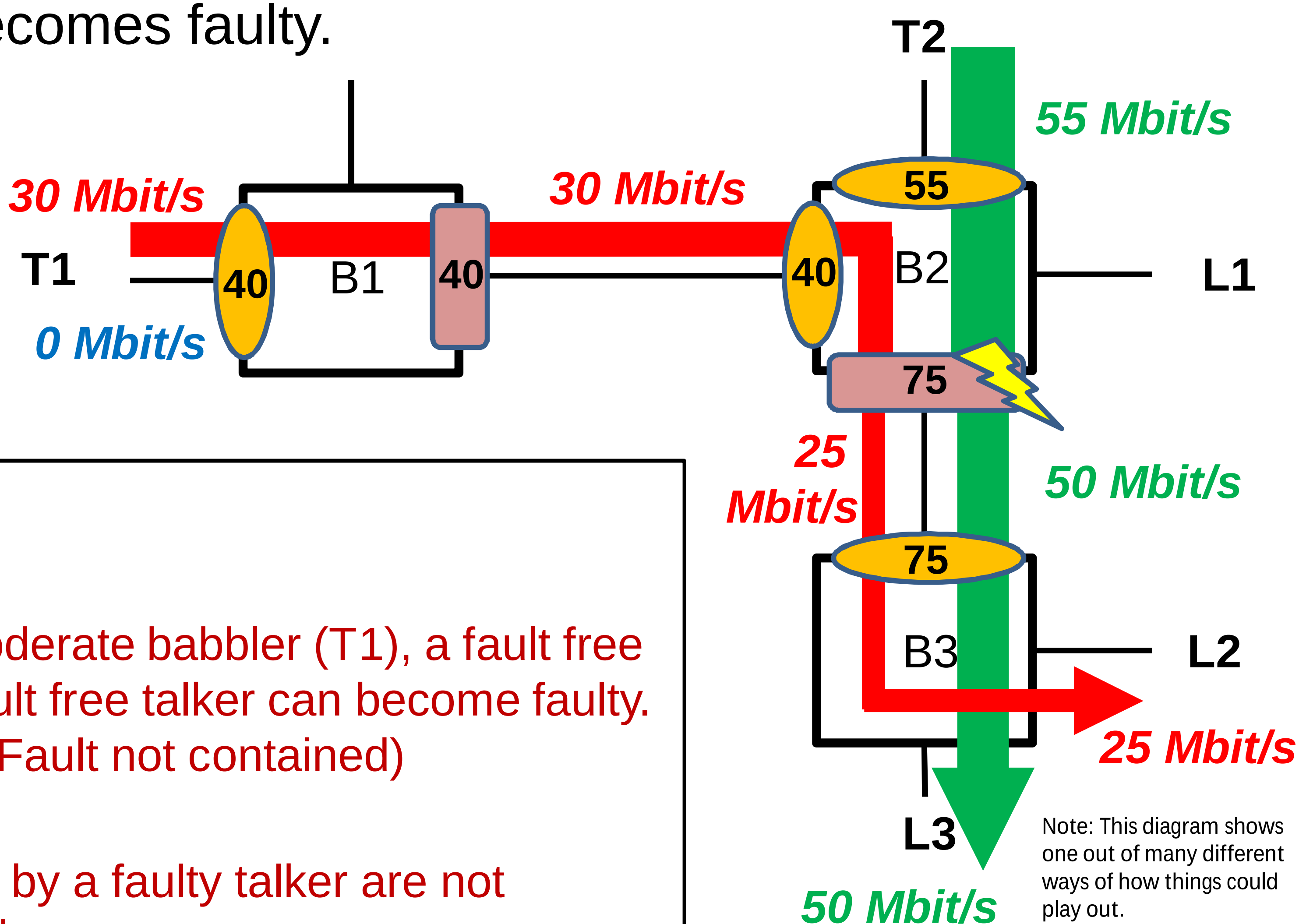
- **More realistic example of a Moderate Babbler:**



