Automotive Ethernet – Opportunities and Pitfalls

Rolf Ernst
Institut für Datentechnik und Kommunikationsnetze
ETFA, Berlin, Sep 7, 2016

Overview

- automotive networks – the Ethernet promise
- Ethernet as a backbone - a closer look
- Ethernet – the safety perspective
- conclusion
Overview

- automotive networks – the Ethernet promise
- Ethernet as a backbone - a closer look
- Ethernet – the safety perspective
- conclusion

Automotive networks – The trends

- Trend 1: New infotainment applications
  - networks with IP traffic via car-to-X communication
  - primarily best effort
- Trend 2: Quickly growing sensor traffic
  - high resolution redundant image sensors for autonomous driving
  - high bandwidth communication using switched high speed network
  - limited network latency (system response times)
- Trend 3: Complex low latency traffic
  - backbone function: legacy, future drives, highly interactive functions, ...
  - low to medium volume, low latency traffic
- the idea: Use Switched Ethernet!
The Ethernet promise

- bandwidth, bandwidth, bandwidth
  - for novel functions with high data rates, such as ADAS or infotainment
  - speed growing with technology 100Mb/s – 1Gb/s -10Gb/s - ...
- open network capabilities
  - open automotive networks towards IP protocol with approved technology
- shared technology cost
  - standard with high volume across industries
  - no headaches with next generation MOST, FlexRay, ...
  - huge engineering platform experience
    - avionics, industry, ...

*but: how to efficiently design with Ethernet?*

*how to reach the required safety?*

---

Where we come from: Bus-based communication

- straightforward support of publisher-subscriber mechanism
- several application specific standards, CAN, FlexRay, LIN, …
  - < 100kbit … 10Mbit (FlexRay, CAN FD) data rate
- predictable scheduling: fixed priority or TDMA or slotted ring (MOST)
- routing by dedicated gateway (GW)
  - low speed allows SW implementation

![Diagram of bus-based communication](image)
Ethernet is different

- **switched network** instead of bus
  - point-to-point connections with dynamic address handling
  - different scheduling mechanisms, flow control
  - *note: original Ethernet bus technology not suitable*
- different communication schemes
  - unicast, multicast, broadcast
  - different identifier assignment
  - not primarily developed for time-critical communication
- complex **multi-level protocol alternatives**
  - many configuration parameters
  - higher overhead than CAN
- *consequence for network properties and design?*

Switched Network – High Flexibility

- all traffic through bidirectional links – *fastest electrical solution*
- all arbitration in the switches – *flexible scheduling*
- arbitrary network topology – *adaptable performance & redundancy*
Bidirectional links for high speed

- automotive: OPEN alliance (BroadR Reach)
  - simple 2-wire physical medium – low cost
  - no link access arbitration necessary!

Adaptable traffic patterns

- variable frame sizes
  - 84 bytes (e.g. control messages) → 1500 bytes (e.g. camera frames)
  - resulting link latencies (non preemptable frames)
    - 100Mbit/s: short frames 6.72 µs → long frames 122 µs
- switch scheduling
  - WRR, static priority, time triggered, different shapers
  - TSN – many additional mechanisms
Ethernet IEEE 802.1Q – Standardization

- **Standard Ethernet (IEEE 802.1Q)**
  - priority based
  - up to 8 priorities and 4096 VLANs
  - static priority scheduling

- **Ethernet AVB (IEEE 802.1Qav)**
  - originally defined for streaming applications
  - adds standardized traffic shaping to IEEE 802.1Q
  - 802.1AS: clock synchronization

- **Time-Sensitive Networking – TSN**
  - set of (draft) Ethernet standards addressing real-time requirements

TSN Arbitration and Shaping

- **frame preemption** (IEEE 802.1Qbu)
  - reduce blocking time by lower-priority frames
  - allow preemption of lower-priority frames (at certain points)

- **ingress filtering** (IEEE 802.1Qci)
  - ensure that traffic streams stay within predefined bounds (fault containment)

- **timing and synchronization** (IEEE 802.1ASbt)
  - extensions to 802.1AS: redundancy, multiple time domains

- **time triggering** (IEEE 802.1Qbv)
  - time aware shaper for low latency, time sensitive traffic

