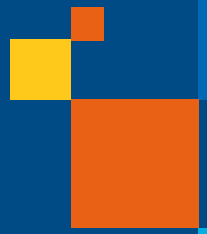
netdev 0x17

synce4l: open-source implementation of Synchronous Ethernet

Michał Michalik
Arkadiusz Kubalewski



intel®

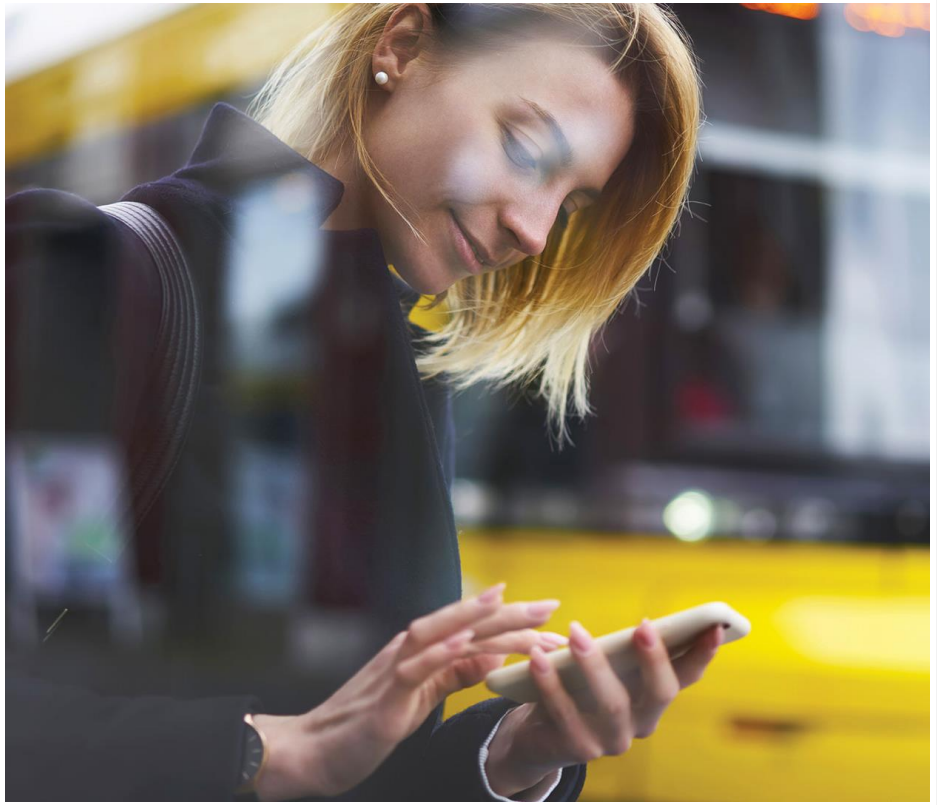
Agenda

1. The need for time, phase and frequency synchronization
    a) Precision Time Protocol (PTP)
    b) Synchronous Ethernet
2. synce4l – implementation of the SyncE signaling protocol
    a) Application overview
    b) Synchronized networks examples
    c) Hardware requirements
    d) Recent changes & DPLL kernel subsystem
    e) Application configuration
    f) Future of project

The need for time, phase and frequency synchronization

Quarter of the Internet traffic is pushed through mobile networks (~75 exabytes / month in 2022).

Achieving that need good synchronization between wired components in 5G/6G.

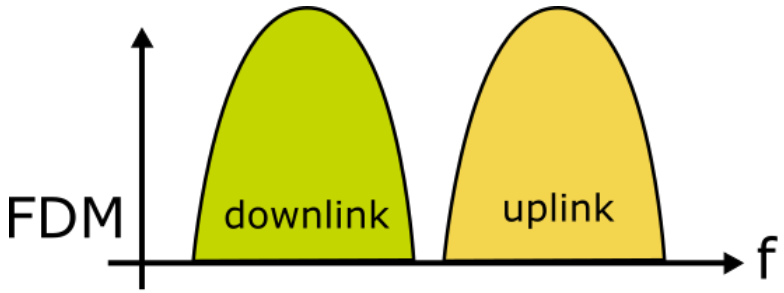


Source: https://twiki.cern.ch/twiki/pub/HEPIX/TechwatchNetwork/HtwNetworkDocuments/white-paper-c11-741490.pdf

