



# HKG18-415: TSN and Linux

Khoronzhuk Ivan

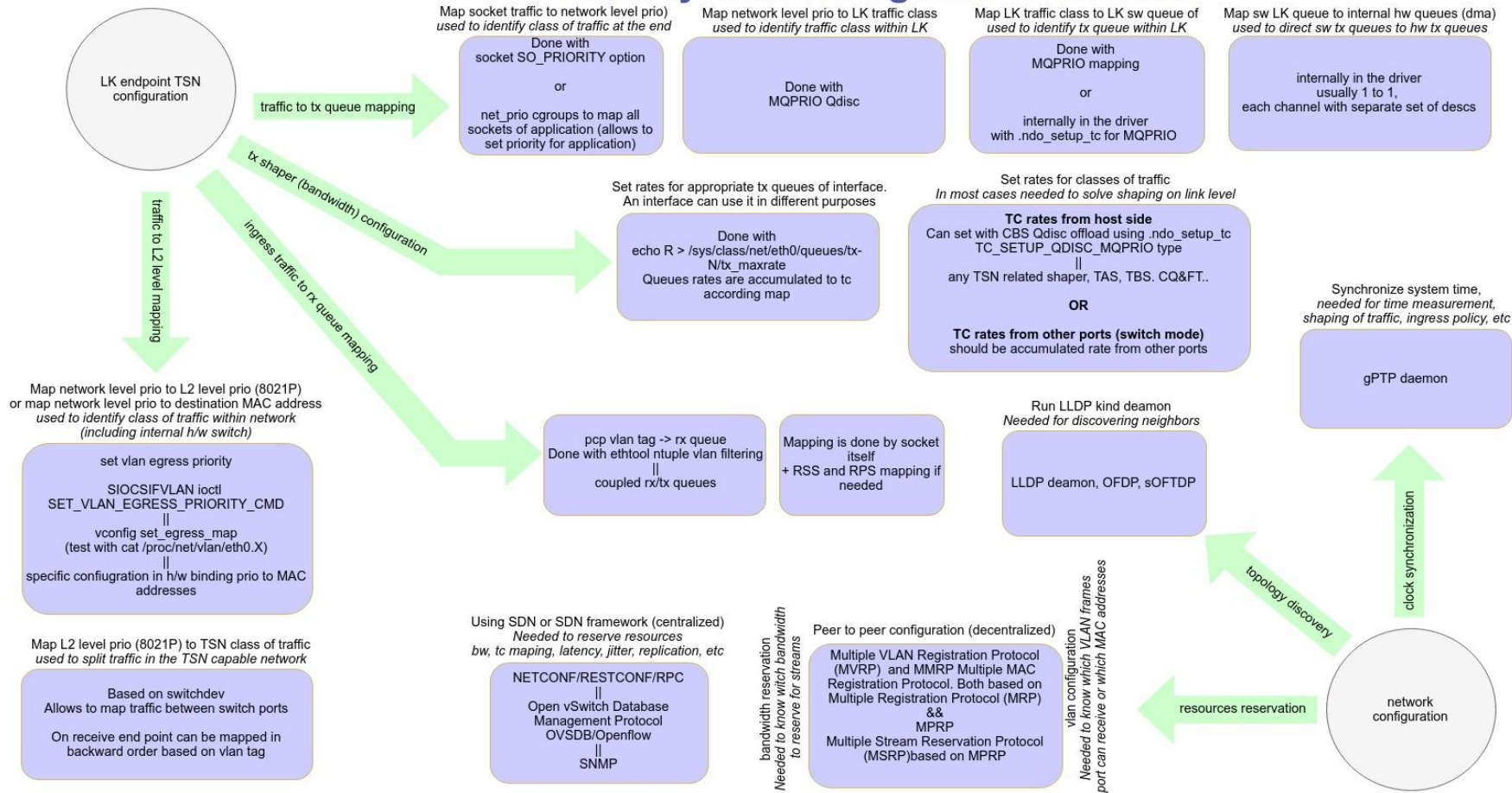


# What is TSN and Intentions?

- TSN is the IEEE802.1Q defined standard technology to provide deterministic messaging on standard Ethernet. TSN technology is centrally managed and delivers guarantees of delivery and minimized jitter using time scheduling for those real-time applications that require determinism.
- Generic configuration interfaces when possible
  - Linux kernel, rich open source community, thus review
  - Allows to reuse tools like ethtool, iproute2, tc, vconfig, bridge, lldp, etc
  - Use generic interfaces for driver configuration, netdev, ethtool, switchdev
  - No need to maintain out of tree drivers
- Main criteria - guaranteed delivery, best-effort coexists



# Overall system configuration scheme



traffic to tx queue mapping

Map socket traffic to network level prio)  
*used to identify class of traffic at the end*

Done with  
socket SO\_PRIORITY option

or

net\_prio cgroups to map all  
sockets of application (allows to  
set priority for application)

Map network level prio to LK traffic class  
*used to identify traffic class within LK*

Done with  
MQPRIO Qdisc

Map LK traffic class to LK sw queue of  
*used to identify tx queue within LK*

Done with  
MQPRIO mapping

or

internally in the driver  
with .ndo\_setup\_tc for MQPRIO

Map sw LK queue to internal hw queues (dma)  
*used to direct sw tx queues to hw tx queues*

internally in the driver  
usually 1 to 1,  
each channel with separate set of desc





traffic to L2 level mapping

Map network level prio to L2 level prio (8021P)  
or map network level prio to destination MAC address  
*used to identify class of traffic within network  
(including internal h/w switch)*

set vlan egress priority

```
SIOCSIFVLAN ioctl
SET_VLAN_EGRESS_PRIORITY_CMD
||
