Insights into the performance and configuration of TCP in Automotive Ethernet Networks

Jörn MIGGE, RealTime-at-Work (RTaW)
Nicolas NAVET, University of Luxembourg

RTaW RealTime-at-Work

2018 IEEE Standards Association (IEEE-SA)
Ethernet & IP @ Automotive Technology Day
9-10 October 2018 | London, England

uni.lu UNIVERSITÉ DU LUXEMBOURG

Use-cases for TCP in future vehicles

Service-Oriented Architectures

Some/IP

Any TCP-based applications or protocols

e.g.: FTP, HTTP, SSH, SIP, car2x, cloud-based services, electric vehicle charging,

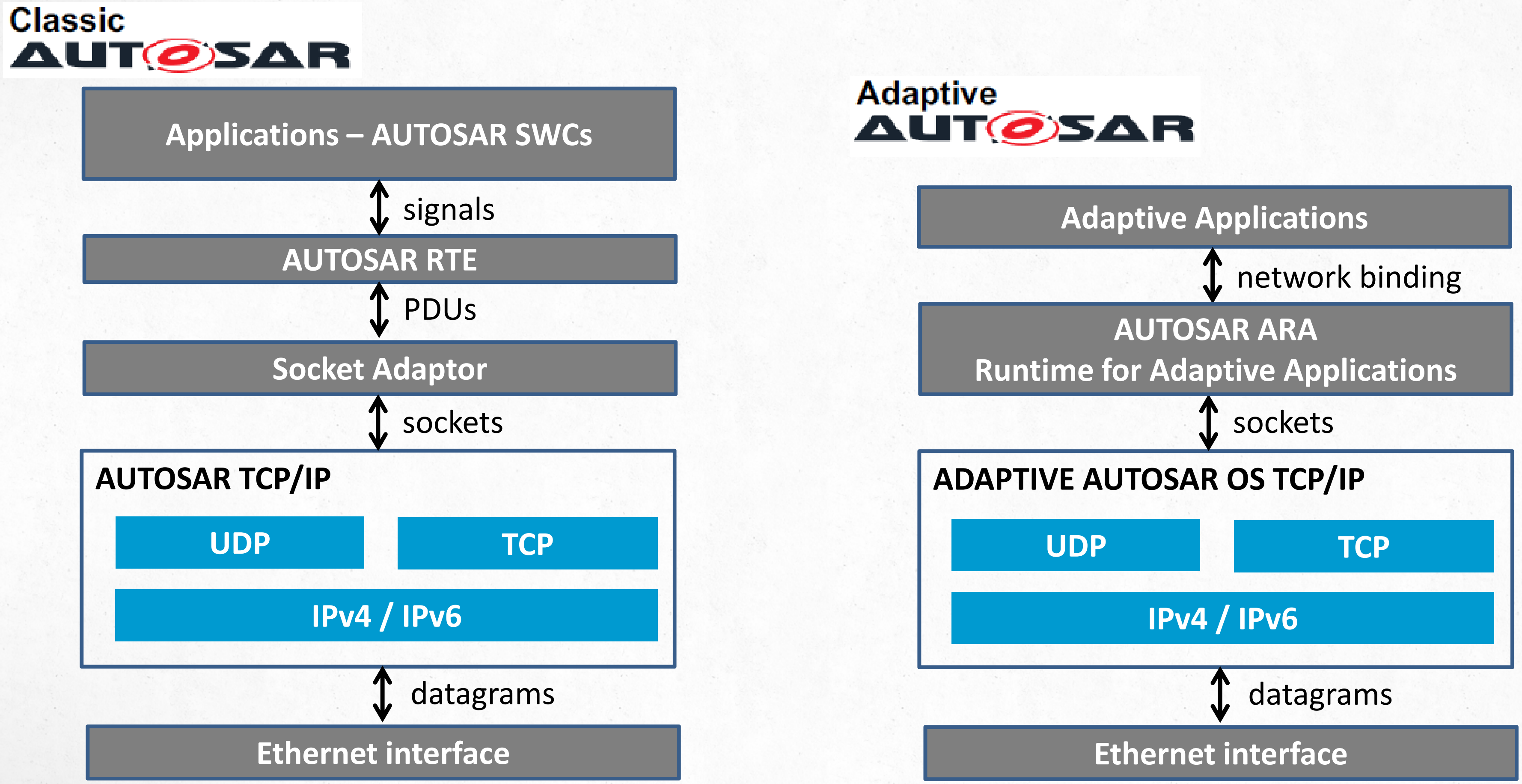
Diagnostics & flashing

DoIP
XCP

Standard TCP/IP protocols + sockets speed-up the development of applications requiring off/on-board reliable communications

Ethernet TSN

AUTOSAR TCP/IP stacks



Objectives

1. AUTOSAR TCP/IP design choices
2. Maximum achievable TCP performances with & without interfering traffic
3. Guidelines for configuring AUTOSAR TCP/IP for on-board communication
4. Impact of shapers on TCP traffic: illustration with CBS used for video

TCP performances and configuration has been studied for 40+ years, but what about TCP – as specified by AUTOSAR – for in-vehicle communication ?