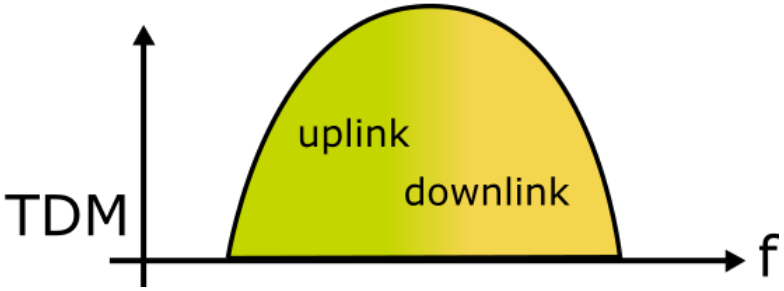
Frequency Division Multiplexing (FDM)

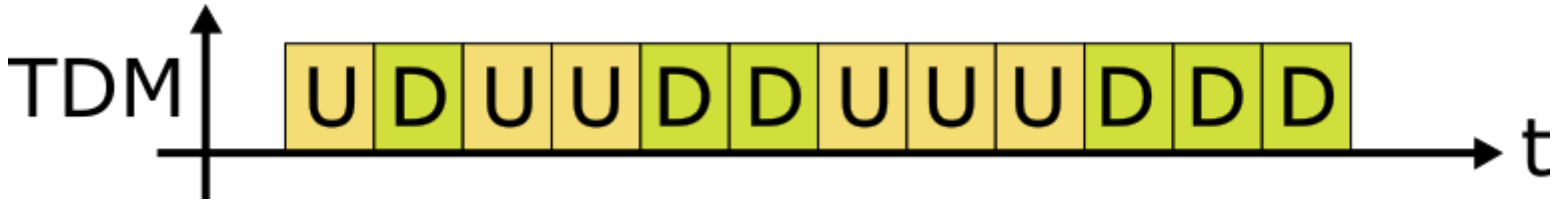
vs.

Time Division Multiplexing (TDM)



TDM uses the same frequency for the uplink and downlink transmission.

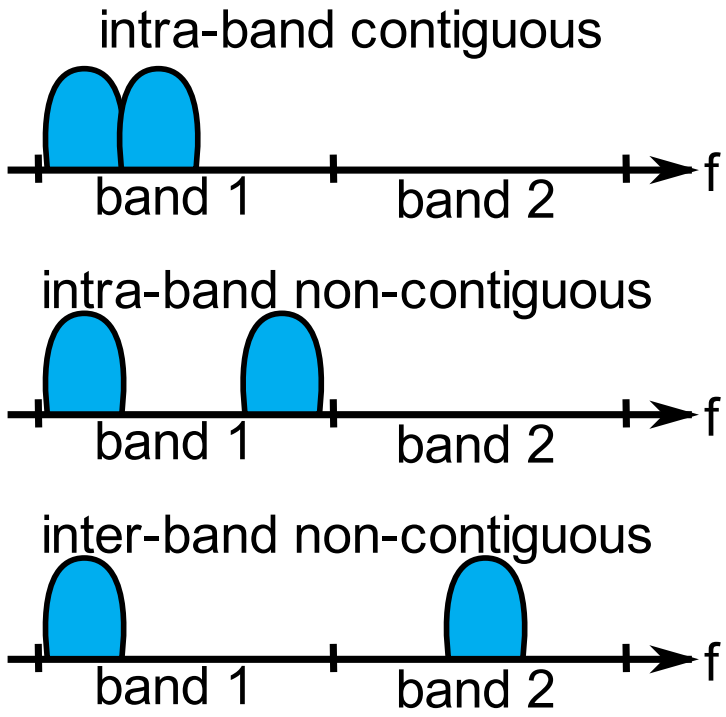




Source: https://www.nokia.com/blog/improving-synchronization-accuracy-with-help-from-synce/

Synchronization requirements for 5G:

Application	Frequency	Phase/time
5G FDM	+/- 16 ppb	N/A
5G TDM	+/- 16 ppb	+/- 1500ns
IB non-contiguous CA FR2 IB contiguous CA FR1	+/- 16 ppb	260 ns (relative)
IB contiguous CA FR2	+/- 16 ppb	130 ns (relative)



Glossary: IB – intra-band, CA – carrier aggregation, FR – frequency range, FR1 – less than 7GHz, FR2 above 24GHz

Sources: https://www.nokia.com/blog/improving-synchronization-accuracy-with-help-from-synce/
https://www.3gpp.org/technologies/101-carrier-aggregation-explained
https://www.nokia.com/about-us/newsroom/articles/5g-carrier-aggregation-explained/

What happens if we fail to satisfy the requirements?

- Call interference / poor call quality / dropped calls
- Problems with call handovers between cells
- Decreased efficiency of bandwidth utilization
- Poor signal quality on the edge of a cell
- Interference between the cells
- Lost packets in data transmission

Source: https://www.nokia.com/blog/improving-synchronization-accuracy-with-help-from-synce/

How to fulfil such a strict requirements?

One possible solution:

GNSS receiver at each node:

- high cost of installation and maintenance,
- prone to signal loss and jamming

More practical solution

Hybrid approach:

GNSS as primary reference clock

PTP + SyncE as a distribution

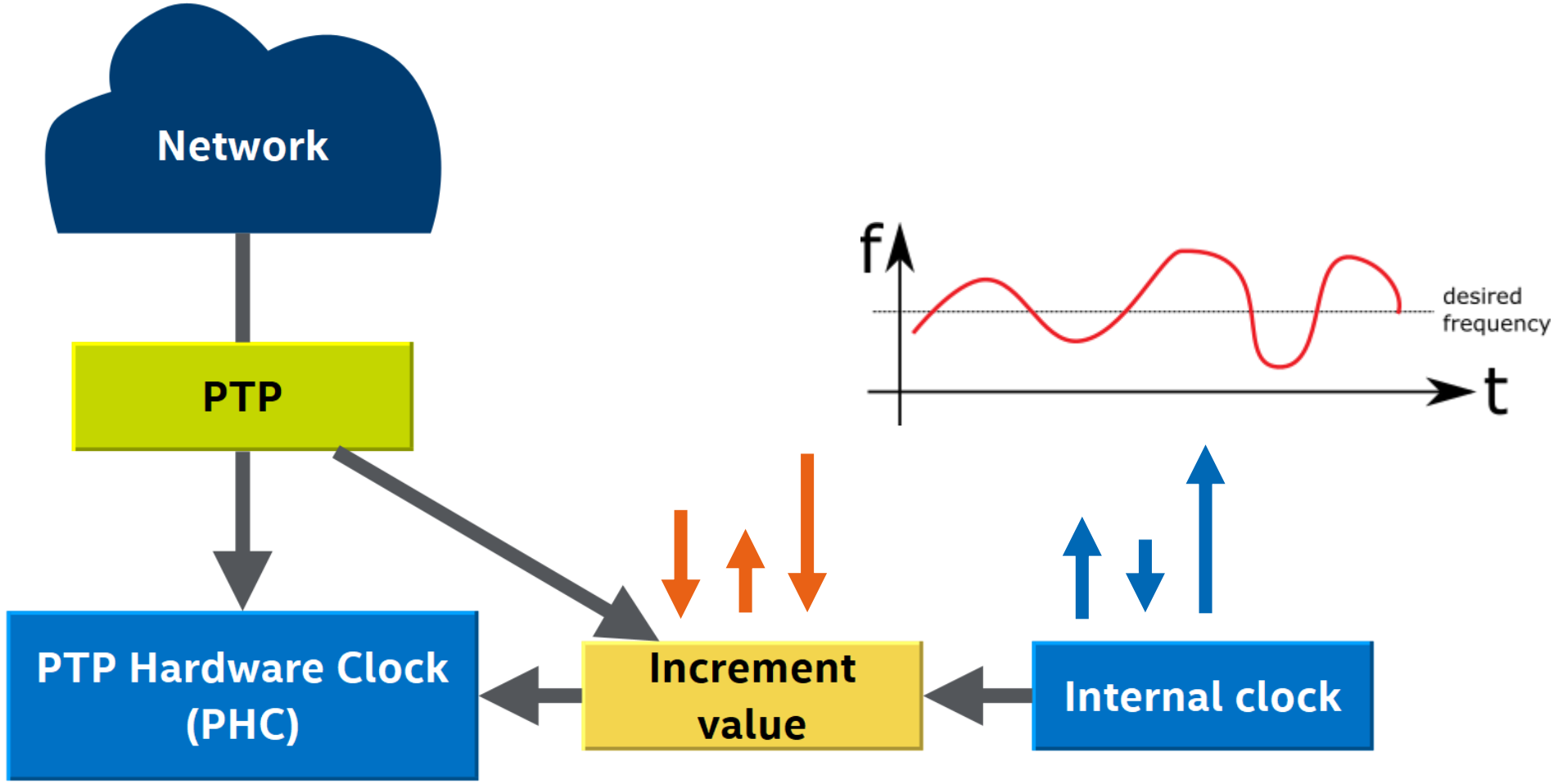
Precision Time Protocol (PTP)

