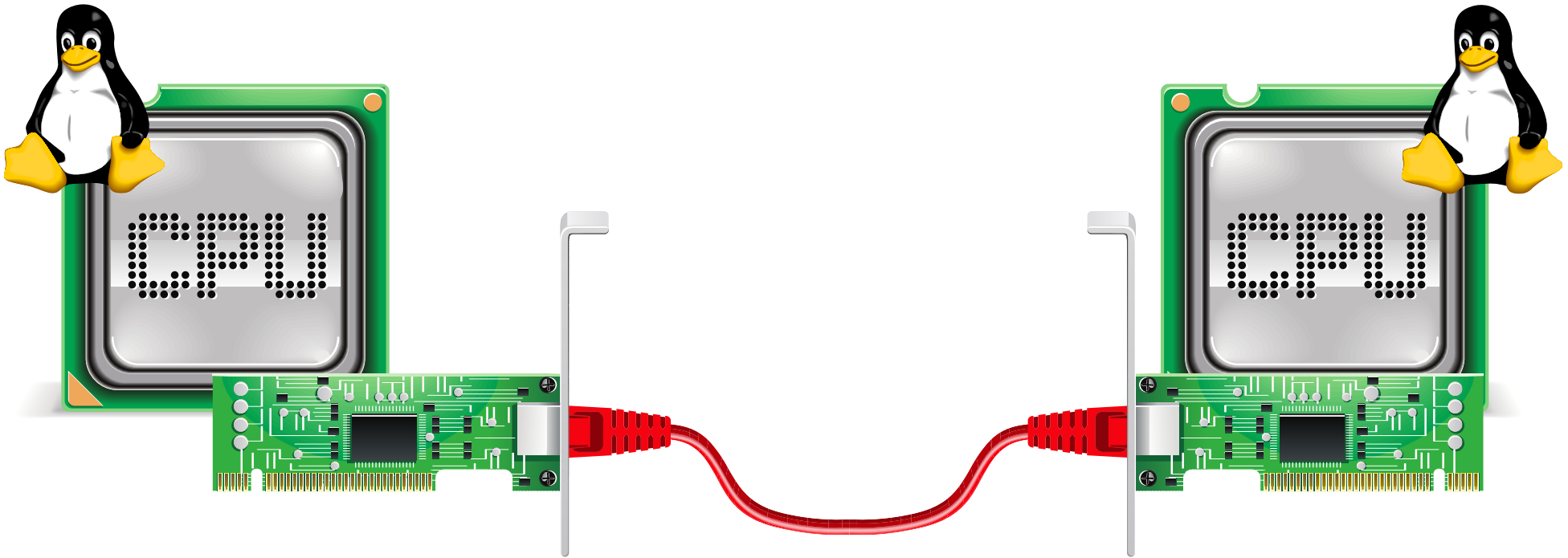
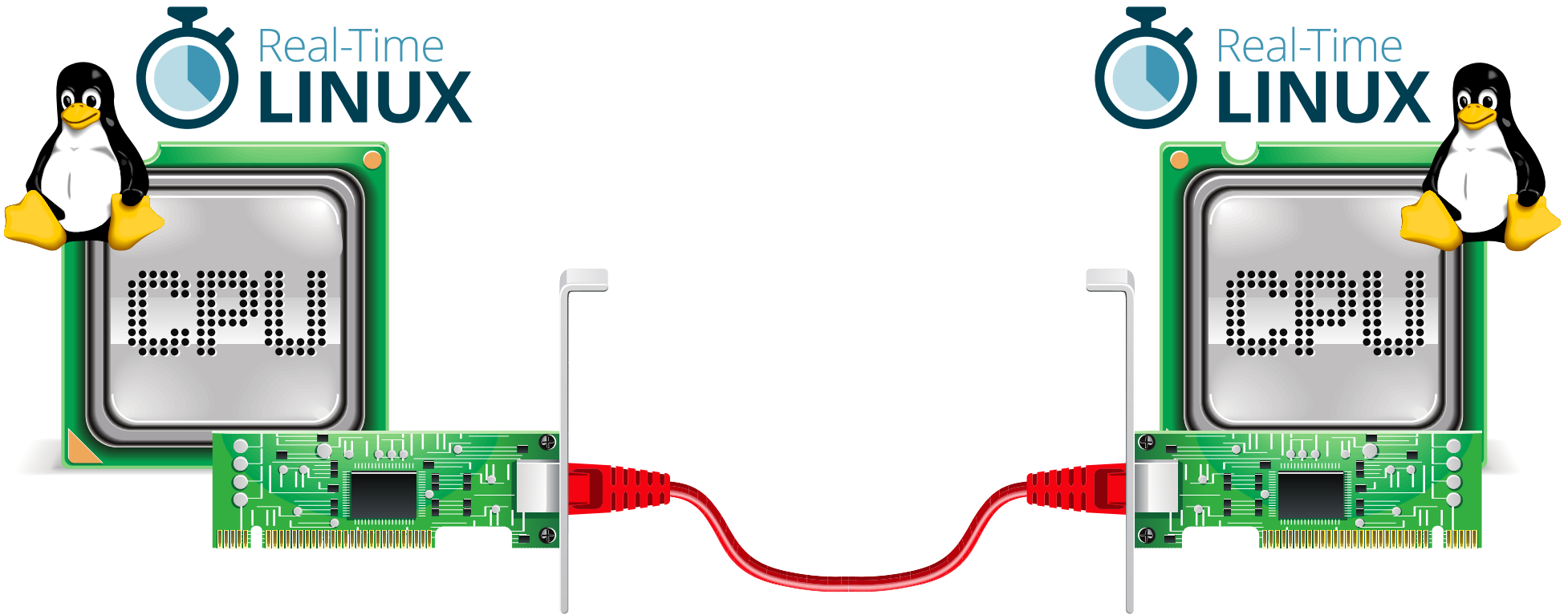
Setting up Real-time Ethernet and TSN with Linux: A 10-step guide

