Worst-case backlog evaluation of Avionics switched Ethernet networks with the Trajectory approach

Henri BAUER
Jean-Luc SCHARBARG
Christian FRABOUL

IRIT/INPT-ENSEEIHT
Université de Toulouse

Université de Toulouse

INP TOULOUSE ENSEEIHT

IRIT
CNRS - INPT - UPS - UT1 - UTM

ECRTS 2012
10-13 July
Pisa, Italy

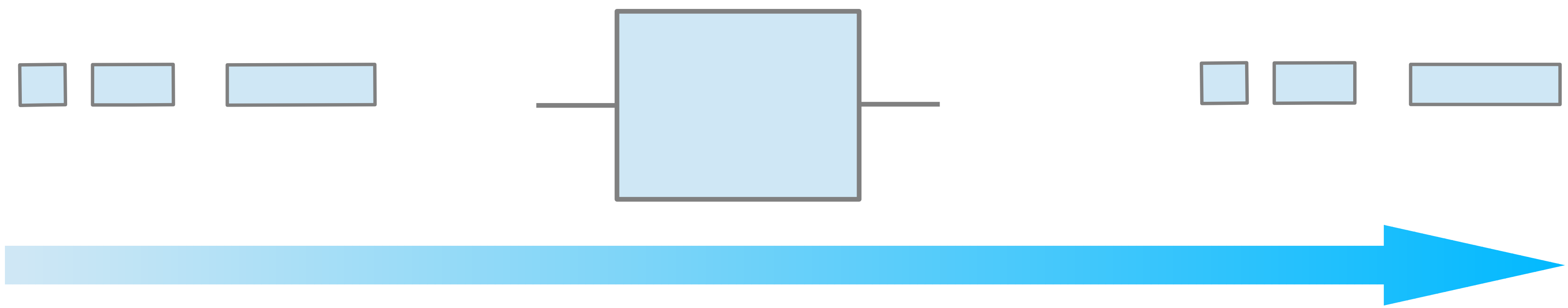
- Avoid over or under **sizing of output port buffers** in store-and-forward switches
- **Industrial concern**: at least as important as bounding end-to-end communication delays (aircraft certification)
- Trajectory approach has **improved bounds** on worst case end-to-end delays (compared to Network Calculus)

→ Compute **worst case buffer occupancy** using the **Trajectory approach** (with a FIFO servicing policy)

→ Application to an **industrial AFDX configuration**

Embedded networks with real time constraints

Frames to be transmitted | Shared com. resources | COM. CHANEL MULTIPLEXING



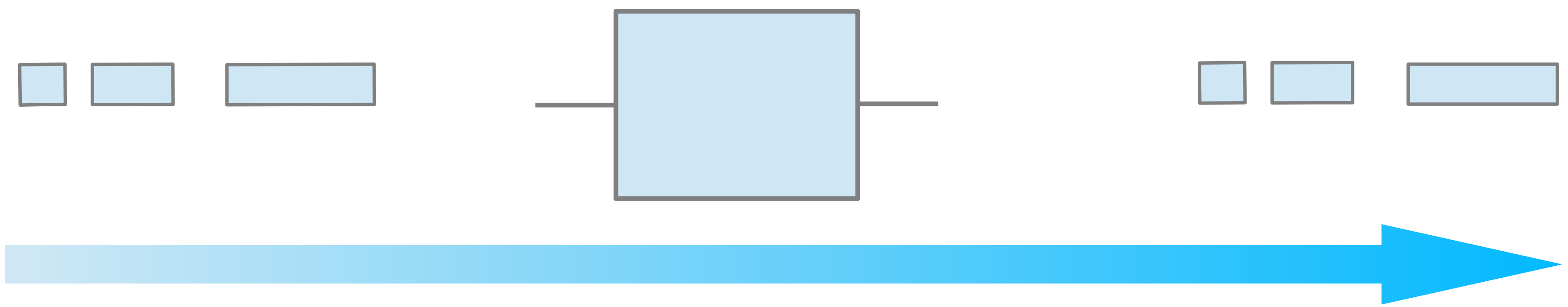
COMMUNICATION DETERMINISM

Bounded frame **transmission time** (CST + VAR) :

- Constant part = technological latency
- Variable part = output buffer occupancy
  → **maximum backlog**