vconfig set_egress_map
(test with cat /proc/net/vlan/eth0.X)
||
specific configuration in h/w binding prio to MAC addresses
```

Map L2 level prio (8021P) to TSN class of traffic  
*used to split traffic in the TSN capable network*

Based on switchdev  
Allows to map traffic between switch ports

On receive end point can be mapped in backward order based on vlan tag



ingress traffic to rx queue mapping

pcp vlan tag -> rx queue  
Done with ethtool ntuple vlan filtering  
||  
coupled rx/tx queues

Mapping is done by socket itself  
+ RSS and RPS mapping if  
needed



tx shaper (bandwidth) configuration

Set rates for appropriate tx queues of interface.  
An interface can use it in different purposes

Done with  
echo R > /sys/class/net/eth0/queues/tx-  
N/tx\_maxrate  
Queues rates are accumulated to tc  
according map

Set rates for classes of traffic  
*In most cases needed to solve shaping on link level*

**TC rates from host side**

Can set with CBS Qdisc offload using .ndo\_setup\_tc  
TC\_SETUP\_QDISC\_MQPRIO type  
||  
any TSN related shaper, TAS, TBS, CQ..

**OR**

**TC rates from other ports (switch mode)**  
should be accumulated rate from other ports



Using SDN or SDN framework (centralized)  
*Needed to reserve resources  
bw, tc mapping, latency, jitter, replication, etc*

NETCONF/RESTCONF/RPC  
||  
Open vSwitch Database  
Management Protocol  
OVSDB/Openflow  
||  
SNMP

bandwidth reservation  
*Needed to know witch bandwidth  
to reserve for streams*

Peer to peer configuration (decentralized)

Multiple VLAN Registration Protocol  
(MVRP) and MMRP Multiple MAC  
Registration Protocol. Both based on  
Multiple Registration Protocol (MRP)  
&&  
MPRP  
Multiple Stream Reservation Protocol  
(MSRP)based on MPRP

vlan configuration  
*Needed to know which VLAN frames  
port can receive or which MAC addresses*