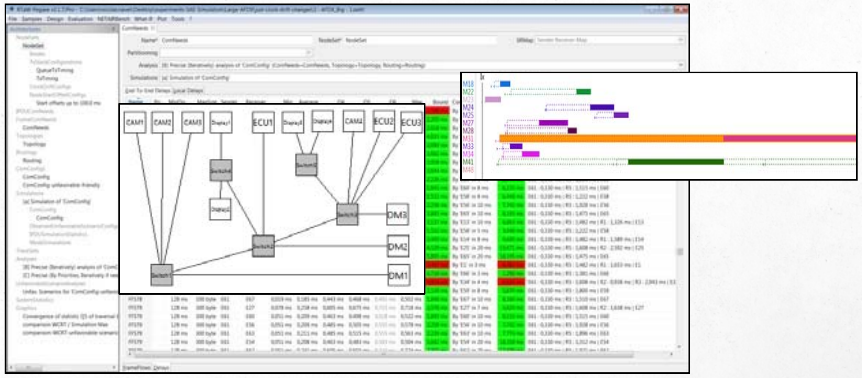
Important study in the literature: "On AUTOSAR TCP/IP Performance in In-Vehicle Network Environments", in IEEE Communications Magazine, vol. 54, no. 12, pp. 168-173, Dec. 2016.

Techniques & toolset

- **Worst-case Traversal Time (WCTT) analysis** – for hard deadline constraints
- **Timing-accurate Simulation** – for TCP throughput constraints
- **Optimization algorithms** for setting the parameters of all supported protocols

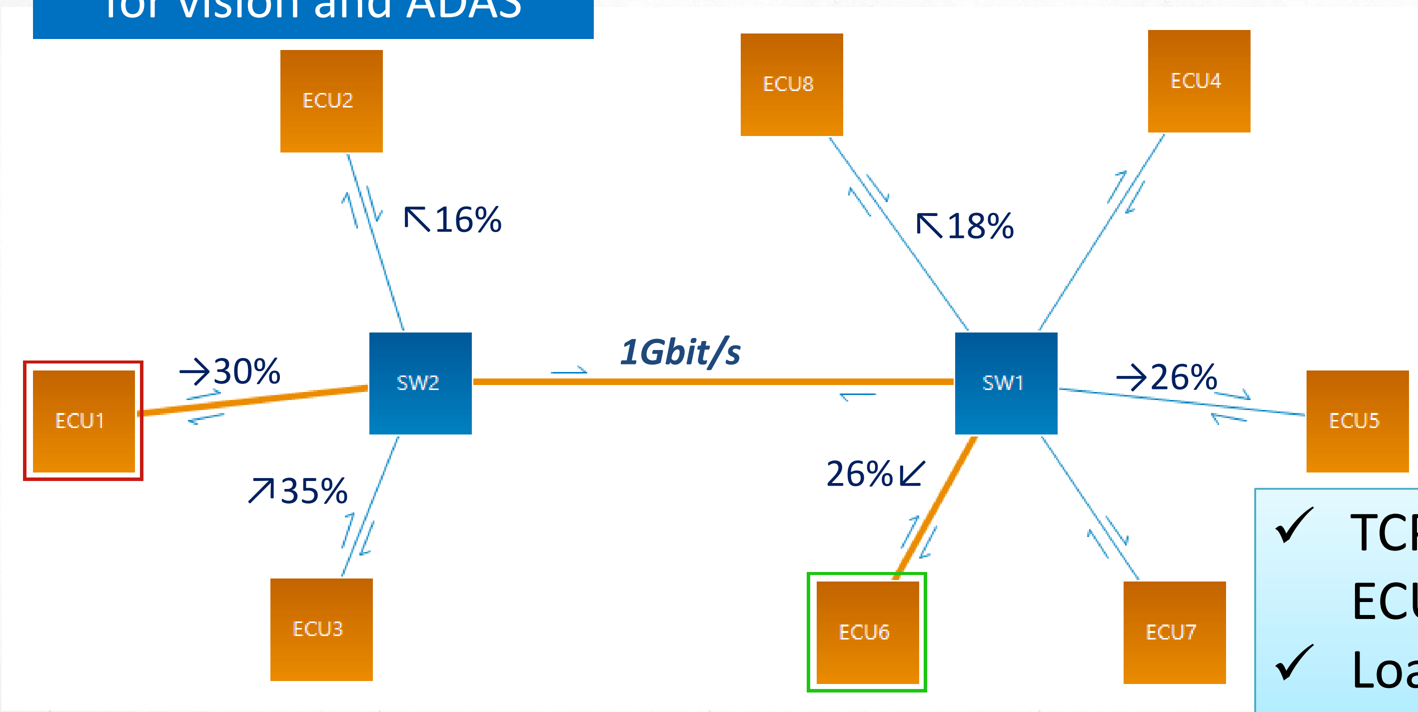
Toolset

- ***RTaW-Pegase*: modeling / analysis / configuration of automotive Ethernet TSN**
- AUTOSAR TCP/IP stack model implemented in RTaW-Pegase



Case-study : Network topology

7 video streams (30 FPS) for vision and ADAS

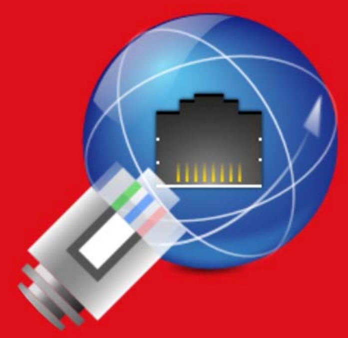


#Nodes	8
#Switches	2
#streams	47 + TCP streams
Load per link (wo TCP streams)	Min: <1%, max:35%
Link data rates	100Mbit/s and 1Gbit/s (1 link)

