Urgency Based Scheduler

- Performance and Characteristics -

Johannes Specht (Univ. Duisburg-Essen)

Background and Content

- **Content** Urgency Based Scheduler (UBS)

  (based on Oct. 2k13 slides)

- **UBS** Flexible Traffic Class?

  (cmp. AAA2C slides)

Topics:

- Egress Operation
- Latency Calculation
- Latency Guarantees in AVB
- Number of Sub Shapers

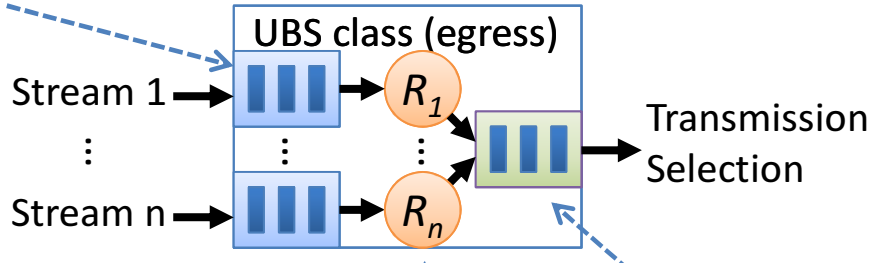
Feedback from TSN call

This presentation

Egress Operation

sub queues

- FIFO



sub shapers

- leaky buckets
- independent/non-blocking

priority queue

- Output highest priority first

```
while (true){
  wait until subQ[i].containsFrame();
  frame = subQ[i].removeHead();
  priorityQ.insert(frame);
  wait for frame.bitSize*linkSpeed/R[i];
}
```

Priority per frame **vs.** **Priority per sub shaper**

Work in progress…
- Performance
- Configuration Complexity
- …

On the next slides

Later slides merge multiple streams per sub queue / sub shaper

Purpose

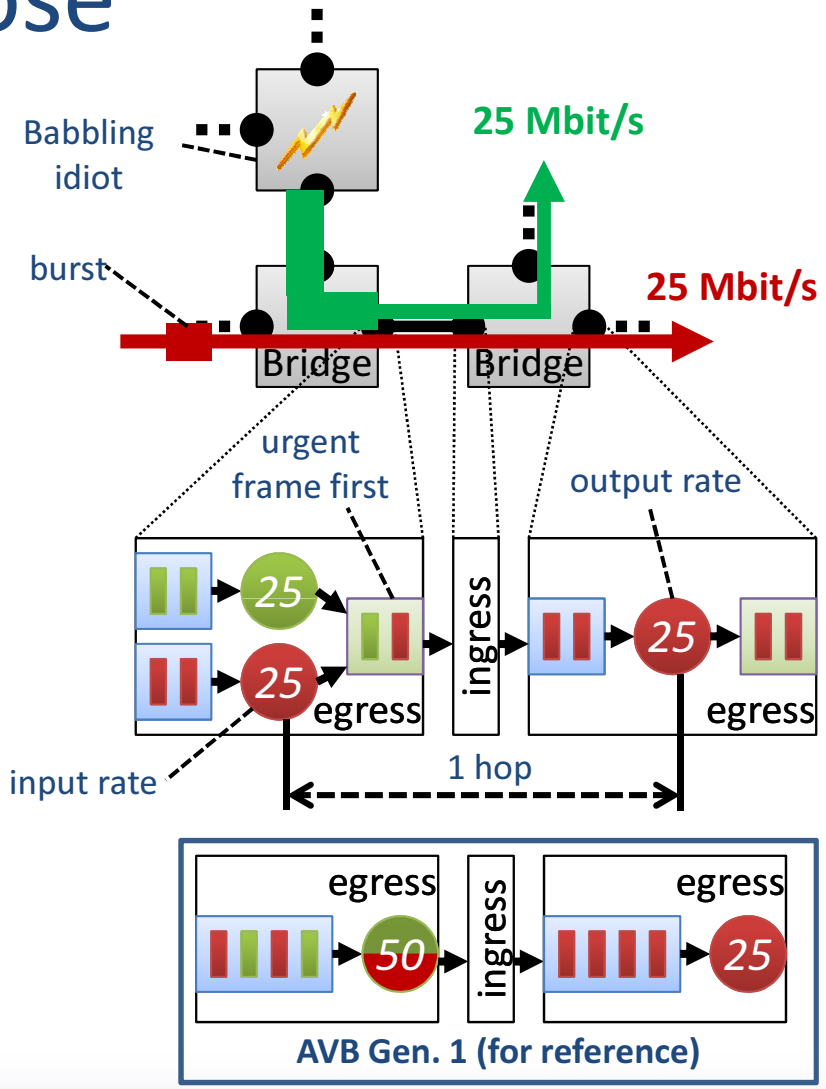
- **Latency calculation per Hop**

  Assure: *input rate* = *output rate*

- **Prevent burst propagation/ accumulation**

- **Accelerate Urgent Traffic**

- **Side-effect:**
  ***Precise* bandwidth policing**



FIFO Queue Prio. Queue R (Sub) Shaper (rate *R*)