- **worst case timing analysis available for most standardized features**
  - (pyCPA, SymTA/S) – see talk by Thiele et al. session T3.1
Higher-layer protocols – Network and Transport

- **IP (layer 3)**
  - IP adds routing support and compatibility
- **UDP (layer 4)**
  - adds software ports on top of IP - connectionless protocol
- **TCP (layer 4)**
  - adds software ports on top of IP - connection oriented, hand shake and flow control

Software stacks and Run-time Environment (RTE)

- **SOME/IP**
  - middleware standardizing data encapsulation in TCP or UDP packets
  - service discovery
- **AUTOSAR Ethernet socked adapter**
  - AUTOSAR: Automotive software standard
  - software adapter to embed Ethernet in AUTOSAR COM stack
  - achieves compatibility to other communication standards
  - more complex than in conventional buses
AUTOSAR Ethernet socket adapter

- AUTOSAR PDU router (Protocol Data Unit)
- Service discovery (Sd)
- Diagnostics over IP (DoIP)

Ethernet socket adapter

- TCP/UDP/IP stack
- communication HW abstraction
- communication drivers

Design complexity and cost – a caveat

- Ethernet: many more parameters and variants than in current networks
  - MAC address management, switch management, protocol selection, packaging, ...
- current standardization addresses compatibility to existing architectures and standards
  - does not limit variety
  - variety easily leads to incompatibilities and inefficiencies – cost!
Overview

- automotive networks – the Ethernet promise
- Ethernet as a backbone - a closer look
- Ethernet – the safety perspective
- conclusion

The Ethernet backbone idea
Evaluation: Design example

- use case
  - traffic pattern according to published BMW use-case [Lim2011]
- ECUs
  - 4 control units
    - each ~72 kbit/s → streamed to Head Unit
  - 4 cameras (Rear, Sides, Front)
    - each 26 Mbit/s (compressed) → streamed to Head Unit
- rear seat entertainment
  - audio: 1.43 Mbit/s
  - DVD video: 12 Mbit/s
  - Internet data / bulk traffic: 11.52 Mbit/s

AVB experiments: Tree topology

- static priority w. shaping
  - latency critical traffic mapped to Class A disabled shaper
  - bandwidth critical traffic mapped to Class B shaper 10% overreservation
  - bulk/internet traffic: lowest priority
- WC response time analysis using SymTA/S (pyCPA)
Larger system [Thiele14] – Results for control

- **correlation has strong influence for control messages**
  - period > latency [5ms ..1s]
  - not considered (NO)
  - stream correlation considered

→ control streams are NOT pipelined in any topology
→ no overwriting in Ingress transmit buffers

Experiments – Results for camera messages

- **frame period < latency** [0.1ms ..1ms]

→ camera stream packets are pipelined
→ buffers for frames in switches needed!
### Topology and bandwidth effects - Conclusion

- high priority control traffic (class A)
  - no risk of message overwriting in transmit buffers
  - if sufficient switch buffer available and message density low
  - holds for all investigated topologies in example
- high priority camera traffic (class B)
  - risk of overwriting in transmit buffer – stream buffering needed
  - holds for all investigated topologies
- *Ethernet AVB scheduling appears sufficient*
  - *no shaper for class A*

---

### Overview

- automotive networks – the Ethernet promise
- Ethernet as a backbone - a closer look
- Ethernet – the safety perspective
- conclusion
Current automotive network – protection

- application of safety standard ISO26262 affects large part of the system
  - „freedom from interference!“
- isolation on mixed critical buses required

\[ \text{ECU}_4 \quad \text{CAN}_1 \quad \text{ECU}_1 \\
\text{ECU}_5 \quad \text{ECU}_2 \\
\text{ECU}_6 \quad \text{ECU}_3 \]

\[ \text{ECU}_7 \quad \text{CAN}_3 \quad \text{ECU}_8 \]

- use priorities to separate criticalities
  - „criticality as a priority“
- allow „occasional“ loss of non (time) critical frames due to overload
  - „less than worst-case design“ possible

\( \rightarrow \text{mature solution to address ISO 26262 requirements} \)

- comparable solutions for FlexRay (TDMA) and MOST (reservation)
Ethernet – The safety perspective

- isolation
  - how well does Ethernet isolate critical from other traffic?
    - „freedom from interference“
- delivery under transmission errors
  - what timing guarantees are possible under errors?
- fail operational
  - how well can network failures be compensated?