- ✓ TCP connection from ECU1 to ECU6
- ✓ Loads shown for interfering traffic only – not TCP streams

Case-study : Traffic

Priority	Traffic	Characteristics
Top priority	Command & Control (CC)	✓ 32 streams, 256 to 1024 byte frames ✓ 5ms to 80ms period and deadlines ✓ Hard deadline constraints
Second priority level	Audio Streams	✓ 8 streams: 128 and 256 byte frames ✓ 1.25ms period and deadline ✓ deadline constraints (soft)
Third priority level	Video streams	✓ 3 streams (vision): 30x1400 byte frames every 33ms – deadline = 33ms ✓ 4 streams (ADAS): 15x1000bytes frames every 33ms – deadline = 10ms ✓ hard and soft deadline constraints
Lowest priority	TCP streams	✓ Bulk data = from 64K to 1MB transfers, or ✓ 100ms periodic PDUs data, e.g. from CAN networks

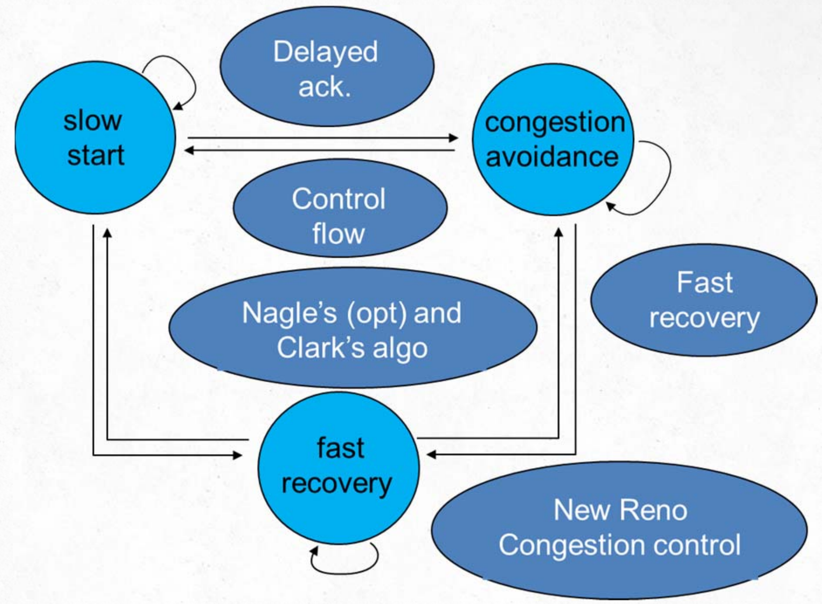


AUTOSAR TCP specification

AUTOSAR TCP design choices

– A full-fledged TCP implementation!

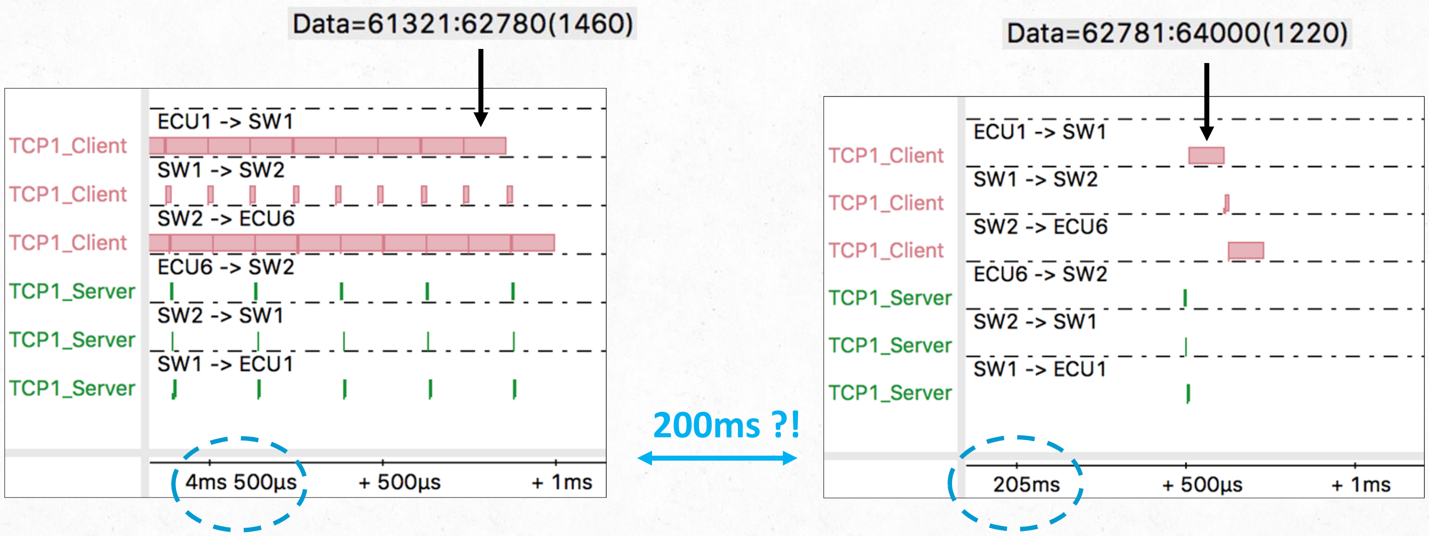
Our view: sound design choices but configuration is difficult because
- application specific
- subtle interactions between parameters: e.g. send/receive windows size, TCP task period, Nagle's algorithm on/off, time-out