Jan Altenberg
Open Source Automation Development Lab (OSADL) eG

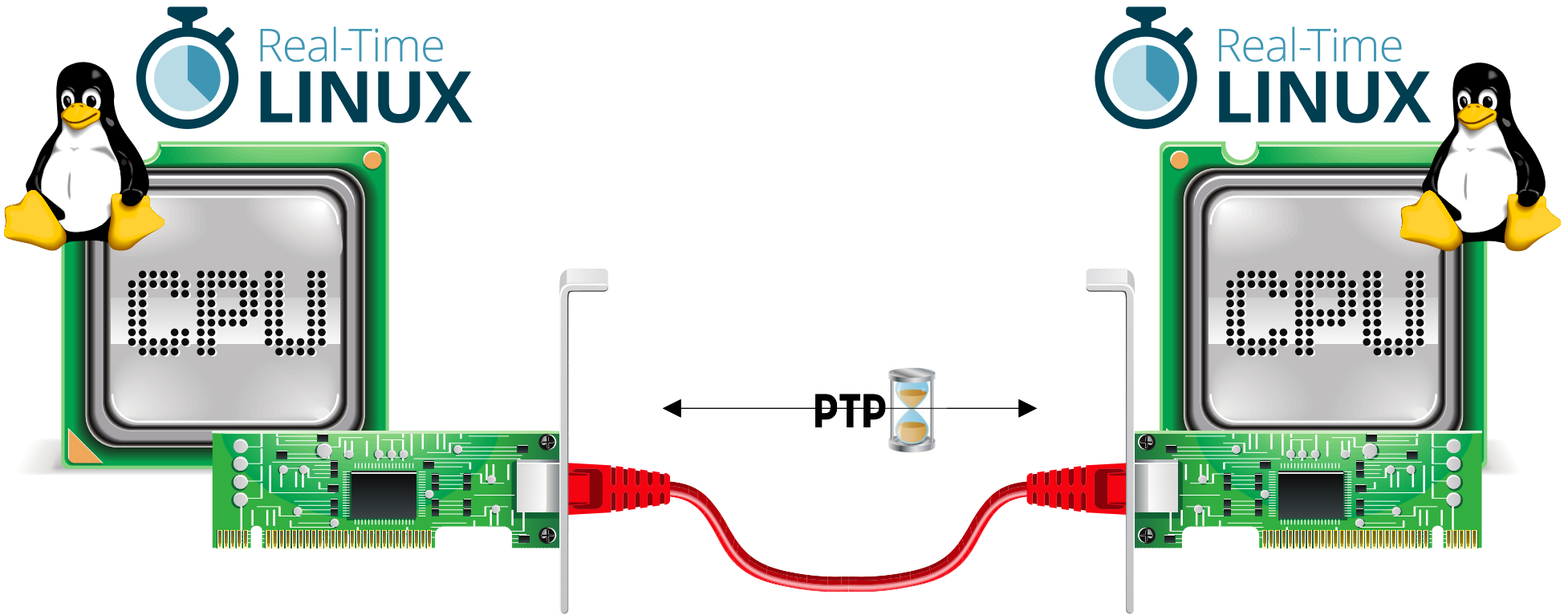
Setup



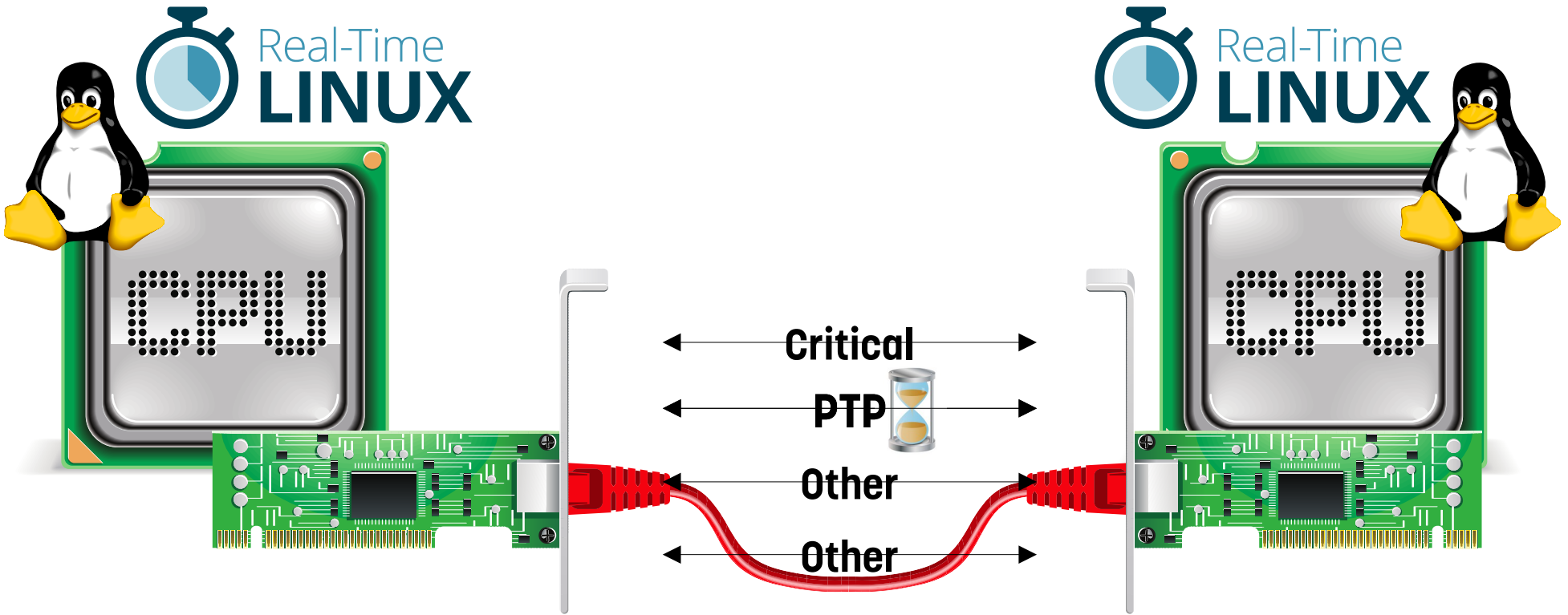
Setup



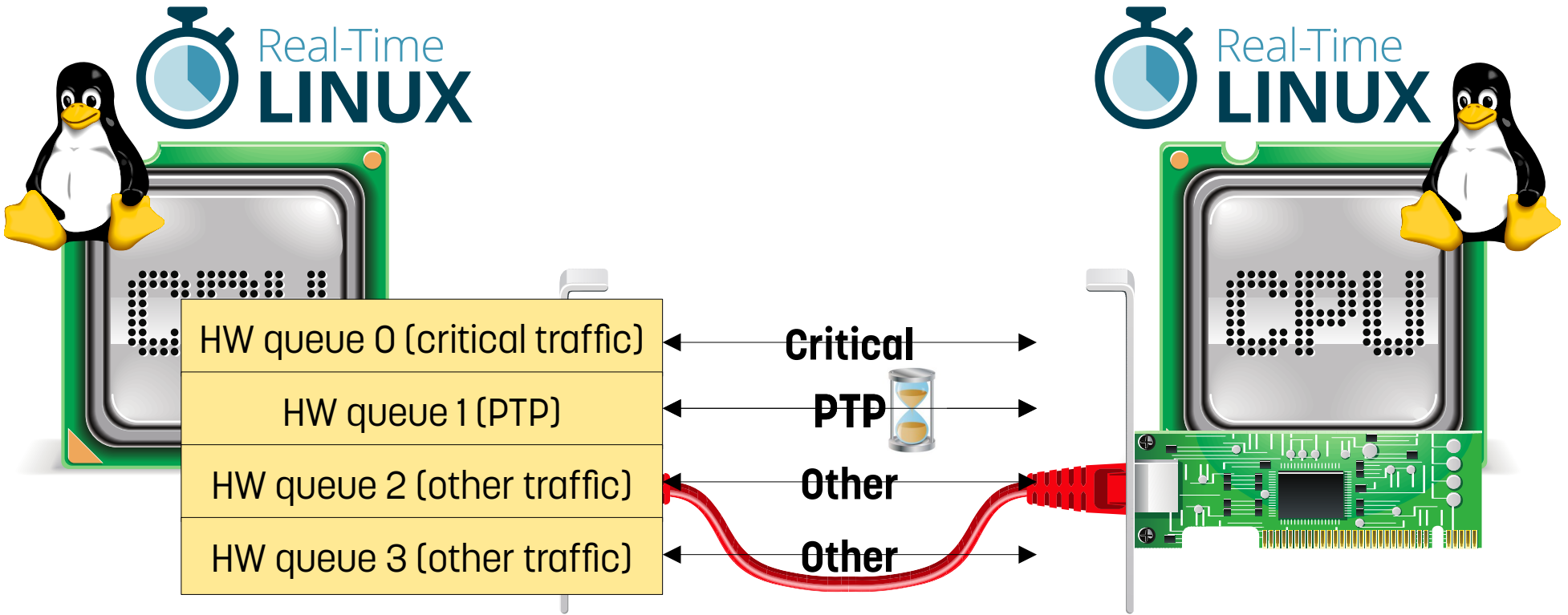
Setup



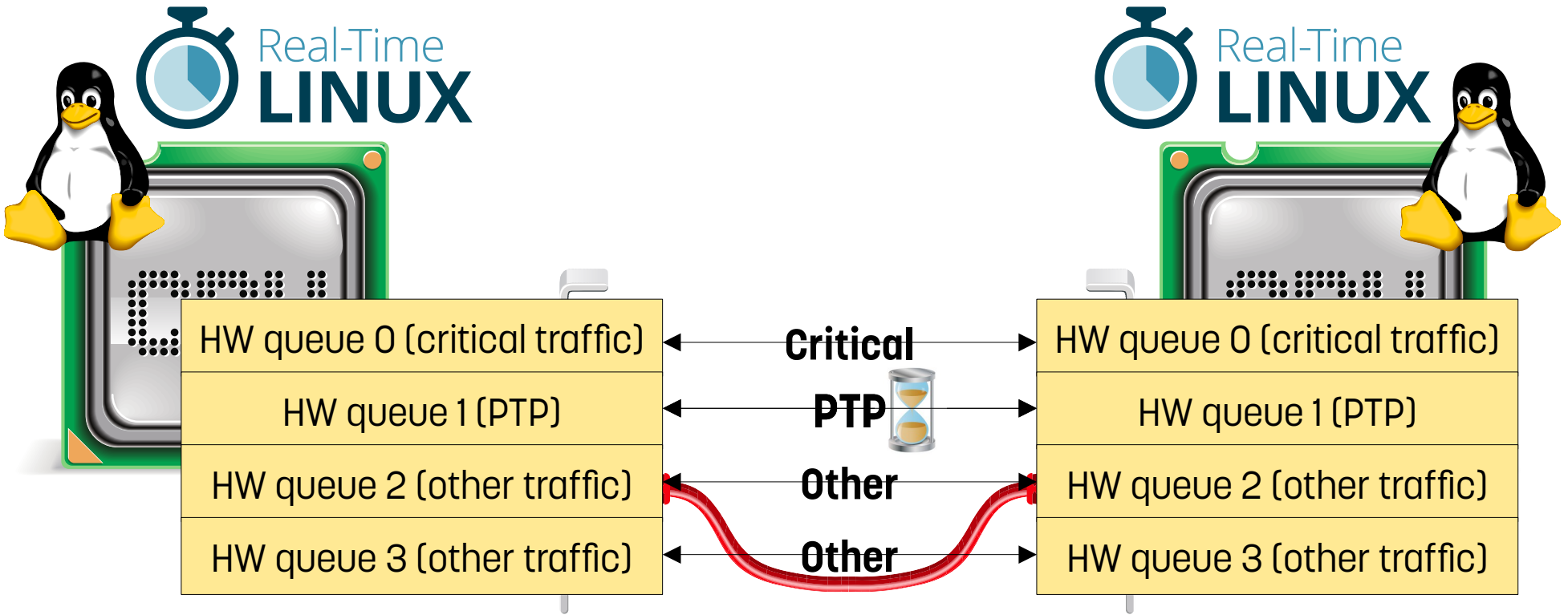
Setup


Real-Time
LINUX
CPU
Critical
PTP
Other
Other
Real-Time
LINUX
CPU

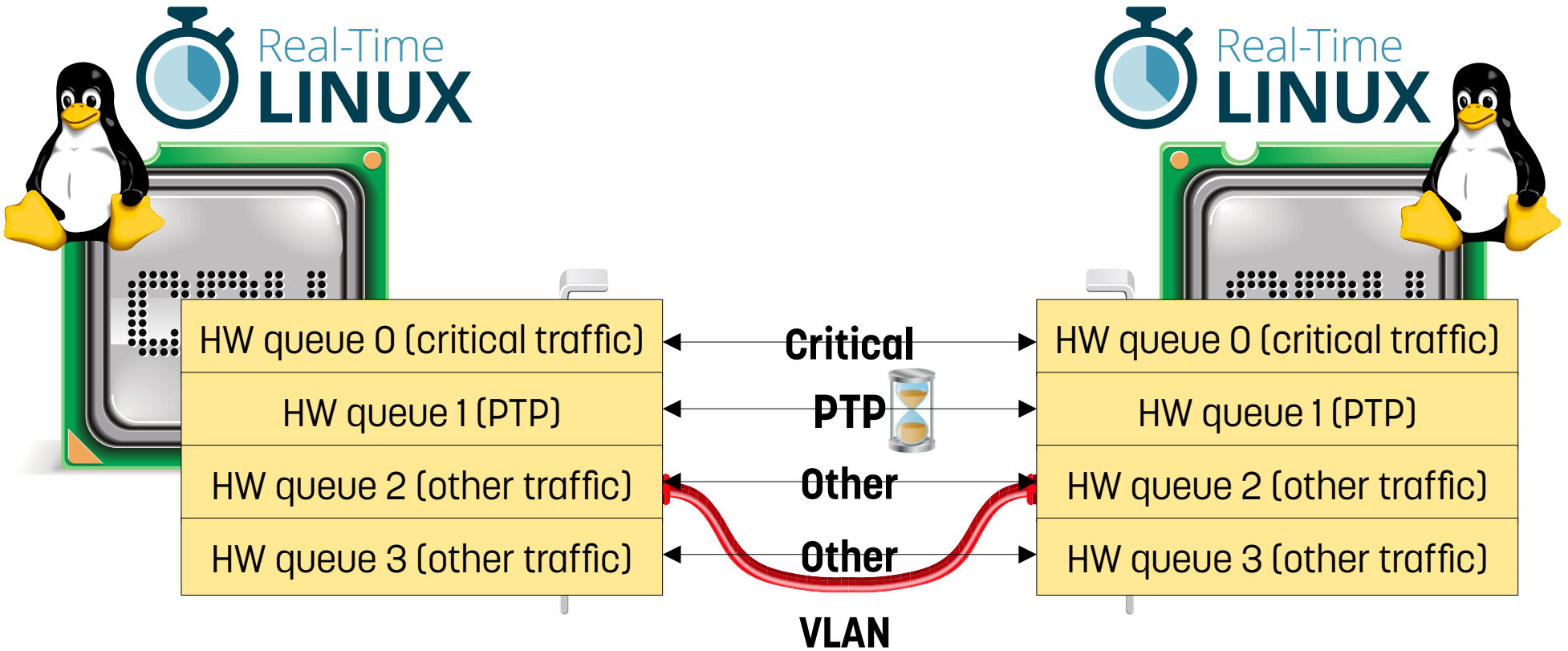
Setup


Real-Time
LINUX
CPU
HW queue 0 (critical traffic)
HW queue 1 (PTP)
HW queue 2 (other traffic)
HW queue 3 (other traffic)
Critical
PTP
Other
Other
Real-Time
LINUX
CPU

Setup


Real-Time
LINUX
Real-Time
LINUX
HW queue 0 (critical traffic)
HW queue 1 (PTP)
HW queue 2 (other traffic)
HW queue 3 (other traffic)
Critical
PTP
Other
Other
HW queue 0 (critical traffic)
HW queue 1 (PTP)
HW queue 2 (other traffic)
HW queue 3 (other traffic)

