Real-time communications over Switched Ethernet supporting dynamic QoS management

Ricardo Marau

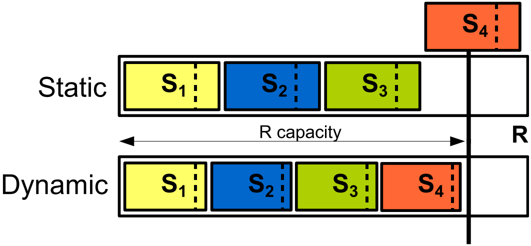
DETI/IEETA, Universidade de Aveiro, Portugal
marau@ua.pt

HaRTES - kickoff meeting

- Trend to support dQoS within the resource management
  - [Artist Design NoE technical annex]

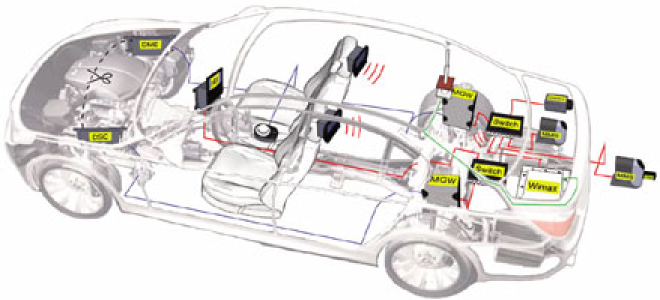
DQOS PROVIDES:

- Integration of multi-scoped applications (different QoS demands),
  - Video, control, multimedia, ...
- Dynamic adaption of the resource requirements;
- Optimal resource utilization - maximizes throughput.



Communication across automotive subsystems

- Powertrain & Safety,
- Comfort & Vehicle Controls,
- Driver Assistance,
- Infotainment & Communications.



dQoS e.g.: In-vehicle communication scenario

Multi-disciplinary scenario with actions crossing different car subsystems.
Multitude of sensors/actuators:

- Cameras (high-speed, rear-view, IR, side mirror), compass/gyro, temperature, wheel speed, sonar, ...
- dashboards, rear-view mirror, brakes, buzzers, fuel injection, steer,...

integrated in services such as:

- Park/Reverse Assist, Night vision, Lane Departure Warning, Adaptive Cruise Control, Collision Warning,...

Why dynamic QoS management?

Services are continuously turned on/off or adapted on demand by changes in the environment.

Requirements for dQoS management

An application model that:

- supports the spontaneous environment changes.
- provides a clear relationship between the metric for QoS and the resource parameters (e.g. bandwidth).

A resource (e.g. network) framework that supports:

- Reconfigurability
- Adaptability

Flexibility

Requirements for dQoS management

An application model that:

- supports the spontaneous environment changes.
- provides a clear relationship between the metric for QoS and the resource parameters (e.g. bandwidth).

A resource (e.g. network) framework that supports:

- Reconfigurability
- Adaptability

Flexibility

Switched Ethernet, on the go for

Recalling the in-vehicle communication example

BMW is making efforts to bring Ethernet (switched/bus topology) as a backbone for RT communications in vehicles.

Ethernet provides:

- Hi-bandwidth
- Cheap components
- Seamless integration
- Robustness

However

It still provides poor determinism for RT applications:

- non-determinism CSMA/CD arbitration for bus topologies,
- FIFO queues in the switch (no priority enforcement) and
- possible memory overflow in the switch.

RT frameworks for Ethernet/Switched-Ethernet

- Ethereal, Profinet: non-COTS
- EtherCat, EPL: no forwarding parallelization (switch=hub)
- Traffic shaping solutions: distributed, hard to adapt consistently and promptly.

Hard real-time and flexibility?

FTT (Flexible Time Triggered) paradigm

RT frameworks for Ethernet/Switched-Ethernet

- Ethereal, Profinet: non-COTS
- EtherCat, EPL: no forwarding parallelization (switch=hub)
- Traffic shaping solutions: distributed, hard to adapt consistently and promptly.

