

Creating high-speed ethernet devices based on Linux & FPGA

Plan

FPGA-based network devices benefits

1G/10G/40G/100G NIC architecture

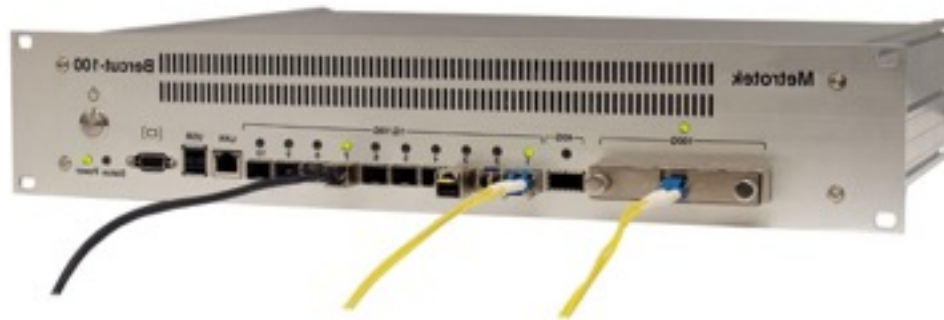
Network driver structure

Company & expertises

13 years of RnD

10M to 100G ethernet

Linux



HTЦ Метротек

Denis Gabidullin, Pavel Kurochkin

FPGA-based network devices benefits

Flexibility

Runtime upgradability

High Performance/Watt

Examples:

Software Defined Networks

Fast packet processing

Network device architecture

DRV: Send/rcv packets + mgmt

CPU I/F: Transfer packets to/from Linux

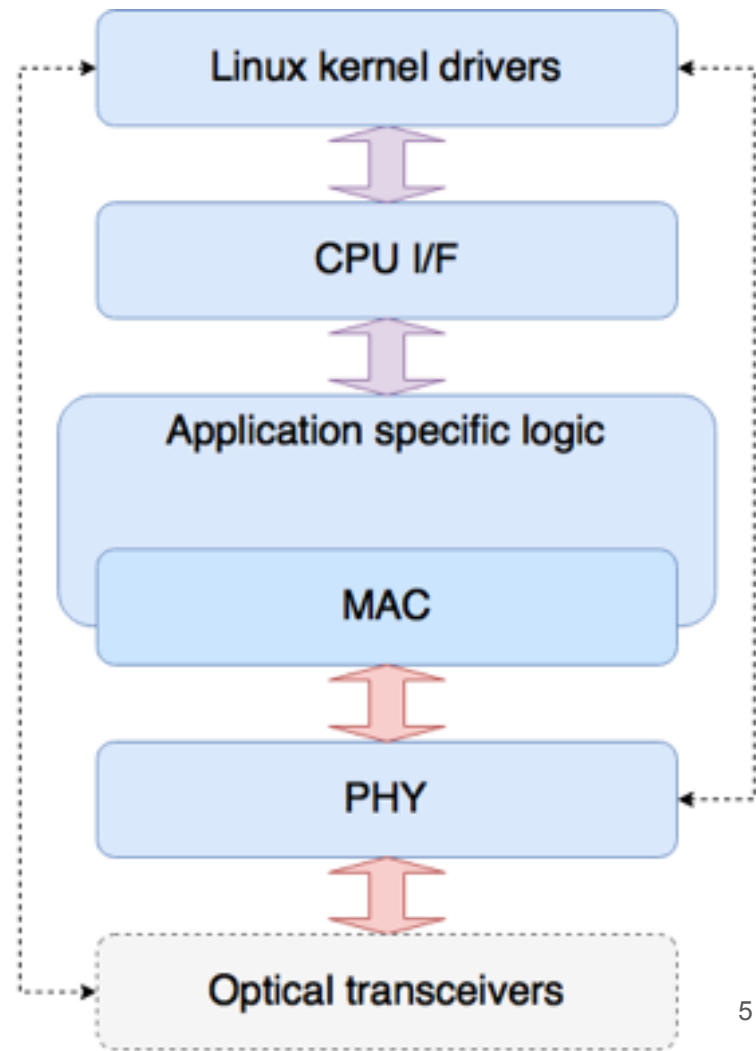
App: flexible logic

MAC: CRC32, Statistics

PHY: 8/10B, 64/66B coding

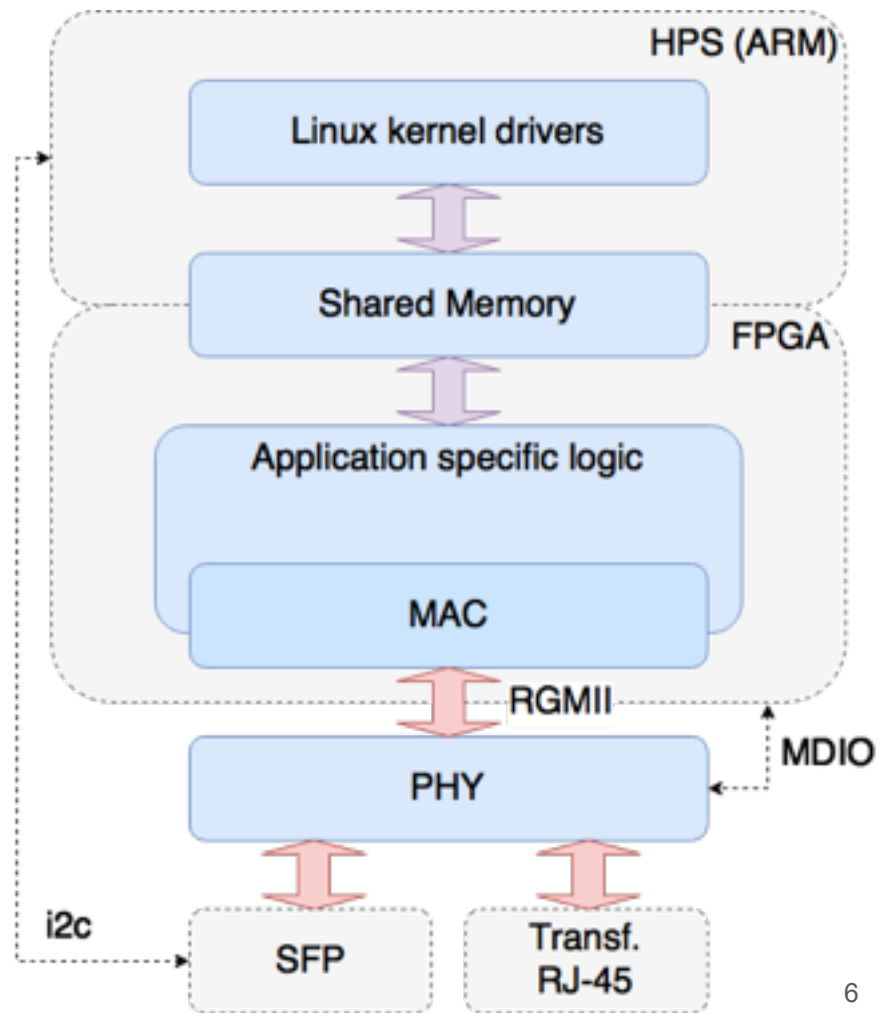
HTЦ Метротек

Denis Gabidullin, Pavel Kurochkin



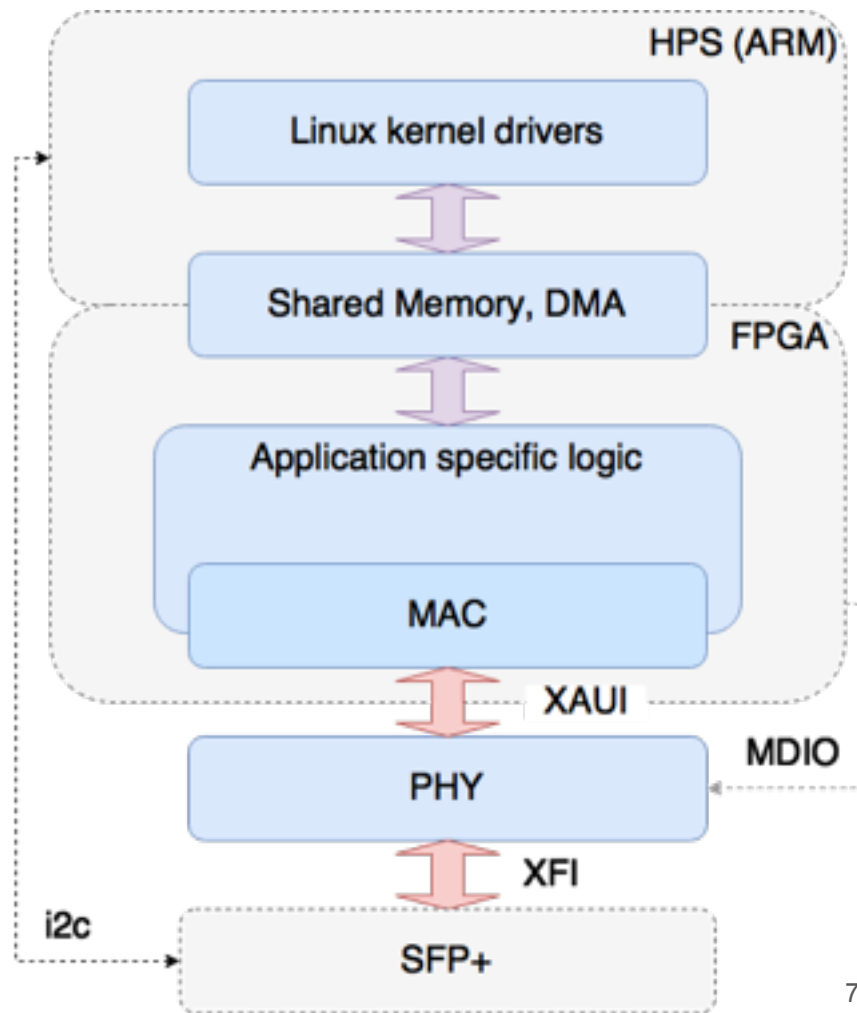
Gigabit Ethernet/SoC

- RGMII - 1 Gpbs
- SFP Serial 1.25 Gbps interface
- External transceiver
- Shared memory