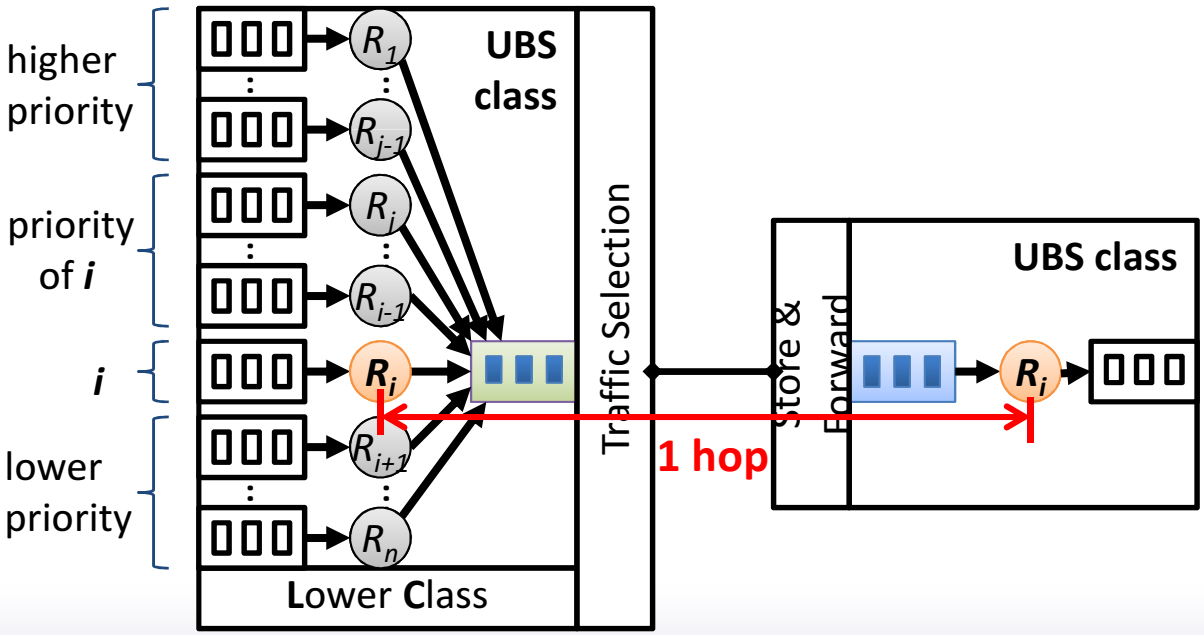
LATENCY CALCULATION

General Calculation

higher of *i* lower

$$W_i^{max} = \frac{\sum_{k=1}^{j-1} l_k^{max} + \sum_{k=j}^{i-1} l_k^{max} + max\{l_{LC}^{max}, l_{i+1}^{max}, ..., l_n^{max}\}}{R - \sum_{k=1}^{j-1} R_k} + \frac{l_i^{max}}{R}$$

interfering traffic S&F



Max. latency of priority *i* [s]

W_i^{max}	per hop

Maximum frame lengths [bit]

$l_1^{max} ... l_{j-1}^{max}$	higher priorities
$l_j^{max} ... l_{i-1}^{max}$	priority of *i*
l_i^{max}	*i*
$l_{i+1}^{max} ... l_n^{max}$	lower priorities
l_{LC}^{max}	lower classes

Shaper/link rates [bit/s]

$R_1 ... R_{j-1}$	higher priorities
R_i	priority *i*
R	link

Notes

Calculation under validation; No link delays and additional device dependent delays included;Calculation for bridge→end station is the same

Frame length includes overhead, interferring traffic frame lengths include IFG

Possible Mappings

$$W_i^{max} = \frac{\sum_{k=1}^{j-1} l_k^{max} + \sum_{k=j}^{i-1} l_k^{max} + max\{l_{LC}^{max}, l_{i+1}^{max}, ..., l_n^{max}\}}{R - \sum_{k=1}^{j-1} R_k} + \frac{l_i^{max}}{R}$$

Traffic Type	$l_{1...n}^{max}$	$R_{1...n}$
AVB Stream	MaxFrameLength	$\frac{\text{MaxFrameLength*MaxIntervalFrames}}{\text{class measurement interval}}$
Periodic Control Stream (const. frame length)	frame length	$\frac{\text{Frame length}}{\text{Period}}$
<Something totally different>	maximum frame length	rate

Max. latency of priority *i* [s]

W_i^{max} per hop

Maximum frame lengths [bit]

$l_1^{max} ... l_{j-1}^{max}$ higher priorities

$l_j^{max} ... l_{i-1}^{max}$ priority of ***i***

l_i^{max} ***i***

$l_{i+1}^{max} ... l_n^{max}$ lower priorities

l_{LC}^{max} lower classes

Shaper/link rates [bit/s]