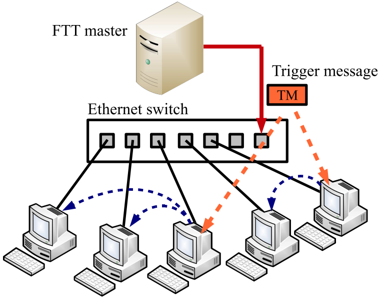
Hard real-time and flexibility?

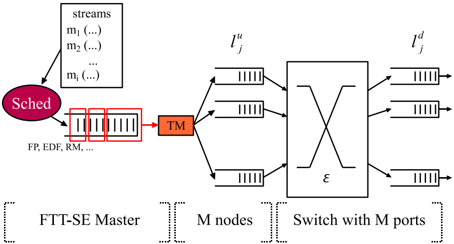
FTT (Flexible Time Triggered) paradigm

FTT-SE (FTT over switched Ethernet)

Features

- Hard real-time, controlling the load submitted to the switch
- Periodic and Aperiodic traffic (efficiently)
- Supports unicast, multicast and broadcast traffic
- Centralized Scheduling
  - Master-slave (or multi-slave)
  - Supports arbitrary traffic scheduling policies

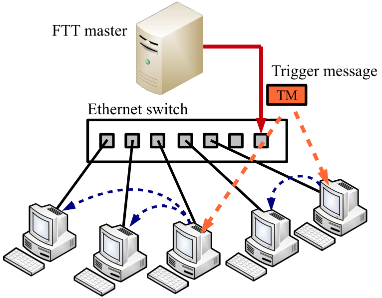


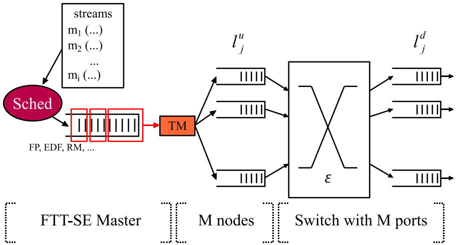


FTT-SE (FTT over switched Ethernet)

Features

- Hard real-time, controlling the load submitted to the switch
- Periodic and Aperiodic traffic (efficiently)
- Supports unicast, multicast and broadcast traffic
- Centralized Scheduling
    - Master-slave (or multi-slave)
    - Supports arbitrary traffic scheduling policies



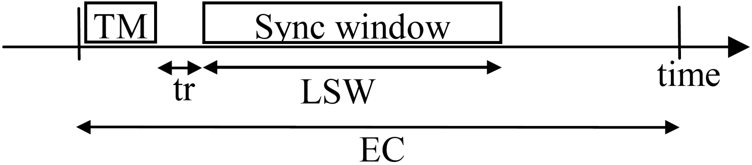


Periodic scheduling model

- Set of periodic streams

$$\Omega = \left\{ \sigma_i : \sigma_i \left(C_i, D_i, T_i, O_i, S_i, \left\{ R_i^1 .. R_i^{ki} \right\} \right), i = 1..N \right\}$$

- Strictly confined to the Synchronous Window per EC
- Scheduling with multiple queues

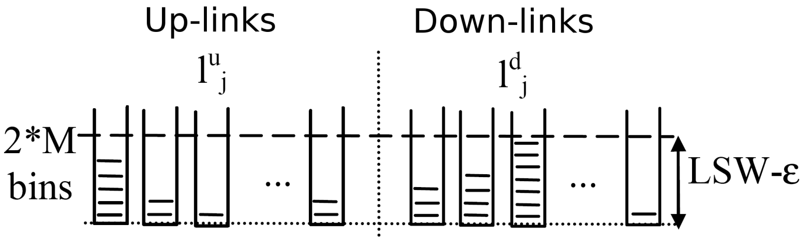


Periodic scheduling model

- Set of periodic streams

$$\Omega = \left\{ \sigma_i : \sigma_i \left(C_i, D_i, T_i, O_i, S_i, \left\{ R_i^1 .. R_i^{ki} \right\} \right), i = 1..N \right\}$$

- Strictly confined to the Synchronous Window per EC
- Scheduling with multiple queues

