cesnet

CzechLight SDN DWDM

ORS 2024

Jan Kundrát

CESNET

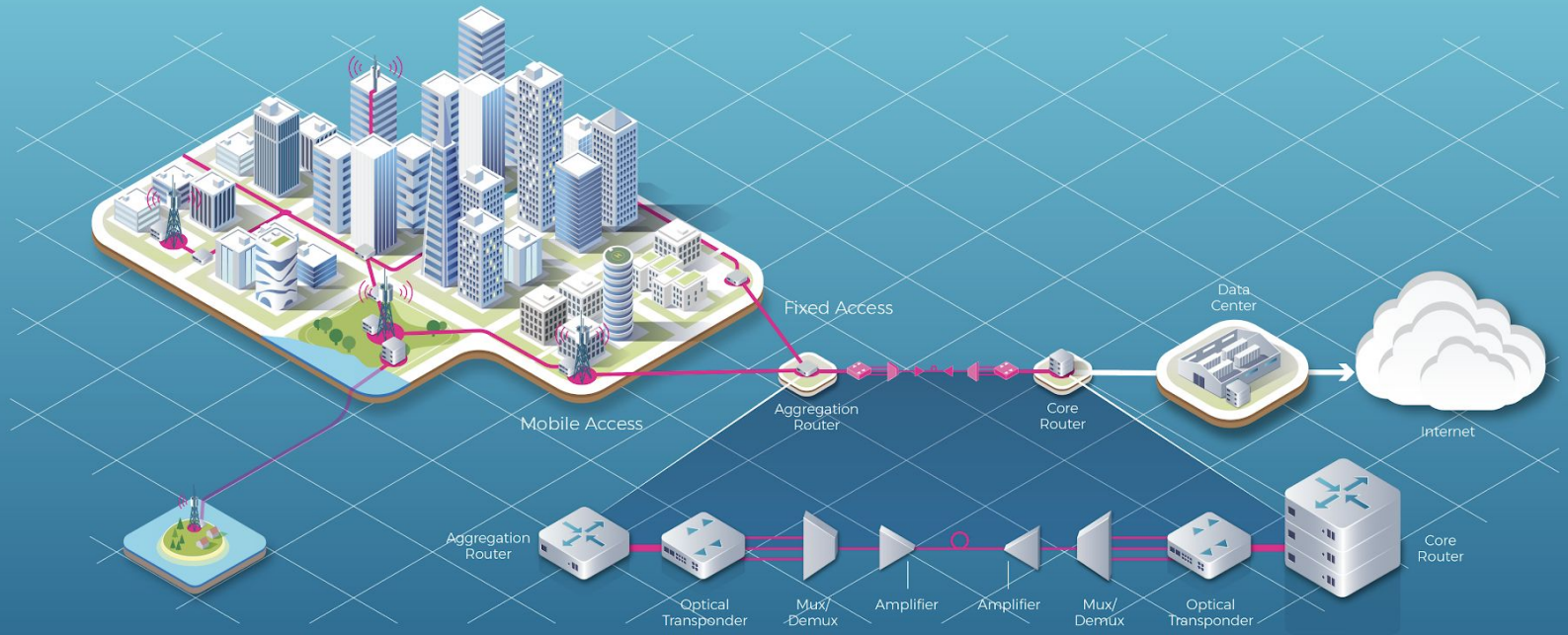
- E-infrastructure service provider
  - Czechia, EU
  - Science, research & education
  - Services
    - Network
    - Compute
    - Storage
    - Collaborative environment
- About me
  - Researcher, SDN at L0

cesnet

A Modern Network

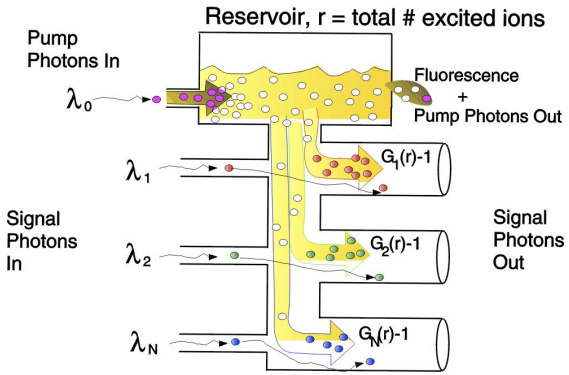
cesnet

Optical Layer



cesnet

Fiber, WDM, EDFA



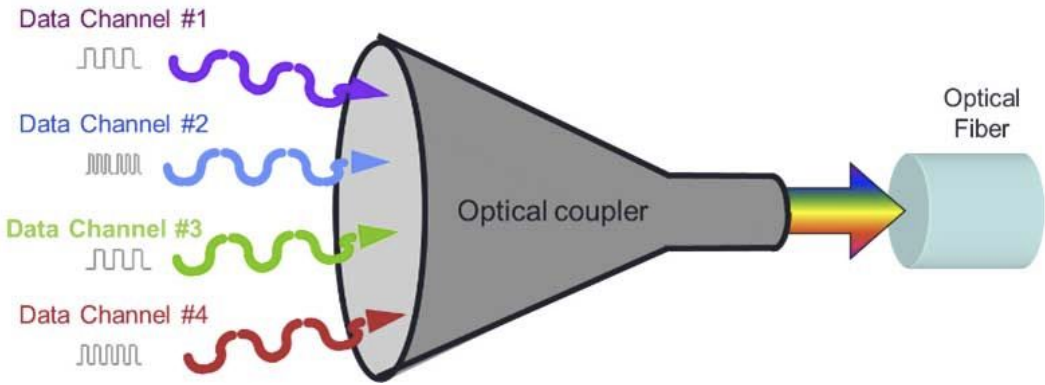
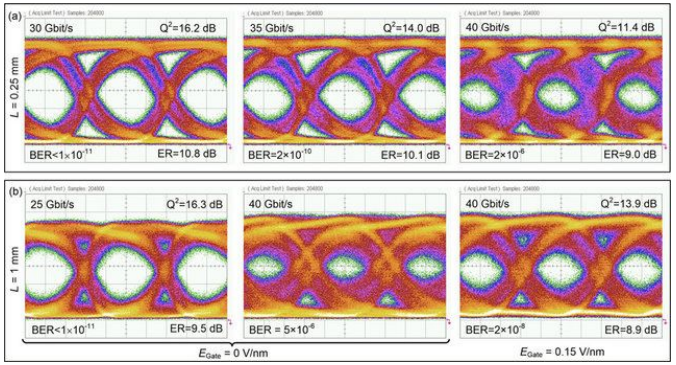
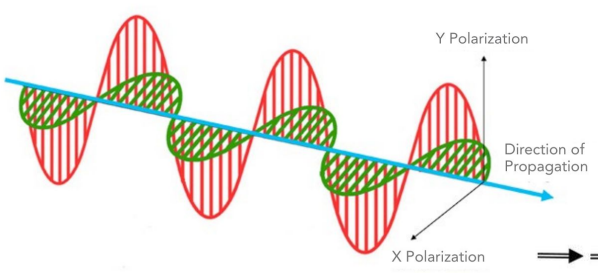
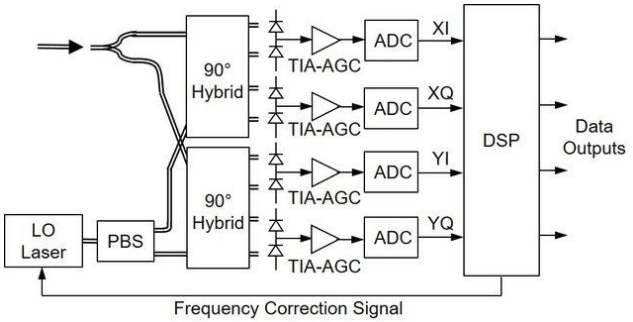