Run LLDP kind daemon  
*Needed for discovering neighbors*

LLDP daemon, OFDP, sOFTDP

Synchronize system time,  
*needed for time measurement,  
shaping of traffic, ingress policy, etc*

gPTP daemon

clock synchronization

topology discovery

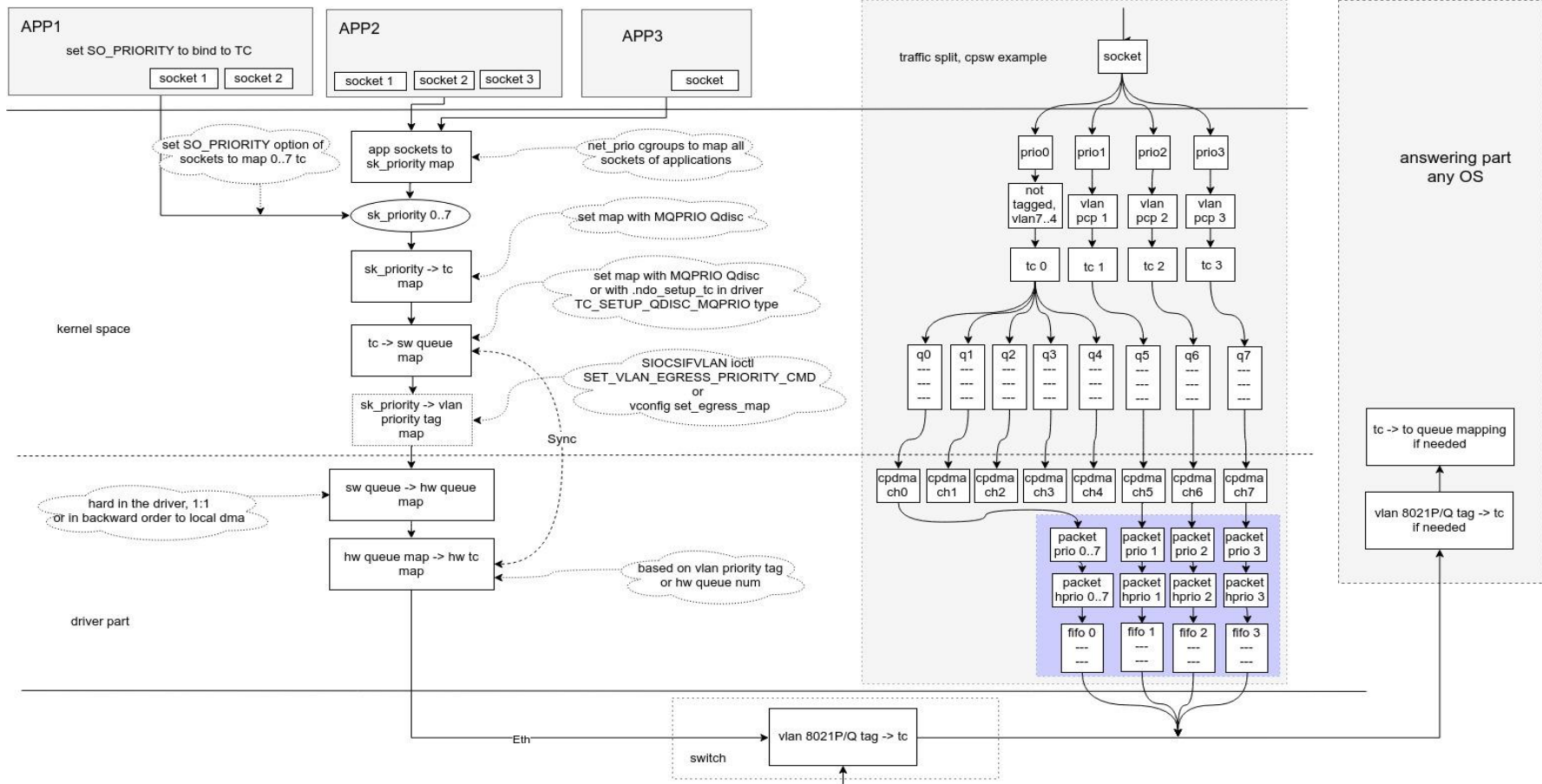
resources reservation

network  
configuration

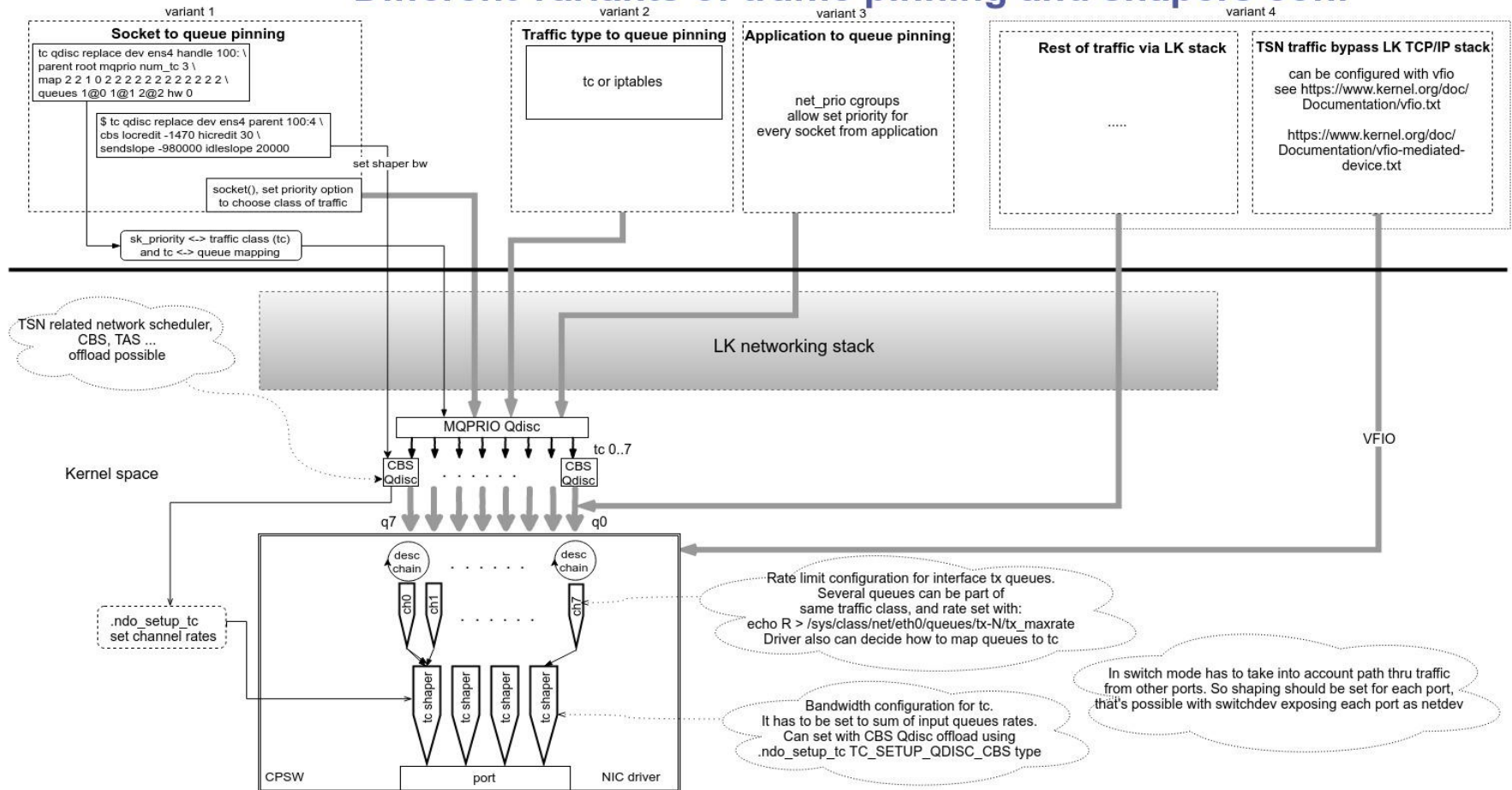