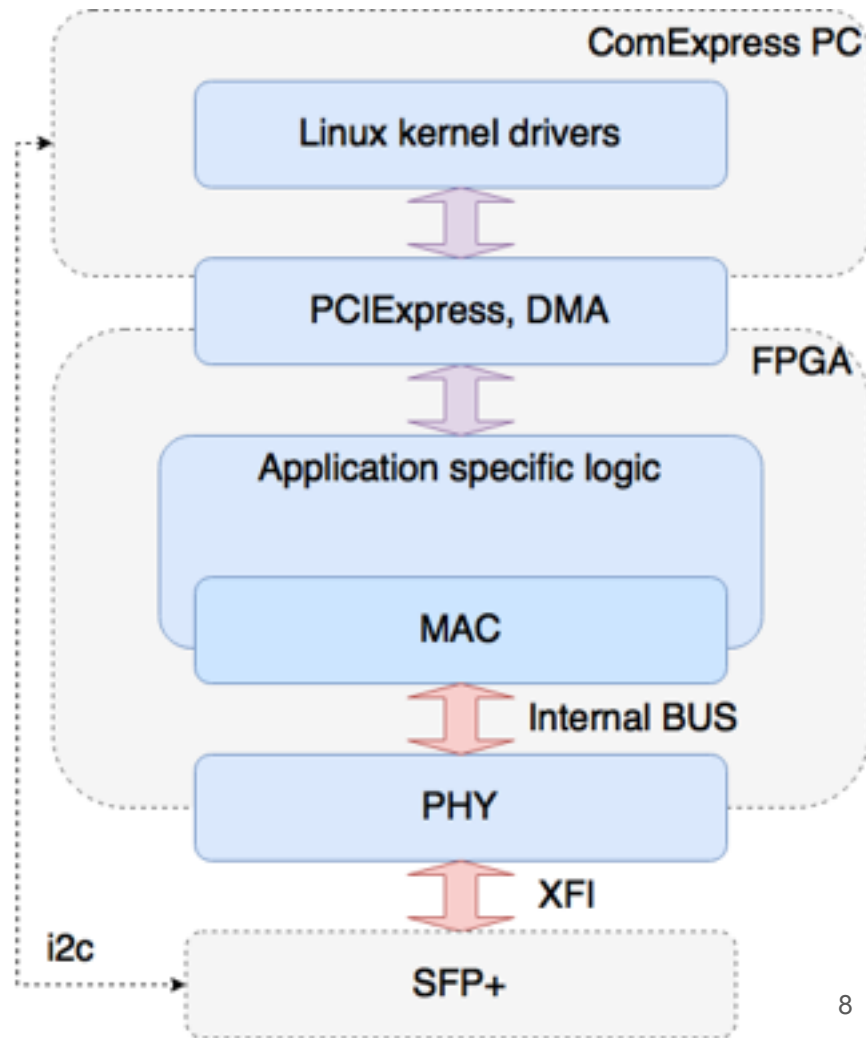
10 Gigabit Ethernet/SoC

- XAUI - 4 x 3.125 Gbps
- XFI - 10.3125 Gbps
- External transceiver
- Shared memory