- defined by IEEE 1588
- allowing precise synchronization of distributed clocks
- synchronize time of day, frequency and phase
- achieves sub-microsecond* accuracy

*not good enough for 5G/6G carrier aggregation (CA) applications 130/260ns

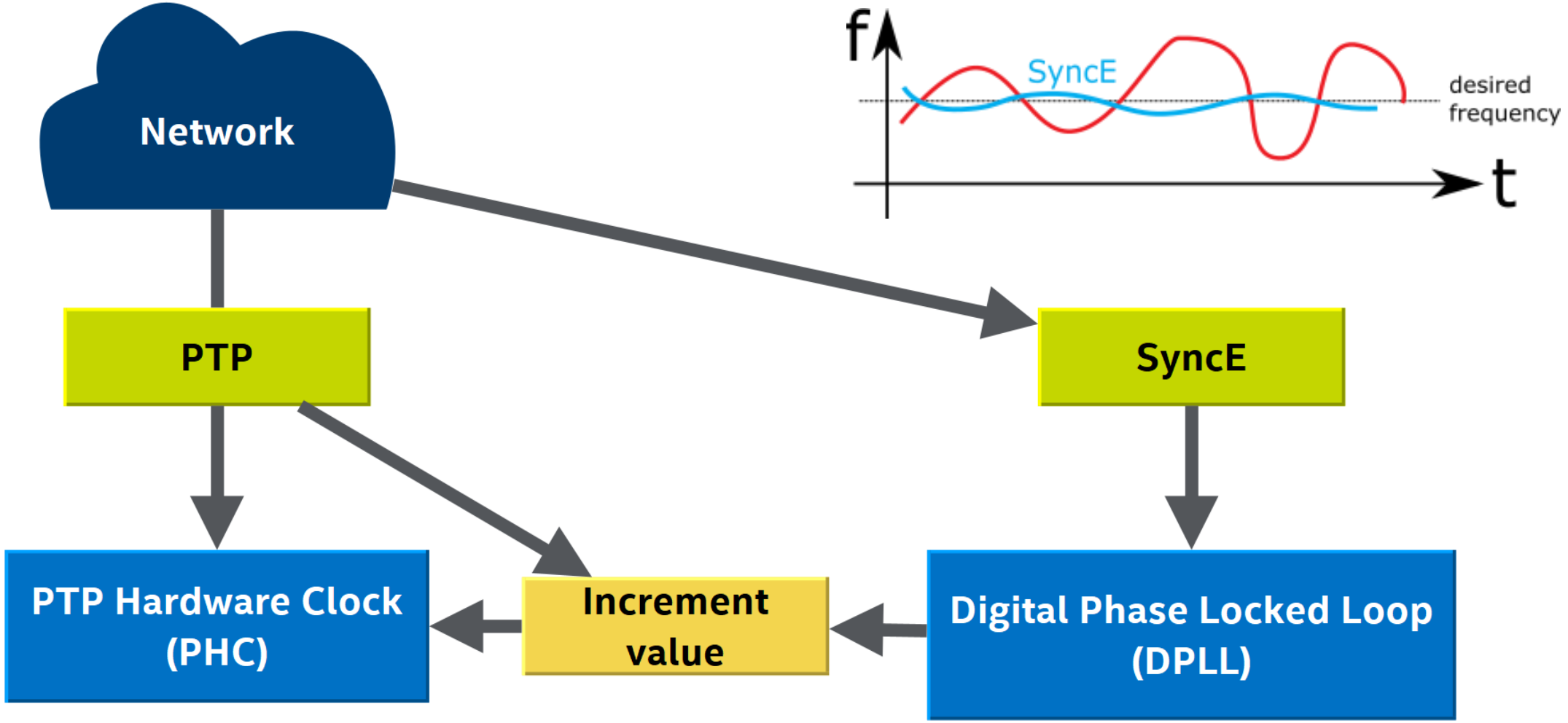
Sources: IEEE 1588-2008: https://standards.ieee.org/ieee/1588/4355/
https://netdevconf.info/0x15/session.html?Precision-Time-Protocol-optimization-using-genetic-algorithm
https://netdevconf.info/0x16/sessions/talk/introduction-to-time-synchronization.html