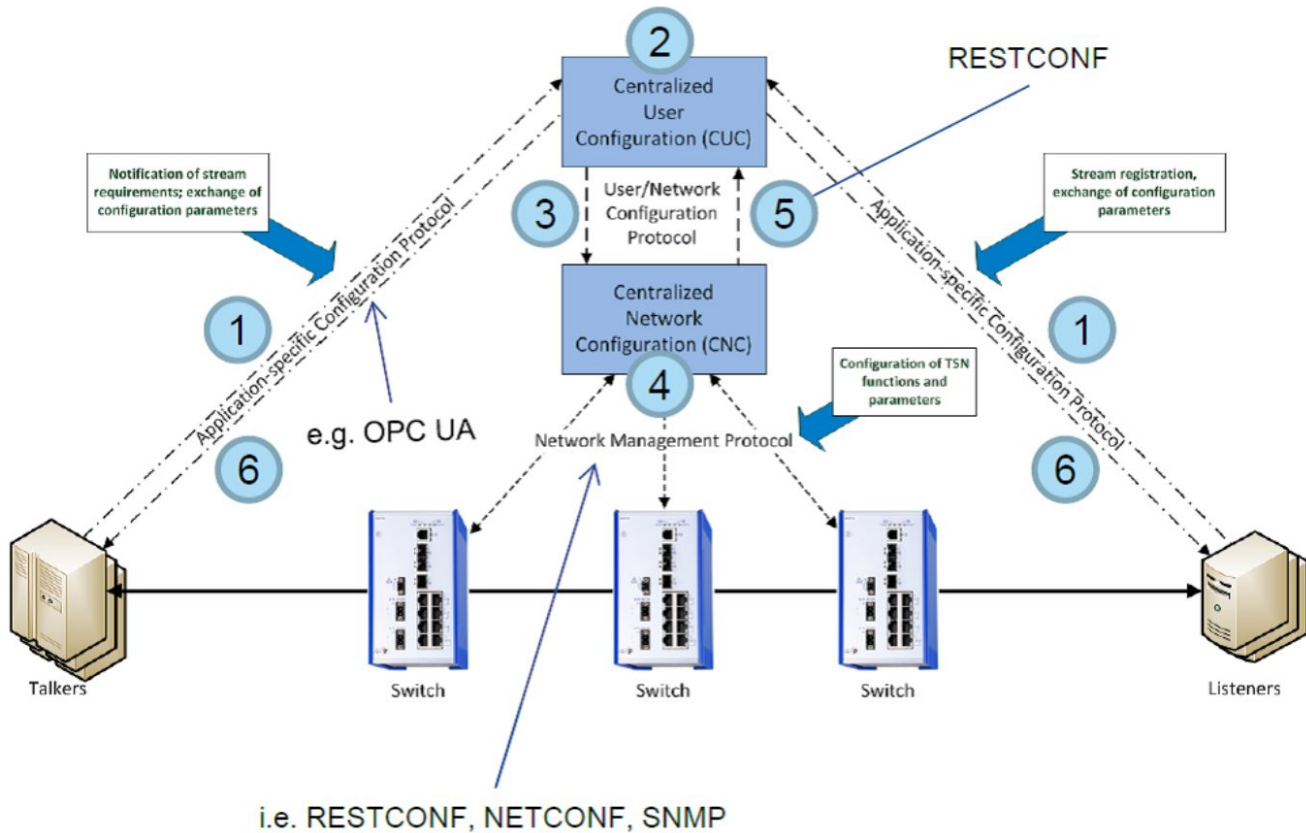
# traffic map with sockets



# Different variants of traffic pinning and shapers conf



# TSSDN model



# Challenges

- Scale impact, a lot of control path messages, LLDP, PTP, NETCONF
- Only secure access to the network and secure configuration
  - mistiming is forbidden
  - topology issue can lead to incorrect latency
  - data plane flooding, sophisticated rx policy
- Network redundancy, can be solved in h/w at NIC level, but configuration
- LK, decrease latency increase stability
  - Separate netspace from best effort traffic
  - RSS, CPU pinning
  - Bypass some part of stack, eBFP
  - XDP eBPF and AF\_XDP (still under development)