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

Per Class X Blocking: Moderate Babbler

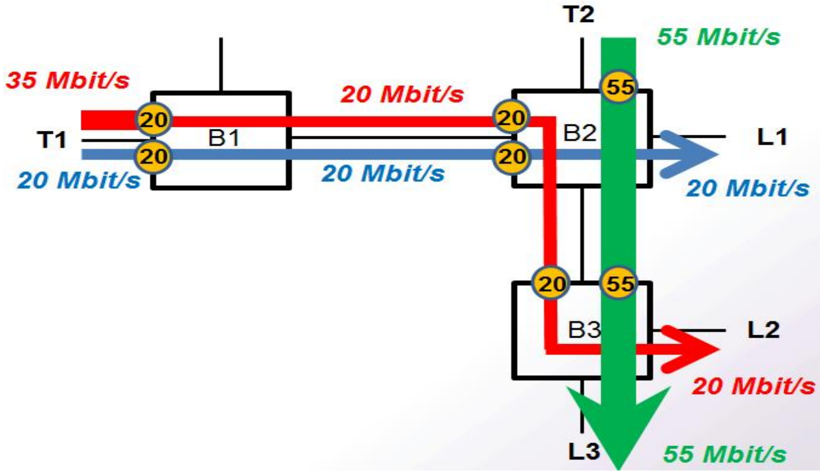
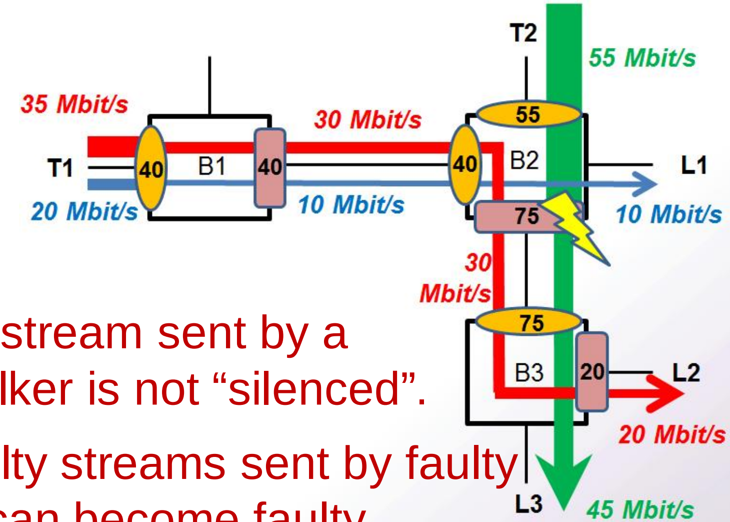
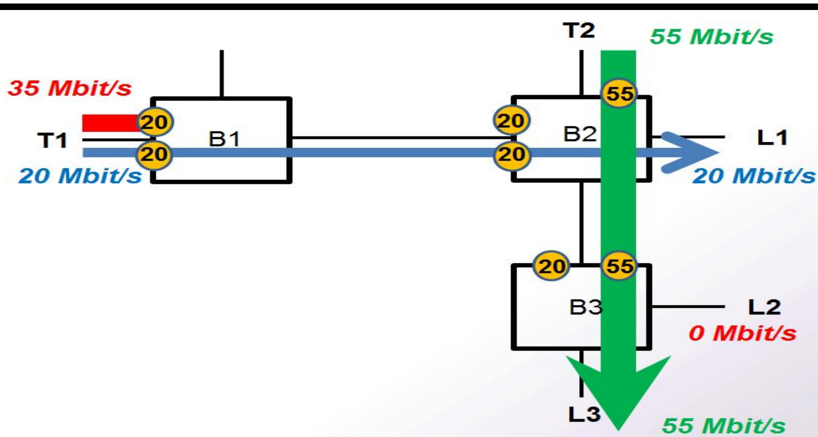
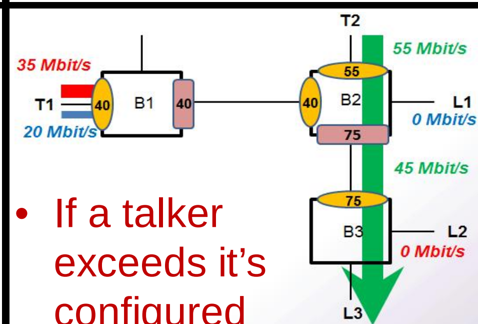
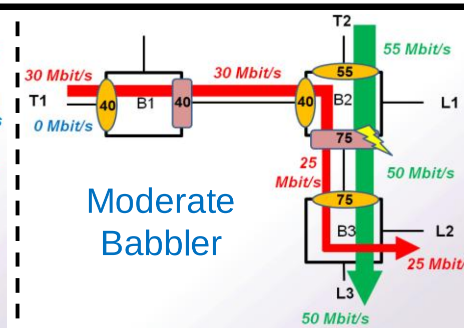
- Moderate Babbler T1: 30 + 0 instead of 20+20
- Shaper at B2 drops frames
  => T2-green becomes faulty.