PTP – basis of the protocol operation



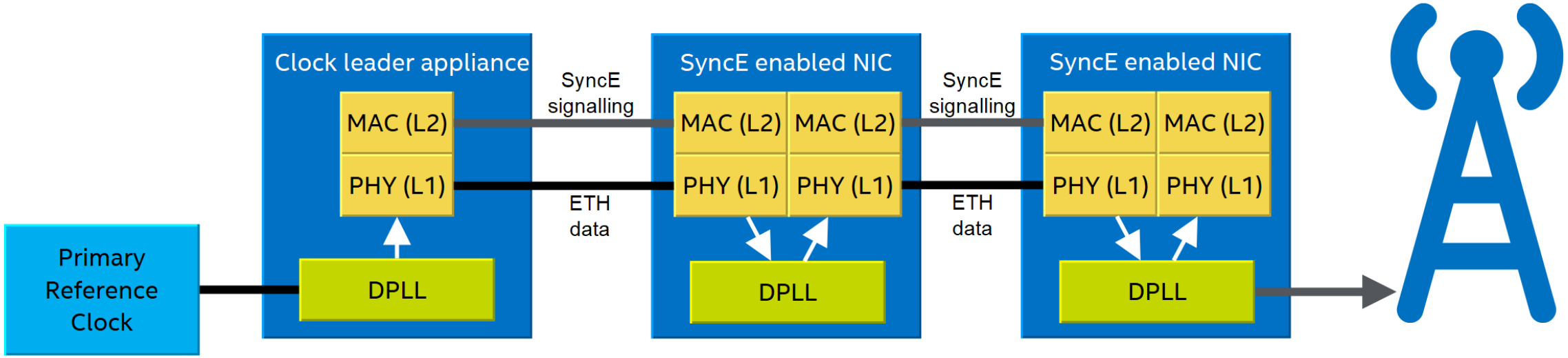
IEEE 1588-2008: https://standards.ieee.org/ieee/1588/4355/

PTP – basis of the protocol operation



IEEE 1588-2008: https://standards.ieee.org/ieee/1588/4355/

Synchronous Ethernet (SyncE), protocol transferring clock signals over the Ethernet physical layer.



Sources : https://en.wikipedia.org/wiki/Synchronous_Ethernet

SyncE was standardized by the ITU-T, in cooperation with IEEE, as three recommendations:

1. **ITU-T Rec. G.8261** - architecture and the wander performance of SyncE networks
2. **ITU-T Rec. G.8262** - specifies clocks for SyncE
3. **ITU-T Rec. G.8264** - specification of Ethernet Synchronization Messaging Channel (ESMC)

Source : https://en.wikipedia.org/wiki/Synchronous_Ethernet

Synchronous Ethernet – system requirements

- **Hardware requirements**

Network interface card (NIC) need to:

1. Recover the clock from PHY
2. Feed the clock to PHYs
3. Read the state of the DPLL

- **Software requirements**

Signalization protocol is needed to broadcast the clock qualities.