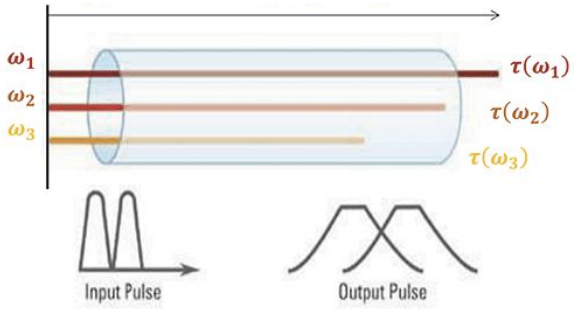
Image credits: © Corning, © A. Bononi, Universita di Parma, © Ciena

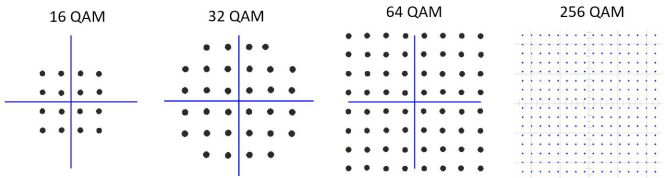
OOK NRZ, IM/DD and Coherent



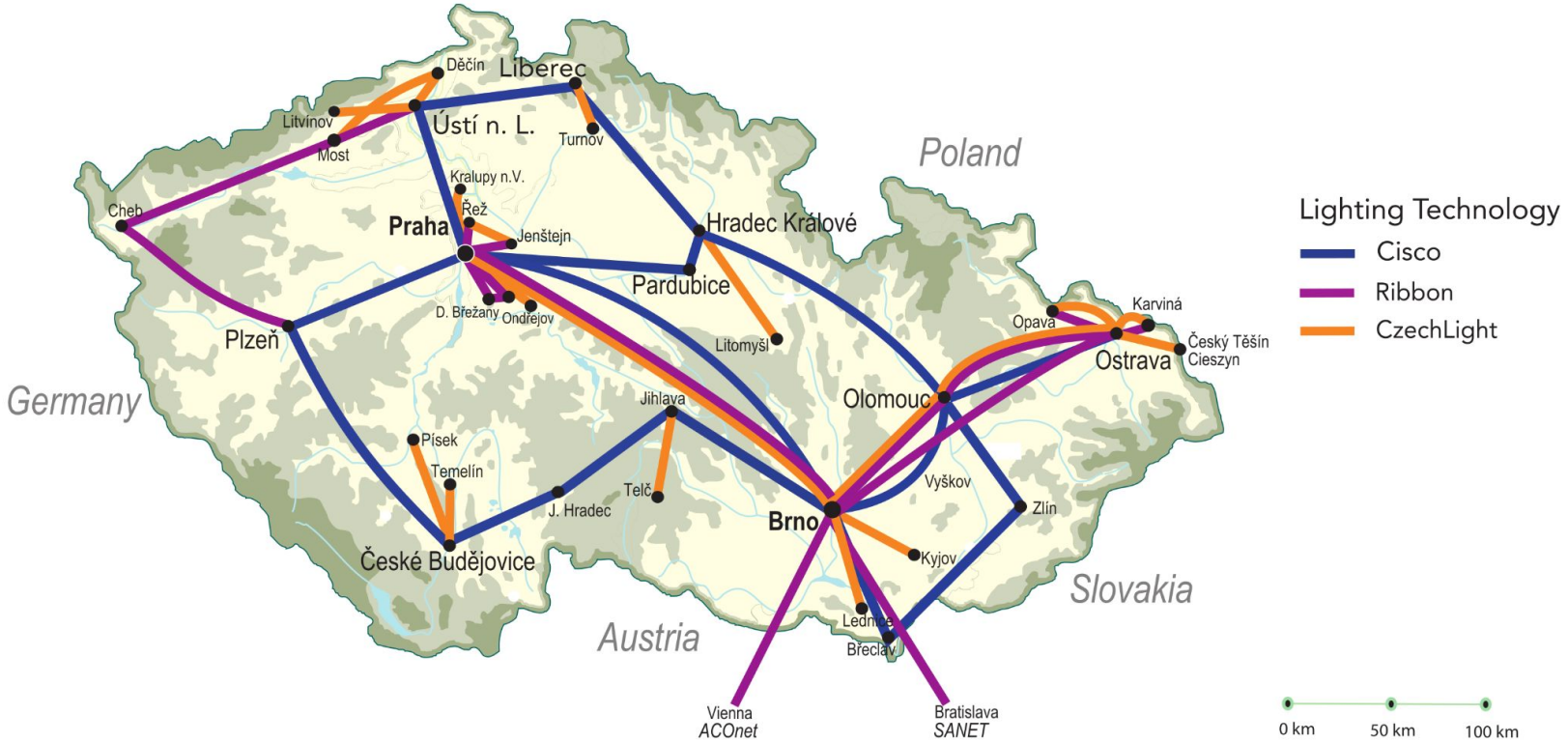






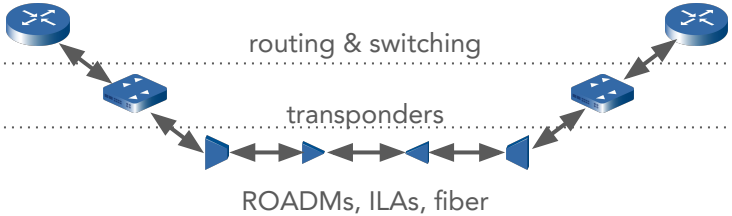


cesnet

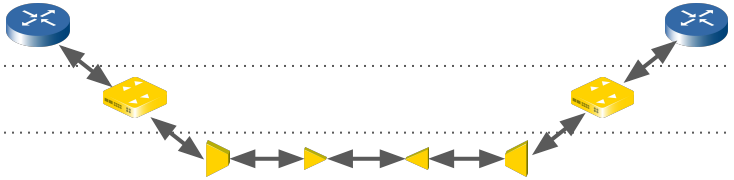

Děčín
Liberec
Litvínov
Ústí n. L.
Most
Turnov
Kralupy n.V.
Řež
Cheb
Praha
Jenštejn
Hradec Králové
Pardubice
D. Březany
Ondřejov
Plzeň
Litomyšl
Opava
Karviná
Ostrava
Český Těšín
Cieszyn
Olomouc
Jihlava
Germany
Písek
Temelín
Telč
J. Hradec
Vyškov
Zlín
Brno
České Budějovice
Kyjov
Slovakia
Austria
Břeclav
Poland
Vienna
ACOnet
Bratislava
SANET
Lighting Technology
Cisco
Ribbon
CzechLight
0 km
50 km
100 km

cesnet

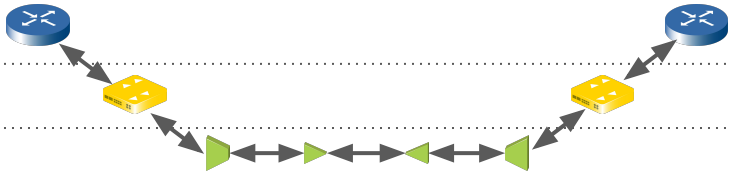
"Disaggregation"