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". • Non-faulty streams sent by faulty talker are not necessarily 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.

Ranking

Note: The following rankings reflect the opinion of the author. They are intended to be rankings for a generic solution.
For a specific system / application a combination that is not preferable as a generic solution may very well be an acceptable / good solution.

	Per Stream (= Potentially higher number of filters per port)	Per Class (= Small number of filters per port)
Threshold Enforcing	Not a preferred solution. <u>Reason:</u> "*Per Stream X Blocking*" and "*Per Stream X Thr. Enforcing*" both require the same HW overhead associated with Per Stream IPFs, but "*Per Stream X Blocking*" exhibits preferable behavior.	Not a preferred solution. <u>Reason:</u> Fault containment related problem. (Whether or not this is acceptable heavily depends on the specific application.)
Blocking	**#1** **Preferred solution !!!** Very effectively addresses the Babbling Idiot problem.	**#2** **Potential alternative to "Per Stream X Blocking"** <u>Reason:</u> Effectively addresses Babbling Idiot problem in many cases with limited HW resources. Potential drawback: Depending on the application, the Moderate Babbler may need to be considered.

Discussion

Discussion within 802.1 TSN

I would kindly like to ask the group to discuss the alternatives.

- Does the group agree with the ranking from a fault tolerance perspective?
- Does the picture change when we take hardware resource / cost consideration into account?

The intention is to identify a reasonable solution that the group can agree on and support!

Golden Streams

How many filters for Per Stream Policing?

- Key question: **How many filters per port ?**
- If we need one IPF per stream on ingress, then this question causes some headache: **(*1)**
  - High number of filters:
    A waste of HW resources if the high number of streams is not required by a given system.
  - Low number of filters:
    The bridge cannot be used if the system requires more streams.

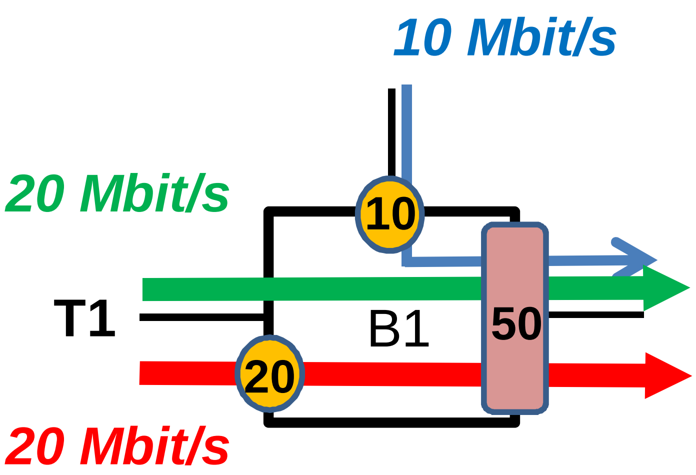
But what if we don't need one IPF per stream even though we do Per Stream Ingress Policing ?

(*1): For AVB1@100Mbit/s it may not be too difficult to define a reasonable number of streams per port, since the maximum number of streams per port is rather limited anyway (13 or 14 streams, see http://www.ieee802.org/1/files/public/docs2013/new-tsn-specht-avb1-automotive-control-20130318.pdf). However, at higher link speeds, or with the introduction of new traffic classes for automotive and industrial control that are currently discussed in 802.1 (e.g. Peristaltic shaper, Urgency based shaper, Burst limiting shaper) and with the desire to enable the credit based shaper to send at lower frequencies,the question remains hard.