Setup


Real-Time
LINUX
Real-Time
LINUX
HW queue 0 (critical traffic)
HW queue 1 (PTP)
HW queue 2 (other traffic)
HW queue 3 (other traffic)
Critical
PTP
Other
Other
HW queue 0 (critical traffic)
HW queue 1 (PTP)
HW queue 2 (other traffic)
HW queue 3 (other traffic)
VLAN

Step 1

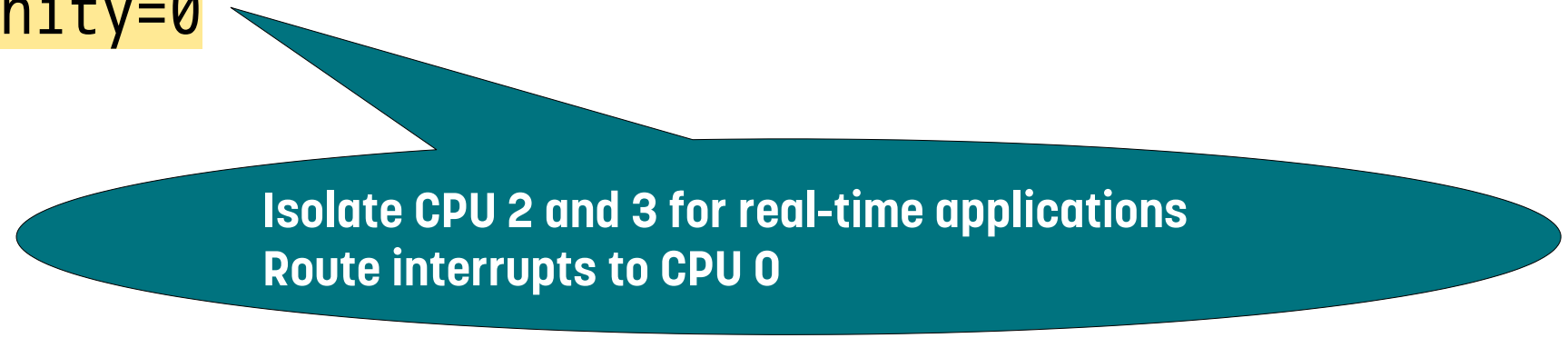
- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

Kernel commandline

```
isolcpus=2,3 rcu_nocbs=2,3 nohz_full=2,3
irqaffinity=0
[...]
```

Kernel commandline

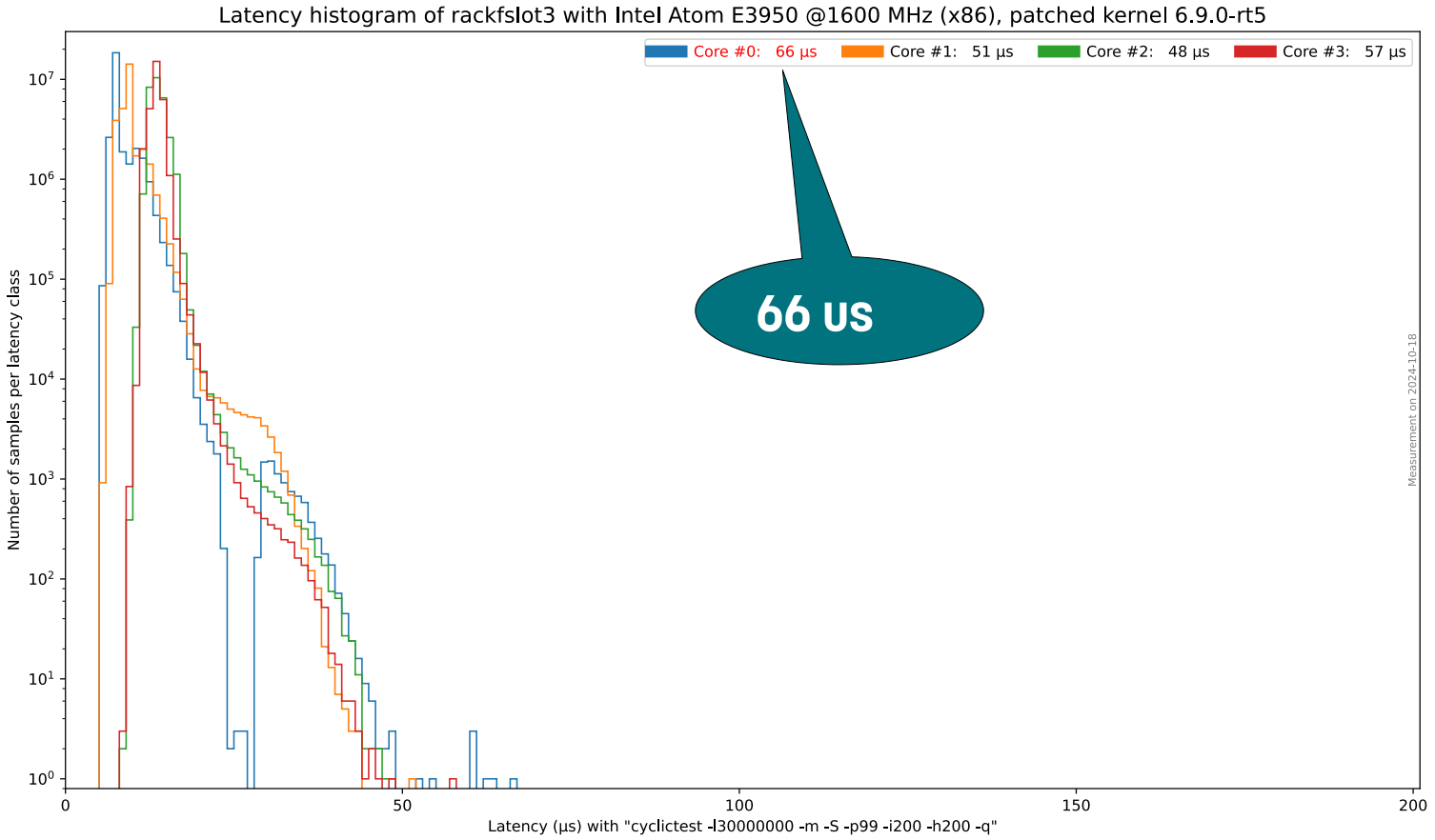
```
isolcpus=2,3 rcu_nocbs=2,3 nohz_full=2,3
irqaffinity=0
[...]
```



Setting up PREEMPT_RT

https://realtime-linux.org/getting-started-with-preempt_rt-guide/

Evaluate the real-time behavior

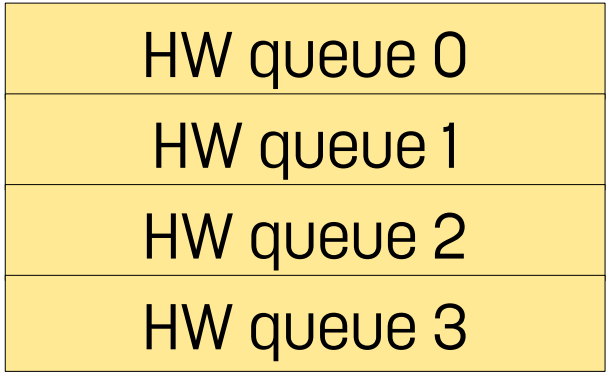


Step 2

- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

Set up 4 combined HW queues

```
$ ethtool -L ${INTERFACE} combined 4
$ ethtool -G ${INTERFACE} rx 4096 tx 4096
```



Disable energy efficiency

```
$ ethtool --set-eee ${INTERFACE} eee off
```

Step 3

- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

Queueing disciplines: mqprio

- Define the number of traffic classes
- Map socket priorities to traffic classes
- Assign the traffic classes to specific hardware queues