$R_1 ... R_{j-1}$ higher priorities

R_i priority ***i***

R link

Notes

Unit conversions not shown explicitly; Additional frame overhead not shown for MaxFrameLength

AVB LATENCY GUARANTEES

Latency Guarantees

Engineered Networks

- Maximum Latency:
  - Requirement per stream
    → Input parameter
  - Known before runtime
- Priority Configuration Goal:
  - Map latency requirements of all streams

Plug and Play (AVB Gen. 1)

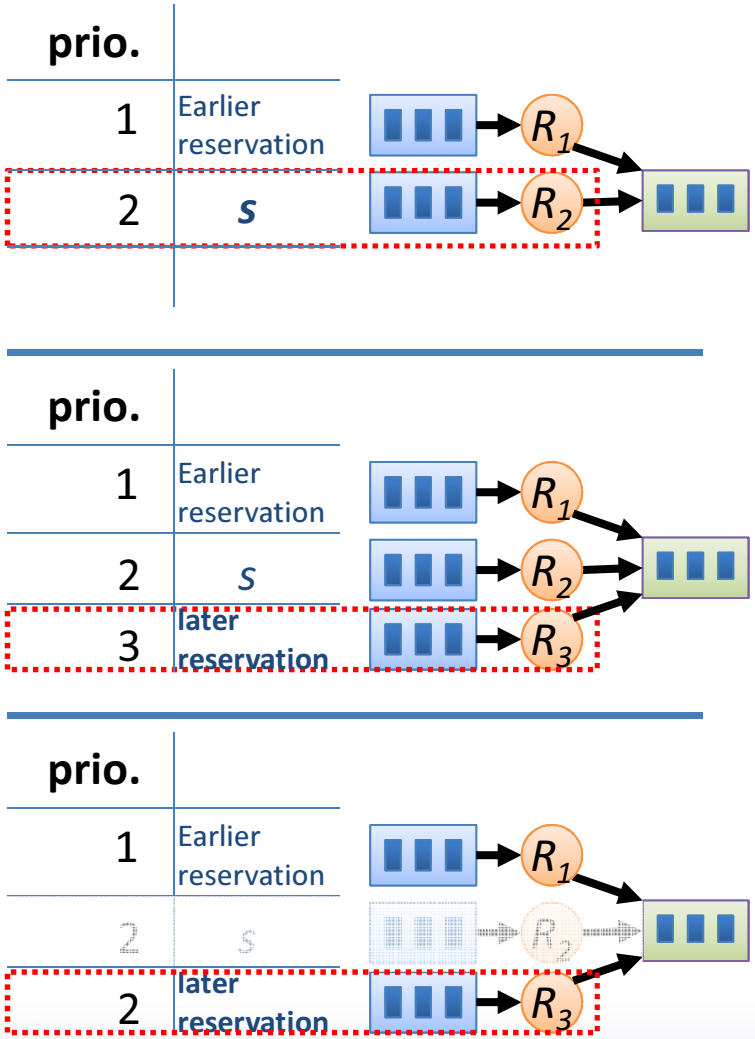
- Maximum Latency:
  - Result of Reservation
    → Output parameter
  - Runtime dependent
- Priority Configuration Goal:
  - Stable maximum latency over reservation lifetime

Possible algorithm per hop on the next slides

Priority Configuration

FCFS Priorities (per Port)

1. **Reservation of stream *s***
   - Highest free priority
2. **Later reservations**
   - Lower priority than *s*
3. **De-reservation of stream *s***
   - Increase priorities of later reservations



Accumulated Latency

Earlier res.
- Higher priorities
- Can only **decrease**

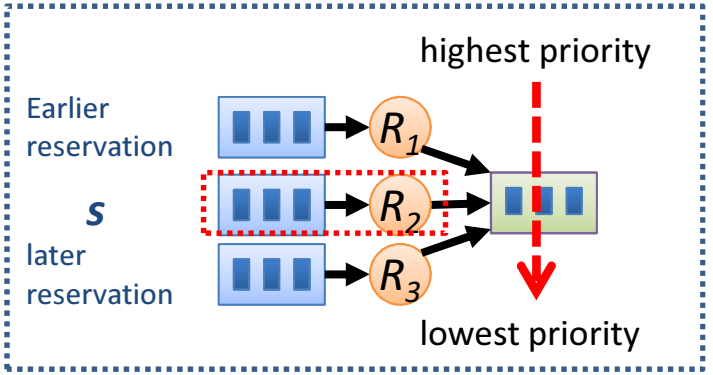
Unused (unique priorities)

Later reservations (not yet known)
→ *Max. frame size*

$$W_i^{max} = \frac{\sum_{k=1}^{j-1} l_k^{max} + \sum_{k=j}^{i-1} l_k^{max} + max\{l_{LC}^{max}, l_{i+1}^{max}, ..., l_n^{max}\}}{R - \sum_{k=1}^{j-1} R_k} + \frac{l_i^{max}}{R}$$

Link Speed - earlier res.
- Can only **increase**

Tspec of *s*



Max. latency of priority *i* [s]