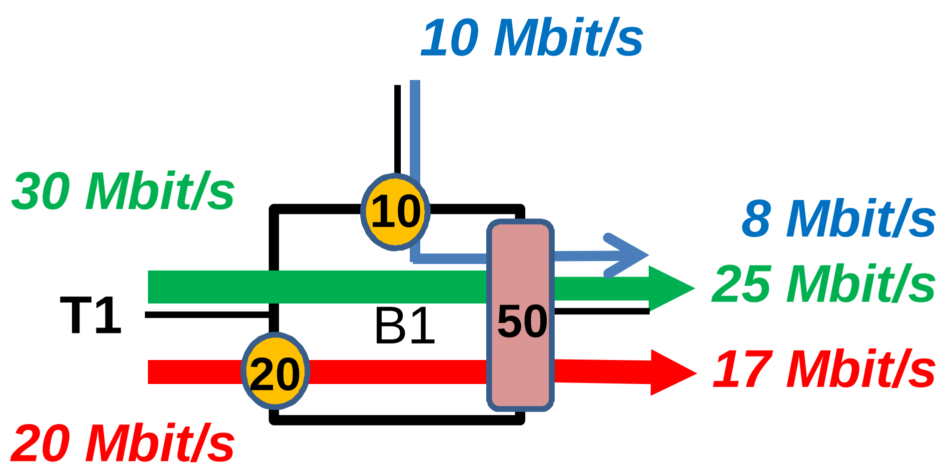
Less than one IPF per stream (1/2)

Again... what if we don't need one IPF per stream even though we do Per Stream Ingress Policing ?

Well... without any modifications on egress that won't work!



Fault free case



Faulty case

(T1-green sends 30 instead of 20)

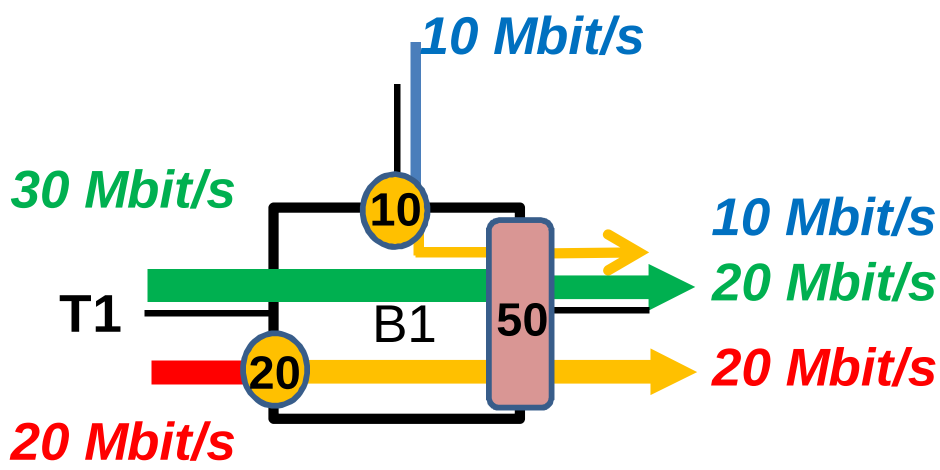
Since there is no IPF for T1-green, the shaper will drop blue, green and red frames on egress!

Note: Like with many other diagrams shown so far, the diagram on the right shows one out of many different ways of how things could play out.

Less than one IPF per stream (2/2)

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.

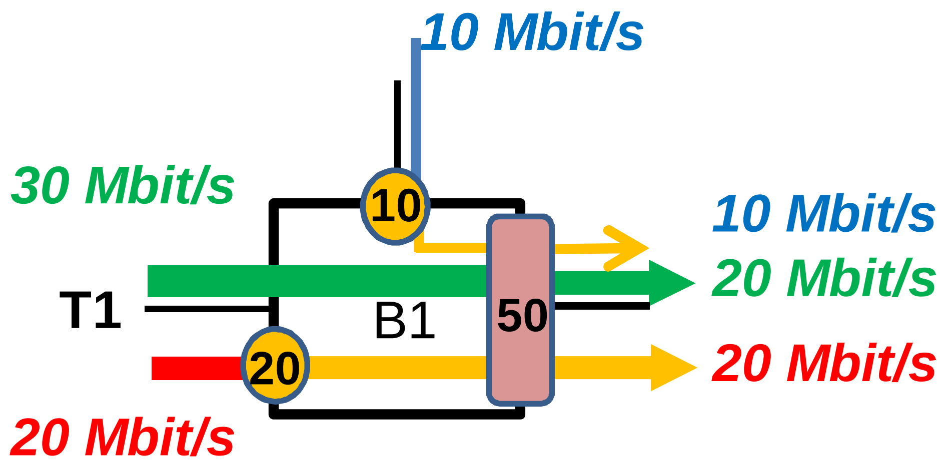


Faulty case

(T1-green sends 30 instead of 20)