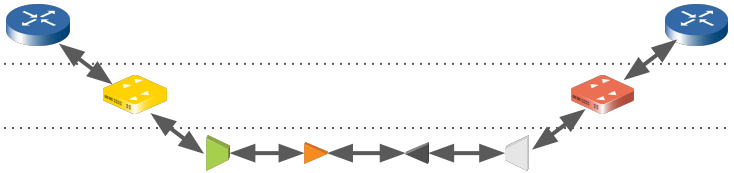
#0: monolithic



#1: turnkey DWDM



#2: transponder disaggregation



#4: full component disaggregation

cesnet

Czech Light ™ SDN ROADM

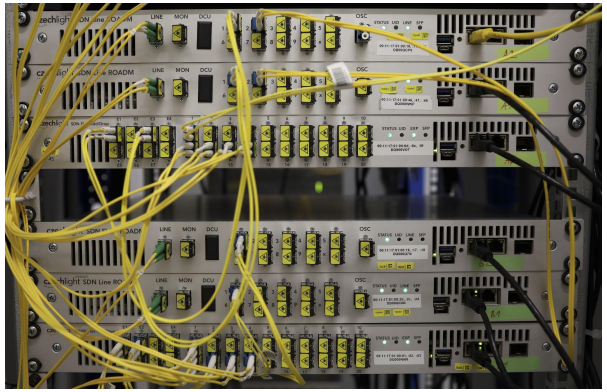
Modular Open Optical Line System

Flexgrid, Colorless, Directionless, Contentionless

SDN Northbound APIs: NETCONF, RESTCONF

How Does it Work?

- Embedded Linux & Optical Modules
  - UART, SPI, I^2C
- No hands-on access, remote housing
  - Management & Telemetry over IP network
- SDN access for everything
  - Optics
  - System settings
  - Local CLI console



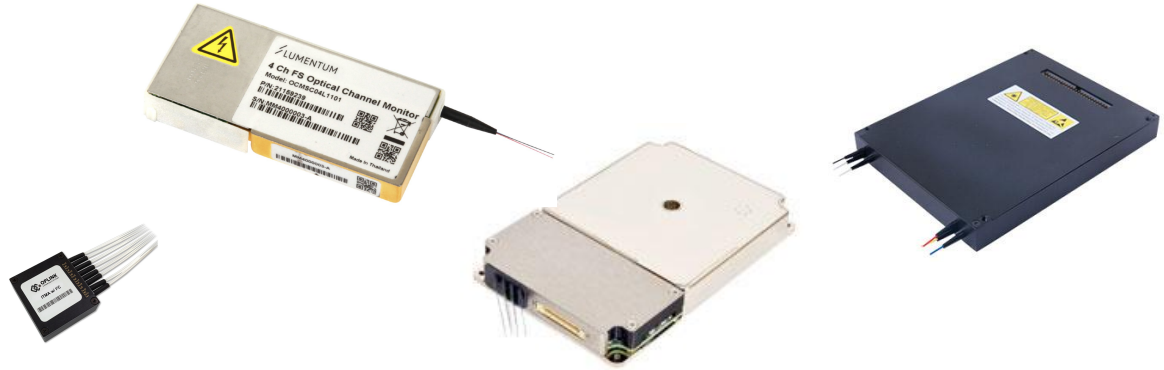


Image source: © Lumentum, © Molex, © Taclink

cesnet

YANG

- Data model
  - Tree structure
  - Configuration vs. State
  - “Leaf” data types
- Defines Data Semantics
- Machine Validation

Inspired by the `ietf-system` YANG model.
Truncated for brevity.

```
module: ietf-system
  +--rw system
  |  +--rw clock
  |  |  +--rw (timezone)?
  |  |   +--:(timezone-name) {timezone-name}?
  |  |   |  +--rw timezone-name?   timezone-name
  |  |   +--:(timezone-utc-offset)
  |  |           +--rw timezone-utc-offset?   int16
  |  +--rw authentication {authentication}?
  |      +--rw user-authentication-order*   identityref
  |      +--rw user* [name] {local-users}?
  |      +--rw name                 string
  |      +--rw password?            iana-crypt-hash:crypt-hash
  |      +--rw authorized-key* [name]
  |               +--rw name        string
  |               +--rw algorithm   string
  |               +--rw key-data    binary
  +--ro system-state
         +--ro platform
         |  +--ro os-name?          string
         |  +--ro os-release?       string
         |  +--ro os-version?       string
         |  +--ro machine?          string
         +--ro current-datetime?    ietf-yang-types:date-and-time
  rpcs:
         +---x set-current-datetime
         |  +---- input
         |        +---w current-datetime     ietf-yang-types:date-and-time
         +---x system-restart
         +---x system-shutdown
```