W_i^{max}	per hop

Maximum frame lengths [bit]

$l_1^{max} ... l_{j-1}^{max}$	higher priorities
$l_j^{max} ... l_{i-1}^{max}$	priority of ***i***
l_i^{max}	***i***
$l_{i+1}^{max} ... l_n^{max}$	lower priorities
l_{LC}^{max}	lower classes

Shaper/link rates [bit/s]

$R_1 ... R_{j-1}$	higher priorities
R_i	priority ***i***
R	link

Acc. Latency

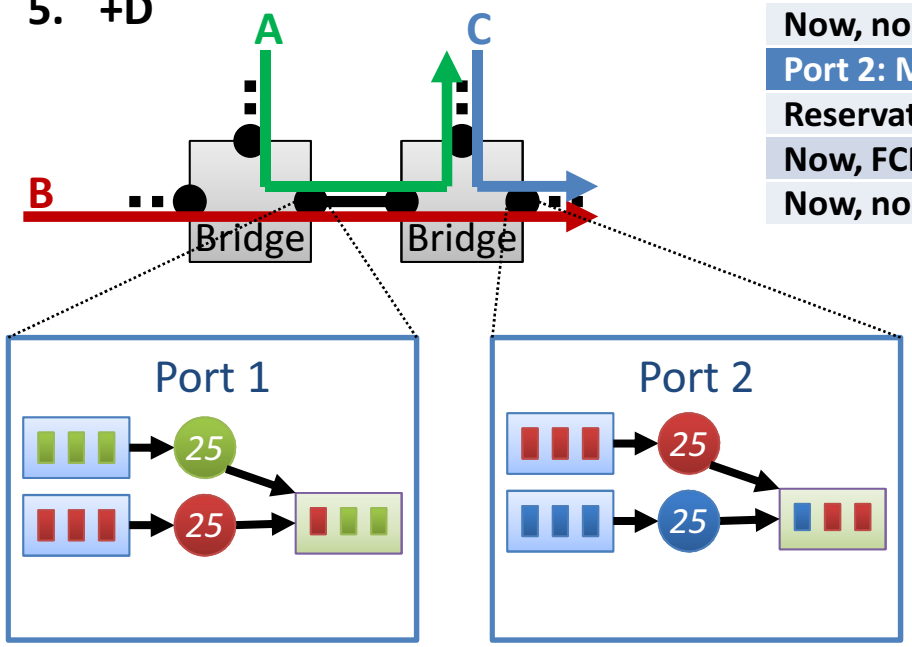
Not increased by later reservations

- Plus additional overhead
- Unit conversions not shown
- Store and forward

Example: +A→+B→+C

Sequence:

1. +A
2. +B
3. +C
4. - B
5. +D



Stream	A	B	C	D
Max. frame size [bit]	1024	2048	4096	
Rate [Mbit/s]	25	25	25	
Port 1: Maximum Additional Latency				
Reservation, FCFS [us]	133,60	198,61	-	
Now, FCFS [us]	133,60	198,61	-	
Now, no prios* [us]	154,08	154,08	-	
Port 2: Maximum Additional Latency				
Reservation, FCFS [us]	-	143,84	232,75	
Now, FCFS [us]	-	143,84	232,75	
Now, no prios* [us]	-	184,80	184,80	

*For comparison only

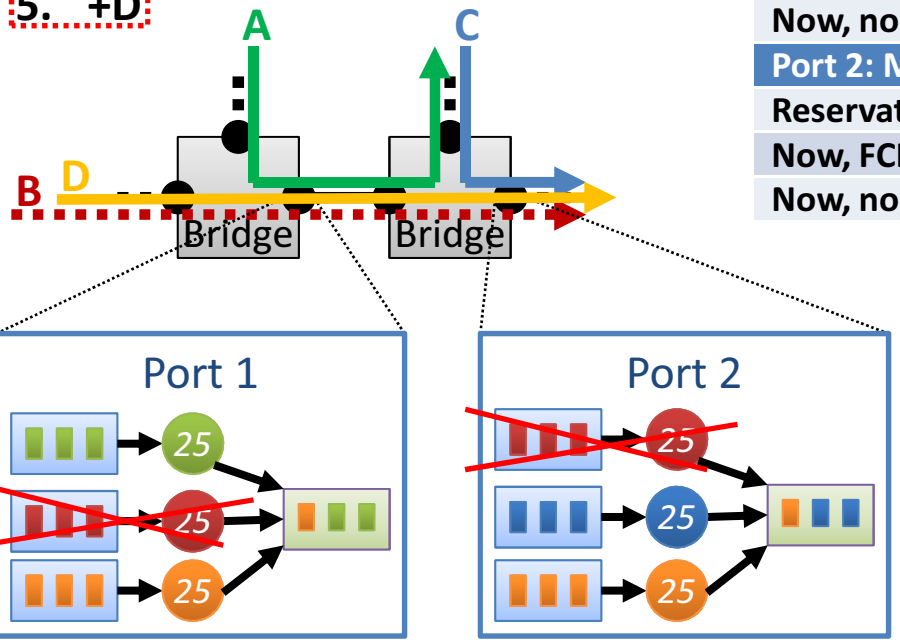
Assumptions

-100 Mbit/s/Fast Ethernet
- Max. Frame size: 1542 Byte
- Store & Forward
- No preemption