# State, plans

- TSN yocto layer +
  - Last in kernel changes
  - Package group needed for TSN system configuration
  - In kernel changes (CBS, TAS, TBS, optimization)
  - System configuration changes, documentation
  - Driver tsn extensions for shaper configuration, switchdev..
- More latency measurements
- LAVA environment for TSN based on yocto layer
- Mininet configuration instead of h/w? For TSSDN test purposes only
- Identifying data model entries: latency, jitter, tc, bw, route and others





# State, plans for TI parts

- cpsw CBS shaper patches (need only upstream)
- netcp multiqueue patches (need only upstream)
- netcp Ethtool ntuples configuration (need add support)
- cpsw ethtool ntuples configuration (need add support)
- cpsw add coupled queues and ingress configuration
- cpsw swithdev support (not sure if need, but should be)
- Test TAS shapers support
- extend tisdsk, probably



# Links

- TAS (time aware shaper) mainline activity
  - <https://patchwork.kernel.org/patch/10029815/>
- TBS (time based packet scheduler)
  - <https://lwn.net/Articles/744797/>
- AF\_XDP (one more try to avoid SKB allocation and apply ZC)
  - <https://lwn.net/Articles/745934/>
- Latency measurements for am572, netcp, x86\_64 i210
  - <https://projects.linaro.org/secure/attachment/12934/TSN%20measurements%20am572x%20k2g.pdf>
  - [https://git.linaro.org/people/ivan.khoronzhuk/tsn\\_latencies.git/tree/](https://git.linaro.org/people/ivan.khoronzhuk/tsn_latencies.git/tree/)
  - [https://projects.linaro.org/secure/attachment/13379/x86\\_lat.pdf](https://projects.linaro.org/secure/attachment/13379/x86_lat.pdf)
- Cpsw, netcp mq, cbs shaper configuration
  - [https://git.linaro.org/people/ivan.khoronzhuk/tsn\\_kernel.git/log/?h=cpsw\\_avb\\_v1](https://git.linaro.org/people/ivan.khoronzhuk/tsn_kernel.git/log/?h=cpsw_avb_v1)
  - [https://git.linaro.org/people/ivan.khoronzhuk/tsn\\_kernel.git/log/?h=mq\\_netcp\\_v6](https://git.linaro.org/people/ivan.khoronzhuk/tsn_kernel.git/log/?h=mq_netcp_v6)



# Related base protocols

- IEEE 802.1Qbu and IEEE 802.3br Frame Preemption
- IEEE Std 802.1Qbv-2015 Enhancements for Scheduled Traffic
- IEEE Std 802.1Qca-2015 Path Control and Reservation
- IEEE 802.1Qcc Central configuration, enhancements and perf improvements
- IEEE 802.1Qci Time-based ingress policing
- IEEE 802.1CB
- IEEE Std 802.1AS-2011 Timing and Synchronization, gPTP
- IEEE Std 802.1Qat-2010 Stream Reservation Protocol (SRP)
- IEEE Std 802.1Qav-2009 Forwarding and Queueing Enhancements (for CBS)
- IEEE Std 802.1BA-2009 Audio Video Bridging (AVB) Systems





# Thank You

## #HKG18

HKG18 keynotes and videos on: [connect.linaro.org](https://connect.linaro.org)

For further information: [www.linaro.org](https://www.linaro.org), [ivan.khoronzhuk@linaro.org](mailto:ivan.khoronzhuk@linaro.org)

