Motivation

- We have plenty of deployed optical HW of varying age
- The legacy SW provided a human-oriented user interface
- We want to modernize that with a proper, machine-readable remote management
- It seems that NETCONF might be helpful
- Enables Software-Defined Networking (SDN) at the optical/photonic layer

NETCONF

- A protocol for configuring and monitoring of stuff – for example,
  network devices
- Standardized by IETF
- XML streams, XPath, YANG for describing the data model
- SSH or TLS as a transport layer

Netopeer and Sysrepo

- An implementation of a NETCONF server
- Open Source, Free Software projects
- A community of contributors and users
- CESNET maintains several important parts

YANG Models

- Description of the configuration data
- Defines what your device supports
- Potential for common, standardized models
- Unfortunately, vendors typically reinvent the wheel
  ▶ ... so we followed suit

```
module: czechlight-roadm-v2
  +--rw roads
    +--rw channel-plan
      |    +--rw center-frequency sint32
      |      +--rw channel-width decimal64
      |      +--rw approximate-wavelength decimal64
      +--rw configuration
        |    +--rw channel [center-frequency]
        |      |    +--rw center-frequency -> ../../../channel-plan/channel/center-frequency
        |      |    +--rw vca-attenuation attenuation
        |      |    +--rw description? string
        |      |    +--rw min-level-input!
        |      |    +--rw critical-level measured-power
        |      |    +--rw critical-hysteresis? power-hysteresis <1.0>
        |    +--rw optical-power
        |      |    +--rw center-frequency
        |      |      +--rw center-frequency -> ../../../channel-plan/channel/center-frequency
        |      |      +--rw input measured-power
        |      |      +--rw drop measured-power
        |      |      +--rw output measured-power
        |      |      +--rw input-signal-lost boolean
        |      |      +--rw aggregate-power
        |      |      +--rw input measured-power
        |      |      +--rw output measured-power

notifications:
  +--n input-signal-lost
  +--n center-frequency? string
  +--n input-signal-restored
  +--n center-frequency? string
```

Buildroot

- Something for producing a deployable image – an embedded Linux distribution
- Industry appears to use either Buildroot or Yocto/OpenEmbedded/Poky
- Buildroot was very straightforward to start with
- Easy to target various boards
- Can produce container images for automated testing
- A replacement of an entire Linux distribution
- You’re in control of what you want to produce

Embedded Hardware – ARM

- Started as a cost-cutting measure, in the end we have a better product
- Industry-grade HW
- BeagleBone Black
- SW ready for embedded use cases – better watchdogs, more intelligent bootloaders
- TODO: dual A/B rootfs for failsafe updates

Demo at the booth

You can see this in action at our booth – ask for an interactive demo!

→ SDN-Controlled Live Optical Path Protection