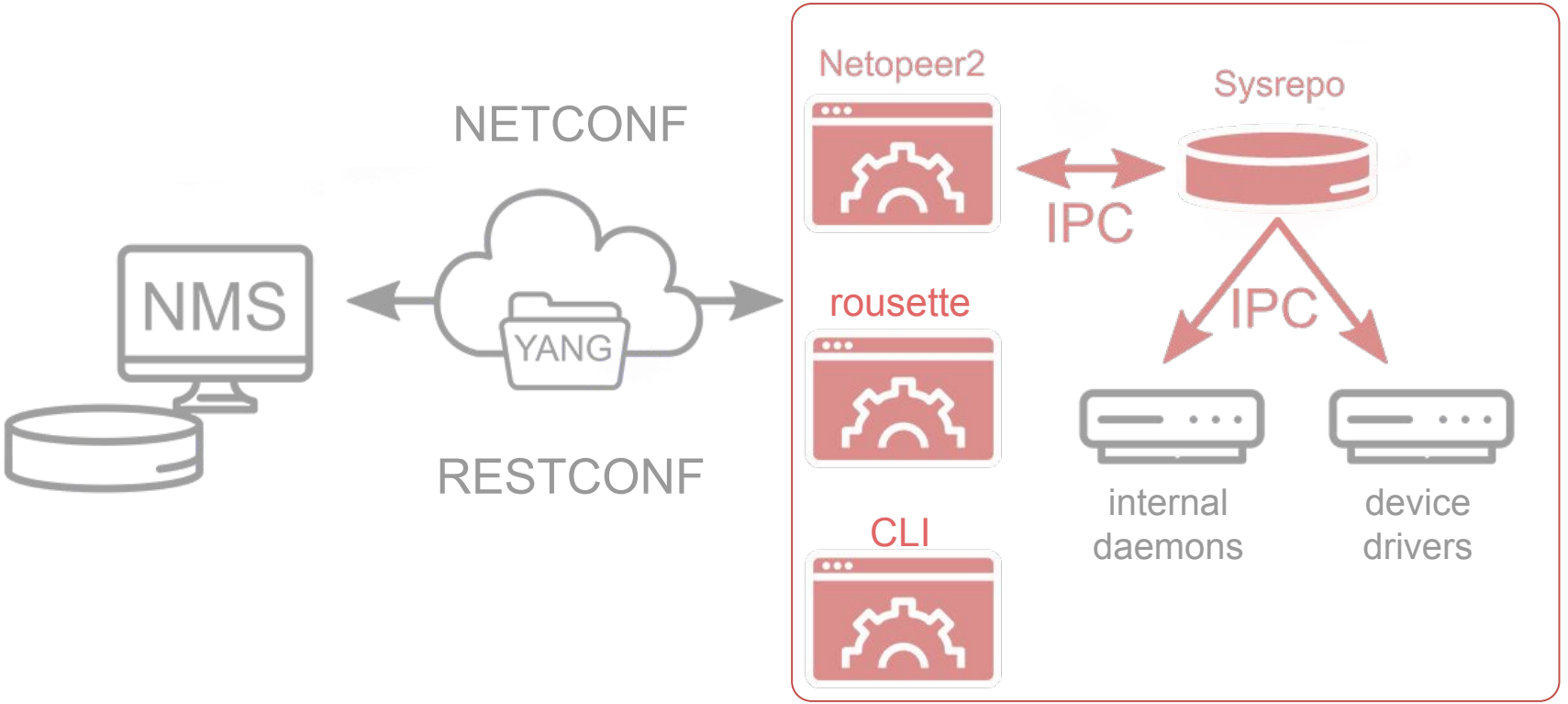

- Bjorklund, Martin. "YANG-a data modeling language for the network configuration protocol (NETCONF)." RFC 6020, 2010.
- Bjorklund, Martin. "The YANG 1.1 data modeling language." RFC 7950, 2016.

cesnet

- YANG
- NETCONF
- RESTCONF
- CLI


sysrepo.org
Storing and managing YANG-based configurations for UNIX/Linux applications

SW Architecture with Sysrepo



cesnet

Configuration via Ansible

```yaml
- hosts: roadm-deg3-foo.bar.example.org
  # Ansible's built-in NETCONF support
  connection: ansible.netcommon.netconf
  tasks:

  # Spectrum Definition
  - CzL_channel_plan:
    name: ciena
    state: present
    lower_frequency: '193950000'
    upper_frequency: '194150000'

  # Spectrum Routing
  - CzL_media_channel:
    name: ciena
    state: present
    leaf_port: 1
    attenuation_add: 0
    attenuation_drop: 10
    description: Ciena
```

ANSIBLE

git

cesnet

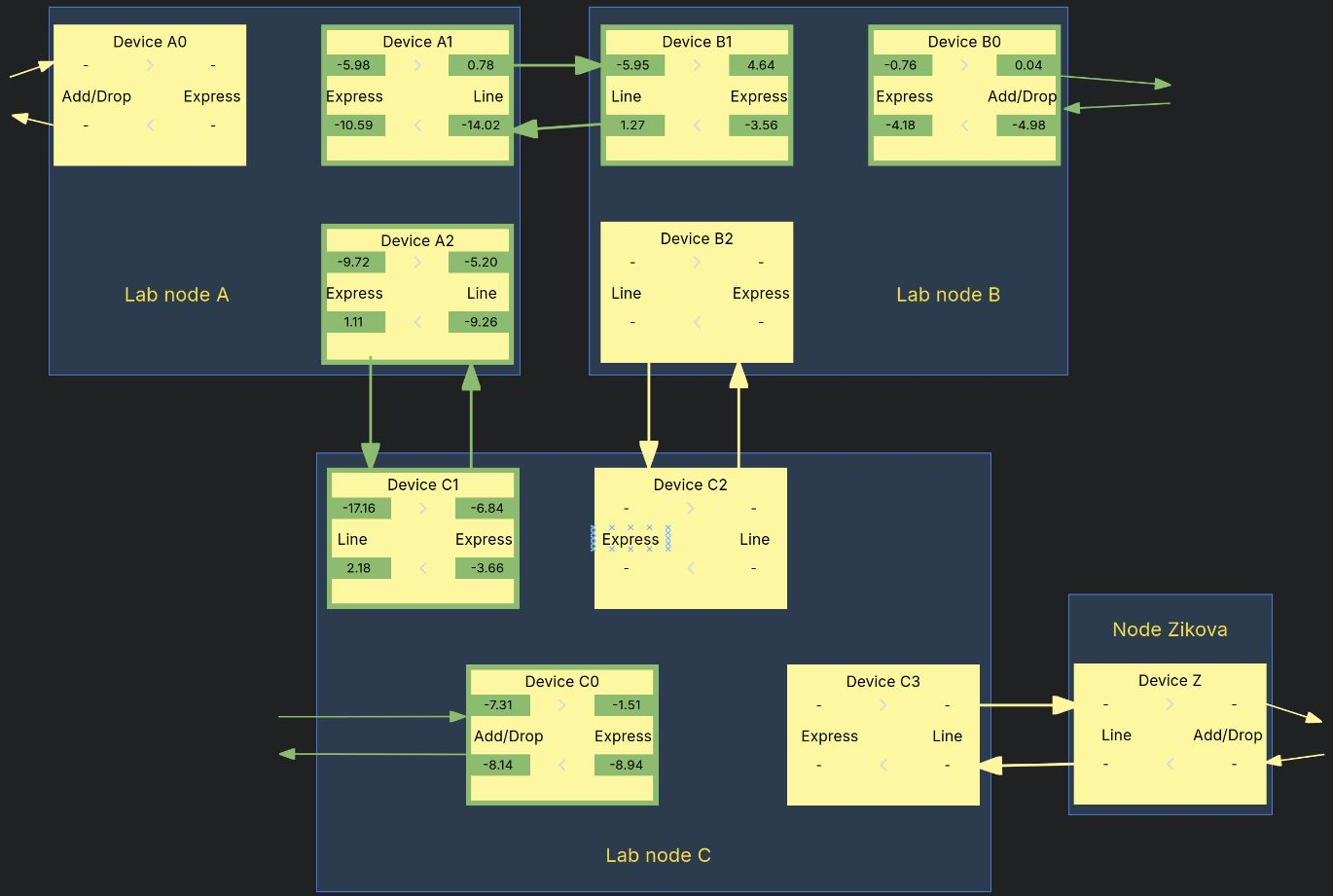
Writing NETCONF in Ansible

```
def config_string(CHname, min_f, max_f, state) -> str:
    '''Turn spectrum channel definition to an XML string'''

    root = ET.Element('{urn:ietf:params:xml:ns:netconf:base:1.0}config' )
    cl = '{http://czechlight.cesnet.cz/yang/czechlight-roadm-device}'
    ET.register_namespace('cl', cl[1:-1])

    channel_plan = ET.SubElement(root, f'{cl}channel-plan')
    channel = ET.SubElement(channel_plan, f'{cl}channel')
    if (state == 'absent'):
        channel.attrib['ns0:operation'] = 'delete'
        ET.SubElement(channel, f'{cl}name').text = CHname
    else:
        ET.SubElement(channel, f'{cl}name').text = CHname
        ET.SubElement(channel, f'{cl}lower-frequency').text = min_f
        ET.SubElement(channel, f'{cl}upper-frequency').text = max_f

    return ET.tostring(root, encoding='unicode', xml_declaration=True)
```