Example: -B→+D

Sequence:

1. +A
2. +B
3. +C
4. - B
5. +D



Stream	A	B	C	D
Max. frame size [bit]	1024	2048	4096	4096
Rate [Mbit/s]	25	25	25	25
Port 1: Maximum Additional Latency				
Reservation, FCFS [us]	133,60	198,61	-	219,09
Now, FCFS [us]	133,60	198,61	-	219,09
Now, no prios* [us]	174,56	154,08	-	174,56
Port 2: Maximum Additional Latency				
Reservation, FCFS [us]	-	143,84	232,75	260,05
Now, FCFS [us]	-	143,84	164,32	260,05
Now, no prios* [us]	-	184,80	205,28	205,28

*For comparison only

stable

stable,
even smaller now:
de-reservation of B …

No prios:
Better for later res.,
but unstable
(was 184,80 for C)

… used by D

Assumptions:
-100 Mbit/s/Fast Ethernet
- Max. Frame size: 1542 Byte

- Store & Forward
- No preemption

Mixing with engineered streams

FCFS “around” engineered streams

- **Reservations**

  Highest priorities unless required latency of eng. streams violated

- **De-reservations**

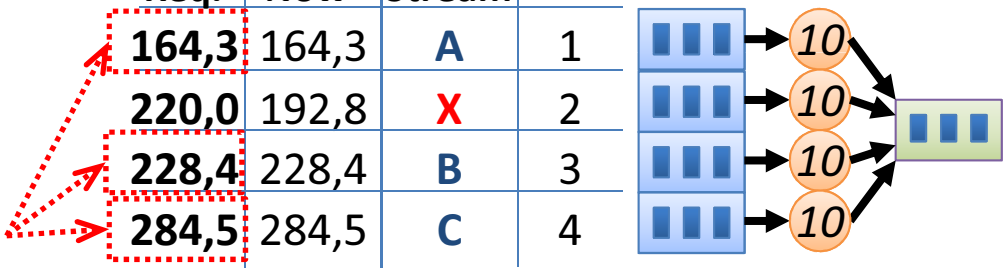
  Increase priorities of lower reservations, if possible

Req.	Now	Stream	Prio.
220,0	133,6	X	1

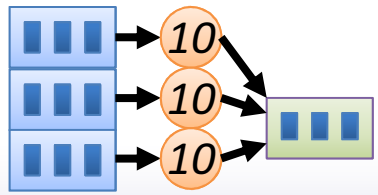
Result of engineering

Req.	Now	Stream	Prio.
164,3	164,3	A	1
220,0	192,8	X	2
228,4	228,4	B	3
284,5	284,5	C	4

Result of reservation
(see prev. slides)



Req.	Now	Stream	Prio.
228,4	133,6	B	1
284,5	168,9	C	2
220,0	213,1	X	3



Assumptions
-100 Mbit/s/Fast Ethernet
- Store & Forward
- Max. Frame size: 1542 Byte
- No preemption

	Type	Max. Frame Length
X	Engineered	1024 bit
A	Reserved	4096 bit
B	Reserved	1024 bit
C	Reserved	2048 bit

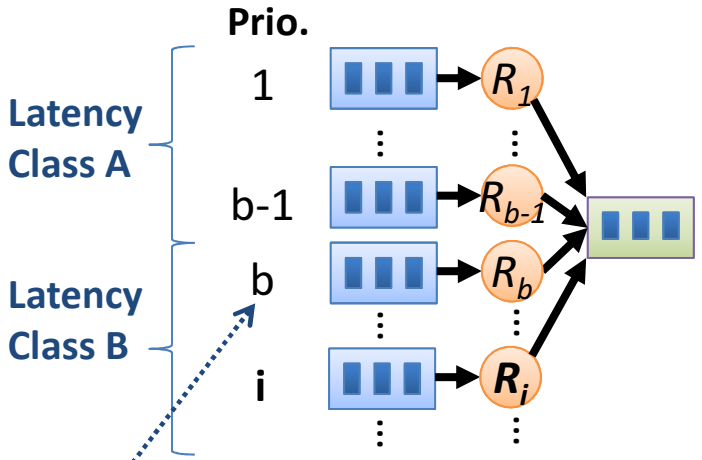
One way to implement classes

Consideration

- Have 2 *Latency Classes* (A and B)
  (both in one UBS class)
- Requires definitions:

  R_{UBS}^{max} : max. UBS Class rate, e.g. 75%

  I_B^{max} : max. interference on class B, e.g. 1 ms



Sum of class A max. frame lengths

$$I_B^{max} = \frac{L_A^{max} + 1542 \cdot 8}{R - R_{UBS}^{max}}$$

Variable during runtime
(increased/decreased by cl. A)

