REAL-TIME COMMUNICATIONS OVER SWITCHED ETHERNET SUPPORTING DYNAMIC QoS MANAGEMENT

Ricardo Marau

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

HaRTES - kickoff meeting
**Dynamic QoS management - motivation for**

- 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.
DQoS e.g.: In-vehicle communication scenario

Communication across automotive subsystems

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

Diagram:
- FTT master
- Ethernet switch
- Trigger message
- Streams: \( m_1(\ldots) \), \( m_2(\ldots) \), ... , \( m_i(\ldots) \)
- Sched
- \( I^u_j \) and \( I^d_j \)
**PERIODIC SCHEDULING MODEL**

- Set of periodic streams
  \[ \Omega = \left\{ \sigma_i : \sigma_i \left( C_i, D_i, T_i, O_i, S_i, \{ R_{1i}^{i..} R_{ki}^{i} \} \right), i = 1..N \right\} \]

- Strictly confined to the Synchronous Window per EC

- Scheduling with multiple queues

---

![Diagram](image-url)
**PERIODIC SCHEDULING MODEL**

- Set of periodic streams

\[ \Omega = \{ \sigma_i : \sigma_i \left( C_i, D_i, T_i, O_i, S_i, \{ R^1_i \ldots R^{ki}_i \} \right), i = 1 \ldots N \} \]

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

\[
\begin{align*}
\max_j \left( \sum_{SM_i \in I_j^u} C_i \right) & \leq LSW - \varepsilon \quad \text{UPlinks} \\
\max_j \left( \max_{SM_i \in I_j^d} (f_i) \right) & \leq LSW \quad \text{DOWNlinks}
\end{align*}
\]

**Memory Bounds**

\[
\max_{j=1..M} \left( \mu_j^u, \mu_j^p \right) < (LSW - \varepsilon) \times \frac{r}{8}
\]
**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
**UPLINKS**

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

\[
\forall_j \text{ uplinks}, \sum_{m_i \in I_j^u} \frac{C_i}{T_i} \leq U_{RM,EDF}^{lub} \times IIT_{factor}
\]

**DOWNLINKS**

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

\[
\forall_j \text{ downlinks}, \sum_{m_i \in I_j^d} \frac{C_i}{T_i} + \frac{\max_{m_i \in I_j^d} J_i}{T_1} \leq U_{RM,EDF}^{lub} \times IIT_{factor}
\]
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: $\text{qos}_i$
The dQoS manager

- Map the service constraints into resource utilization:
  \[\{(C^n_i, T^n_i), \ldots\} \mapsto [U_{\min i}, U_{\max i}]\]

- Capacity (BW) distribution:
  \[\{(U_{\min i}, U_{\max i}, \text{qos}_i), \forall i\} \mapsto \{BW_i, \forall i\}\]

- Re-map into operational parameters:
  \[\{BW_i, \forall i\} \mapsto \{(C_{fi}, T_{fi}), \forall i\}\]

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:
  \(\{(C^n_i, T^n_i), \ldots\} \mapsto [U_{min_i}, U_{max_i}]\)

- Capacity (BW) distribution:
  \(\{(U_{min_i}, U_{max_i}, qos_i), \forall i\} \mapsto \{BW_i, \forall i\}\)

- 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:
  \[\{(C^n_i, T^n_i), \ldots\} \mapsto [Umin_i, Umax_i]\]

- 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:
  \[ \{(C_i^n, T_i^n), \ldots \} \rightarrow [U_{\text{min}}_i, U_{\text{max}}_i] \]

- Capacity (BW) distribution:
  \[ \{(U_{\text{min}}_i, U_{\text{max}}_i, \text{qos}_i), \forall i \} \rightarrow \{\text{BW}_i, \forall i\} \]

- Re-map into operational parameters:
  \[ \{\text{BW}_i, \forall i\} \rightarrow \{(C_{f_i}, T_{f_i}), \forall i\} \]

WITHIN FTT-SE

- Multi-resource distribution problem:
  - \((n \text{ uplinks} + n \text{ downlinks}), \) 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:
  \[ \{(C^n_i, T^n_i), \ldots \} \mapsto [U_{min_i}, U_{max_i}] \]

- Capacity (BW) distribution:
  \[ \{(U_{min_i}, U_{max_i}, qos_i), \forall i \} \mapsto \{BW_i, \forall i\} \]

- 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.