– Not included in the specification: selective ack (sack) and timestamp options, recent congestion control algorithms

Bulk traffic: Nagle's algorithm and delayed ack

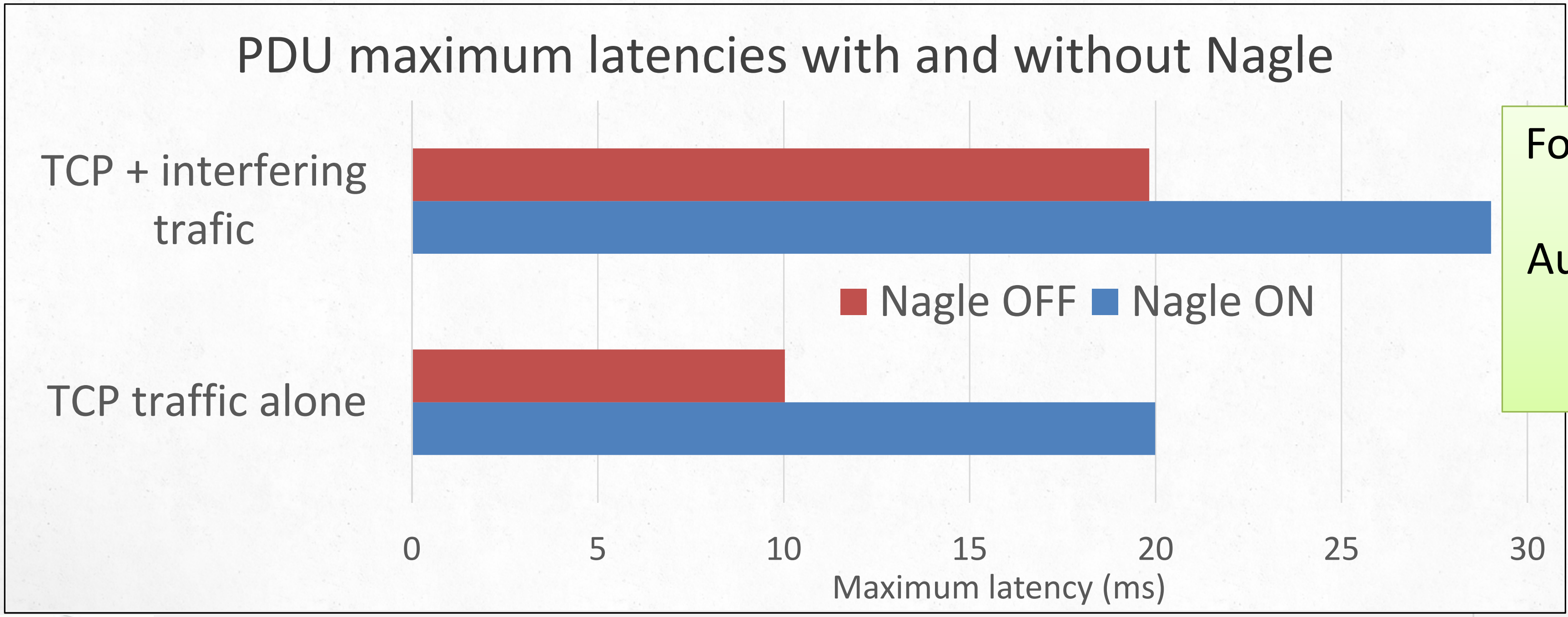
- Improve TCP efficiency by postponing both sending of data and sending of ack → buffering on both the sending and receiving sides



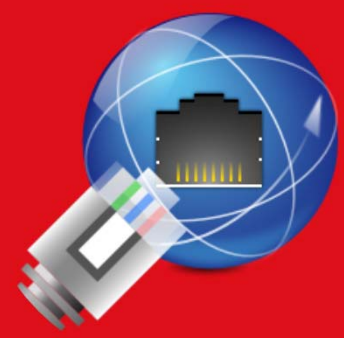
64kB = first 43 segments of 1460bytes … 200ms later comes last segment → because of Nagle's algorithm, TCP waits for the delayed ack (200ms). Solution in Autosar is to turn off "Nagle".

PDU traffic: Nagle detrimental as well

– 3 PDU streams over a TCP connection | 8, 20 and 64bytes at the lowest priority level
– Maximum latency: from the time the PDU is written in the socket, until receiver reads it