Stable guarantees for Latency Class B

- FCFS below class A with adjusted math:

$$W_i^{max} = \frac{L_A^{max} + \sum_{k=b}^{i-1} l_k^{max} + 1542 \cdot 8}{R - \left(R_{UBS}^{max} - R_i\right)} + \frac{l_i^{max}}{R}$$

Check on class A reservations
(reject cl. A reservations beyond this limit)

$$L_A^{max} = I_B^{max} \cdot (R - R_{UBS}^{max}) - 1542 \cdot 8$$

Latency Guarantees: Summary

FCFS per egress port

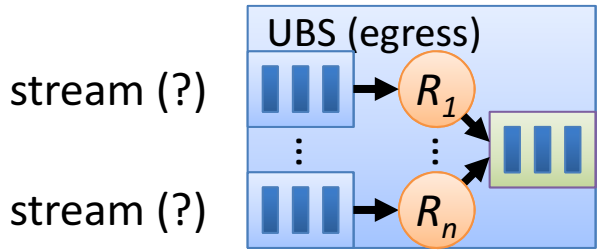
- Possible local configuration algorithm
- Stable Latency Guarantees

One single traffic class: UBS

- Multiple latency classes …
- … mixed with engineered streams

NUMBER OF SUB SHAPERS

Number of Sub Shapers



Efficient bridge implementations

- One sub shaper per stream

→ ***Best Performance***

Scale down to bridge capabilities

- Merge streams/flows
- Manage sub shaper allocation

→ ***Reduced Performance***

Scaling options on the next slides

Merge Option 1

Merge f_1 and f_2, if

- *talker(f_1) = talker(f_2)*

  AND

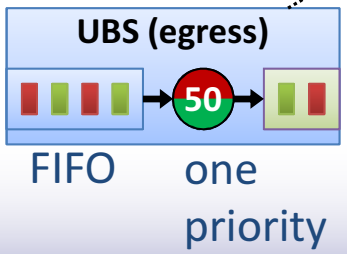
- *listeners(f_1) = listeners(f_2)*

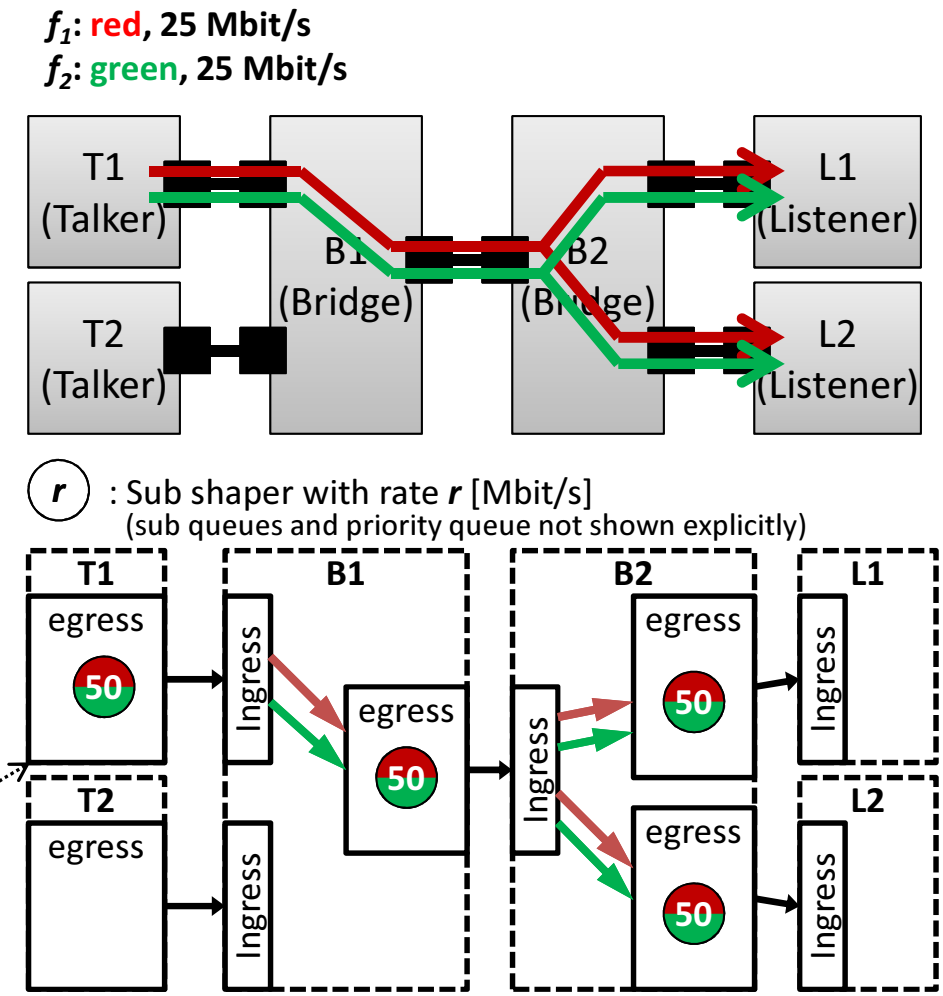
No later "split"

