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
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**
Bounded frame transmission time \((\text{CST} + \text{VAR})\) :

- Constant part  = technological latency

- Variable part  = output buffer occupancy 
  \(\rightarrow\) maximum backlog
Backlog and transmission delay

Frames to be transmitted

Shared com.
ressources

COM. CHANNEL
MULTIPLEXING

maximum
transmission time

APPLICATION
LEVEL

FUNCTION
SUPPLIER

HARDWARE
LEVEL

maximum backlog

SYSTEM
INTEGRATOR

a dual problem
End systems send frames through VLs

AFDX switch

AFDX End system

AFDX switch

Traffic Policing

Traffic Filtering

S_{max}

BAG

VL

S_{max}

FIFO servicing policy

AFDX
Computing worst case upper-bounds for backlog and delay

Network Calculus

Incoming frames | Network element output port | Delayed frames

Incoming frames | Network element output port | Delayed frames

ARRIVAL CURVE | SERVICE CURVE | (1) maximum virtual delay
(2) maximum virtual backlog

ARIVAL CURVE (leaky bucket)

SERVICE CURVE (rate-latency)
Existing approach

Proposed method

Network Calculus

Trajectory (vs. NC)

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

(Previous work)

Worst case end-to-end delay computation on an industrial configuration
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.
The Trajectory approach

Amount of frames delaying $m$ in a given node

Busy period

Considering:
- Flows going through the studied node,
- Maximal jitter of the incoming frames
- Maximum traffic contract of each flow ($s_{\text{max}}$, 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

\[ \text{busy period} = \text{superset of all the frames generating backlog} \]

\[ \sum_{\text{flows}_j} \left( 1 + \left[ \frac{t + A_{i,j}}{BAG_j} \right] \right)^+ S_{\text{max}_j} \]

Maximum jitter up to the studied node

Flows meeting frame m up to the studied node

Traffic contract of flow j
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**
Worst case backlog computation

Why is it a superset?

Impact of the serialization effect

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
Worst case backlog computation

Which frames should be discarded?

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

no backlog generated for frame $m$.
Worst case backlog computation

Principle of the calculation

Busy period of \( m \) in a given node (Traj. approach)

Set of frames that cannot 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

output port backlog

reverse order

maximum backlog reduction

early start

\[ m \]

\[ \Delta \]
Worst case backlog computation

Worst case backlog in a node

Output port

Worst case backlog for a frame of flow

\[ \tau_i, \tau_j, \tau_k, \tau_l \]

flows crossing a given node

\[ \max \]

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

On an industrial network

Normalized to 100 (NC)

Backlog bound

Worst case backlog bound improvement

Average gain: 9%

Improvement for 82% of the output ports

Advantage to NC

Advantage to Trajectory approach

Switch output ports sorted by decreasing improvement

Average gain: 9%
→ 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?

:-)