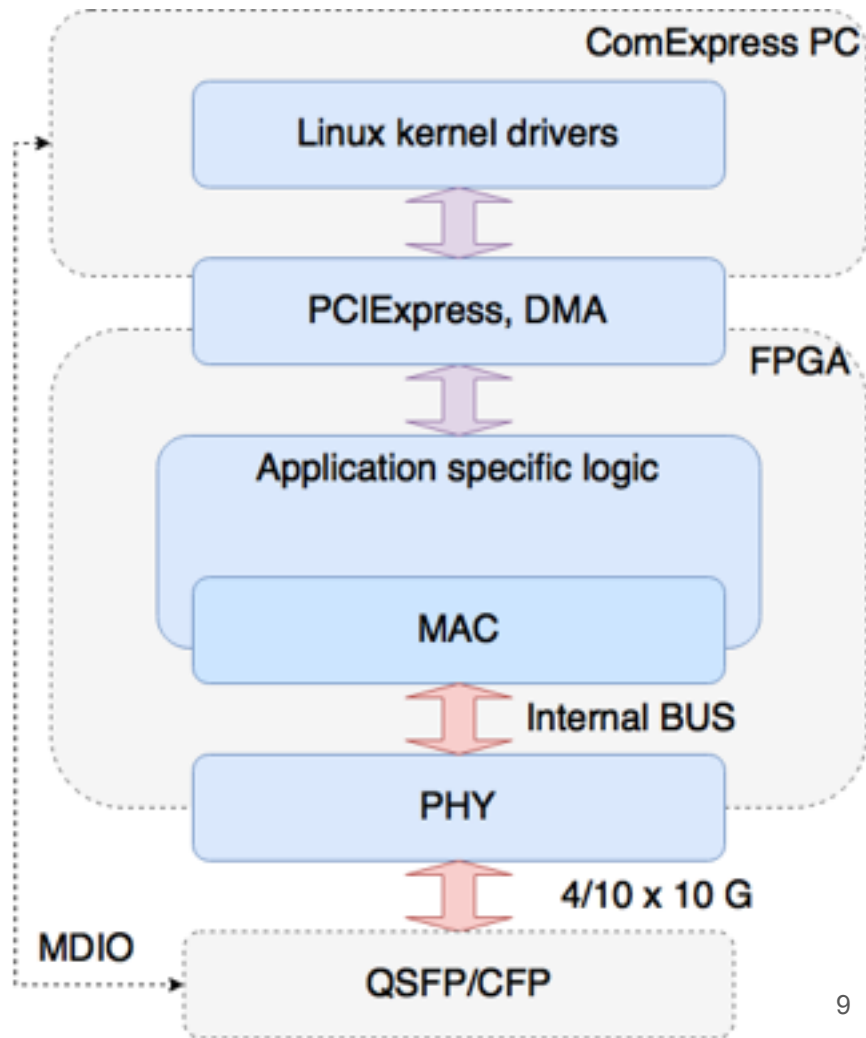
10 Gigabit Ethernet/PCIe

- XFI - 10.3125 Gbps
- Built-in transceiver
- PCIe
- GX-series FPGA



40/100 gigabit ethernet

- Interface - 4/10 x 10 Gbps
- Built-in transceivers
- PCIe
- High-end FPGA

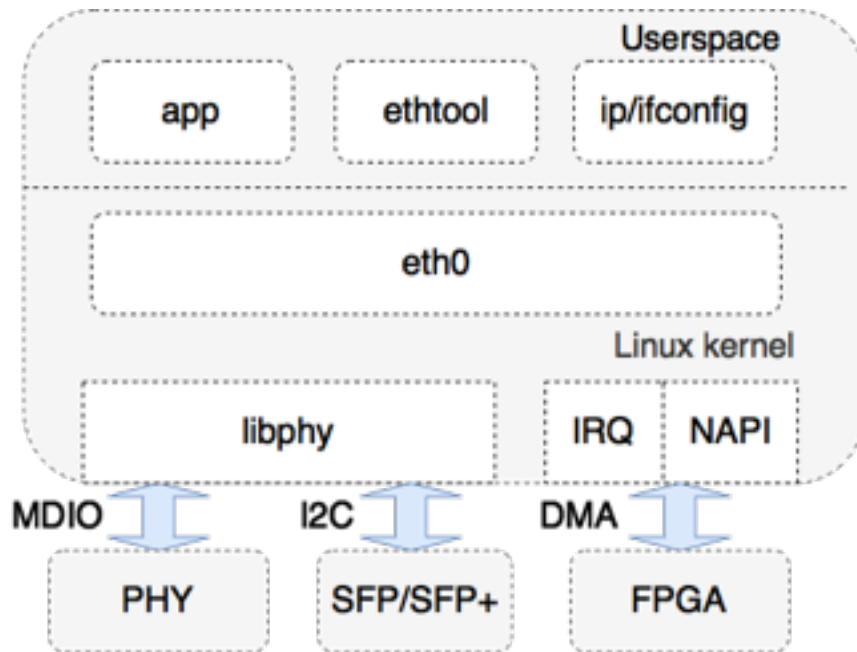


Network Driver Components

Packet transfer: NAPI,
IRQ, DMA

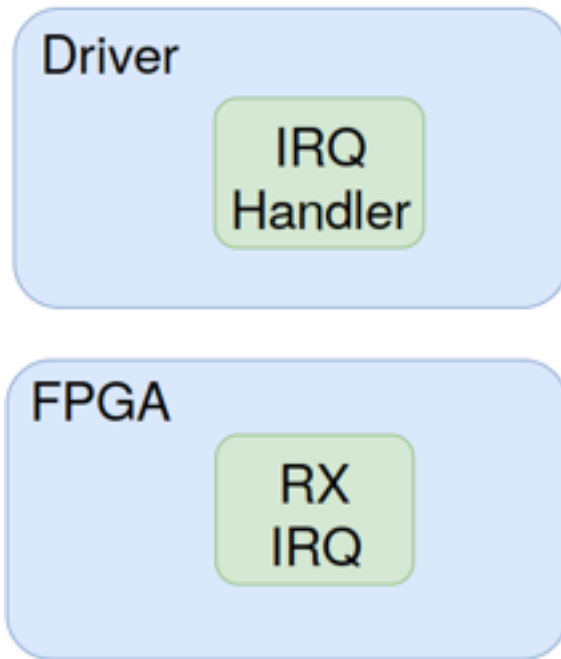
Phy/Link management: SFP,
CFP, MDIO

User space mgmt: ethtool
support

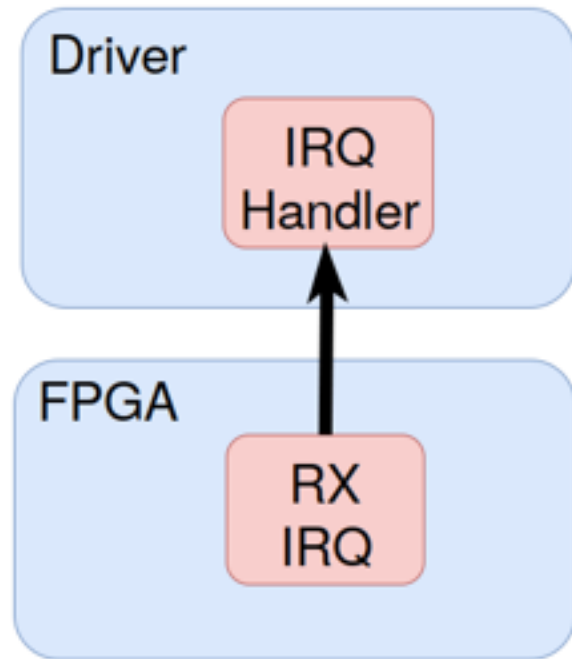


RX packets transfer -- IRQ only

1. Waiting RX packets

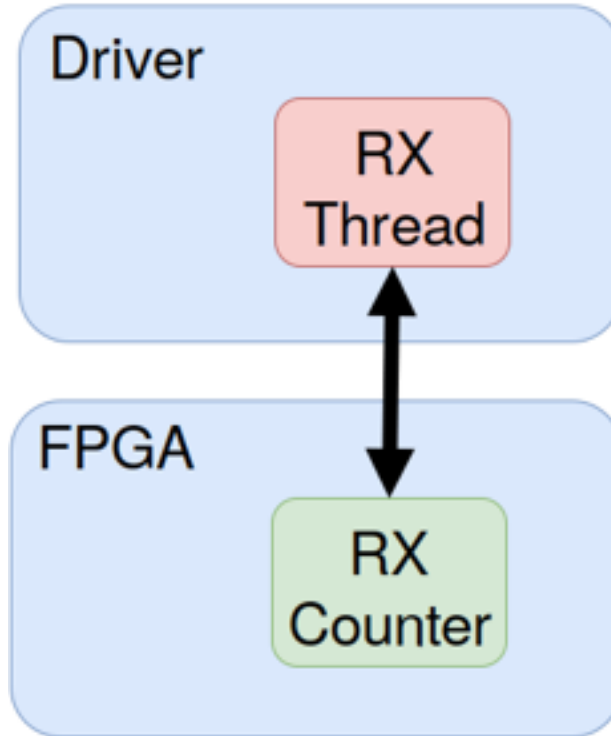