Backlog and transmission delay — a dual problem

Frames to be transmitted	Shared com. ressources	COM. CHANEL MULTIPLEXING



APPLICATION LEVEL

maximum transmission time

FUNCTION SUPPLIER

HARDWARE LEVEL

maximum backlog

SYSTEM INTEGRATOR

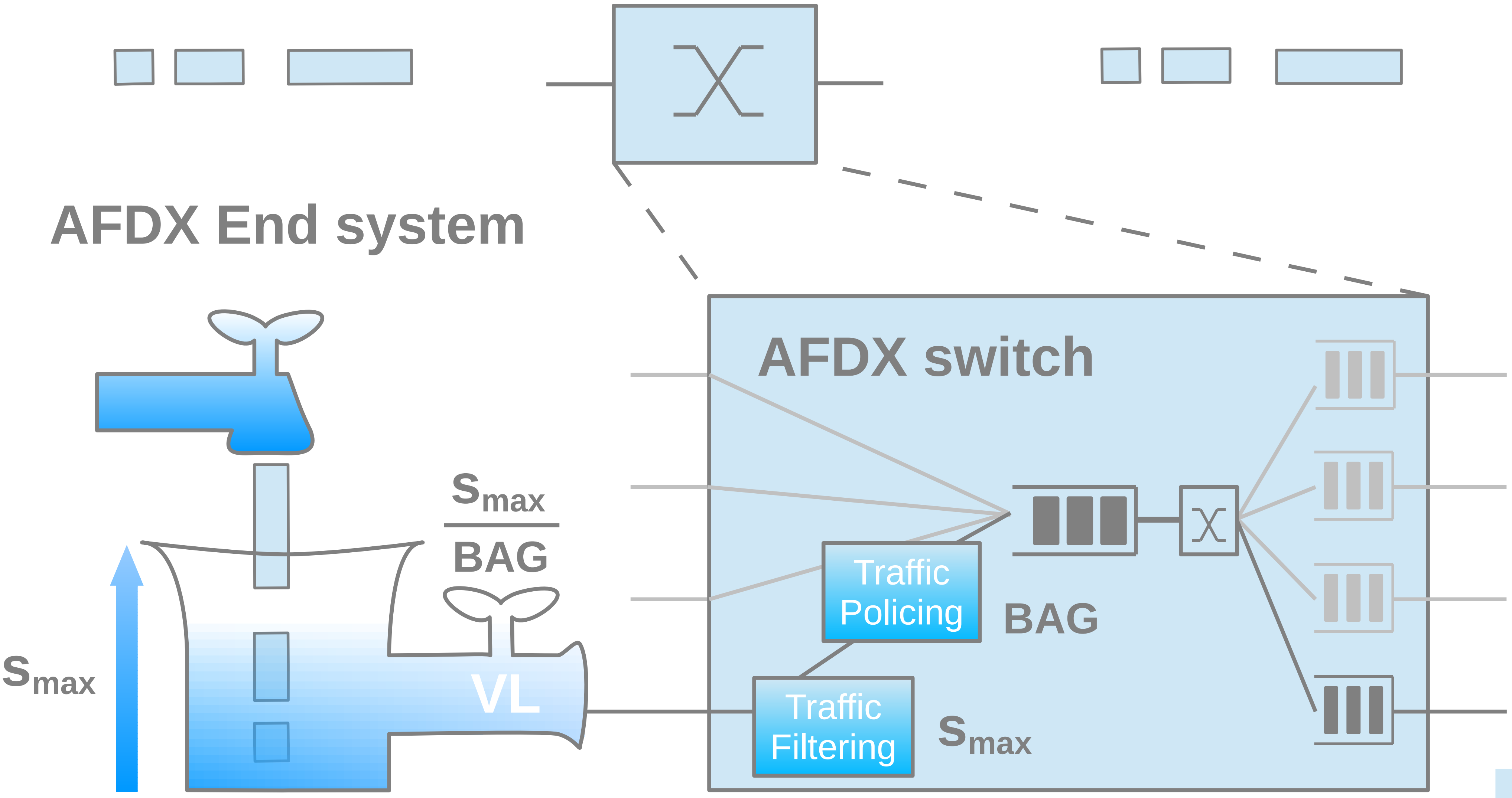
Avionics Full Duplex Switched Ethernet

AFDX

End systems send frames through VLs

AFDX switch

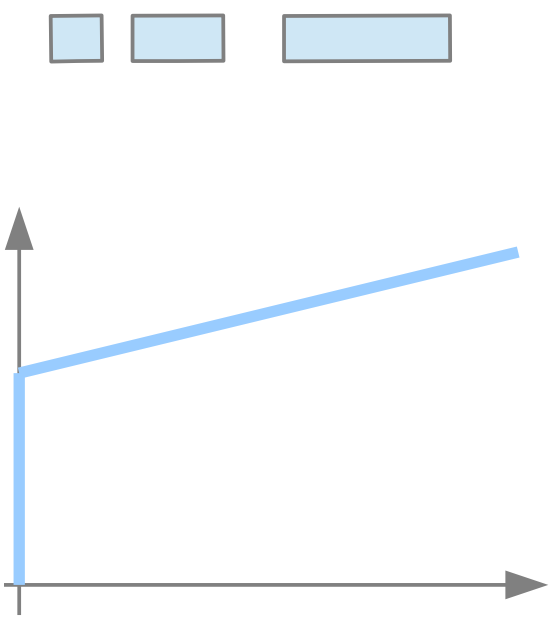
FIFO servicing policy



Computing worst case upper-bounds for backlog and delay

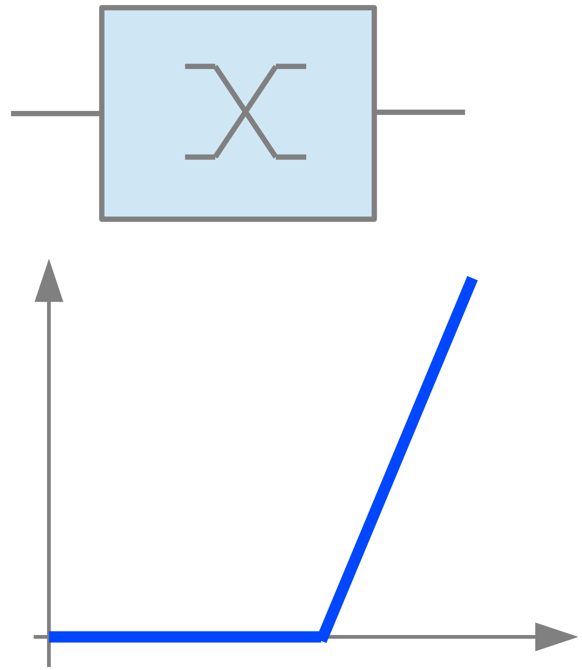
Network Calculus

Incoming frames	Network element output port	Delayed frames



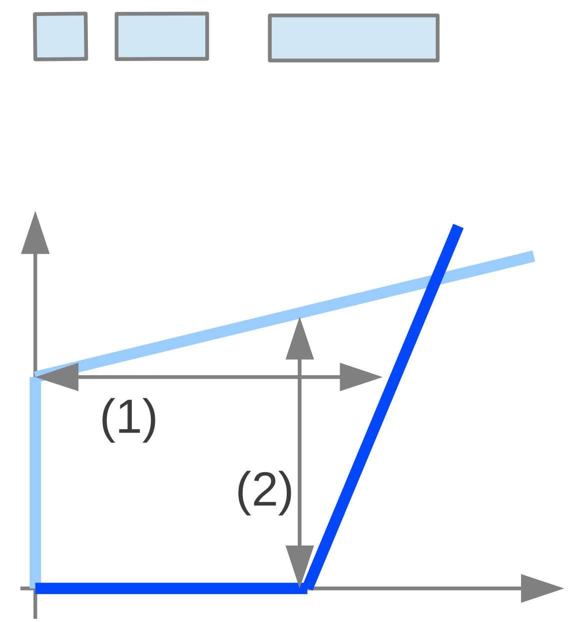
ARRIVAL CURVE

(leaky bucket)