Here comes synce4l!

synce4l – implementation of the SyncE signaling protocol

- hardware agnostic open-source project
- implementation of SyncE signaling
- base on ITU-T Recommendation G.8264
- process Ethernet Synchronization Messaging Channel (ESMC)
- control Ethernet Equipment Clock (EEC)

Repository: https://github.com/intel/synce4l

Ethernet Synchronization Messaging Channel (ESMC)

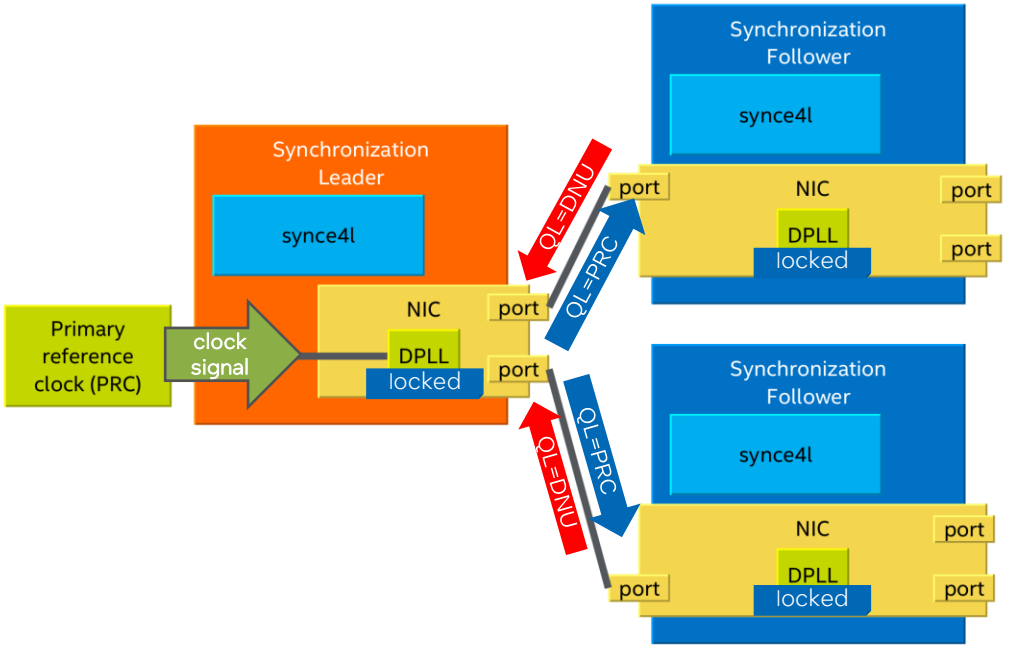
- Organization specific slow protocol (OSSP)
- defined in ITU-T Rec. G.8264
- distribute timing information
- carries clock quality on the receiving link
- transfer infrequent information through a single link
- slow protocols defined in Annex 57A and 57B of IEEE Std 802.3-2022.

Sources: https://www.itu.int/rec/T-REC-G.8264, https://standards.ieee.org/ieee/802.3/10422/
https://netdevconf.info/0x15/session.html?Introduction-to-time-synchronization-over-Ethernet