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

The number of traffic classes to be used

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Mapping the socket priorities to a traffic class

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Priority **0**

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Priority 1

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Priority **2**

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Priority **15**

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Priority **0** maps to traffic class **2**

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Priority **2** maps to traffic class **1**

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Priority **3** maps to traffic class **0**

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \

num_tc 3 \

map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \

queues 1@0 1@1 2@2 \

hw 0
```

Map the traffic classes into hardware queues (4 queues are being used here)

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Traffic class **0**

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Traffic class **1**

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

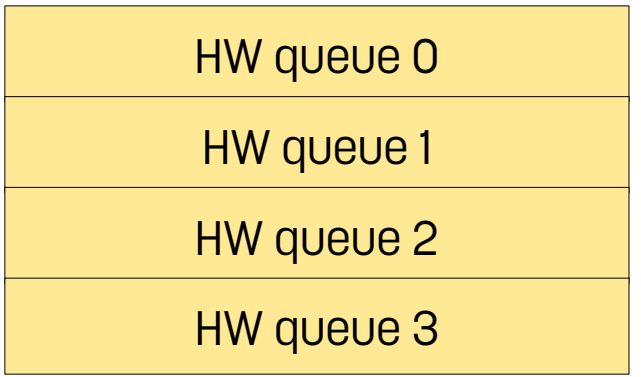
```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

Traffic class **2**

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

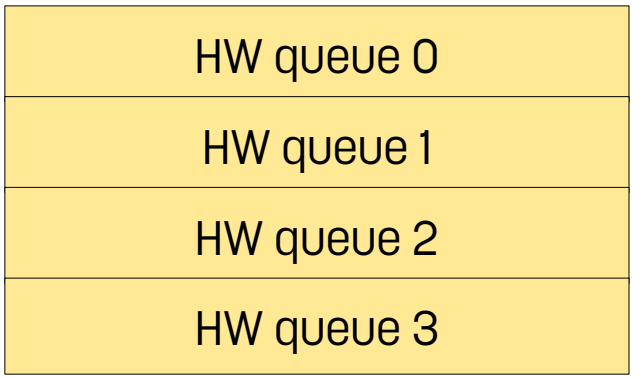


Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

TC0 goes into **1** queue

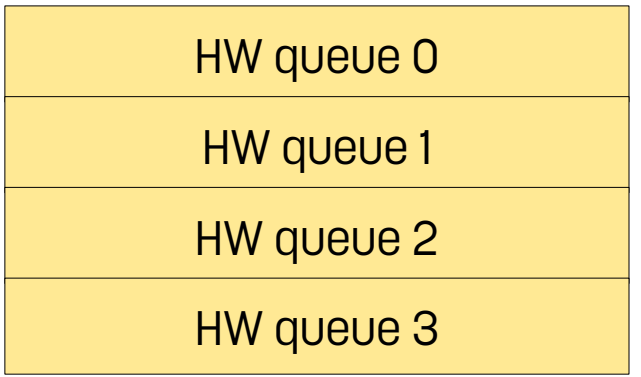


Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \

num_tc 3 \

map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \

queues 1@0 1@1 2@2 \

hw 0
```

TC0 goes into 1 queue starting from queue **0**



Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

TC0 goes into **1** queue starting from queue **0**

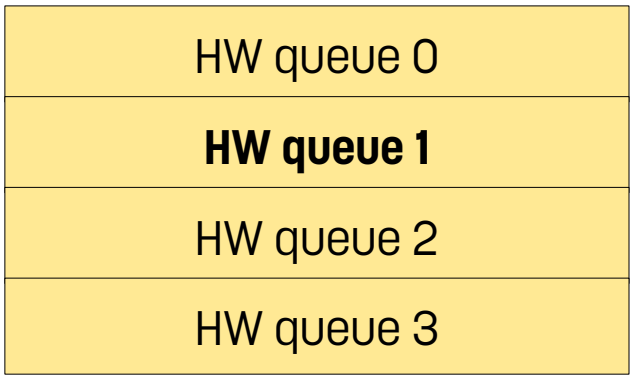
HW queue 0
HW queue 1
HW queue 2
HW queue 3

Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \

num_tc 3 \

map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \

queues 1@0 1@1 2@2 \

hw 0
```

TC1 goes into **1** queue starting from queue **1**

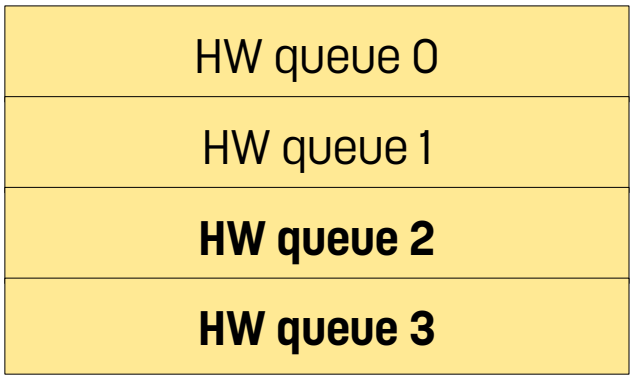


Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

TC2 goes into **2** queues starting from queue **2**



Queueing disciplines: mqprio

Manage qdiscs using the application **tc**:

```
tc qdisc add dev eth1 handle 100: parent root mqprio \
num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
queues 1@0 1@1 2@2 \
hw 0
```

0: No hardware offloading
1: Hardware offloading

```
# Check for hardware support
ethtool -k eth1 | grep hw-tc-offload
hw-tc-offload: on
```

Step 4

- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

Prioritization of the relevant threads

- Enable NAPI threading
- IRQ routing
- IRQ thread priorities
- NAPI thread priorities

Threaded NAPI

```
$ echo 1 > /sys/class/net/${INTERFACE}/threaded
```

CPU pinning and priorities

```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
[...]
  13874 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/3:1H-kblo
2055005 root     -76   0  238944  46456   2304 S  50.0   1.2   0:56.79 pubsub_TSN_publ
2055008 root     -76   0  238944  46456   2304 S   0.0   1.2   0:11.79 pubsub_TSN_publ
[...]
    666 root     -71   0       0      0      0 S   0.0   0.0   0:07.24 irq/132-enp4s0
    667 root     -99   0       0      0      0 S   0.0   0.0  25:06.80 irq/133-enp4s0-TxRx-0   2
    668 root     -99   0       0      0      0 S   0.0   0.0   0:31.43 irq/134-enp4s0-TxRx-1   2
    793 root     -99   0       0      0      0 S   0.0   0.0   0:30.76 napi/enp4s0-8198        2
    794 root     -99   0       0      0      0 S  50.0   0.0  27:21.56 napi/enp4s0-8197        2
1975955 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/2:1H            2
2055006 root     -79   0  238944  46456   2304 S   0.0   1.2   0:23.29 pubsub_TSN_publ         2
2055007 root     -82   0  238944  46456   2304 S   0.0   1.2   0:21.59 pubsub_TSN_publ         2
```

HW queue 0 (critical traffic)

HW queue 1 (PTP)

HW queue 2 (other traffic)

HW queue 3 (other traffic)

Step 5

- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

ETF

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \
clockid CLOCK_TAI \
delta 66000 \
offload
```

ETF

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \
clockid CLOCK_TAI \
delta 66000 \
offload
```

handle of the parent qdisc (defined before)

ETF

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \
clockid CLOCK_TAI \
delta 66000 \
offload
```

Traffic class:
1: TC0 2: TC1 3: TC2

ETF

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \
clockid CLOCK_TAI \
delta 66000 \
offload
```

Traffic class:
1: TC0 2: TC1 3: TC2

ETF

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \

clockid CLOCK_TAI \

delta 66000 \

offload
```

The clock to be used for scheduling events and measuring the time

ETF

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \
clockid CLOCK_TAI \
delta 66000 \
offload
```

Can be used to take the scheduler latency into account

ETF

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \

clockid CLOCK_TAI \

delta 66000 \

offload
```

Can be used to take the scheduler latency into account
Next wake up = next tx time - **delta**

ETF

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \

clockid CLOCK_TAI \

delta 66000 \
```

`offload` Use the “Launch Time” feature of the NIC (disabled by default)

Queueing disciplines: taprio

- Implements a simplified version of the scheduling state machine.