SERVICE CURVE

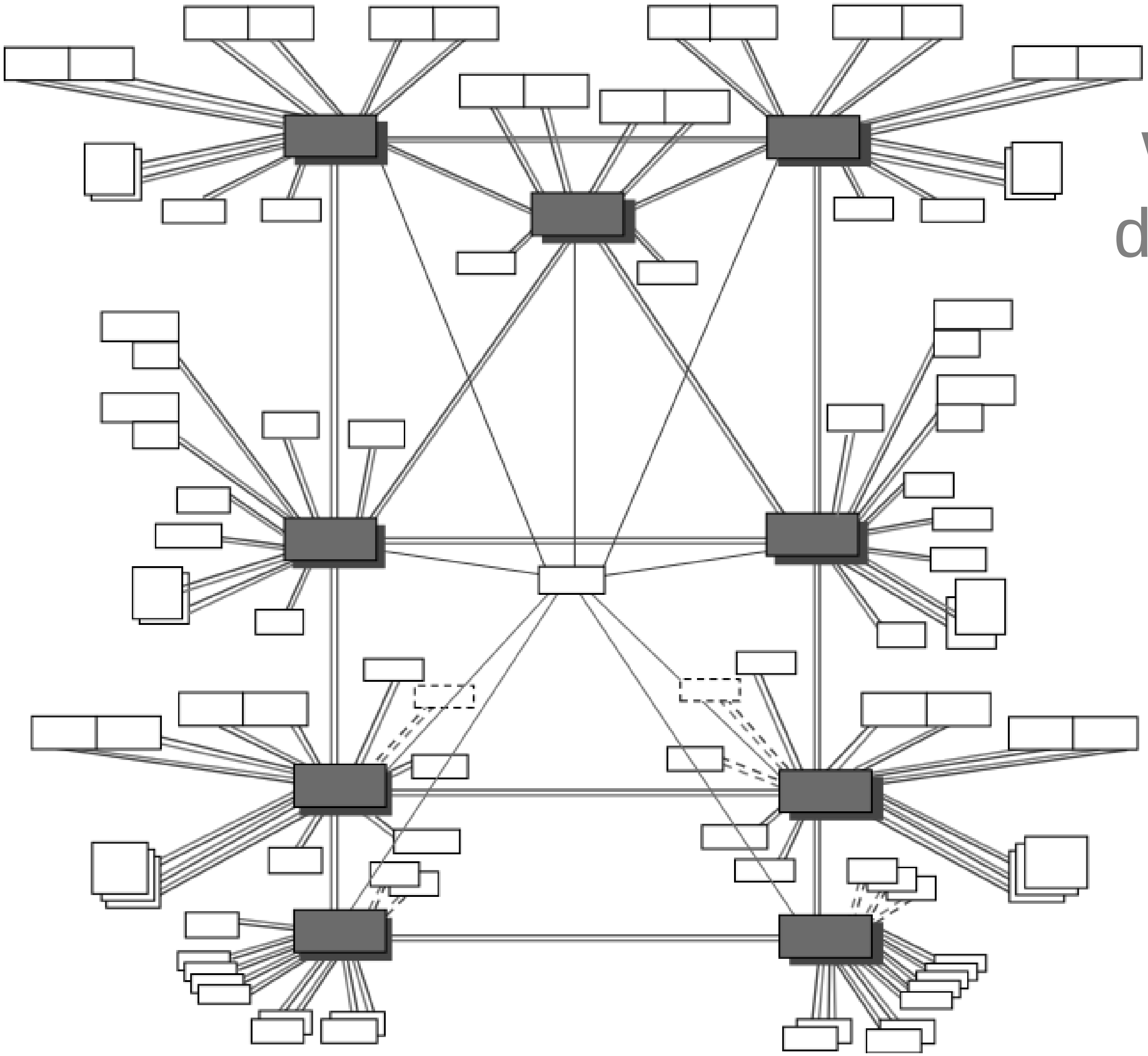
(rate-latency)



(1) maximum virtual delay

(2) maximum virtual backlog

Existing approach	Network Calculus
Proposed method	Trajectory approach



Worst case end-to-end delay computation on an industrial configuration

Trajectory (vs. NC)

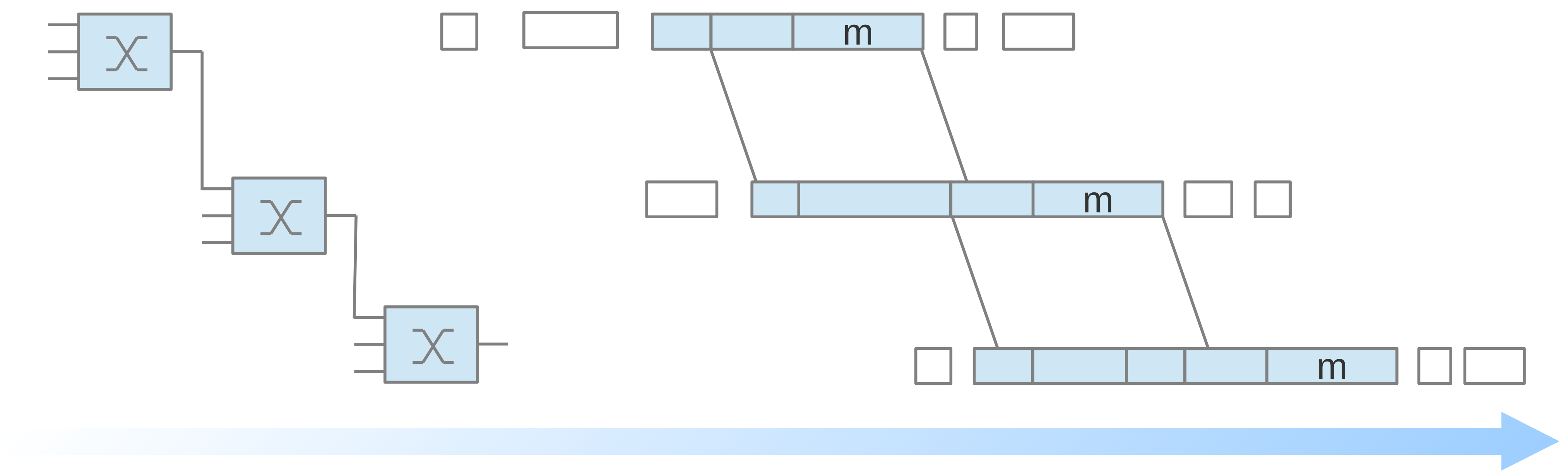
- Average: **> 10 %**
- Maximum : **> 34%**

(Previous work)