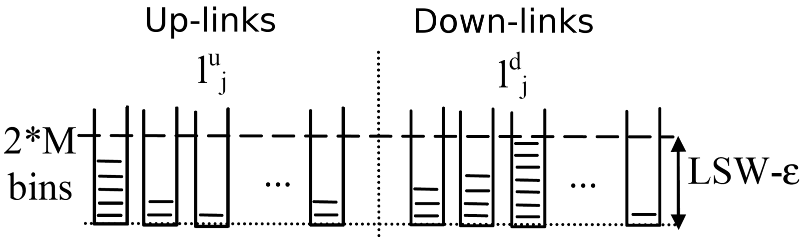


Scheduling equation

$$\begin{cases} \max_j \left(\sum_{SM_i \in l_j^u} C_i \right) \leq LSW - \varepsilon & \text{UPlinks} \\ \max_j \left(\max_{SM_i \in l_j^d} (f_i) \right) \leq LSW & \text{DOWNlinks} \end{cases}$$

Memory bounds

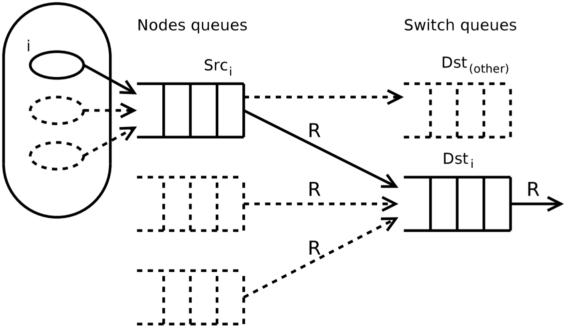
$$\max_{j=1..M} \left(\mu_j^n, \mu_j^p \right) < (LSW - \varepsilon) * r/8$$



Current analysis

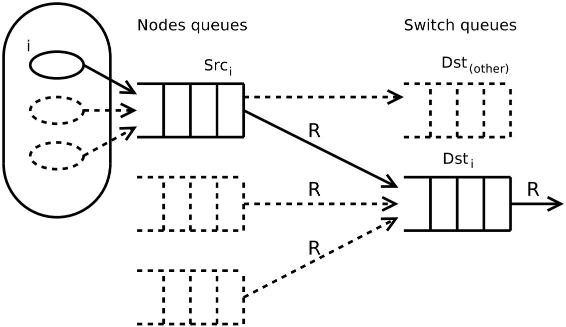
- Network Calculus
- Response-time

are too complex to execute on-line and do not provide a feasibility bound to help distributing the resource capacity.



...AND SO WE NEED

Utilization-based analysis



FTT-SE: Schedulability analysis (ctd.)

Uplinks

- Periodic traffic interference
  - traffic from within the same sending node.

$$\forall_{j\ uplinks}, \sum_{m_i \in I_j^u} \frac{C_i}{T_i} \leq U^{lub}_{RM,EDF} \times IITfactor$$

Downlinks

- Traffic aggregation from several uplinks
- Traffic no longer strictly periodic
  - Interference at the uplinks from messages going elsewhere.
  - Release jitter

$$\forall_{j\ downlinks}, \sum_{m_i \in I_j^d} \frac{C_i}{T_i} + \frac{\max_{m_i \in I_j^d} J_i}{T_1} \leq U^{lub}_{RM,EDF} \times IITfactor$$

Recalling the requirements for dQoS management

FTT-SE network framework

- Online Reconfigurability and Adaptability
  - Flexible framework,
  - Admission control (utilization-based),
  - An effective centralized approach (FTT-master):
    - Scheduling,
    - Aggregation of all traffic requirements.