Queueing disciplines: taprio

- Implements a simplified version of the scheduling state machine.
- Define the number of traffic classes
- Map socket priorities to traffic classes
- Assign the traffic classes to specific hardware queues
- **Define a sequence of gate states (for traffic classes)**

Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
   num_tc 3 \
   map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
   queues 1@0 1@1 1@2 \
   base-time 0000 \
   sched-entry S 4 100000 \
   sched-entry S 2 150000 \
   sched-entry S 1 200000 \
   flags 2
```

Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```

Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```

Defines the start time of the schedule

Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```

Define the schedule

Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```

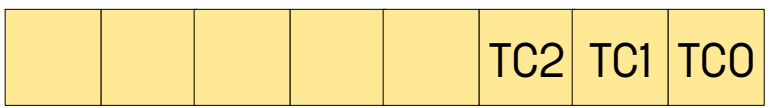
Command: SetGateStates

Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```

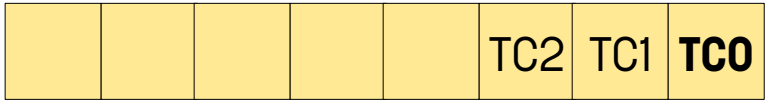
Traffic class assignment (bitmask)



Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

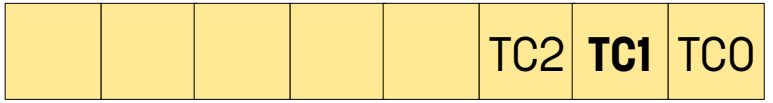
```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```



Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

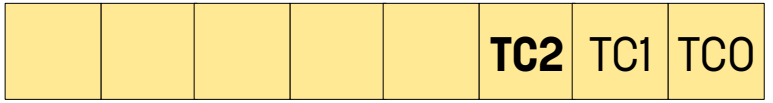
```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```



Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

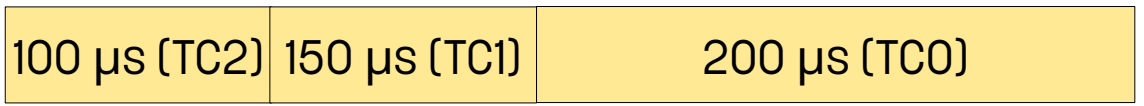
```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```



Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

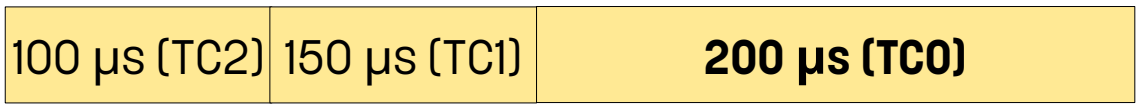
```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```



Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

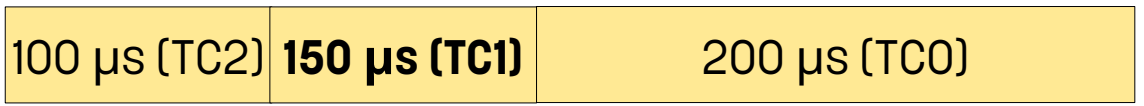
```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```



Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

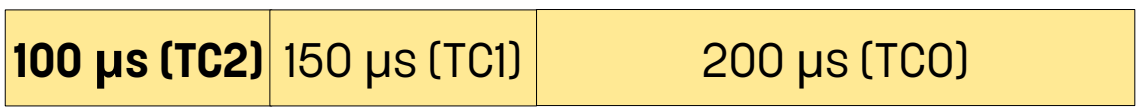
```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```



Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```



Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```

taprio can operate in different modes:
0x1: Enable the txassist feature (utilizing ETF)
0x2: Enable the full offload feature
(schedule is executed by the NIC)
0x1 and 0x2 are mutually exclusive!

Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```

taprio can operate in different modes:
0x1: Enable the txassist feature (utilizing ETF)
0x2: Enable the full offload feature
(schedule is executed by the NIC)
0x1 and 0x2 are mutually exclusive!

Assign ETF to the related queues when using txassist

Queueing disciplines: taprio

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
    num_tc 3 \
    map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 \
    queues 1@0 1@1 1@2 \
    base-time 0000 \
    sched-entry S 4 100000 \
    sched-entry S 2 150000 \
    sched-entry S 1 200000 \
    flags 2
```

clockid must be omitted when using offload mode

(schedule is executed by the NIC)

0x1 and 0x2 are mutually exclusive!

Assign ETF to the related queues when using txassist

Using taprio with txassist

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
   [...]
   base-time 0000 \
   sched-entry S 2  50000 \
   sched-entry S 1 200000 \
   flags 1 \
   txtime-delay 150000 \
   clockid CLOCK_TAI
```

Using taprio with txassist

Manage the taprio qdisc using the application **tc**:

```
tc qdisc replace dev eth1 parent root handle 100 taprio \
   [...]
   base-time 0000 \
   sched-entry S 2  50000 \
   sched-entry S 1 200000 \
   flags 1 \
   txtime-delay 66000 \
   clockid CLOCK_TAI
```

Using taprio with txassist

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \

clockid CLOCK_TAI \

delta 66000 \

offload \

skip_sock_check
```

ETF and txassist

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \

clockid CLOCK_TAI \

delta 66000 \

offload \

skip_sock_check
```

Without this option ETF will drop any packet that does not have a socket assigned or if the socket does not have SO_TXTIME set.

ETF and txassist

Manage qdiscs using the application **tc**:

```
tc qdisc replace dev eth1 parent 100:1 etf \

clockid CLOCK_TAI \

delta 66000 \

offload \

skip_sock_check
```

In the case of taprio the transmit time is set by another qdisc and no socket is associated. So, we need to skip this check.

Step 6

- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

VLAN

```
auto enp4s0.8
iface enp4s0.8 inet static
address 192.168.4.1
netmask 255.255.255.0
```

VLAN

```
auto enp4s0.8
iface enp4s0.8 inet static
address 192.168.4.2
netmask 255.255.255.0
```

Step 7

- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

Setup virtual networks (VLAN)

- Using qdiscs traffic classes are mapped to hardware queues and outgoing traffic can be scheduled.
- In a next step outgoing packets have to be “marked”, so the different traffic classes can be differentiated on the network.
- This is done using the Port Control Protocol (PCP) field.
- Socket priorities can be mapped to the PCP field using VLANs.

Setup virtual networks (VLAN)

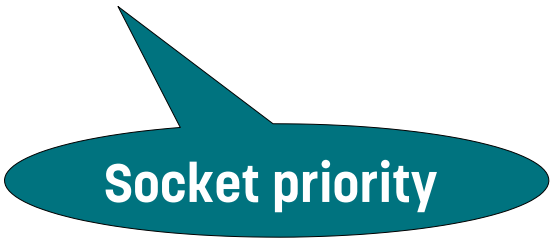
Map the traffic classes to the PCP field:

```
# ip link set enp4s0.8 type vlan \
egress 2:2 3:3
```

Setup virtual networks (VLAN)

Map the traffic classes to the PCP field:

```
# ip link set enp4s0.8 type vlan \
egress 2:2 3:3
```



Setup virtual networks (VLAN)

Map the traffic classes to the PCP field:

```
# ip link set enp4s0.8 type vlan \
egress 2:2 3:3
```

Step 8

- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

Filtering incoming traffic

```
ethtool -K ${INTERFACE} ntuple on

# PCP 3 -> HW queue 0
ethtool -N ${INTERFACE} flow-type ether vlan 0x6000 m 0x1fff action 0
# PCP 2 -> HW queue 1
ethtool -N ${INTERFACE} flow-type ether vlan 0x4000 m 0x1fff action 1

# The rest goes to HW queue 2 and 3
ethtool -N ${INTERFACE} flow-type ether vlan 0xc000 m 0x1fff action 2
[...]
ethtool -N ${INTERFACE} flow-type ether vlan 0xe000 m 0x1fff action 3
```

Filtering incoming traffic

```
ethtool -K ${INTERFACE} ntuple on

# PCP 3 -> HW queue 0
ethtool -N ${INTERFACE} flow-type ether vlan 0x6000 m 0x1fff action 0
# PCP 2 -> HW queue 1
ethtool -N ${INTERFACE} flow-type ether vlan 0x4000 m 0x1fff action 1

# The rest goes to HW queue 2 and 3
ethtool -N ${INTERFACE} flow-type ether vlan 0xc000 m 0x1fff action 2
[...]
ethtool -N ${INTERFACE} flow-type ether vlan 0xe000 m 0x1fff action 3
```

Filtering incoming traffic

```
ethtool -K ${INTERFACE} ntuple on

# PCP 3 -> HW queue 0
ethtool -N ${INTERFACE} flow-type ether vlan 0x6000 m 0x1fff action 0
# PCP 2 -> HW queue 1
ethtool -N ${INTERFACE} flow-type ether vlan 0x4000 m 0x1fff action 1

# The rest goes to HW queue 2
ethtool -N ${INTERFACE} flow-type ether vlan 0x2000 m 0x1fff action 2
[...]
ethtool -N ${INTERFACE} flow-type ether vlan 0xe000 m 0x1fff action 3
```

PCP field 2 goes to HW queue 1

Filtering incoming traffic

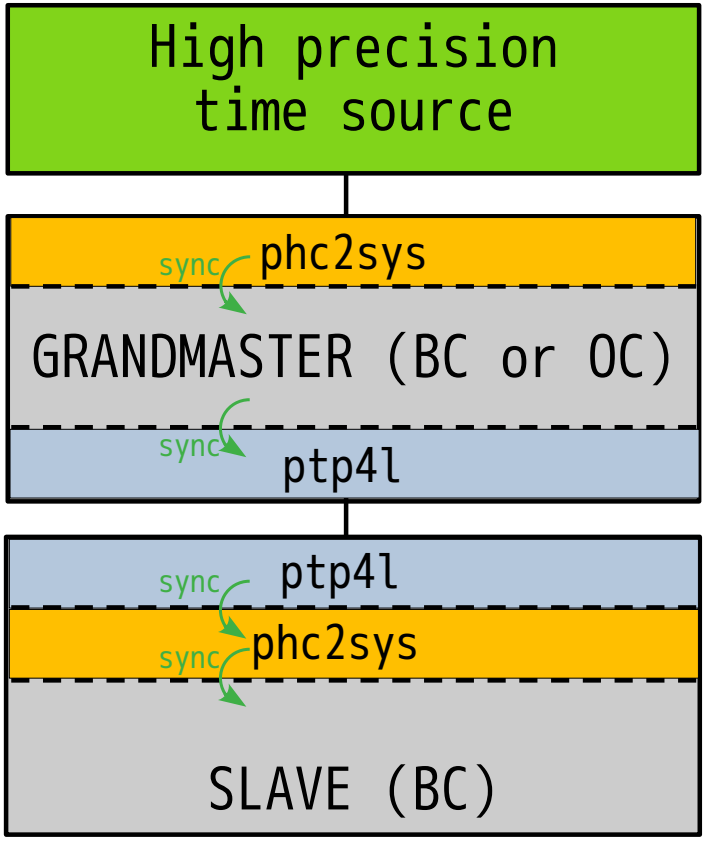
```
ethtool -K ${INTERFACE} ntuple on

# PCP 3 -> HW queue 0
ethtool -N ${INTERFACE} flow-type ether vlan 0x6000 m 0x1fff action 0
# PCP 2 -> HW queue 1
ethtool -N ${INTERFACE} flow-type ether vlan 0x4000 m 0x1fff action 1

# The rest goes to HW queue 2 and 3
ethtool -N ${INTERFACE} flow-type ether vlan 0xc000 m 0x1fff action 2
[...]
ethtool -N ${INTERFACE} f1                    0x1fff action 3
```

Other traffic goes to HW queues 2 and 3

Step 9

- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

Time synchronization with PTP



- Exactly one time provider is needed per network (the “grandmaster”). This grandmaster knows the time, since it
  - has a built-in highly accurate clock,
  - obtains the correct time from an external trusted source, or
  - does both.
- As many time consumers (“slaves”) as required may be installed.
- Slaves may be connected peer-to-peer or via switches.

PTP synchronization on the Grandmaster

Send network time information via PTP

```
$ ptp4l -A -i enp1s0  --socket_priority=2 -f /etc/linuxptp/ptp4l.conf \
  -2 -mq >/var/log/ptp4l.out 2>/var/log/ptp4l.err
```

Synchronize system time

```
$ phc2sys -c enp1s0 -s CLOCK_REALTIME \
-w -mq >/var/log/phc2sys.out 2>/var/log/phc2sys.err
```

PTP synchronization on the Grandmaster

Send network time information via PTP

```
$ ptp4l -A -i enp1s0  --socket_priority=2 -f /etc/linuxptp/ptp4l.conf \
  -2 -mq >/var/log/ptp4l.out 2>/var/log/ptp4l.err
```

Synchronize system time

```
$ phc2sys -c enp1s0 -s CLOCK_REALTIME \
-w -mq >/var/log/phc2sys.out 2>/var/log/phc2sys.err
```

HW queue 0 (critical traffic)
HW queue 1 (PTP)
HW queue 2 (other traffic)
HW queue 3 (other traffic)

PTP time synchronization on the slave

Receive network time information via PTP

```
ptp4l -A -i enp1s0 --socket_priority=2 -2 -mq -s \
-f /etc/linuxptp/ptp4l.conf >/var/log/ptp4l.out 2>/var/log/ptp4l.err &
```

Synchronize system time

```
phc2sys -s enp1s0 -w -mq >/var/log/phc2sys.out 2>/var/log/phc2sys.err &
```

PTP time synchronization on the slave

Receive network time information via PTP

```
ptp4l -A -i enp1s0 --socket_priority=2 -2 -mq -s \
-f /etc/linuxptp/ptp4l.conf >/var/log/ptp4l.out 2>/var/log/ptp4l.err &
```

run as slave

Synchronize system time