---

Ethernet generally supports similar techniques

- priority assignment according to traffic class
  - combine with shaping where needed (AVB or TSN)
  - supports combination of design styles

<table>
<thead>
<tr>
<th>Class</th>
<th>Time Critical</th>
<th>Non-Time Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>time critical</td>
<td>general traffic</td>
</tr>
<tr>
<td>Class B</td>
<td>less time critical</td>
<td>camera traffic</td>
</tr>
<tr>
<td>other</td>
<td></td>
<td>other</td>
</tr>
</tbody>
</table>

- other techniques: TSN time triggering (cp. FlexRay)

*but: is the isolation effective?*
Isolation – The switch matters

combined unit

switched network

terminal node

switch

link

terminal node

switch

network interface

terminal node

Ethernet switch structure

- configure switch
- assign frame buffer memory
- update forwarding table

- parse the packets
- lookup output ports
- send to output queue

memory

program & data

forwarding table

queueing buffers

switch management

packet handling

switch fabric

switch input links

switch output links

Ingress stage

store & forward

Egress stage

link arbitration
Switching interference challenges

- **forwarding table**
  - limited index space leads to indexing conflicts
    - loss of timing, *interference*
  - requires appropriate MAC address management

- **queueing buffers**
  - limited buffer space
    - message drop, *interference*
  - flow control
    - same priority blocking, increased delay & buffer
  - few queues - few priorities
    - head-of-line-blocking, *interference*
  - queueing effects require system level end-to-end analysis

---

**Ethernet switch – address mapping**

- forwarding resolved via indexing

```plaintext
destination address space
(globally assigned)

index calculation (hashing)

index value space
table lookup

forwarding table entries

forwarding port (map)
```

- forwarding table
- program & data
- queueing buffers
- scheduler

---

7.9.2016 | R. Ernst, “Automotive Ethernet – Opportunities and Pitfalls” | Page 31

---

7.9.2016 | R. Ernst, “Automotive Ethernet – Opportunities and Pitfalls” | Page 32
Ethernet switch – address mapping

- forwarding resolved via indexing

<table>
<thead>
<tr>
<th>destination address space (globally assigned)</th>
</tr>
</thead>
<tbody>
<tr>
<td>index calculation (hashing)</td>
</tr>
<tr>
<td>index value space</td>
</tr>
<tr>
<td>table lookup</td>
</tr>
<tr>
<td>forwarding table entries</td>
</tr>
<tr>
<td>conflict</td>
</tr>
<tr>
<td>when all table entries for index are full</td>
</tr>
<tr>
<td>forwarding port (map)</td>
</tr>
</tbody>
</table>

destination address

\[ \geq 2^{48} \]

\[ \sim 2^7 - 2^{10} \]

alternative table entries (set associative)

switch architecture and indexing usually not published

Isolation in a backbone – The **gateway** influence

**Body**

CAN (-FD)

- ECU
- ECU

**Advanced Driver Assistance**

- CAM
- CAM

**Switched Ethernet Network**

**Powertrain** CAN (-FD)

- ECU
- ECU

- Domain Gateway

**Infotainment**

- FlexRay

- ECU
- ECU

- ECU
- ECU
CAN frame packaging scenario - Backbone

- complex protocol choices
  - SOME/IP – UDP – IP – MAC
  - TCP – IP – MAC, ...
- packaging is further source of interference

CAN-to-Ethernet-to-CAN – Frame grouping

- by destination - minimize multicast overhead
- by priority (e.g. CAN ID) – enable QoS for different traffic classes
- by period or deadline - minimize sampling delay
Frame grouping - Triggering and interference

- buffer timeout (AUTOSAR)
  - frame is sent periodically
  - no frame interference
- buffer full event (AUTOSAR)
  - frame transmitted if buffer full
  - interference
- trigger frames (AUTOSAR)
  - certain CAN IDs immediately release frame
  - interference
- per-frame timeout
  - send upon single frame timeout

Interference in automotive Ethernet – Conclusion

- numerous sources of interference
  - switch operation, prioritization, frame grouping, triggering
  - no standardized solutions
  - partly based on non-disclosed parameters
- careful evaluation and design required
  - don’t rely on standards only!
Ethernet under errors – HW fault probabilities