The Trajectory approach

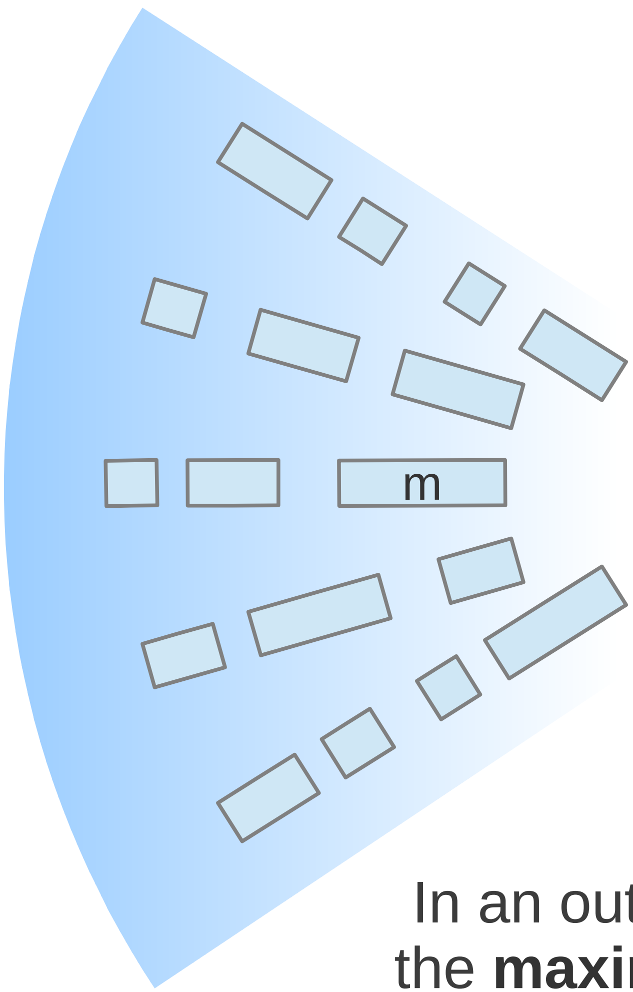
Study packet m
throughout its trajectory

Distributed
system



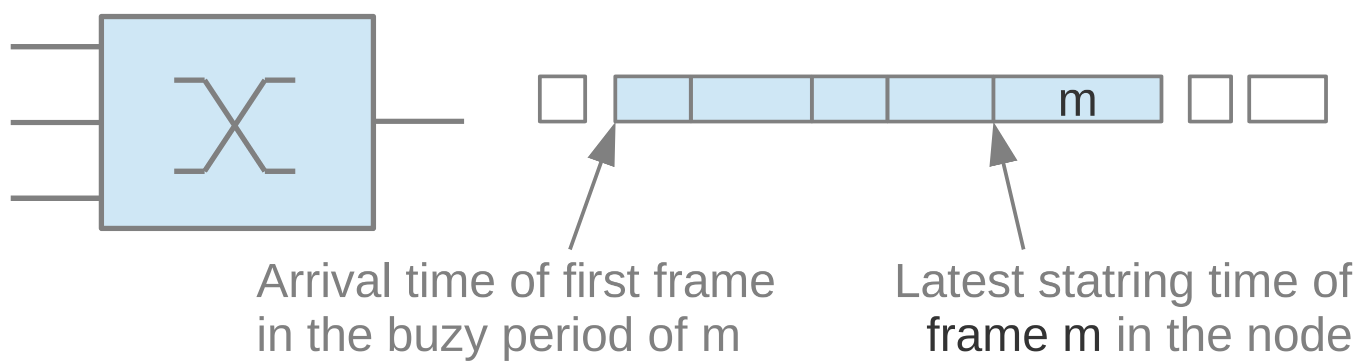
The Trajectory approach consists in:

- Defining a global equivalent node,
- Counting the occurrences of all the frames that can delay frame m on its path.



Considering:

- **Flows** going through the studied node,
- **Maximal jitter** of the incoming frames
- Maximum **traffic contract** of each flow (s_{max}, BAG)



In an output port h, for a given flow (to whom frame **m** belongs) the **maximum backlog** occurs when **m incurs maximum delay**

Worst case backlog computation

Frames belonging to the busy period

Superset

buzy period

=

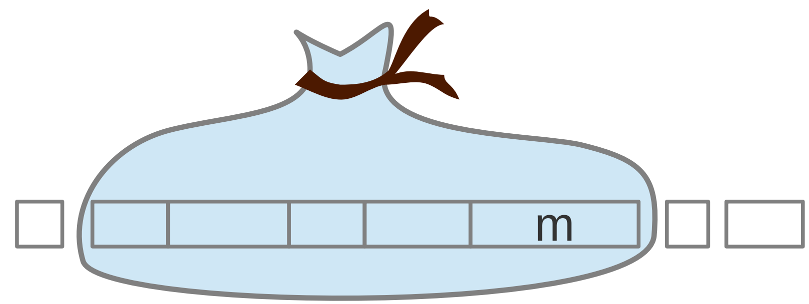
superset of all the frames generating backlog

Maximum jitter up to the studied node

$$\sum_{flows_j} \left(1 + \left\lfloor \frac{t + A_{i,j}}{BAG_j} \right\rfloor \right)^{+} s_{max j}$$