Conclusion

- As a result we can reduce the number of IPFs to the anticipated maximum number of safety critical streams per port
- The overall maximum number of streams is no longer limited by the Per Stream IPF concept !



Faulty case

(T1-green sends 30 instead of 20)

Are Golden Streams an idea the 802.1TSN group wants to consider ?

Does that change our conclusion on whether Per-Stream or Per-Class Ingress Policing is preferable?

Filter Latency

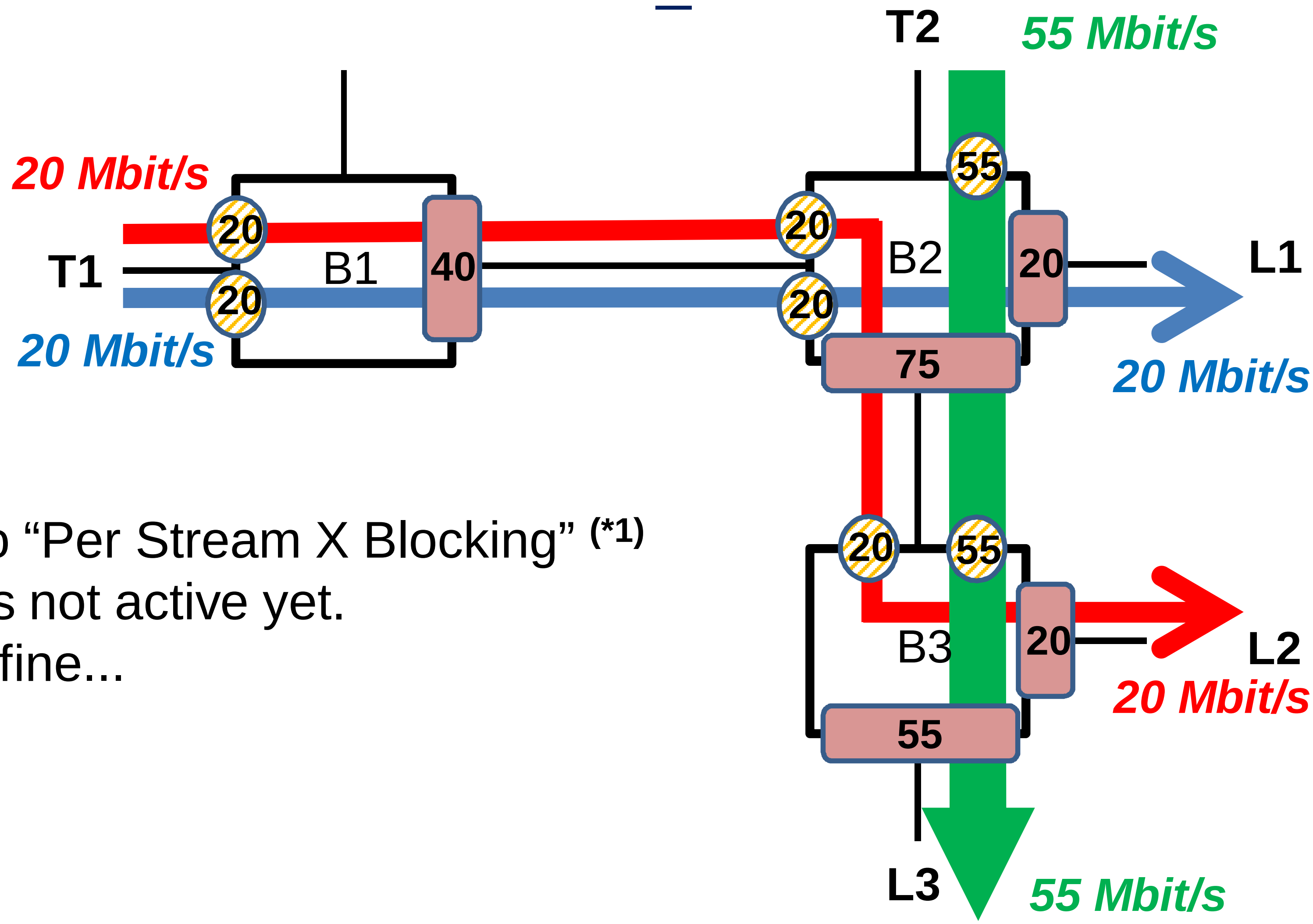
IPF Latency

t0	Node starts babbling
Δt_IPF_decide	Time it takes to decide that an IPF threshold is exceeded. (Not necessarily a constant).
Δt_IPF_activate	Time it takes to activate the IPF response (E.g. "threshold enforcing" or "blocking").
Δt_IPF	The IPF Latency or Filter Latency Δt_IPF = Δt_IPF_decide + Δt_IPF_activate

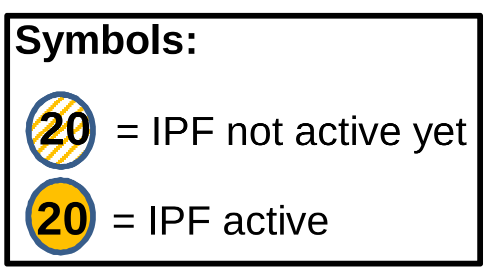
What happens between t0 and t0 + Δt_IPF ?

Let's look at three different time intervals:

t < t0: Fault has not yet occurred

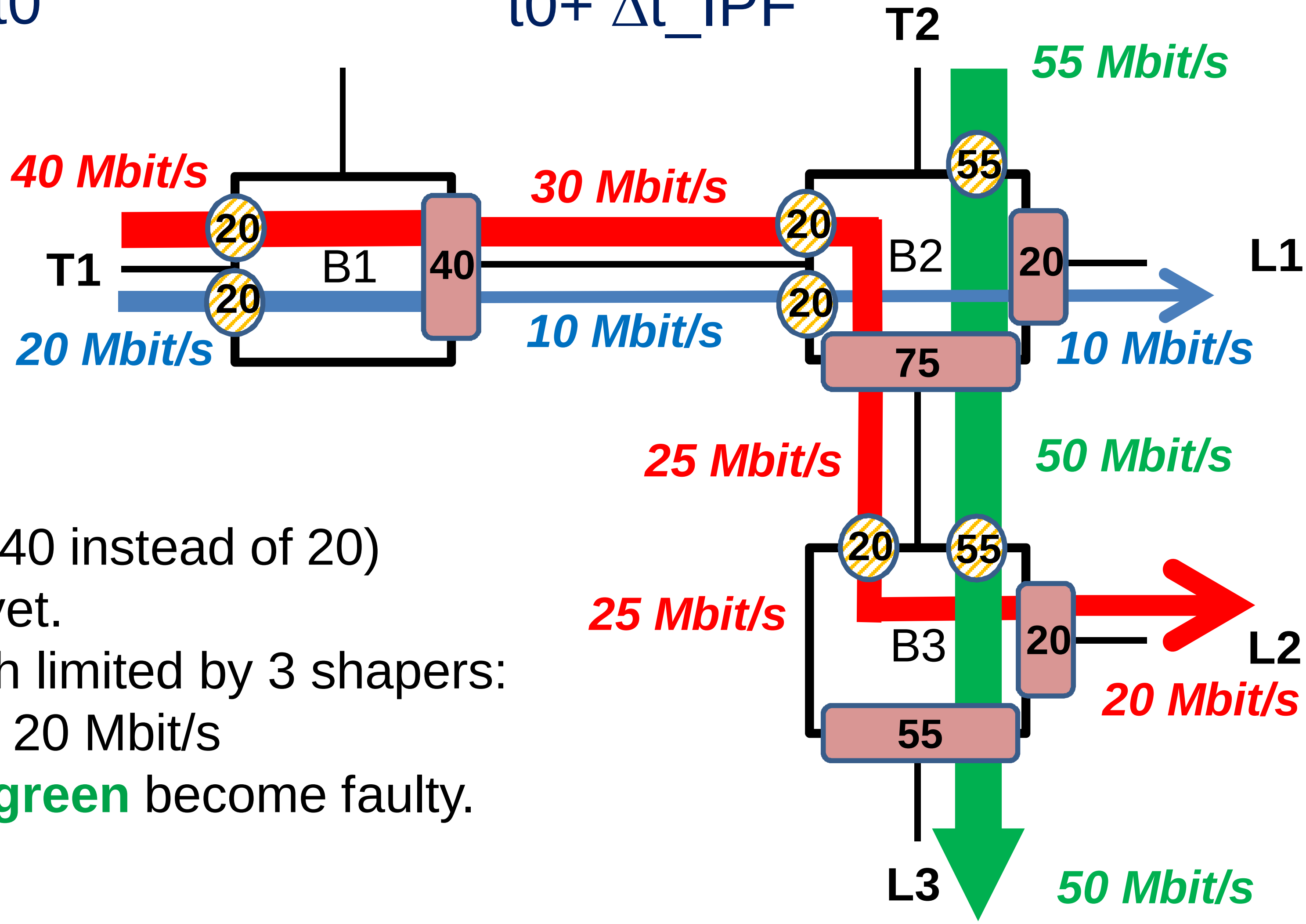


- Assumption: IPFs configured to "Per Stream X Blocking" (*1)
- No fault yet... IPFs not active yet.
- Everything works fine...



(*1): All four combinations {Per Stream, Per Class} X {Threshold enforcing, Blocking} could be