2. Get IRQ, pass packet to Network Stack

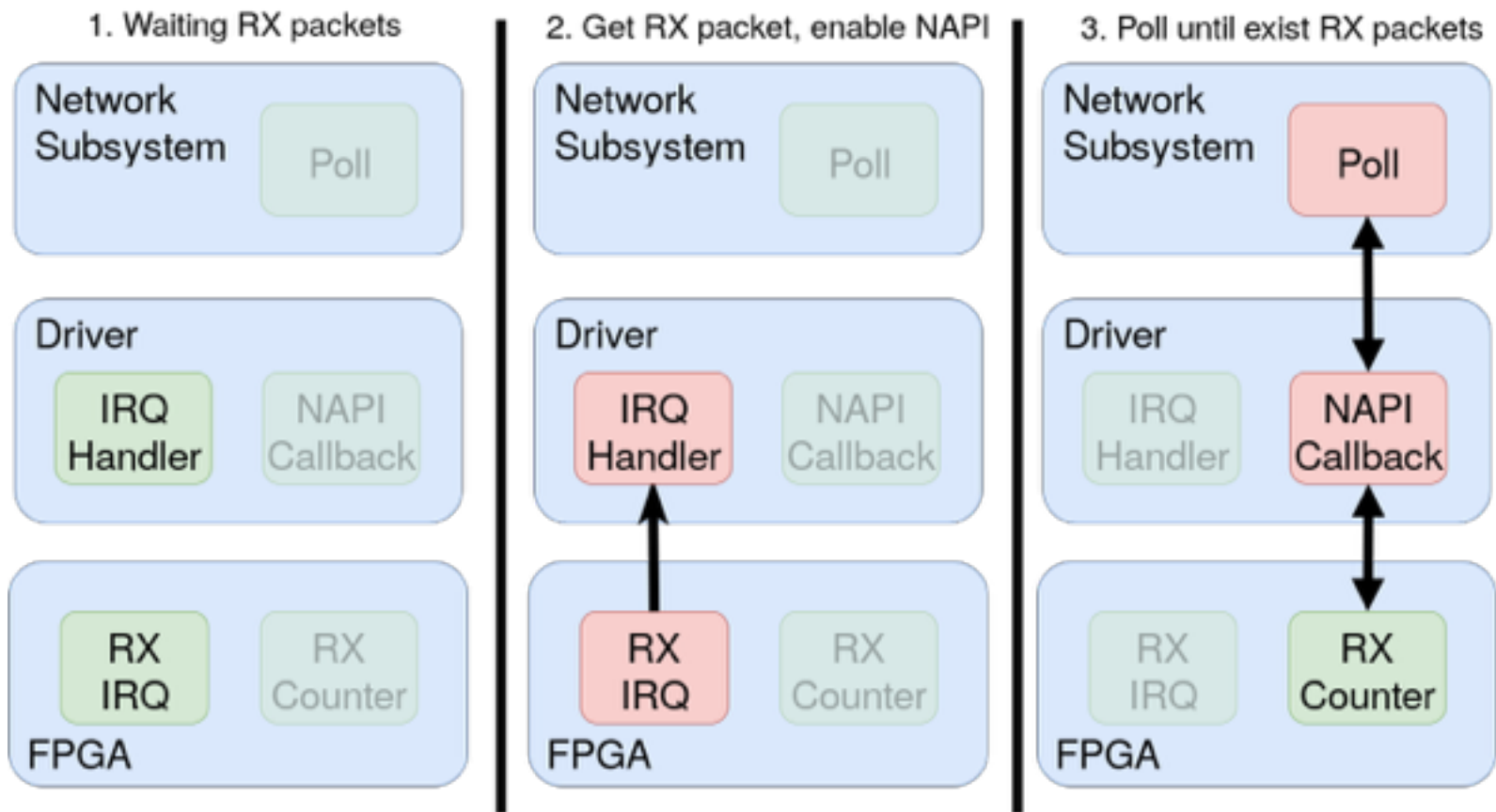


RX packets transfer -- polling only

1. Always poll RX packets



RX packets transfer -- NAPI



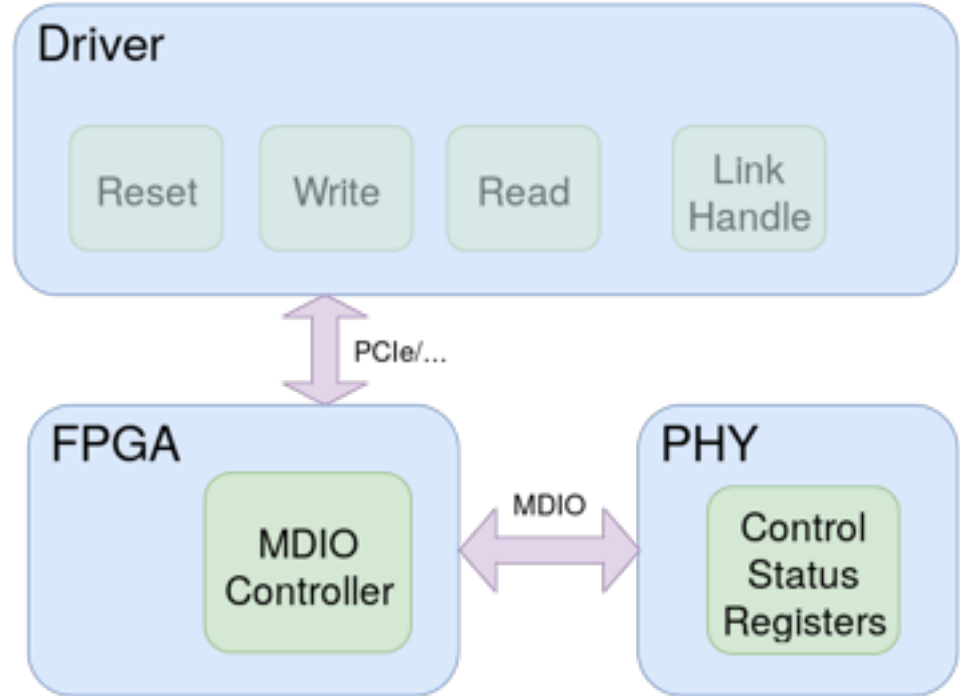
MDIO Control Interface

Access to PHY registers

Used by kernel PHY framework

Link monitoring

Auto-negotiation & Speed



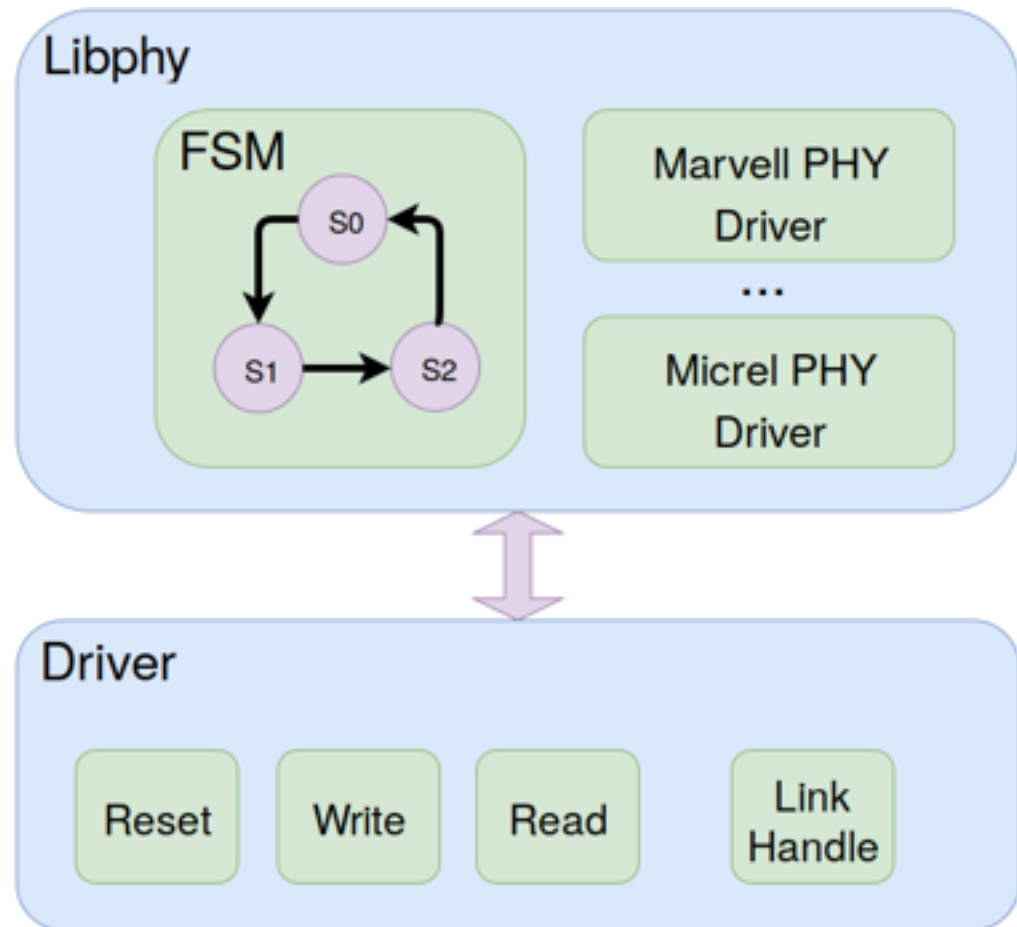
Kernel PHY framework

libphy

Does all “dirty” work

Support many PHYs

PHY discovery



Libphy usage example

```
mdio->name = "etn-mdio";  
mdio->read = &fpga_mdio_read;  
mdio->write = &fpga_mdio_write;  
mdio->reset = &fpga_mdio_reset;  
...  