Live Monitoring with Grafana

Channel_number ciena

Power levels monitoring: Channel ciena



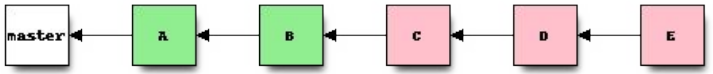
Grafana

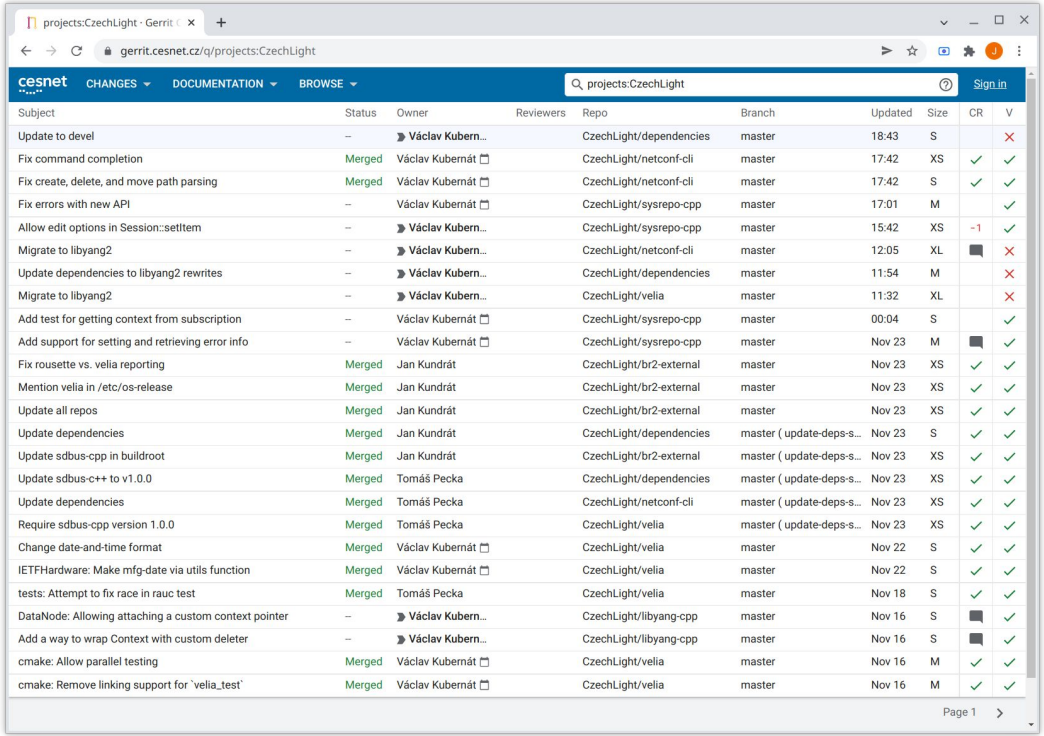
VICTORIA METRICS

Source Code + CI/CD

https://gerrit.cesnet.cz/

br2-external docs





cesnet

Open Source Software

- sysrepo and libyang
  - YANG software stack
  - config/ops database
- libnetconf2 and Netopeer2
  - NETCONF server
- rousette
  - RESTCONF server
  - telemetry
- netconf-cli
  - <Tab>-driven interactive console
- velia: system management
  - health tracking
  - system management
  - firewall
  - hardware
  - network (L2/L3)
- sysrepo-ietf-alarms
  - alarm management
- cla-sysrepo
  - drivers for optical modules
  - ROADM logic

cla-sysrepo: APIs & Test Suite

```
145         session->session_switch_ds(SR_DS_RUNNING);
146     }
147
148     SECTION("leaf port properties") {
149         session->set_item_str("/czechlight-roadm-device:leaf-ports[port='E3']/description", "testing");
150         session->apply_changes(DEFAULT_TIMEOUT, true);
151         waitForCompletionAndBitMore(seq1);
152     }
153
154     SECTION("symmetric channel plan")
155     {
156         // Add a channel
157         {
158             session->set_item_str("/czechlight-roadm-device:media-channels[channel='14.0']/add/port", "E6");
159             session->set_item_str("/czechlight-roadm-device:media-channels[channel='14.0']/add/attenuation", "1.2");
160             session->set_item_str("/czechlight-roadm-device:media-channels[channel='14.0']/drop/port", "E2");
161             session->set_item_str("/czechlight-roadm-device:media-channels[channel='14.0']/drop/attenuation", "2.3");
162             props = {
163                 {"waveplan/0/channel/0/center-MHz", int32_t{191'400'000}},
164             };
165             FAKE_WRITE_PROPS_AT(wss, props);
166             props = {
167                 {"waveplan/0/channel/0/bandwidth-MHz", int32_t{50'000}},
168             };
169             FAKE_WRITE_PROPS_AT(wss, props);
170             props = {
171                 {"wss/1/channel/0/packed/port-and-atten", uint16_t{(2 << 8) | 23}},
172             };
173             FAKE_WRITE_PROPS_AT(wss, props);
174             props = {
175                 {"wss/2/channel/0/packed/port-and-atten", uint16_t{(6 << 8) | 12}},
176             };
177             FAKE_WRITE_PROPS_AT(wss, props);
178
179             props = {
180                 {"plan/2/count", uint16_t{1}},
181                 {"plan/2/0/start-frequency", uint32_t{191'375'000}},
182                 {"plan/2/0/end-frequency", uint32_t{191'425'000}},
183             };
184             nextOcmResult = {
185                 {"scan/plain/1/channel/0/frequency", uint32_t{191'406'000}},
186                 {"scan/plain/1/channel/0/power", int16_t{-1759}},
187                 {"scan/plain/1/channel/0/presence", uint16_t{1}},
188                 {"scan/plain/2/channel/0/frequency", uint32_t{191'392'000}},
189                 {"scan/plain/2/channel/0/power", int16_t{0}},
190                 {"scan/plain/2/channel/0/presence", uint16_t{1}},
191                 {"scan/plain/3/channel/0/frequency", uint32_t{191'395'000}},
192                 {"scan/plain/3/channel/0/power", int16_t{-2998}},
193                 {"scan/plain/3/channel/0/presence", uint16_t{1}},
```

- Header only C++14 mocking framework: `https://github.com/rollbear/trompeloeil`
- Björn Fahller. "Making a Tool of Deception." In Overload, 23(125):7-9, 2015.
- The fastest feature-rich C++11/14/17/20 single-header testing framework: `https://github.com/onqtam/doctest`
- Viktor Kirilov. "Mix Tests and Production Code With Doctest - Implementing and Using the Fastest Modern C++ Testing Framework." In CppCon, 2017.