For PDU's as well, solution in Autosar is to turn off Nagle's algorithm



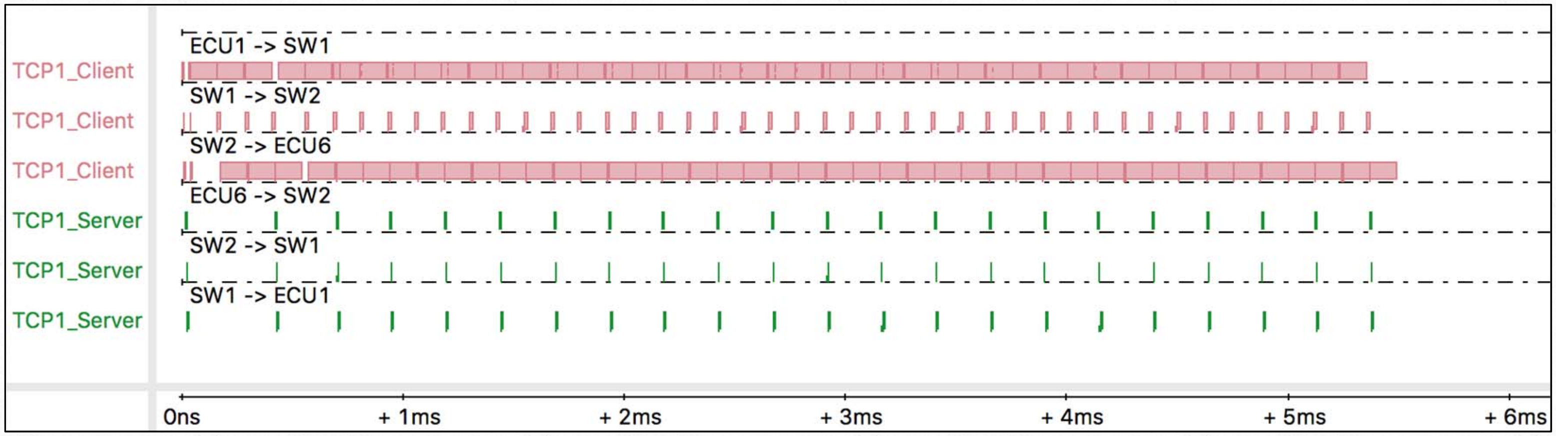
Max. achievable performances with TCP

– throughput for bulk traffic

– latencies for PDU traffic

Throughput – no interfering traffic

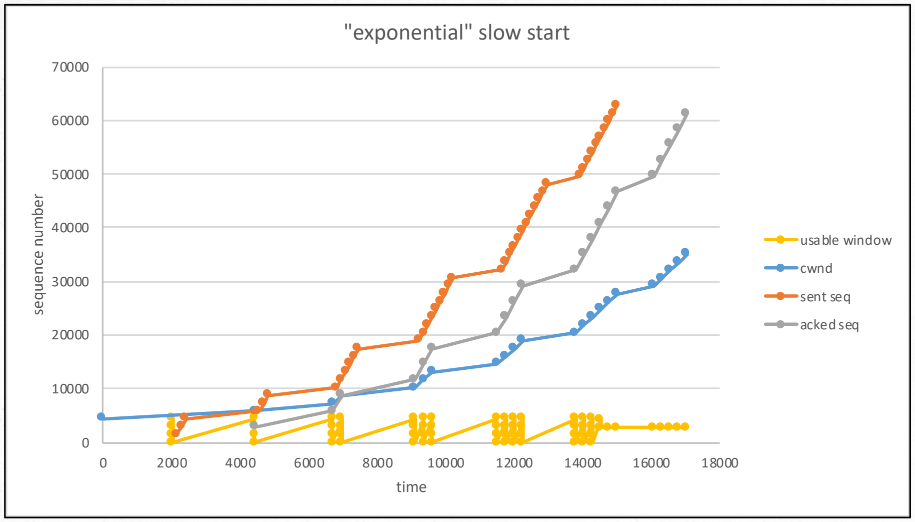
– Experimental conditions: all mechanisms on but Nagle, event-triggered management of TCP stacks, receive window larger than data, no packet loss



- ✓ Max. throughput is quickly reached: 96Mbps of TCP data over 100Mbps links!
- ✓ With interfering traffic (not shown), remaining available bandwidth can be fully used too
- ✓ But no exponential increase during slow-start ?!

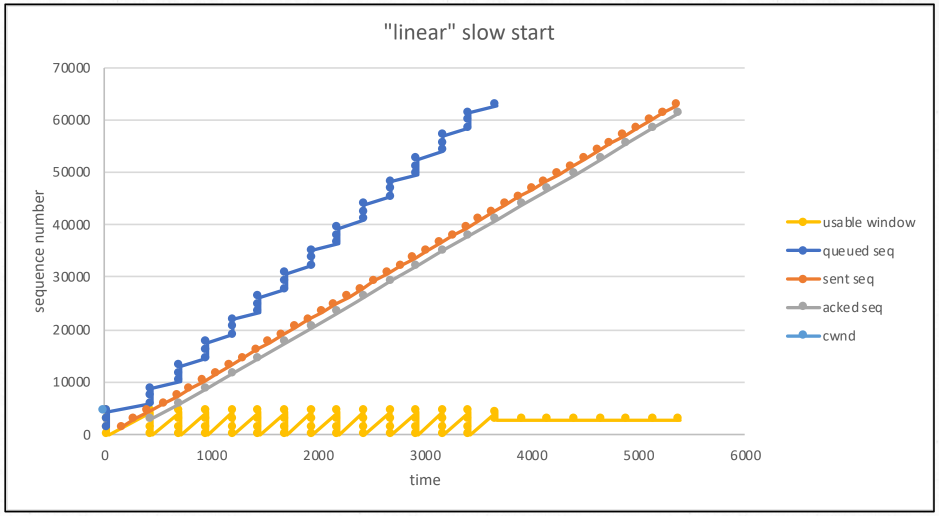
Almost no "slow-start" phase

Expected: throughput growth



[Round-trip times of 2ms]