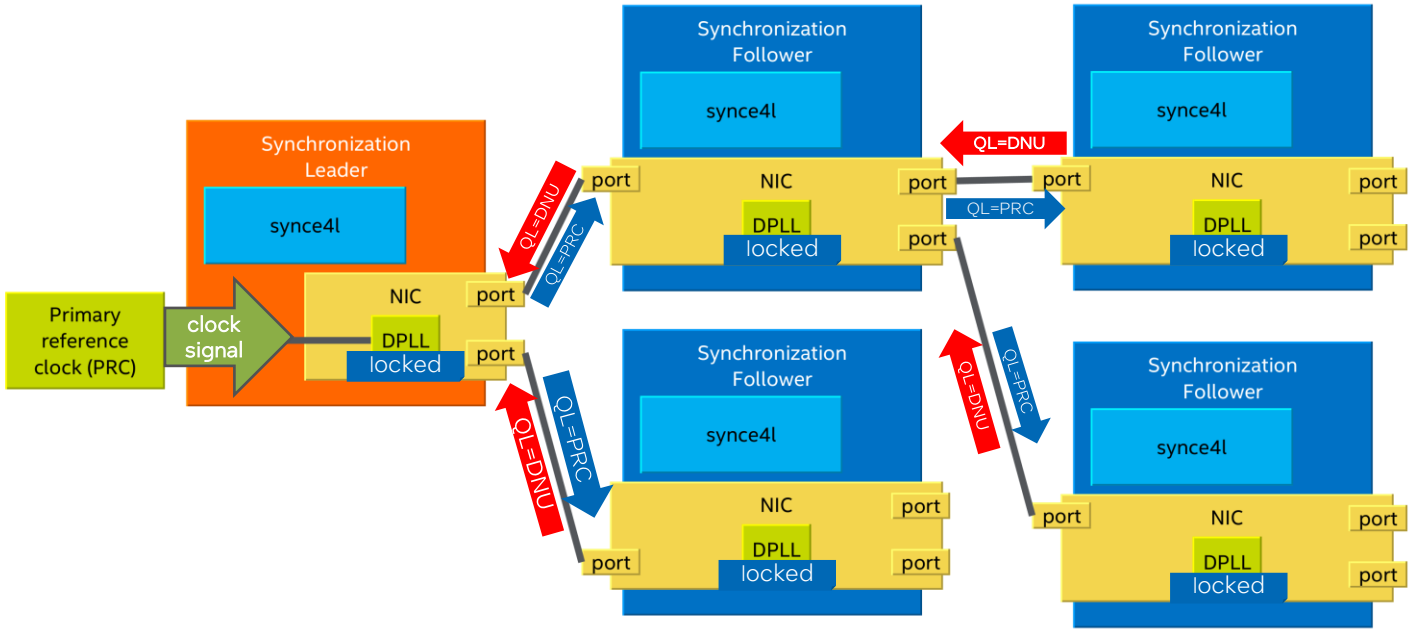
synce4l overview

- similarities to linuxptp/ptp4l
- a daemon application
- possible tracing to stdout and/or to syslog
- requires a configuration file
- requires RAW socket and it configures hardware (sudo it)

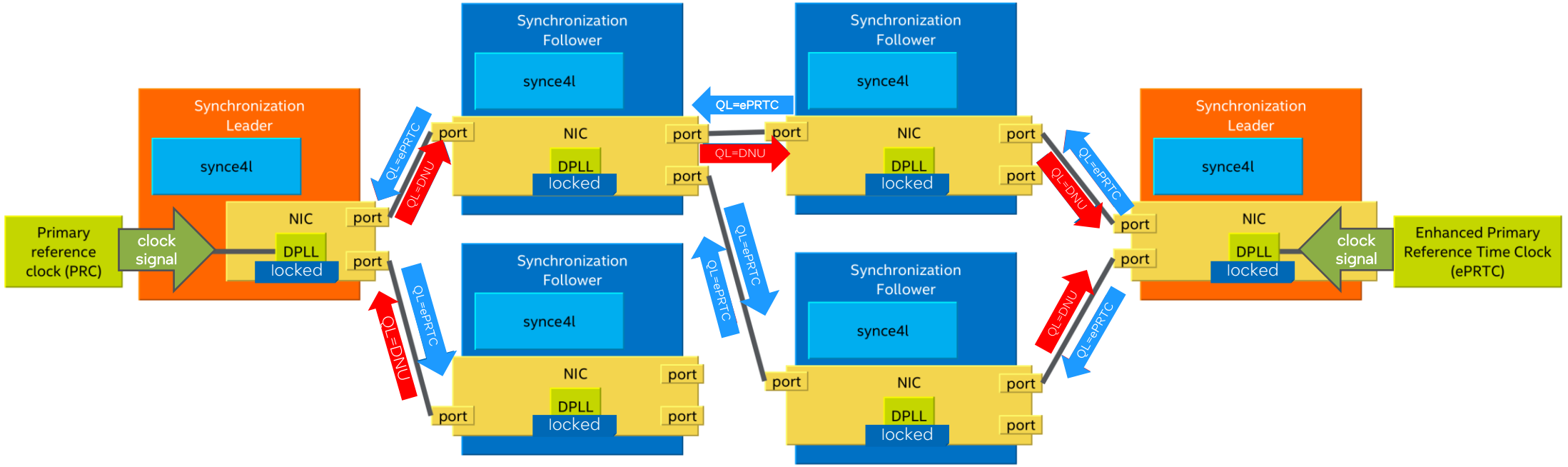
synce4l – Synchronized network - example 1