```
phc2sys -s enp1s0 -w -mq >/var/log/phc2sys.out 2>/var/log/phc2sys.err &
```

Step 10

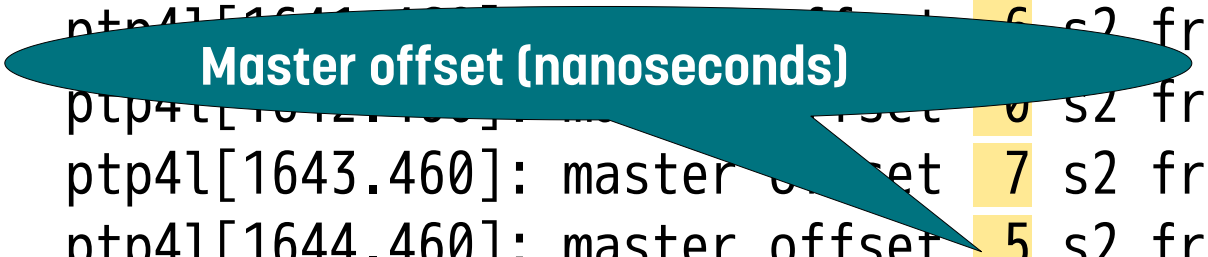
- Real-time operating system setup (PREEMPT_RT)
- Hardware configuration
- Traffic classes
- System level configuration
- Queuing disciplines (QDISCs)
- Networking setup
- VLAN mapping
- Routing incoming traffic
- Time synchronization
- Evaluation

Time synchronization

```
ptp4l[1640.460]: master offset -1 s2 freq  -11427 path delay -13
ptp4l[1641.460]: master offset -6 s2 freq  -11432 path delay -13
ptp4l[1642.460]: master offset  0 s2 freq  -11428 path delay -17
ptp4l[1643.460]: master offset  7 s2 freq  -11421 path delay -18
ptp4l[1644.460]: master offset  5 s2 freq  -11421 path delay -18
ptp4l[1645.460]: master offset -8 s2 freq  -11433 path delay -18
ptp4l[1646.460]: master offset -7 s2 freq  -11434 path delay -18
ptp4l[1647.460]: master offset  8 s2 freq  -11421 path delay -18
```

Time synchronization

```
ptp4l[1640.460]: master offset -1 s2 freq  -11427 path delay -13
ptp4l[1641.460]: master offset  6 s2 freq  -11432 path delay -13
ptp4l[1642.460]: master offset  0 s2 freq  -11428 path delay -17
ptp4l[1643.460]: master offset  7 s2 freq  -11421 path delay -18
ptp4l[1644.460]: master offset  5 s2 freq  -11421 path delay -18
ptp4l[1645.460]: master offset -8 s2 freq  -11433 path delay -18
ptp4l[1646.460]: master offset -7 s2 freq  -11434 path delay -18
ptp4l[1647.460]: master offset  8 s2 freq  -11421 path delay -18
```



Time synchronization

```
# pmc -u -b 0 'GET CURRENT_DATA_SET'
sending: GET CURRENT_DATA_SET
4c0289.fffe.184d72-0 seq 0 RESPONSE MANAGEMENT CURRENT_DATA_SET
stepsRemoved     1
offsetFromMaster 26.0
meanPathDelay    -20.0
```

Time synchronization

```
# pmc -u -b 0 'GET CURRENT_DATA_SET'
sending: GET CURRENT_DATA_SET
4c0289.fffe.184d72-0 seq 0 RESPONSE MANAGEMENT CURRENT_DATA_SET
stepsRemoved     1
offsetFromMaster 26.0
meanPathDelay    -20.0
```

Example: Testing taprio

```
# plget -m rx-lat \
-f ipgap \
-n 16 \
-i enp4s0.8 \
-t udp \
-u 5000
```

Example: Testing taprio

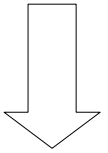
```
# plget -i enp4s0.8 \
-m pkt-gen \
-p1 \
-t udp \
-u 5000 \
-n 16 \
-a 192.168.4.2 \
-l 1500
```

```
# plget -m rx-lat \
-f ipgap \
-n 16 \
-i enp4s0.8 \
-t udp \
-u 5000
```

Example: Testing taprio

```
# plget -i enp4s0.8 \
-m pkt-gen \
-p1 \
-t udp \
-u 5000 \
-n 16 \
-a 192.168.4.2 \
-l 1500
```

```
# plget -m rx-lat \
-f ipgap \
-n 16 \
-i enp4s0.8 \
-t udp \
-u 5000
```



```
gap of hw rx time, us: packets 16:
-----------------------------------------------------------------------
      0 |  12.193 | 12.192 | 12.193 | 12.192 | 12.193 | 12.192 | 12.193 |
 12.192 | 402.465 | 12.192 | 12.193 | 12.192 | 12.193 | 12.192 | 12.193 |
-----------------------------------------------------------------------
```

Example: Testing taprio

```
# plget -i enp4s0.8 \
-m pkt-gen \
-p1 \
-t udp \
-u 5000 \
-n 16 \
-a 192.168.4.2 \
-l 1500
```

```
# plget -m rx-lat \
-f ipgap \
-n 16 \
-i enp4s0.8 \
-t udp \
-u 5000
```

```
gap of hw rx time, us: packets 16:
-----------------------------------------------------------------------
     0 |  12.193 | 12.192 | 12.193 | 12.192 | 12.193 | 12.192 | 12.193 |
12.192 | 402.465 | 12.192 | 12.193 | 12.192 | 12.193 | 12.192 | 12.193 |
-----------------------------------------------------------------------
```

Example: Testing taprio

```
# plget -i enp4s0.8 \
-m pkt-gen \
-p1 \
-t udp \
-u 5000 \
-n 16 \
-a 192.168.4.2 \
-l 1500
```

```
sched-entry S 4 100000
sched-entry S 2 150000
sched-entry S 1 250000
```

```
# plget -m rx-lat \
-f ipgap \
-n 16 \
-i enp4s0.8 \
-t udp \
-u 5000
```

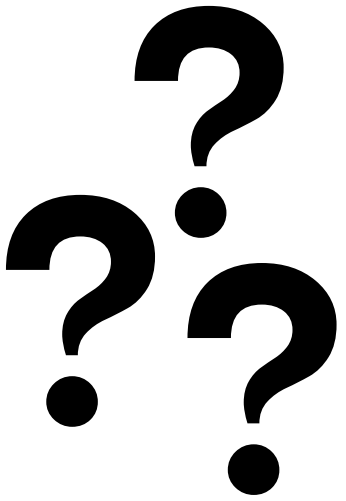
```
gap of hw rx time, us: packets 16:
-------------------------------------------------------------------------
     0 |  12.193 | 12.192 | 12.193 | 12.192 | 12.193 | 12.192 | 12.193 |
12.192 | 402.465 | 12.192 | 12.193 | 12.192 | 12.193 | 12.192 | 12.193 |
-------------------------------------------------------------------------
```

Example: Testing taprio

```
# plget -i enp4s0.8 \
-m pkt-gen \
-p1 \
-t udp \
-u 5000 \
-n 16 \
-a 192.168.4.2 \
-l 1500
```

```
sched-entry S 4 100000
sched-entry S 2 150000
sched-entry S 1 250000
```

```
# plget -m rx-lat \
-f ipgap \
-n 16 \
-i enp4s0.8 \
-t udp \
-u 5000
```

```
gap of hw rx time, us: packets 16:
-----------------------------------------------------------------------
     0 |  12.193 | 12.192 | 12.193 | 12.192 | 12.193 | 12.192 | 12.193 |
12.192 | 402.465 | 12.192 | 12.193 | 12.192 | 12.193 | 12.192 | 12.193 |
-----------------------------------------------------------------------
```

Questions?