LAS16-409 - Time Sensitive Networking - kernel modifications for automotive:industrial

Pekka Varis
Texas Instruments
Time Sensitive Networking (TSN)

Started with Audio/Video (eAVB in ~2005). Automotive and Industrial markets are driving a similar set of enhancements to Ethernet. Much of this is limited to switching and MAC layer only, but couple require SW stack support, some in the data path and some in control.

See https://www.iol.unh.edu/sites/default/files/knowledgebase/UNH-IOL_TSN-Overview.pdf for an overview. First ~25 slides are who, middle ~35 are why, final ~40 are how. Credit to Bob Noseworthy UNH.
Time Sensitive Networking (TSN)

TSN is roughly:

- **802.1AS**: Time Synchronization
- **802.1Qav**: Forwarding and Queuing Enhancements
- **802.1Qaz**: Enhanced Transmission Selection
- **802.1Qbb**: Priority Based Flow Control
- **802.1Qbv**: Time Aware Queuing
- **802.1Qbu**: Frame Preemption
- **P802.1Qch**: Cyclic queueing
- **P802.1Qci**: Input gating
- **P802.1CB**: Seamless redundancy
- **802.3br**: Interspersing Express Traffic
- **802.1Qat**: Distributed “stream reservation protocol”, extended in P802.1Qcc
Markets and examples of companies active in TSN

Audio/Video at consumer and home
  Broadcom, Intel, Apple, ...
Professional Audio/Video such as AV production or large venues (e.g. ESPN Bristol)
  Reidel, Harman, Bosch, Cisco, ...
Automotive Driver Assist (ADAS), Infotainment, Chassis, Powertrain, Body
  Audi, BMW, Continental, Broadcom, TI, Intel, ...
Industrial control, Robotics
  Siemens, Rockwell, Hirschmann, Schneider, ...

Just an some examples of companies who have contributed or have been referenced in for example IEEE in no particular order
Why should Linaro care?

This is traditional embedded computing, ARM is the dominant architecture. Traditional domain of RTOS and even bare-metal, but several factors are driving towards Linux.

- Industrial Internet of Things/ Industry 4.0 (buzz word of your choice) needs full featured connectivity and security
- Availability of devices capable of running Linux at relevant power and price points
- Acceleration (relative to legacy 10 year) of development
- Younger generation of engineers expect Linux

Opportunity for Linaro and LNG to contribute broader than in just in networking centric end products
The application model for control traffic is different.

Typical traffic pattern for control traffic:
- Control Cycle / Transmission Period
- Communicate
- Compute
- Transmission Order
- Typical traffic pattern for industrial closed-loop applications

Controller e.g. PLC

Transmission of Inputs

Closed Loop Control

Transmission of Outputs

Sensors

Actuators

Physical Process

Possible areas to work on

Not chasing highest throughput, but **controlled maximum latency for the application** and good enough performance for the realtime traffic with all the “best effort” that Linux enables available in parallel.

There is already activity like [https://lkml.org/lkml/2016/6/11/214](https://lkml.org/lkml/2016/6/11/214) (Cisco), first step would be to map what is already being worked on.

Example LNG activity:

LNG-282 Investigate the work for kernel multi queue egress on TI parts

Do not target the most extreme requirements in latency and safety, those will keep using RTOSes or just give ownership of the peripheral directly to user space.
Possible areas to work on

Data plane

Transmit side (talker) support for SoC’s capable of Time Aware Shaper and Frame Pre-emption

Existing implementations in Linux on Intel HW and Macs (?)

Testing and benchmarking

What is the latency profile, correctness includes concept of time

Time synchronization improvements (possibly already at a good level)

E.g. support for 1-step timing over packet HW, PTPv2

Maybe support for integrated switch, called bridge in TSN terminology (switchdev?)

Control and Management

Maybe admission control and resource reservation