mdiobus_register(etn_port->mdio_bus);  
...  
etn_port->phy_dev = phy_find_first(etn_port->mdio_bus);
```


Ethtool

Speed/duplex/autoneg

HW offload features

DMA ring size

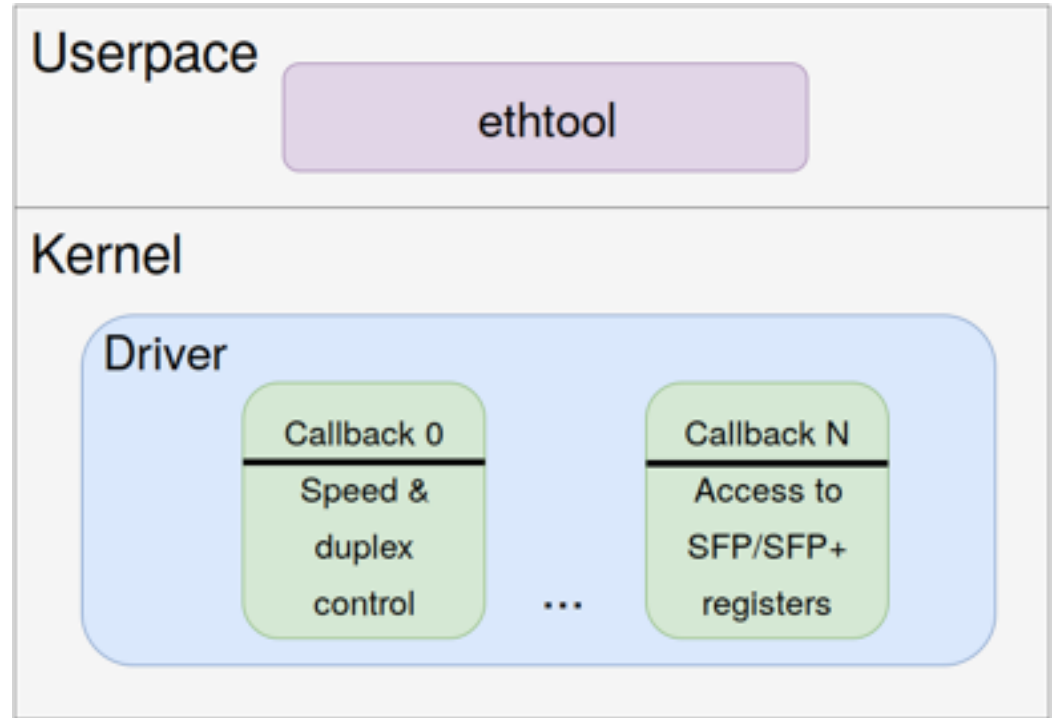
Interrupt moderation

Access to PHY registers

SFP/SFP+ EEPROM

HTL Metrotek

Denis Gabidullin, Pavel Kurochkin



ethtool_ops

```
const struct ethtool_ops etn_ethtool_ops = {  
    .get_drvinfo                = etn_ethtool_get_drvinfo,  
    .get_msglevel              = etn_ethtool_get_msglevel,  
    .set_msglevel              = etn_ethtool_set_msglevel,  
    .get_settings              = etn_ethtool_get_settings,  
    .set_settings              = etn_ethtool_set_settings,  
    .get_ts_info               = etn_ethtool_get_ts_info,  
    .get_link                  = ethtool_op_get_link,  
    .get_module_info          = etn_get_module_info,  
    .get_module_eeprom        = etn_get_module_eeprom,  
};
```

get_module_eeprom()

```
etn_get_module_eeprom(...) {  
    struct memory_accessor *macc, *ext_macc;  
    u8 buffer[512];  
  
    macc = port->sfp.macc;  
    ext_macc = port->sfp.ext_macc;  
  
    macc->read(macc, buffer, 0, 256);  
    ext_macc->read(ext_macc, buffer + 256, 0, 256);  
  
}
```