- *f_1 and f_2* : One FIFO

  → *Same priority on path*

  (varying priority per port of merged flow still possible)





Merge Option 2

Merge f_1 and f_2, if

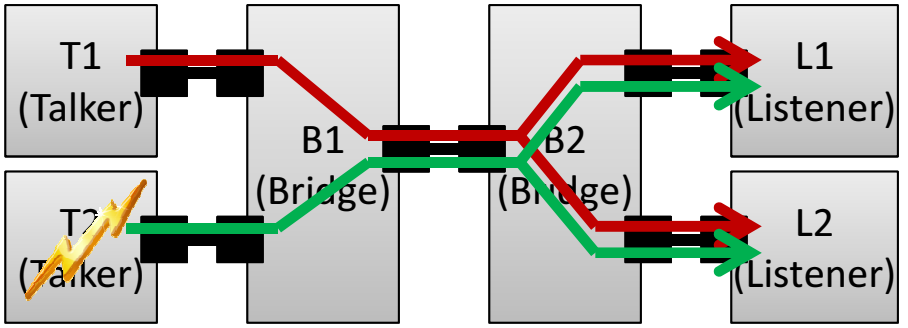
- *listeners(f_1) = listeners(f_2)*

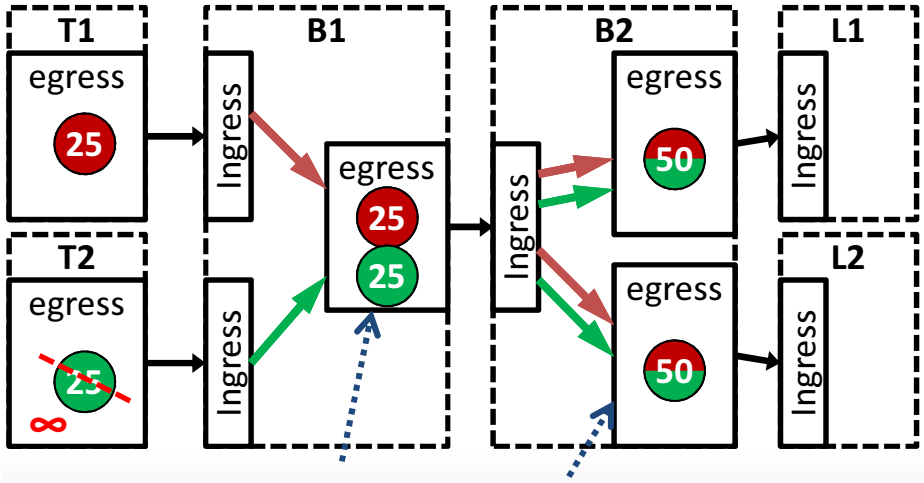
AND

- *ingressPort(f_1) = ingressPort(f_2)*

Maintains bandwidth policing

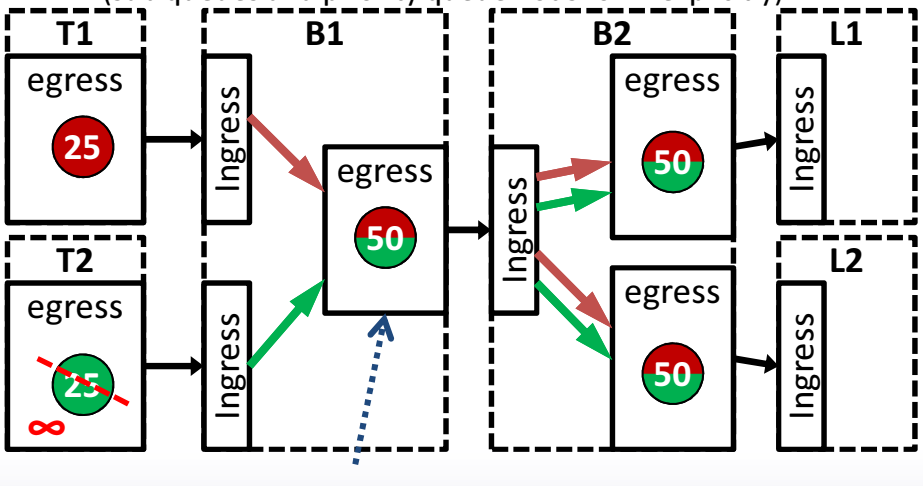
f_1: red, 25 Mbit/s
f_2: green, 25 Mbit/s





Maintains bandwidth policing

(r) : Sub shaper with rate *r* [Mbit/s]
(sub queues and priority queue not shown explicitly)