evaluated on after the other. However, this is of minor relevance since the behavior during the critical phase between t0 and t0 + Δt_IPF (next slide) does not depend on the configuration of the IPFs.

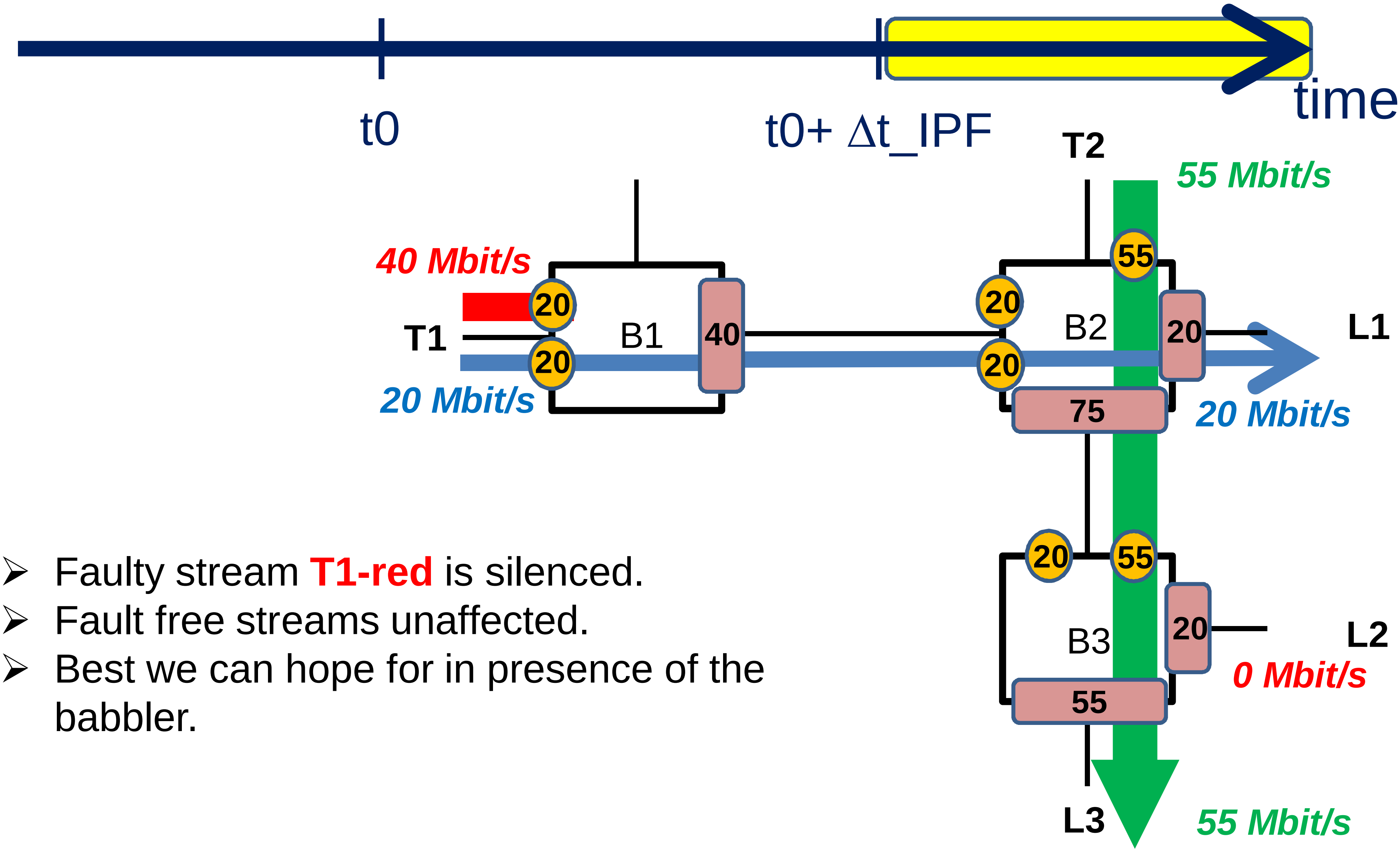
t0 < t < t0 + Δt_IPF: Fault active, IPFs not yet filtering