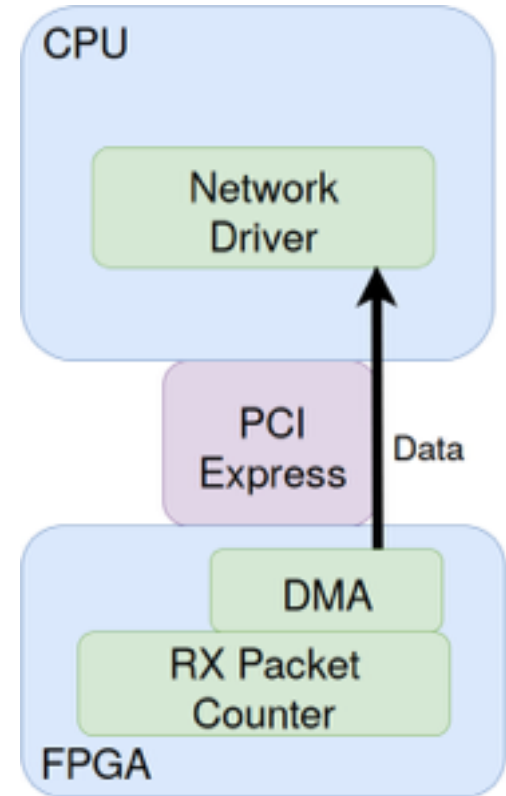
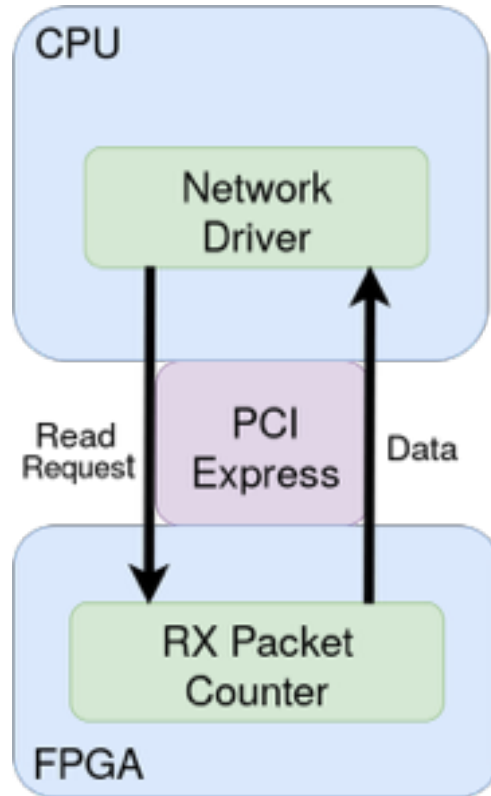
ethtool -m eth0

```
Identifier          : 0x03 (SFP)
Extended identifier : 0x04 (GBIC/SFP defined by 2-wire interface ID)
Transceiver type   : Ethernet: 1000BASE-T
Encoding           : 0x01 (8B/10B)
BR, Nominal        : 1200MBd
Rate identifier     : 0x00 (unspecified)
Length (62.5um)    : 0m
Length (Copper)    : 100m
Length (OM3)       : 0m
Laser wavelengt    : 0nm
Vendor name        : FINISAR CORP.
Vendor OUI         : 00:90:65
Vendor PN          : FCLF-8521-3
Vendor rev         : A
```

Base optimization example

Read -- 500K
transactions per
second

Target -- 14.8M
packets per second



Summary

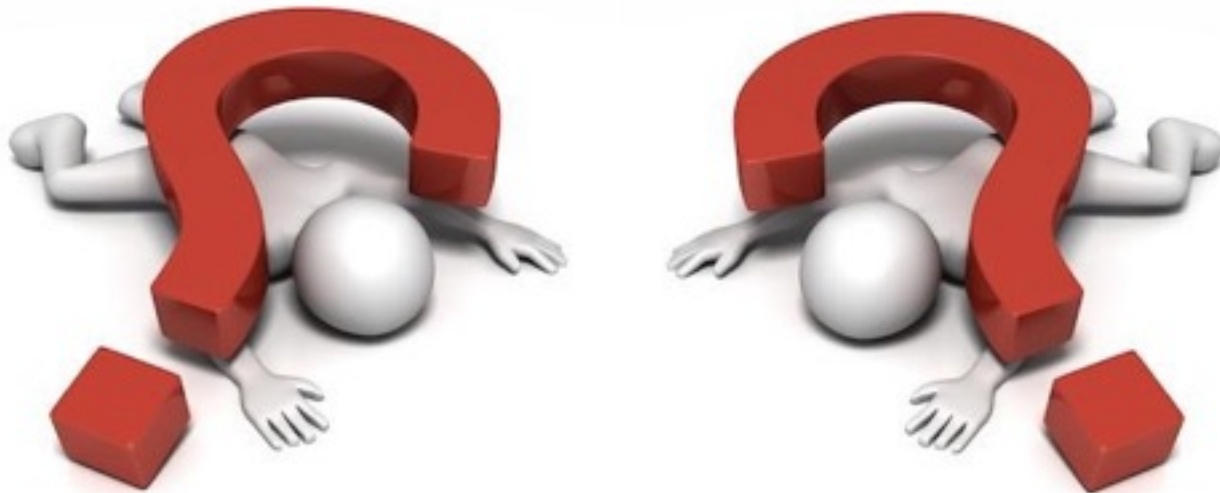
1G/10G/100G NIC Architectures

Network driver components

NAPI, libphy

Examples

Thanks! Questions?



Links

<https://wiki.linuxfoundation.org/networking/napi>

<http://metrotek.spb.ru/files/ethond/3.5.2/src/>