Flows meeting frame m up to the studied node

Traffic contract of flow j



Superset

Worst case backlog computation

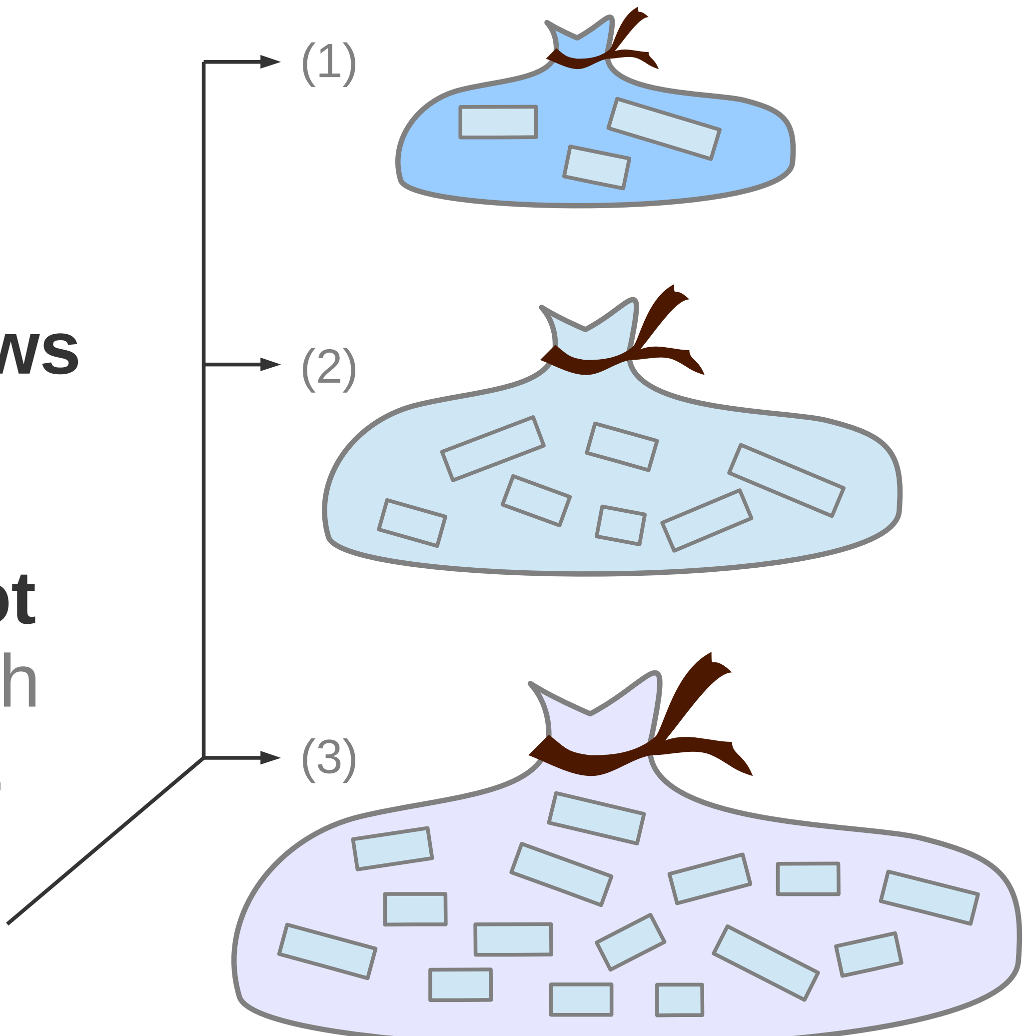
Frames belonging to the busy period

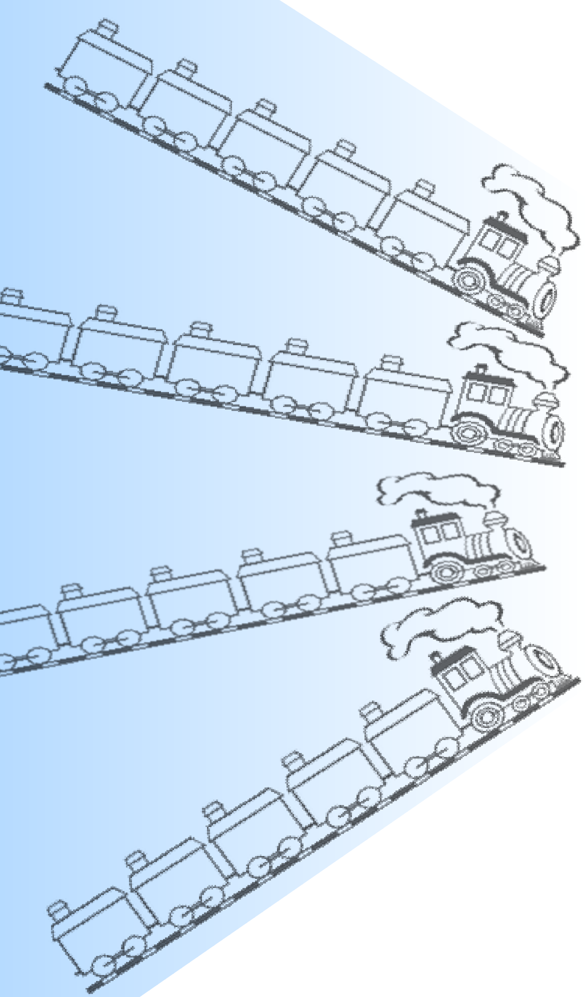
Superset

Major challenges:

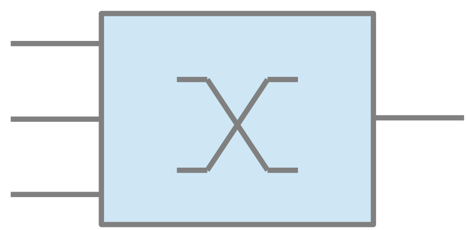
- With increasing loads, **several frames per flows** have to be counted;
- The worst case does **not necessarily** happen with **the first frame** in a row.