No bandwidth policing

Merge Option 3

Merge f_1 and f_2, if

- $listeners(f_1) \neq listeners(f_2)$

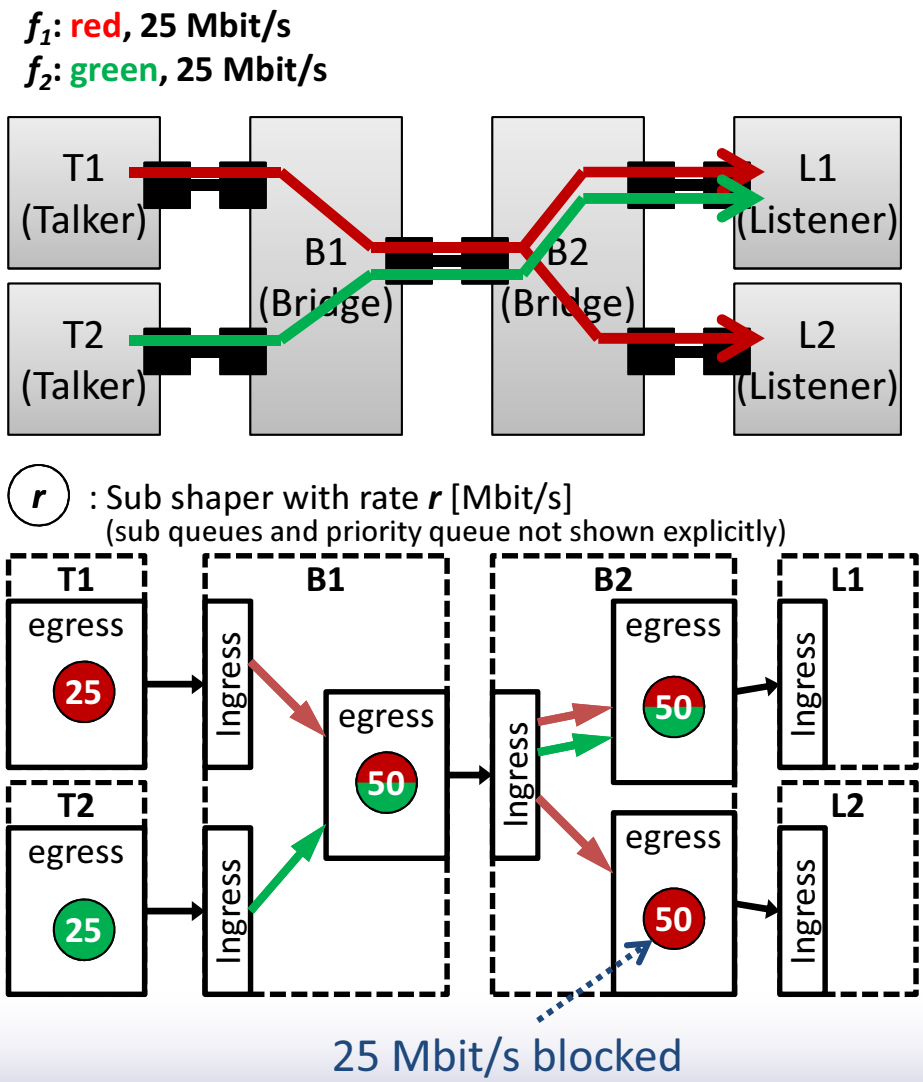
AND

- $ingressPort(f_1) = ingressPort(f_2)$

Maintains bandwidth policing

Requires "sub shaper routing"

- Different metrics:
  - $listeners(f_1)\ \Delta\ listeners(f_2)$
  - blocked bandwidth
  - …
- …



Merge Options: Summary

#	Rule	BW Policing	Blocks BW
1	*listeners(f_1) = listeners(f_2)* **AND** *talker(f_1) = talker(f_2)*	Yes	No
2a	*listeners(f_1) = listeners(f_2)*	No	No
2b	*listeners(f_1) = listeners(f_2)* **AND** *ingressPort(f_1) = ingressPort(f_2)*	Yes	No
3a	*listeners(f_1) ≠ listeners(f_2)*	No	Yes
3b	*listeners(f_1) ≠ listeners(f_2)* **AND** *ingressPort(f_1) = ingressPort(f_2)*	Yes	Yes

- The above rules can be applied, if
  - a device runs out of ressources and
  - The stream requirements allow it

Summary

Proposal for Flexible Traffic Class:

- Urgency Based Scheduler

Previous Slides:

- Egress Operation
- Latency Calculation
  (for priorities per Sub Shaper)
- Simple plug and play configuration algorithm
- Downscaling Options

Further Work:

- Global configuration algorithms
- Analysis of per frame priority algorithms

Thank you for your Attention!

Questions, Opinions, Ideas?

Johannes Specht

Dipl.-Inform. (FH)

Dependability of Computing Systems
Institue for Computer Science and
Business Information Systems (ICB)
Faculty of Economics and
Business Administration
University of Duisburg-Essen

Schützenbahn 70
Room SH 502
45127 Essen
GERMANY
T +49 (0)201 183-3914
F +49 (0)201 183-4573

specht@dc.uni-due.de
http://dc.uni-due.de