- **T1-red** babbles (40 instead of 20)
- IPF's not active yet.
- **T1-red** bandwidth limited by 3 shapers: 40 -> 30 -> 25 -> 20 Mbit/s
- **T1-blue** and **T2-green** become faulty.

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

t0 + Δt_IPF < t : Fault active, IPFs are filtering



- Faulty stream **T1-red** is silenced.
- Fault free streams unaffected.
- Best we can hope for in presence of the babbler.

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

Summary of the observations

- 1 - Everything OK.

- 2 -

a) A faulty stream (**T1-red)** sent by a faulty talker was not “silenced”.

b) Non-faulty stream (**T1-blue)** sent by faulty talkers became faulty.

c) A fault free stream **T2-green** sent by a fault free talker became faulty. (Fault propagation. Fault not contained)

- 3 - Best outcome we can hope for in presence of a babbler.

IPF Latency Conclusions

- The temporary effects that occur during **- 2 -** are severe!
- IPF latency must be as small as possible to limit the duration of these effects **(*1)**:

1) **The IPF algorithm(*2) must not introduce unnecessary delay.**

   E.g.: A credit based IPF algorithm with a bucket that is initialized with a substantial initial credit that is gradually / slowly drained when a babbler becomes active may respond too slow, since it takes time to "consume" the initial credit.

   It must be possible to configure a "reasonably nervous IPF".

   => More work on IPF algorithms and their configuration is required !

2) **No unnecessary delay between "babbling detected" and "filter action"**

   E.g.: An bridge implementation that signals "babbling detected" but requires an external micro to command a "start blocking the port" action would introduce substantial / unnecessary delay. The IPF must be able to act autonomously.

(*1): Depending on the application the vehicle level control algorithm may need to be designed to tolerate the effects for limited time (= IPF latency).
(*2): More work on an adequate IPF algorithm is required.

Wrapping up . . .

Presented:

- Ingress Policing for various Traffic Classes
- Pros and Cons of
  {Threshold Enforcing, Blocking} X {Per Port, Per Stream}
- Golden Streams
- IPF Latency and resulting requirements

Objectives for Dallas Plenary:

- Agreement on which concept is preferred.
- Common understanding / agreement on how to move forward with the topic in 802.1

Backup

Babbling Bridges

- So far the slide deck focuses on scenarios where the babbler is an end station.
- The fundamental conclusions that have been drawn may not change all that much when assuming that the babbler is a bridge which has the potential to babble on multiple ports.
- However, this scenario deserves to be needs to be analyzed more thoroughly. => Future work!