Observed: max. throughput from start

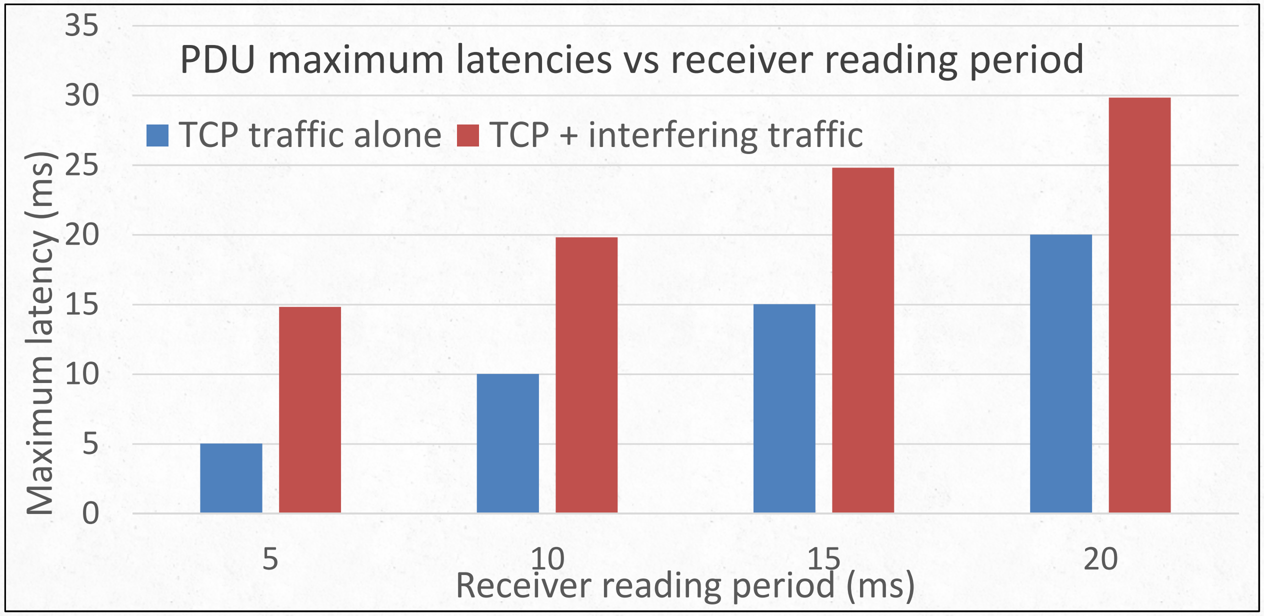


[typical automotive round-trip times]

- ✓ Reduced automotive round-trip times changes the usual behavior of TCP
- ✓ Re-examine what we can expect from TCP in the automotive context

PDU latencies vs receiver reading period

– PDU TCP streams = 8, 20 and 64 bytes at the lowest priority level
– Maximum latency | Nagle off | window update sent asap after receiver reads buffer



PDU maximum latencies can be controlled by adjusting receiver's reading period

Maximum latency ≈ traversal time + reading period
TCPMainFunction may further delay data transfer to application