synce4l – Synchronized network - example 2



synce4l – Synchronized network - example 3



Linux dpll subsystem

- No to HW/vendor specific solutions
- Common APIs/subsystem

Thanks to community efforts mostly of Meta, Nvidia, RHEL and Intel the common DPLL API has been accepted in the upstream net-next tree:

https://lwn.net/ml/netdev/169494842736.21621.10730860855645661664.git-patchwork-notify%40kernel.org/

Linux dpll subsystem

The common dpll API in the upstream focus on:

- dpll device and pins configuration
- status reporting

Interface is based on the generic netlink for the transport of the commands and events notifications.

synce4l – latest version 1.0.0

The 1.0.0 version introduces two major features:

- unified approach for handling Line and External sources, allow mixing sources on one EEC,
- add support for Linux dpll subsystem - recently merged into net-next tree, still waiting for merge in Linus tree 6.7.

synce4l – work modes

synce4l interacts with a hardware with one of two different approaches:

- **legacy mode** (based on Linux system commands)
- **dpll mode** (based on Linux dpll subsystem)

synce4l – configuration

Configuration file consist of sections, there are 3 section types:

- global - `[global]`
- device - `[<device1>]`
- source - `[if_name], [{ext_clk_name}]`

The source can be one of 2 types:

- line (a NIC port) - `[eth0]`
- external (anything the hardware can support: GNSS/SMA/etc) - `[{GNSS}]`

synce4l – configuration – global section

```
[global]
logging_level    7
use_syslog       0
verbose          1
message_tag      [synce4l]
```

synce4l – configuration – device section

```
[<device_cmd_based>]
extended_tlv          1
network_option        1
recover_time          30
eec_holdover_value    4
eec_locked_ho_value   3
eec_locked_value      2
eec_freerun_value     1
eec_invalid_value     0
eec_get_state_cmd     cat /sys/class/net/eth2/device/dpll_0_state
```

```
[<device_dpll_based>]
extended_tlv      0
network_option    2
recover_time      60
clock_id          4658613174691613800
module_name       ice
dnu_prio          0xff
```

synce4l – configuration – multiple devices

```
[<Device1>]
...
[eth0]
...
[eth1]
...
[{GNSS}]
...
[<Device2>]
...
[eth4]
...
[{SMA1}]
```

synce4l – configuration – source section – legacy mode

```
[enp1s0f0]
tx_heartbeat_msec           1000
rx_heartbeat_msec           500
recover_clock_enable_cmd    echo 1 0 > /sys/class/net/enp1s0f0/device/phy/synce
recover_clock_disable_cmd   echo 0 0 > /sys/class/net/enp1s0f0/device/phy/synce

[enp1s0f1]
recover_clock_enable_cmd    echo 1 0 > /sys/class/net/enp1s0f1/device/phy/synce
recover_clock_disable_cmd   echo 0 0 > /sys/class/net/enp1s0f1/device/phy/synce

[{SMA1}]
input_QL                    0x2
input_ext_QL                0xFF
external_enable_cmd         echo 2 1 > /sys/class/net/enp1s0f0/device/ptp/ptp*/pins/SMA1
external_disable_cmd        echo 0 1 > /sys/class/net/enp1s0f0/device/ptp/ptp*/pins/SMA1
```

synce4l – configuration – source section – dpll mode

```
[ens2f0np0]
tx_heartbeat_msec    1000
rx_heartbeat_msec    500

[ens2f1np1]

[{SMA1}]
input_QL             0x2
input_ext_QL         0xFF
board_label          SMA1
#panel_label         <name>
#package_label       <name>
```

Future of synce4l

Plans for further releases of synce4l:

- Introduce support for live interaction with the configuration
- Introduce support for chaining devices
- Introduce support for MANUAL mode dpll device

Q/A

contact points:
michal.michalik@intel.com
arkadiusz.kubalewski@intel.com


intel