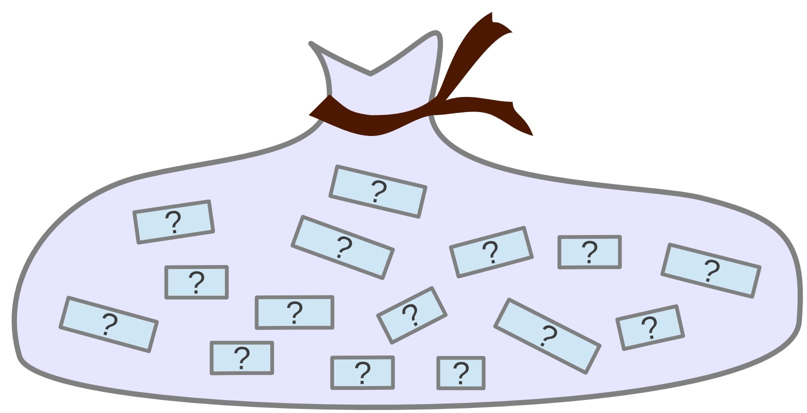
→ **iterative calculation**





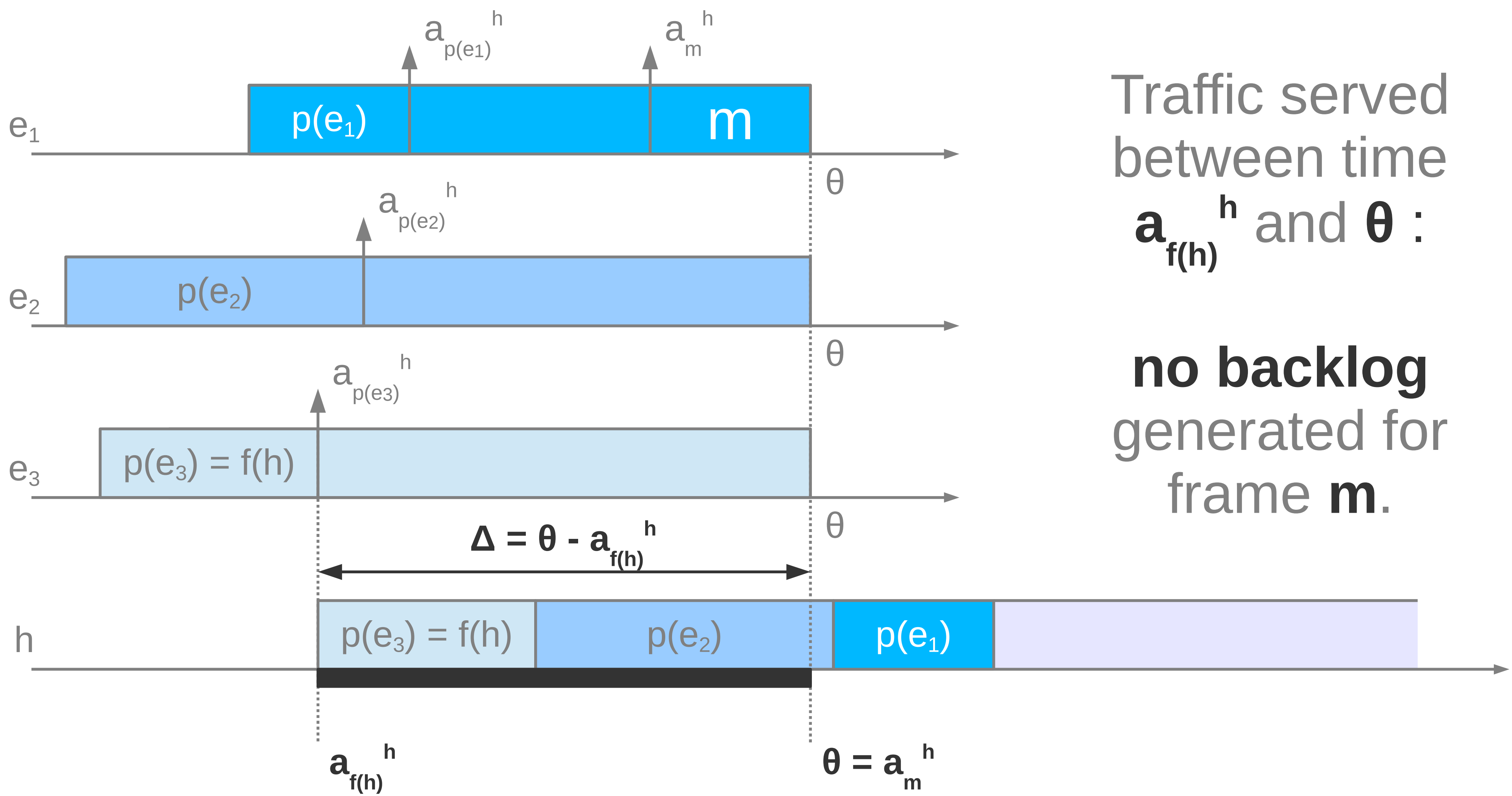
Frames coming from a previous node through the **same input link** are already **serialized**





Frames belonging to the same ”train” do not delay each other more than once

Which frames should be discarded?



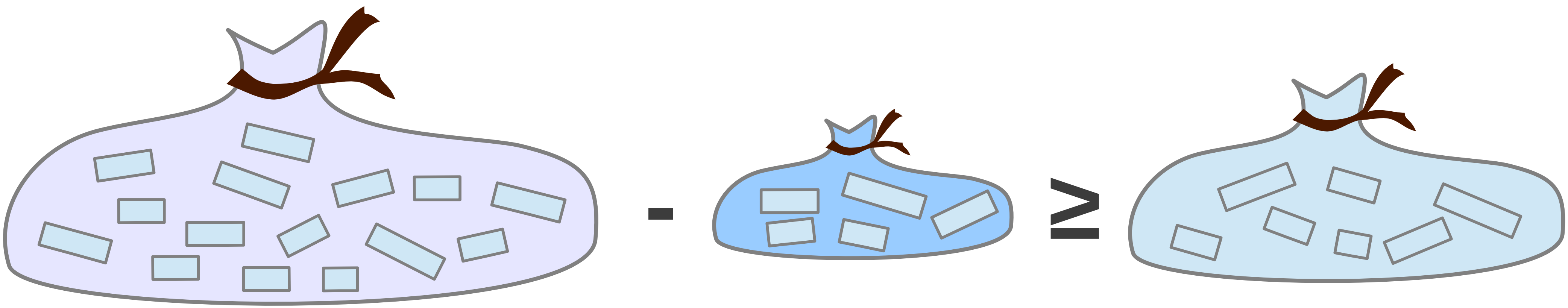
Traffic served between time $a_{f(h)}^h$ and θ :

no backlog generated for frame **m**.

Worst case backlog computation

Principle of the calculation

Inequality



Busy period of m in a given node

(Traj. approach)