TCP configuration in a TSN network

Experimental setup

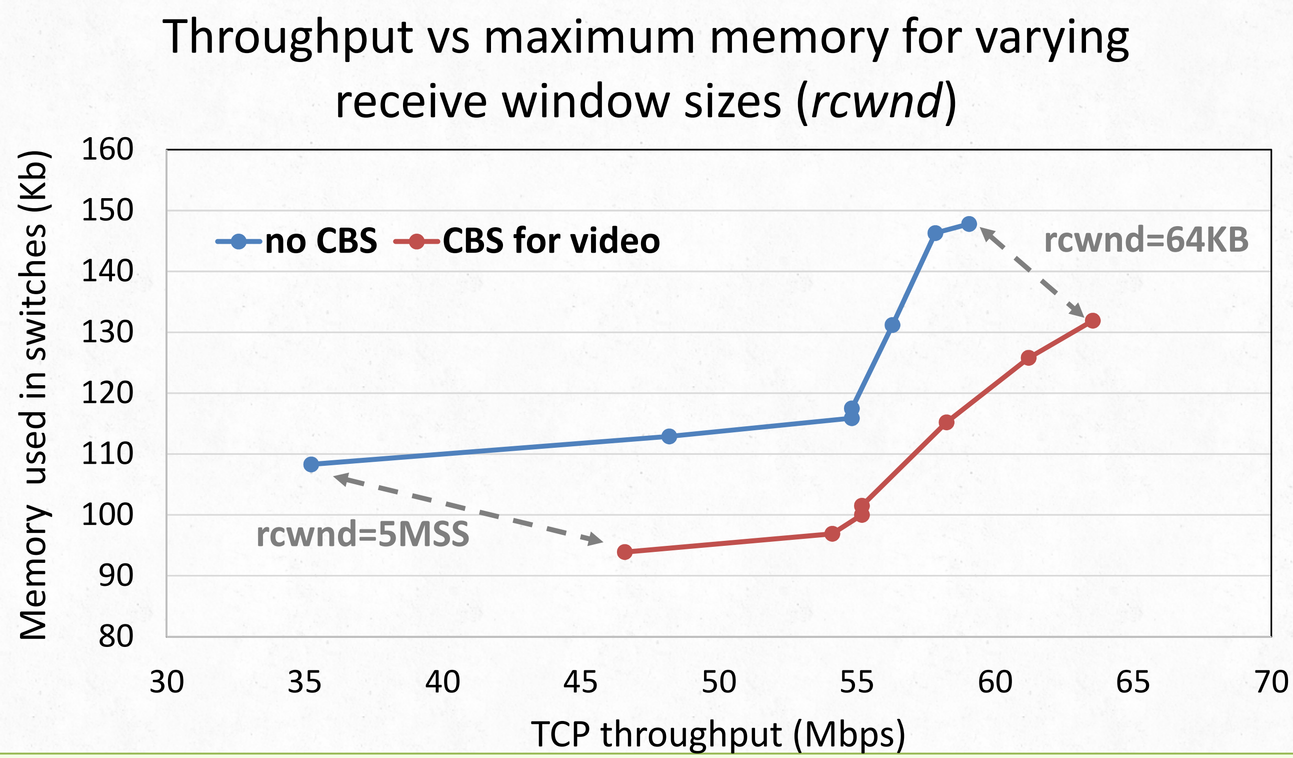
Config #1: video streams not shaped

Config #2: End-to-end shaping with TSN Credit-Based Shaper (CBS)

- Interfering streams configured so as to meet their latency constraints
- Video under CBS configured with *Tight-IdleSlope* algorithm = minimum Idle-Slopes allowing to meet deadline constraints
- TCP traffic: 1MB transfers (=685 segments) between ECU1 and ECU6 every 1s
- Minimum throughput over all TCP transfers collected over long simulations: sample of 36000 data points (12 hours of functioning)
- Receiver reads TCP buffer every 1ms

Shaping improves TCP throughput and reduces switch memory usage

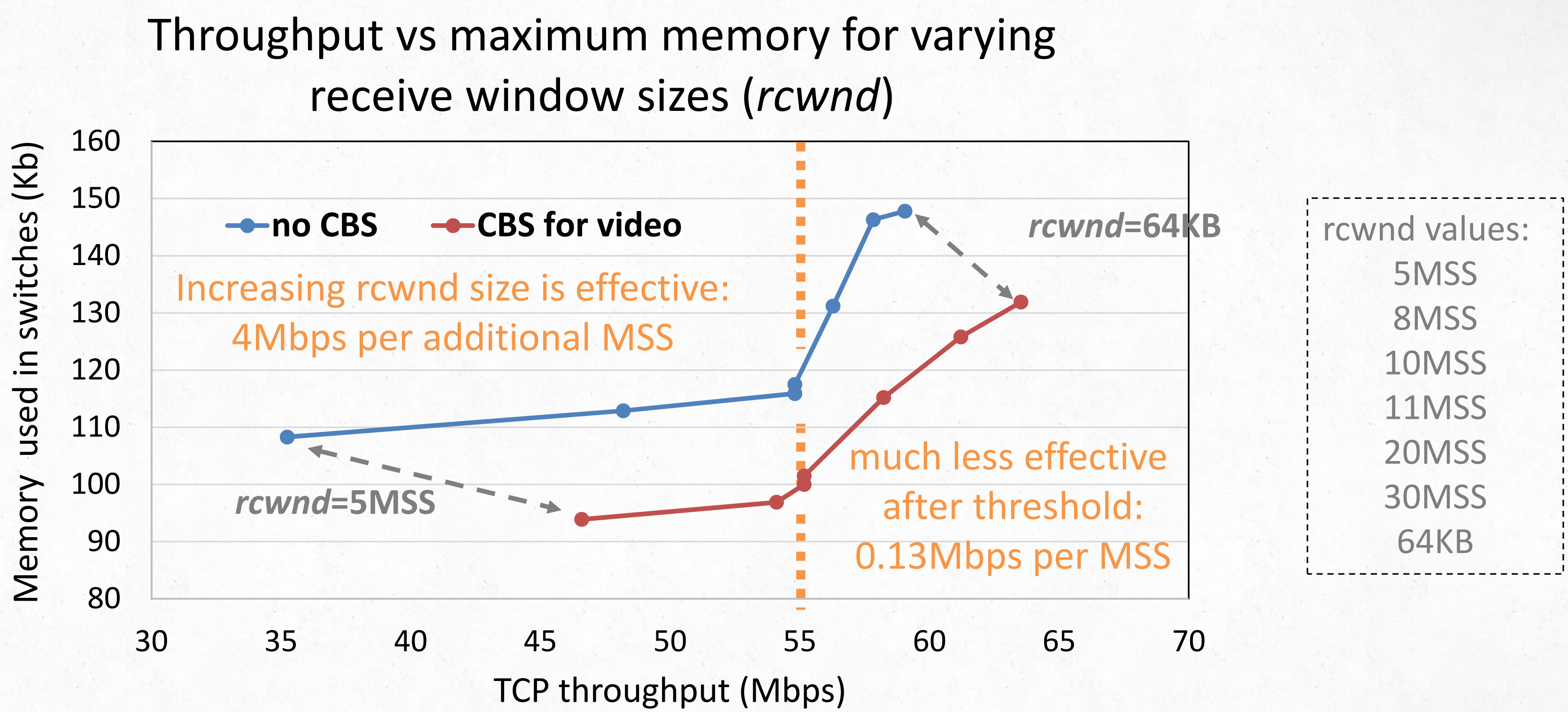
Larger *rcwnd* improves throughput but more memory required in the switches to not lose packets



rcwnd values:
5MSS
8MSS
10MSS
11MSS
20MSS
30MSS
64KB

CBS improves TCP throughput (up to 30%) and reduces memory requirement (up to 14%) for all parameters – larger gains with smaller TCP transfers more subject to bursts of interfering traffic