The application model

- QoS requirements for a service i include:
  - Operational constraints: $\{(C_i^1, T_i^1), \ldots, (C_i^n, T_i^n)\}$
  - Differentiation parameter: qos_i

THE DQOS MANAGER

- Map the service constraints into resource utilization: $\{(C_i^n, T_i^n), \ldots\} \mapsto [Umin_i, Umax_i]$

- Capacity(BW) distribution: $\{(Umin_i, Umax_i, qos_i), \forall_i\} \mapsto \{BW_i, \forall_i\}$

- Re-map into operational parameters: $\{BW_i, \forall_i\} \mapsto \{(Cf_i, Tf_i), \forall_i\}$

WITHIN FTT-SE

- Multi-resource distribution problem:
  - (n uplinks + n downlinks), each a semi-independent resource.
  - Services use multiple resources (distribution-related)

REMARK

The admission control guarantees the minimum performance level.
The QoS manager improves that level.

The dQoS manager

- Map the service constraints into resource utilization: $\{(C_i^n, T_i^n), \ldots\} \mapsto [Umin_i, Umax_i]$

- Capacity(BW) distribution: $\{(Umin_i, Umax_i, qos_i), \forall_i\} \mapsto \{BW_i, \forall_i\}$

- Re-map into operational parameters: $\{BW_i, \forall_i\} \mapsto \{(Cf_i, Tf_i), \forall_i\}$

within FTT-SE

- Multi-resource distribution problem:
  - (n uplinks + n downlinks), each a semi-independent resource.
  - Services use multiple resources (distribution-related)

Remark

The admission control guarantees the minimum performance level. The QoS manager improves that level.

THE DQOS MANAGER

- Map the service constraints into resource utilization: $\{(C_i^n, T_i^n), \ldots\} \mapsto [Umin_i, Umax_i]$

- Capacity(BW) distribution: $\{(Umin_i, Umax_i, qos_i), \forall_i\} \mapsto \{BW_i, \forall_i\}$

- Re-map into operational parameters: $\{BW_i, \forall_i\} \mapsto \{(Cf_i, Tf_i), \forall_i\}$

WITHIN FTT-SE

- Multi-resource distribution problem:
  - (n uplinks + n downlinks), each a semi-independent resource.
  - Services use multiple resources (distribution-related)

REMARK

The admission control guarantees the minimum performance level.
The QoS manager improves that level.

THE DQOS MANAGER

- Map the service constraints into resource utilization: $\{(C_i^n, T_i^n), \dots\} \mapsto [Umin_i, Umax_i]$

- Capacity(BW) distribution: $\{(Umin_i, Umax_i, qos_i), \forall_i\} \mapsto \{BW_i, \forall_i\}$

- Re-map into operational parameters: $\{BW_i, \forall_i\} \mapsto \{(Cf_i, Tf_i), \forall_i\}$

WITHIN FTT-SE

- Multi-resource distribution problem:
  - (n uplinks + n downlinks), each a semi-independent resource.
  - Services use multiple resources (distribution-related)

REMARK

The admission control guarantees the minimum performance level.
The QoS manager improves that level.

THE DQOS MANAGER

- Map the service constraints into resource utilization: $\{(C_i^n, T_i^n), \ldots\} \mapsto [Umin_i, Umax_i]$

- Capacity(BW) distribution: $\{(Umin_i, Umax_i, qos_i), \forall_i\} \mapsto \{BW_i, \forall_i\}$

- Re-map into operational parameters: $\{BW_i, \forall_i\} \mapsto \{(Cf_i, Tf_i), \forall_i\}$

WITHIN FTT-SE

- Multi-resource distribution problem:
  - (n uplinks + n downlinks), each a semi-independent resource.
  - Services use multiple resources (distribution-related)

REMARK

The admission control guarantees the minimum performance level.
The QoS manager improves that level.

Conclusion

Achieved...

- A Hard RT framework with flexibility requirements supporting online Reconfigurability and Adaptability.

Which provides...

- Seamless integration for a dQoS manager
  - where several distribution policies may apply.

On-going and future work

- Server-based scheduling,
- Hierarchical composition,
- FTT-SE master redundancy,
- Synchronization between FTT-SE networks,
  - scalability and composability.

CONCLUSION

Thank you.

ON-GOING AND FUTURE WORK

- Server-based scheduling,
- Hierarchical composition,
- FTT-SE master redundancy,
- Synchronization between FTT-SE networks,
  - scalability and composability.