cesnet

YANG × C++: decimal64

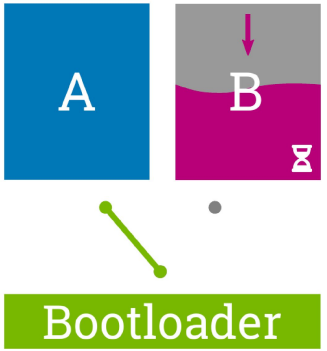
```cpp
/** @short Fixed-point YANG decimal number */
struct Decimal64 {
    int64_t number;
    uint8_t digits;

    explicit constexpr operator double() const
    {
        return number * impl::pow10double(-digits);
    }

    constexpr Decimal64 operator-() const
    {
        return Decimal64{-number, digits};
    }

    template<uint8_t digits>
    constexpr static Decimal64 fromRawDecimal(const int64_t value)
    {
        static_assert(digits >= 1);
        static_assert(digits <= 18);
        return Decimal64{value, digits};
    }

    template<uint8_t digits>
    constexpr static Decimal64 fromDouble(const double value)
    {
        static_assert(digits >= 1);
        static_assert(digits <= 18);
        return Decimal64{impl::llround(value * impl::pow10int(digits)), digits};
    }
private:
    explicit constexpr Decimal64(const int64_t number, const uint8_t digits)
    : number(number)
    , digits(digits)
    {
    }
    template <int64_t V, uint8_t IntegralDigits, uint8_t FractionDigitsPlusOne>
    friend constexpr Decimal64 impl::make_decimal64();
};
```

```cpp
namespace impl {
template <int64_t V, uint8_t IntegralDigits, uint8_t FractionDigitsPlusOne>
constexpr Decimal64 make_decimal64()
{
    static_assert(IntegralDigits <= 18);
    static_assert(IntegralDigits + FractionDigitsPlusOne <= 20);
    if constexpr (FractionDigitsPlusOne < 2) {
        return Decimal64::fromRawDecimal<1>(V * 10);
    } else {
        return Decimal64::fromRawDecimal<FractionDigitsPlusOne - 1>(V);
    }
}
template <int64_t V, uint8_t IntegralDigits, uint8_t FractionDigitsPlusOne, char C, char...
constexpr Decimal64 make_decimal64()
{
    static_assert((C >= '0' && C <= '9') || C == '.', "Invalid numeric character for Decima
    // more than one '.' is rejected by the lexer, apparently
    if constexpr (C == '.') {
        return make_decimal64<V, IntegralDigits, 1, Cs...>();
    } else if constexpr (FractionDigitsPlusOne > 0) {
        return make_decimal64<V * 10 + C - '0', IntegralDigits, FractionDigitsPlusOne + 1,
    } else {
        return make_decimal64<V * 10 + C - '0', IntegralDigits + 1, 0, Cs...>();
    }
}
}
inline namespace literals {
template <char ... Cs>
```

```cpp
static_assert(1.00_decimal64 == Decimal64::fromRawDecimal<2>(100));
static_assert(1.000_decimal64 == Decimal64::fromRawDecimal<3>(1000));
static_assert(1.000000000000000000_decimal64 == Decimal64::fromRawDecimal<18>(1000
static_assert(-1.000000000000000000_decimal64 == Decimal64::fromRawDecimal<18>(-10
static_assert(1.2_decimal64 == Decimal64::fromRawDecimal<1>(12));
static_assert(12.3_decimal64 == Decimal64::fromRawDecimal<1>(123));
static_assert(456.7_decimal64 == Decimal64::fromRawDecimal<1>(4567));
static_assert(456.78_decimal64 == Decimal64::fromRawDecimal<2>(45678));
static_assert(456.789_decimal64 == Decimal64::fromRawDecimal<3>(456789));
static_assert(456.7890_decimal64 == Decimal64::fromRawDecimal<4>(4567890));
static_assert(-456.7890_decimal64 == Decimal64::fromRawDecimal<4>(-4567890));
```

System Image and Bootloader

- Atomic system updates
  - System built as a single image: http://buildroot.org/
- A/B software slots
  - http://rauc.io/
  - Integrated with HW watchdog



yocto PROJECT

onie

- Sirotkin, Alexander. "Roll your own embedded Linux system with buildroot." Linux Journal 2011, no. 206: 7.
- Petazzoni, Thomas. "Buildroot: a nice, simple and efficient embedded Linux build system." In Embedded Linux System Conference, vol. 2012. 2012.
- Sherwood, Rob. "Tutorial: White box/bare metal switches." In Open Networking User Group meeting, New York. 2014.
- Zawada, A., et al. "ATCA Carrier Board with IPMI supervisory circuit." In 2008 15th Intl. Conf. on Mixed Design of Integrated Circuits and Systems, pp. 101-105. IEEE, 2008.

cesnet

Design for Reliability

- Read-only `rootfs`
  - Stateless system
    - Except the YANG database (and some bits for the early boot)
- Userland based on `systemd`
  - Dependencies of units
  - Automatic service restarts

- Sünter, Indrek, Andris Slavinskis, Urmas Kvell, Andres Vahter, Henri Kuuste, Mart Noorma, Johan Kutt et al. "Firmware updating systems for nanosatellites." IEEE Aerospace and Electronic Systems Magazine 31, no. 5 (2016): 36-44.
- Langiu, Antonio, Carlo Alberto Boano, Markus Schuß, and Kay Römer. "UpKit: An Open-Source, Portable, and Lightweight Update Framework for Constrained IoT Devices." In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 2101-2112. IEEE, 2019.
- Blass, John, and John Roberts. "Stateless Provisioning: Modern Practice in HPC." In In HPCSYSPROS18: HPC System Professionals Workshop. Dallas, TX. 2018.
- Westerberg, Ellinor. "Efficient delta based updates for read-only filesystem images: An applied study in how to efficiently update the software of an ECU." (2021).

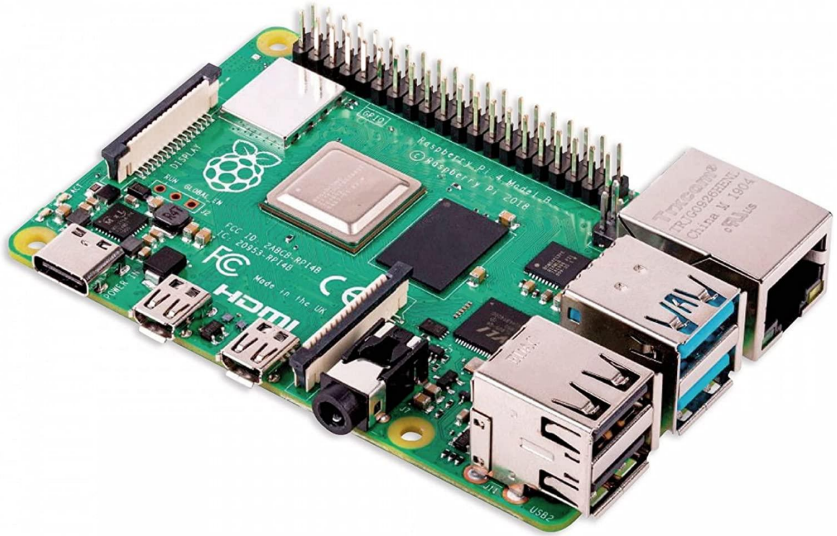
HW: SolidRun Clearfog Base

- Marvell Armada A38x SoC
  - Dual core 32-bit ARM
  - 1GB RAM, 4 GB eMMC
- SFP built-in
  - Native I^2C access
  - No need to route SGMII or PCIe
- Well-supported upstream
  - Same SoC as Turris Omnia
  - But factory-shipped U-Boot horribly outdated
- Not that many GPIOs