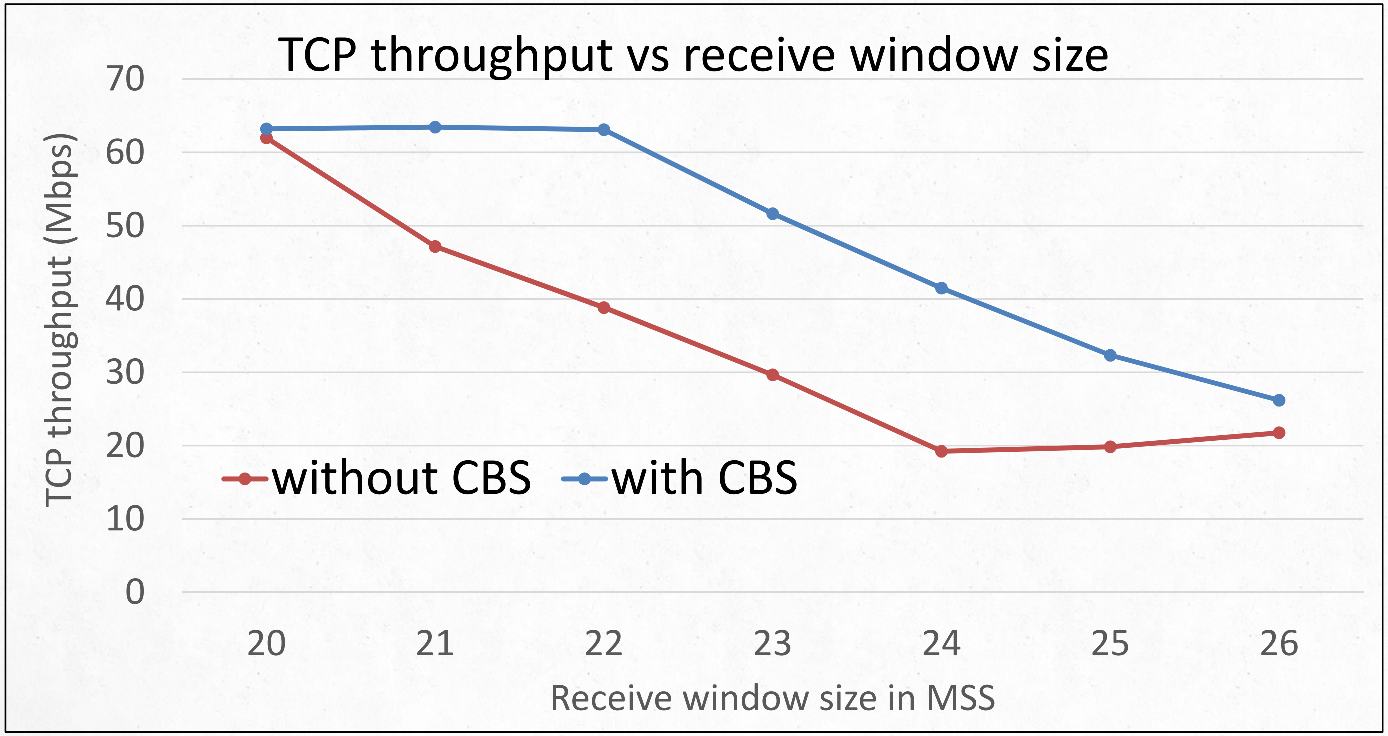
Configuring TCP receive window size – efficiency areas



With or without CBS, larger receive windows improve throughput – the gain drops after a threshold that depends on how often receive buffer is read

In practice, larger receive windows can be detrimental!

– Memory for switch port SW2 to ECU6 set to 30Kb, packet is dropped if memory full
– TCP bulk traffic |average latency | Nagle off



Larger receive windows means more “in-flight” data. Packet losses in switches lead to retransmission after time-out (1s) and drop in throughput!

Receive window size should be set wrt switch memory

Takeways

1 **AUTOSAR specifies a full-fledged TCP protocol**
Need to re-examine what we know about TCP in the automotive context

2 **AUTOSAR TCP is able to use all of the available bandwidth with minimal latencies – if properly configured and enough memory**

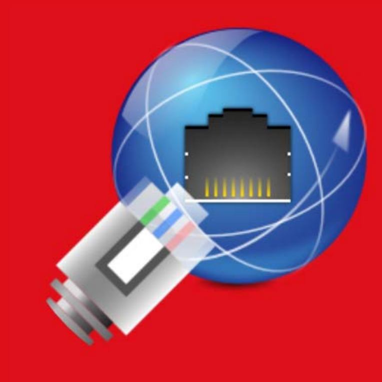
- ✓ TCP for soft real-time only as one can just obtain statistical guarantees (i.e., no worst-case analysis)
- ✓ The use of TSN shapers at higher priority levels improves TCP performance and reduces overall memory requirement

3 AUTOSAR TCP **configuration choices make a huge difference, parameters cannot be set in isolation**

- ✓ E.g. best choices for receive window size & polling period depend on switch memory size

Thank you for your attention!

Questions? Feedback? contact us at jorn.migge@realtimeatwork.com
nicolas.navet@uni.lu



References

References

1. S. Han and H. Kim, “On AUTOSAR TCP/IP Performance in In-Vehicle Network Environments”, in IEEE Communications Magazine, vol. 54, no. 12, pp. 168-173, Dec. 2016.
2. AUTOSAR, “Specification of TCP/IP Stack”, release 4.3.1., 2017.