Set of frames that can not generate backlog in the worst case

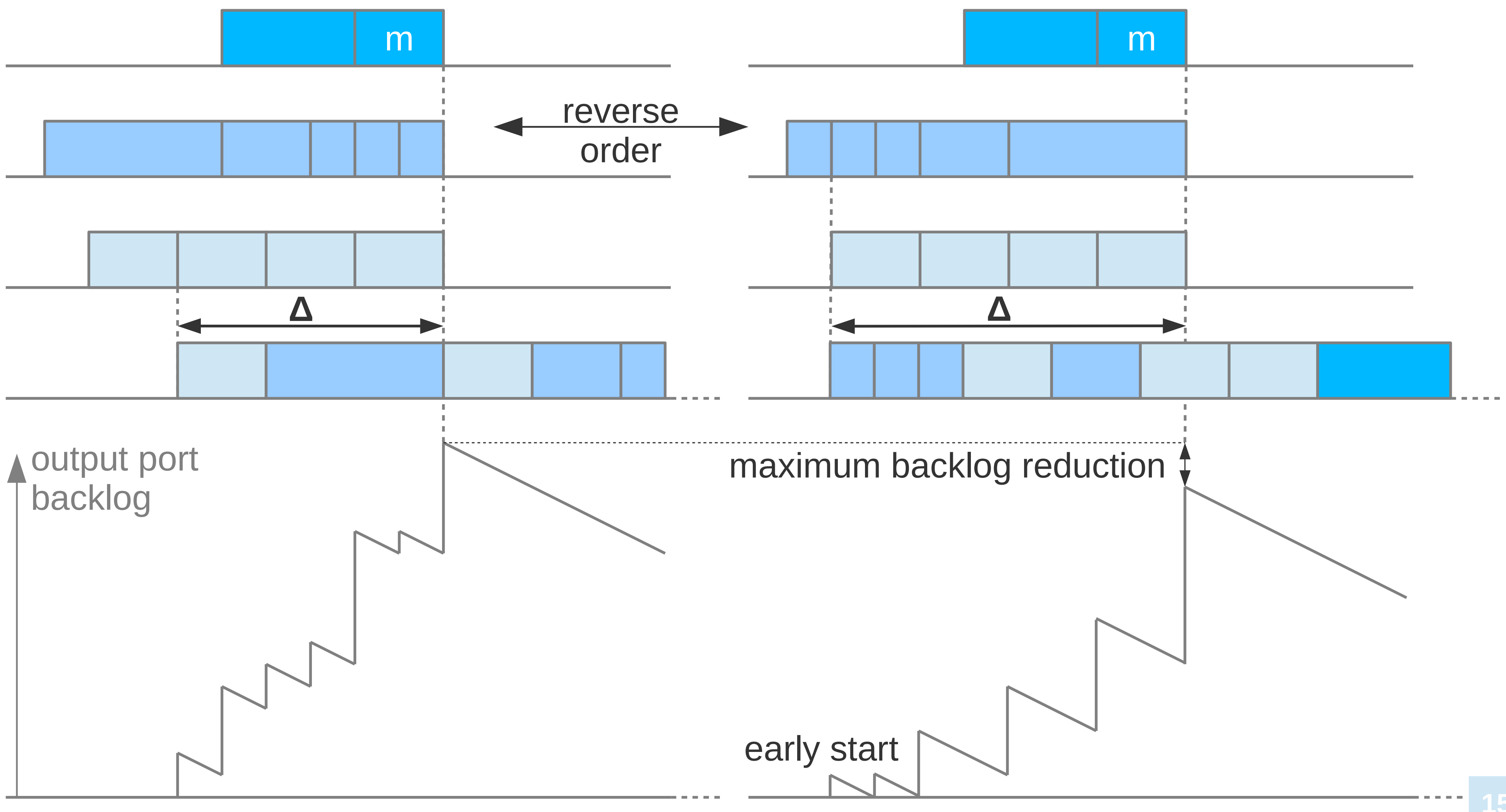
Worst case backlog for m in the node

Maximize this term, such as the inequality holds...