Image source: © SolidRun

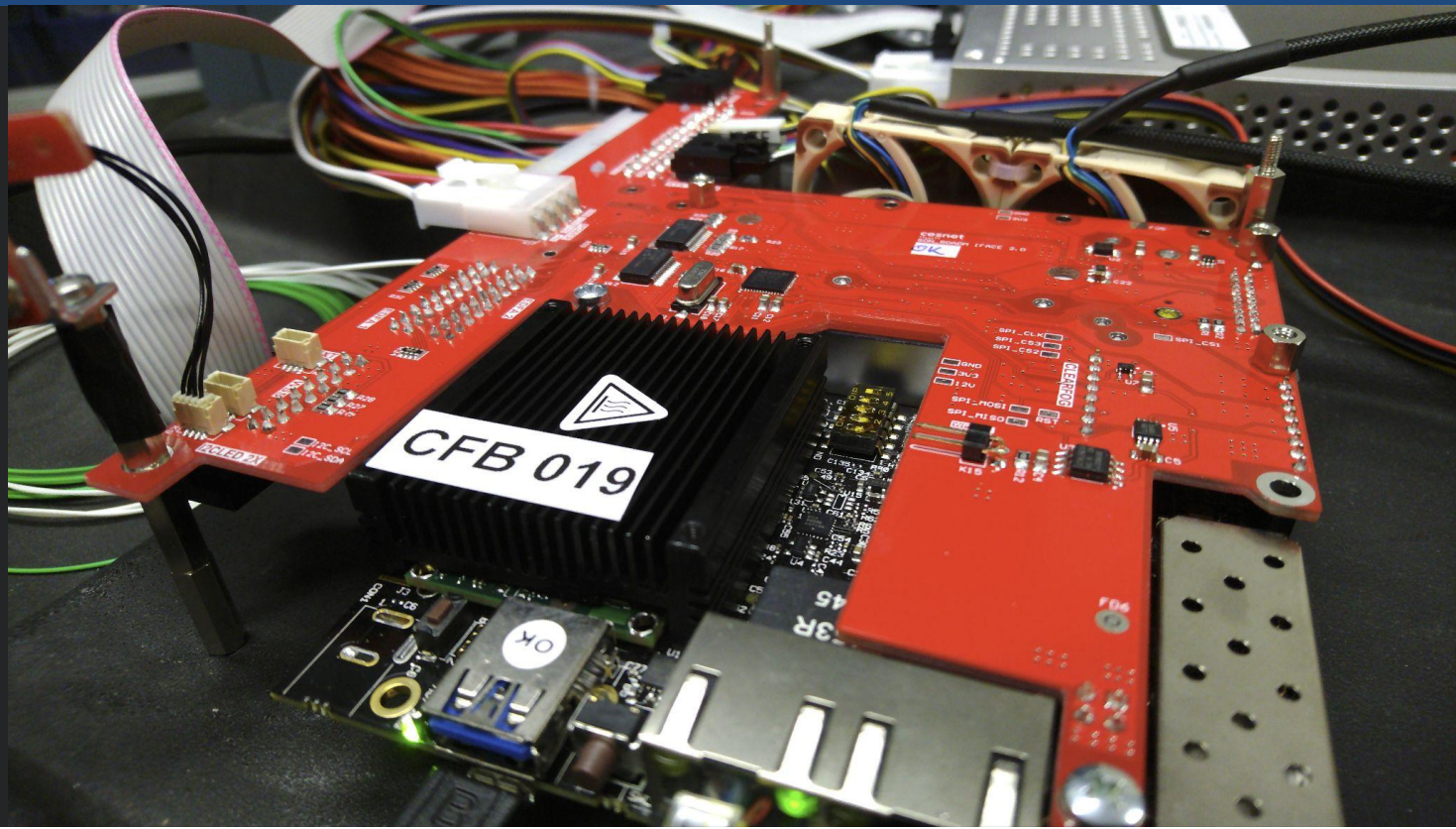
cesnet

Not a Pi



Custom Electronics

10 separate board layouts

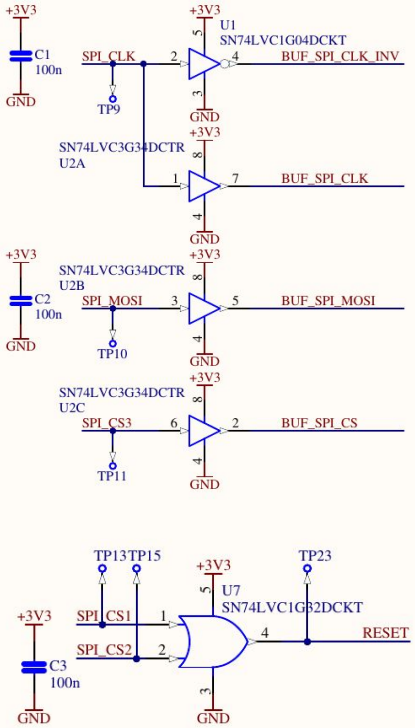


Jan Kundrát (36):
- serial: max310x: Fix invalid memory access during GPIO init
- serial: max310x: Do not hard-code the IRQ type
- serial: max310x: Use level-triggered interrupts
- serial: max310x: Support IRQ sharing with other devices
- serial: max310x: Document clock setup
- serial: max310x: use a batch write op for UART transmit
- serial: max310x: Use batched reads when reasonably safe
- serial: max310x: Reduce RX work starvation
- i2c: gpio: Enable working over slow can_sleep GPIOs
- gpio: serial: max310x: Support open-drain configuration for GPIOs
- pinctrl: mcp23s08: spi: Fix regmap debugfs entries
- pinctrl: mcp23s08: spi: Add HW address to gpio_chip.label
- pinctrl: mcp23s08: spi: Fix duplicate pinctrl debugfs entries
- spi: orion: Rework GPIO CS handling
- spi: orion: Make the error message greppable
- spi: orion: Prepare space for per-child options
- gpio: serial: max310x: Use HW type for gpio_chip's label
- pinctrl: mcp23s08: Kconfig: update to reflect supported features
- pinctrl: mcp23s08: debugfs: Do not restore the INTF register
- spi: orion: fix CS GPIO handling again
- serial: max310x: Check the clock readiness
- i2c: algos: bit: make the error messages grepable
- spi: spidev: Enable control of inter-word delays
- spi: orion: Support spi_xfer->word_delay_usecs
- gpiolib: export devprop_gpiochip_set_names()
- pinctrl: mcp23s08: debugfs: remove custom printer
- pinctrl: mcp23s08: Do not complain about unsupported params
- ARM: mvebu_v7_defconfig: fix Ethernet on Clearfog
- tty: max310x: fix off-by-one buffer access when storing overrun
- hwmon: (pmbus) Fix page vs. register when accessing fans
- Revert "spi: orion: Prepare space for per-child options"
- spi: orion: Respect per-transfer bits_per_word settings
- tty: max310x: Fail probe when external clock crystal is not stable
- leds: tlc591xx: SW reset during initialization
- pinctrl: mcp23s08: work around GPIO line naming
- igb: unbreak I2C bit-banging on i350

Václav Kubernát (6):
- hwmon: Add driver for fsp-3y PSUs and PDUs
- hwmon: (max31790) Rework to use regmap
- hwmon: (max31790) Fix and split pwm*_enable
- hwmon: (max31790) Show 0 RPM/fault when input disabled
- hwmon: (max31790) Allow setting fan*_div
- hwmon: (max31790) Update documentation