- System reaction must be tailored to requirements (performance & safety)
- Transient communication faults dominate
  - Transient error handling must be part of regular communication

Communication under transient faults

- System must be capable of real-time operation
  - Even under occasional transmission errors (cp. CAN, FlexRay, ...)
  - Transient error protocol timing must be part of regular operation
- Suggest end-to-end error control
  - Overhead can be limited to critical messages
  - Covers all error types (link, tail-drop, ...)
- Automatic Repeat Request (ARQ)
Automatic Repeat Request (ARQ)

- various flavors of ARQ [Tanenbaum2002]
  - Stop and Wait ARQ (e.g. CAN)
  - Go-Back-N (HDLC, X.25 used in wide-area packet switched networks)
  - Selective Repeat (TCP) – not considered here due to complexity

- challenge: return channel timing (ACK)
- efficient worst case analysis for ARQ meanwhile available [Axer 2014]

Results [Axer 2014]

- line topology w. 5 switches
- 20 frames à 1024 bytes payload
- congestion: 5 additional terminals send to RX 1024 bytes every 0.5ms

worst case analysis [ms]

observed timing in simulation [ms]
ARQ in an automotive use case [Axer14]

- Go-Back-N end-to-end latency guarantees for \( N=5 \) and \( 10 \)

![Graph showing latency with Go-Back-N for \( n_{max}=5 \) and \( n_{max}=10 \)]

Handling permanent component failures

- introduction and control of component redundancy
  - multipath routing – TSN
  - zero extra delay
  - permanent overhead
  - automated path detection and routing
    - standard approach
    - large and unpredictable delay

- alternative: centralized configuration
  - possible solution: Software Defined Networking (SDN)
  - introduces control plane
  - widely used: OpenFlow protocol
  - fast enough?
**Software Defined Networking - Principle**

- uses network to communicate switch configuration
- access control, reconfiguration, ...
- explicit control or preconfigured
- control redundancy **must be added**

**SDN architecture**

**Feasibility study for SDN [Thiele 2016]**

- protocol timing for access control
  - depends on load, number conf. requests
  - explicit configuration: 1ms ...6ms
  - preconf: < 1.3ms
  - feasible approach for automotive

- more research needed
  - **H2020 project, www.safure.eu**

**SAFURE**
Overview

- automotive networks – the Ethernet promise
- Ethernet as a backbone - a closer look
- Ethernet – the safety perspective
- conclusion

Ethernet standard development – Some comments

- many new features introduced in TSN
  - addressing interests from many industries: industrial, automotive, media
  - some additions seem to be redundant compared to 802.1Qav (AVB)
    - peristaltic and burst limited shapers (worst WCRT - cp. Thiele)
  - for automotive applications
    - min. end-to-end latency typically > 1..2 ms
      - feasible with AVB - or w. additional preemption
      - clock synchronization already in AVB (802.1AS)
  - additions increase protocol and circuit complexity
    - increases switch, terminal and network assurance effort
    - increases switch cost

  be selective with new standard features!
Design complexity and cost revisited
- The cost-of-ownership trap

- many more parameters and variants than in current networks
  - MAC address management, switch management, protocol selection, packaging, ...
  - new standards in TSN even increase feature set
- current software standardarization does not limit variety
  - nor does TSN
- unified automotive solutions needed
  - different solutions and incompatibilities increase design process costs, tool costs, ...
  - cost of variety management at all levels generates new costs-of-ownership for automotive industry

Conclusion

- Ethernet is a viable basis for automatic driving
  - adaptable bandwidth and latency, supports integration
  - predictability for hard real-time systems
  - flexible end-to-end control for transient errors
  - extensions for resilient and secure networks w. fast reconfiguration
- many traps require highly systematic approach for risk mitigation
  - high-level standards needed for integration
  - solutions to individual problems, such as scheduling, are not sufficient
- research should address effective and efficient mechanisms for
  - mitigating and bounding interference on all levels (not only time)
  - providing analysis for end-to-end timing (worst case)
  - predictable dynamic network control, such as SDN
References

- Literature
  
  
  

- Acknowledgements
  - some of the slide contents have been provided by Daniel Thiele, Robin Hofmann, and Philip Axer

Thank you!