Worst case backlog computation

Worst case happens when frames are sorted by decreasing size

Frame order matters

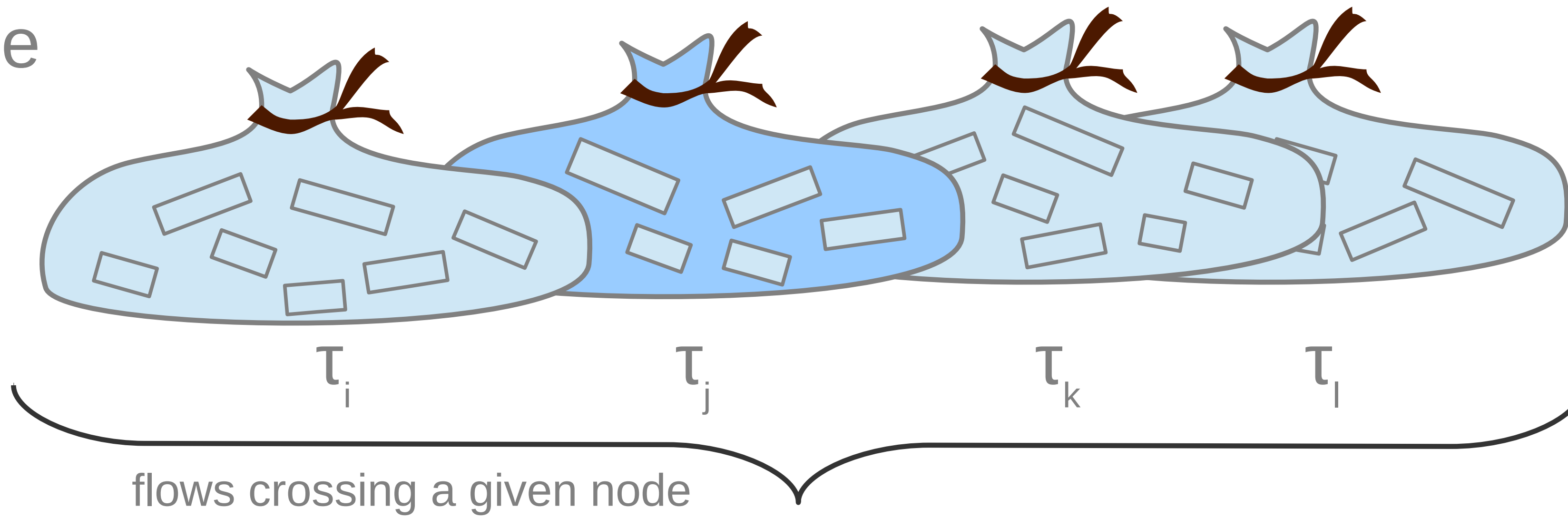


Worst case backlog computation

Worst case backlog in a node

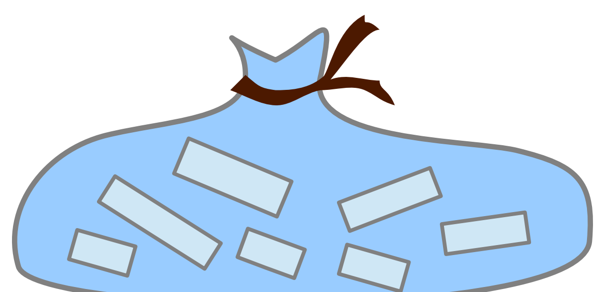
Output port

Worst case backlog for a frame of flow



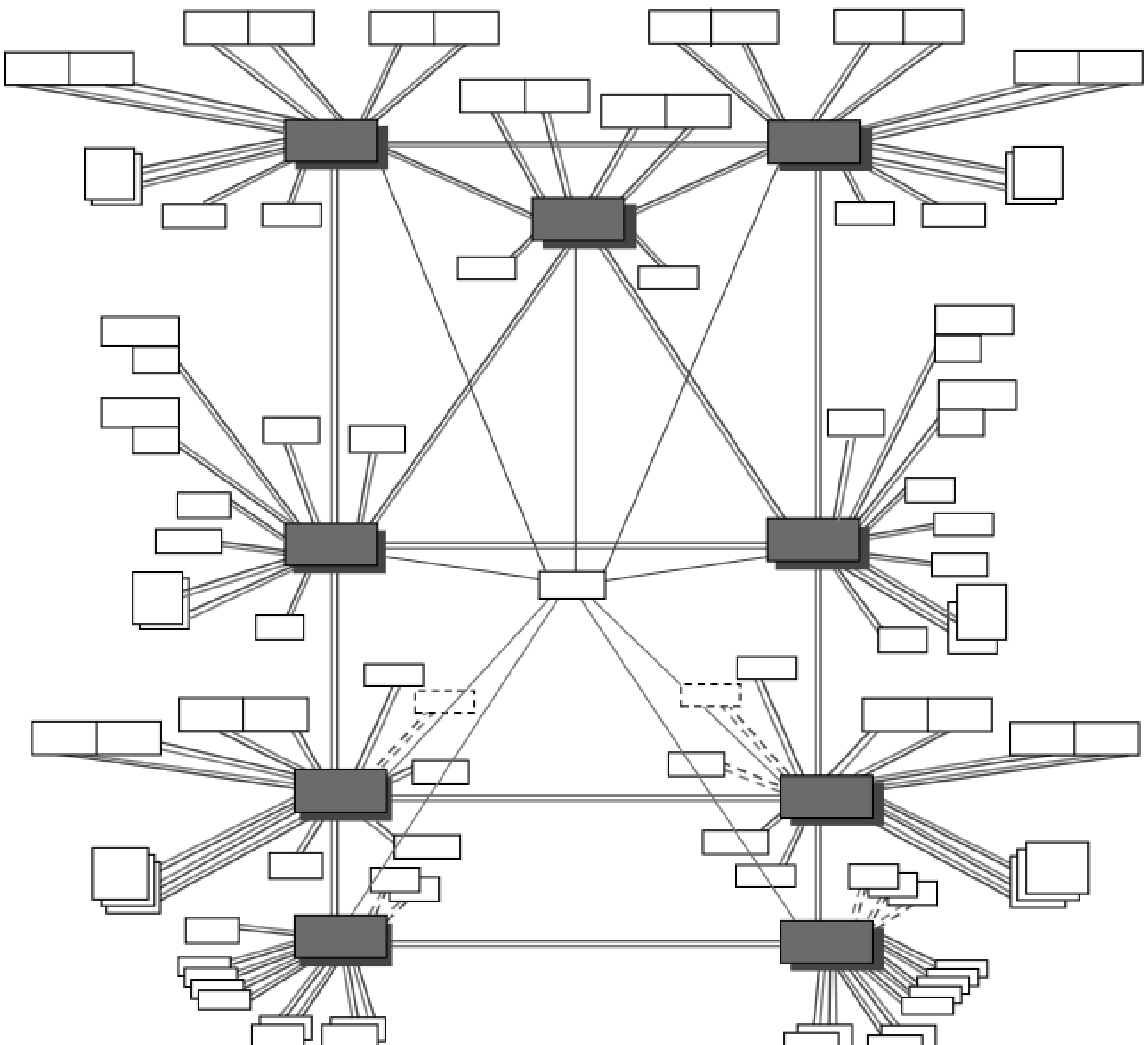
max

=



Worst case backlog in the node (AFDX switch output port)

AFDX avionics configuration



Some figures

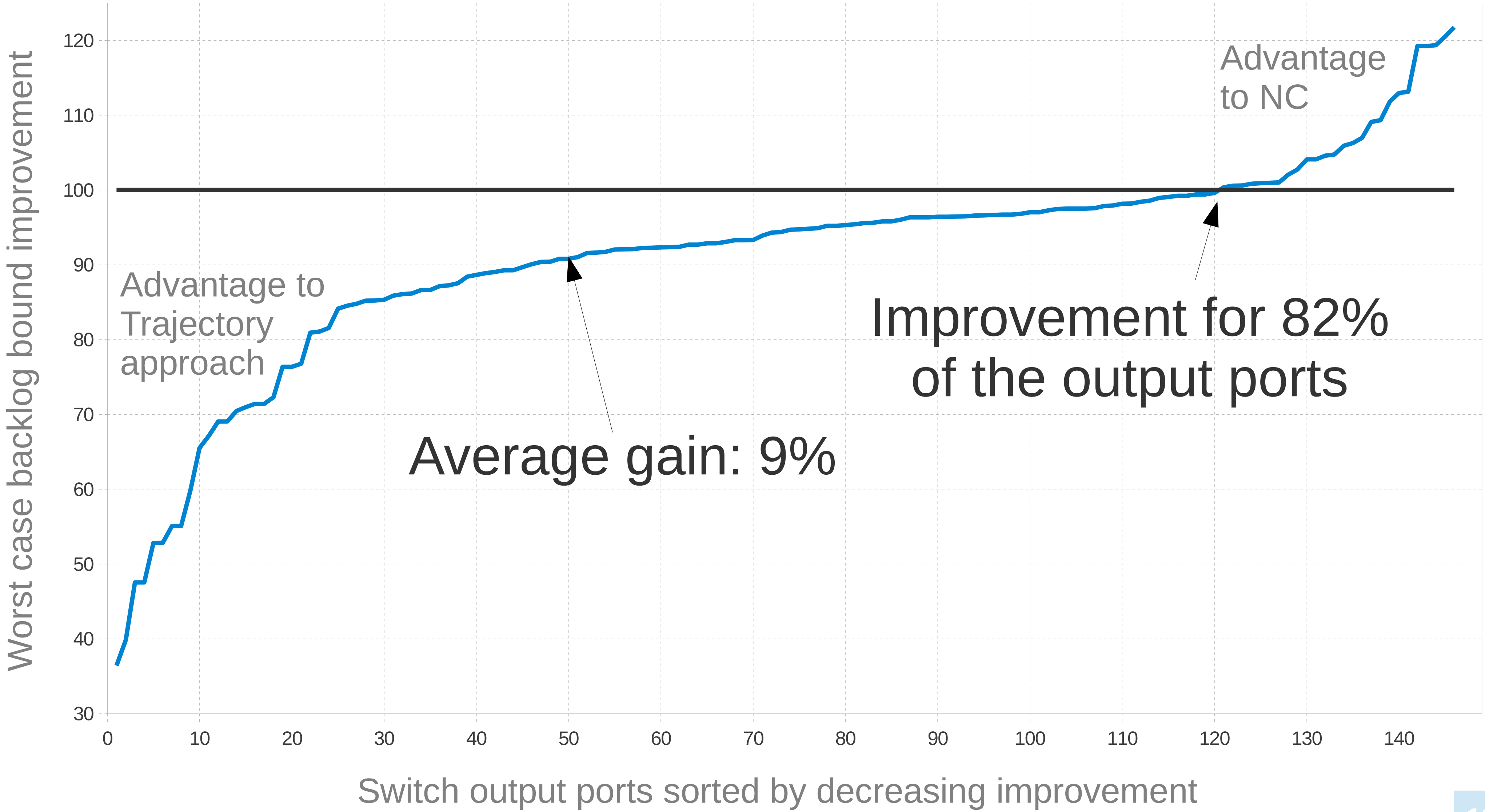
- 126 End Systems (ES)
- 18 switches (24 ports each)
- >1000 Virtual Links (VL)
- >15000 paths (due to multicast paths)

Comparison with Network Calculus

Backlog bound

On an industrial network

Normalized to 100 (NC)



→ Compute **worst case buffer occupancy** using the **Trajectory approach**

- OK for AFDX with FIFO servicing policy
- Could be extended to QoS-aware servicing policies
- Improve bound on busy period earliest starting time

→ Application to an **industrial AFDX configuration**

- Computation scales up to real case networks
- Improvement consistent with previous results on delay
- Reduction of switch maximal buffer size requirement

Thank you for your attention!

:-)

Questions?