Hardware Fun

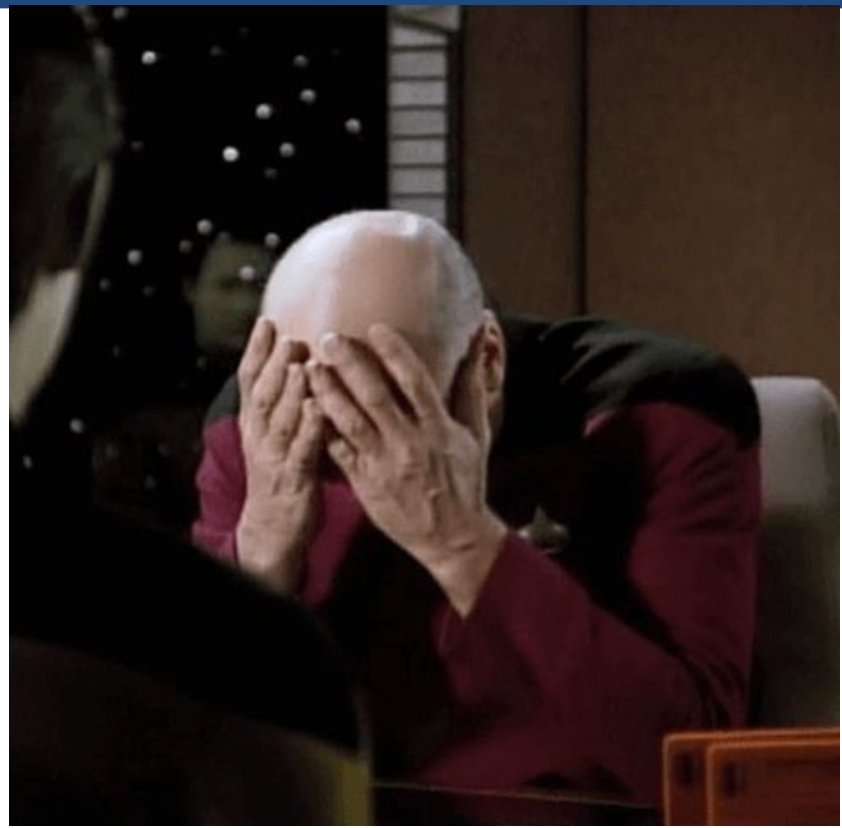
- GPIOs
  - Pin Mux: “IRQ” doesn’t mean “safe to drive at PoR”
  - Virtual “Reset GPIO”
- I2C PMBus Address Clash
  - SFP diagnostics vs. PMBus and mandatory PEC
- Fans
  - AC-coupled 2-wire fans, EOLed chip
- SPI
  - Polarity bugs, extra inter-byte sleep
  - 16-bit bus width instead of DMA



cesnet

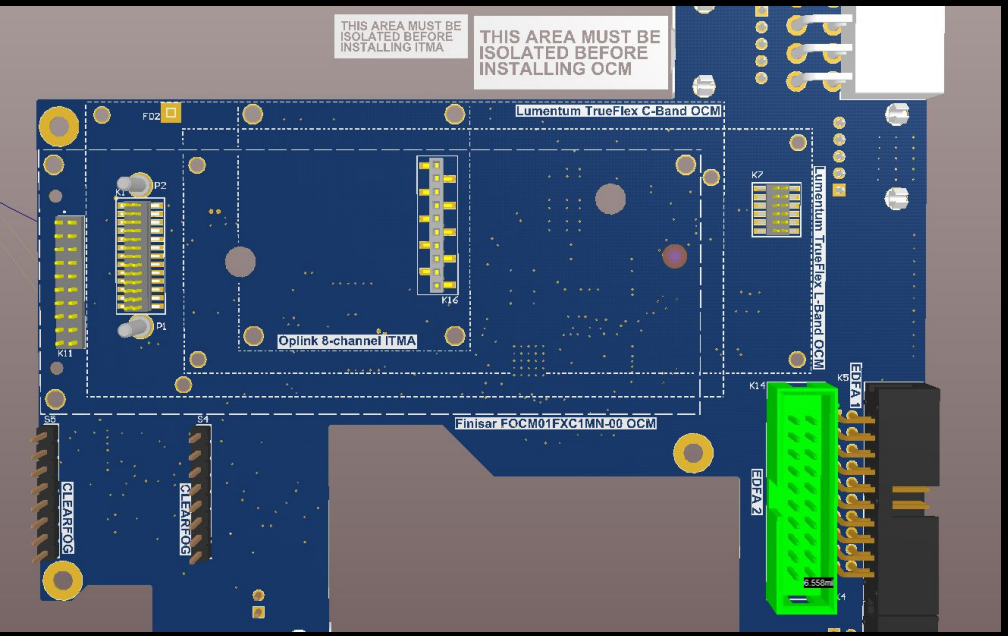
Software Fun

- PMBus
  - Firmware fun with `VOUT_MODE`
  - MCU lockups
- UART FIFOs
  - Buffer overruns
  - 200ms >> 10ms
  - `CONFIG_PREEMPT` for the win
- Service restarts
  - `systemd` to the rescue



cesnet

Space Constraints

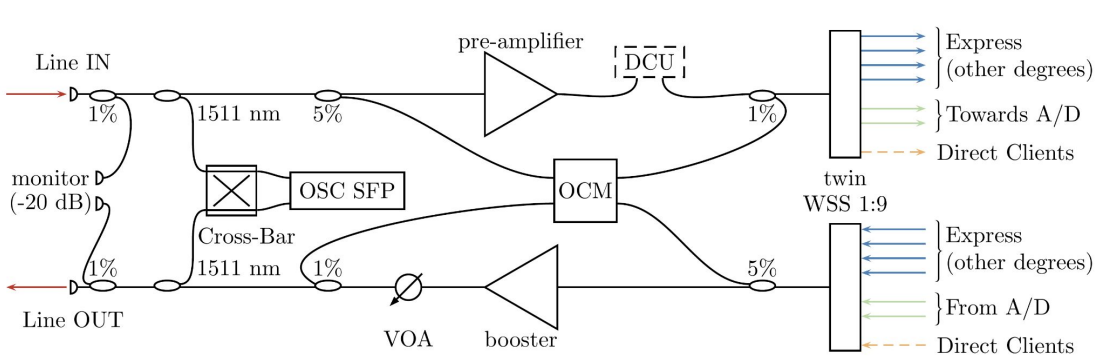


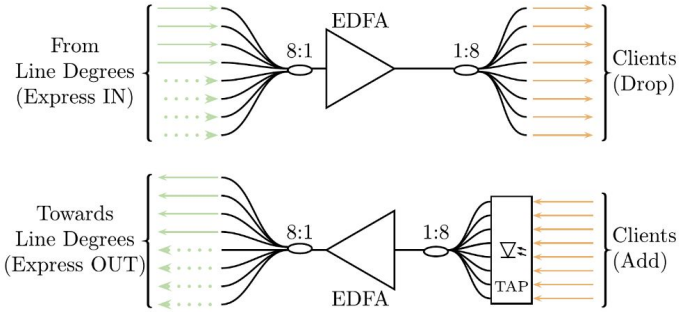
cesnet

Thank you

jan.kundrat@cesnet.cz

Is Optics Complex?





(b)