NETWORKING



Standardizing OCS Interfaces for Seamless Integration



Richard Yu *Staff Software Engineer - Google*

Brian O'Connor *Staff Software Engineer - Google*

Developed in collaboration with the OCP OCS Working Group

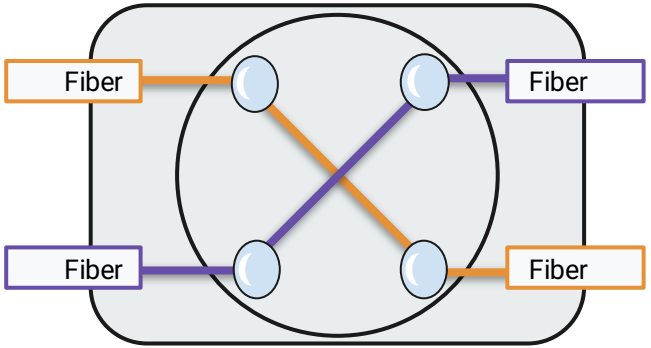
Optical Circuit Switching (OCS)

What is an OCS?

Optical switch that provides configurable pure optical paths between optical input and output ports.

Multiple technologies available: Robotic, Beam Steering, Silicon Photonic

Why use OCS?



Traffic-agnostic

- Compatible with current and future link speeds
- Compatible with any current and future protocols
- No Optical-Electrical-Optical Conversions
  - No choke points, queues, or buffers to overflow
  - Security: No traffic snooping

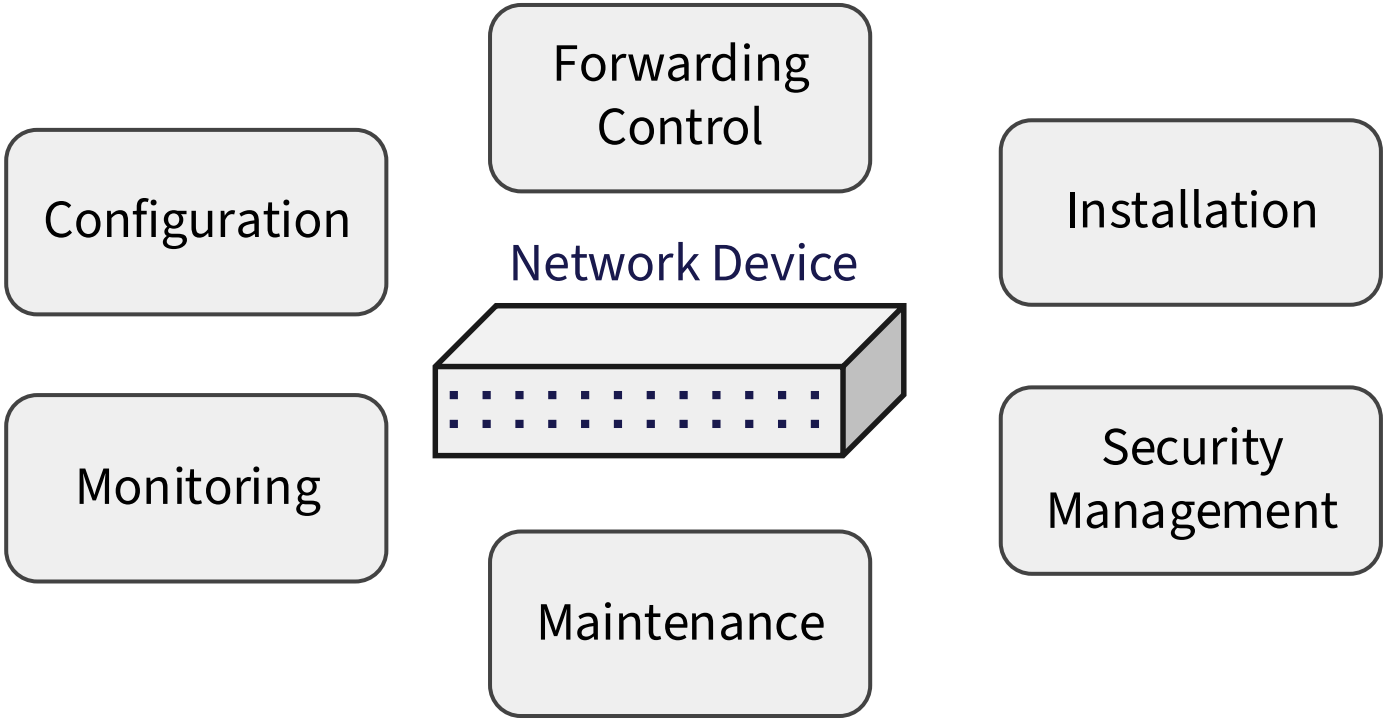
Low latency

- Speed of light vs. ASIC switching latency

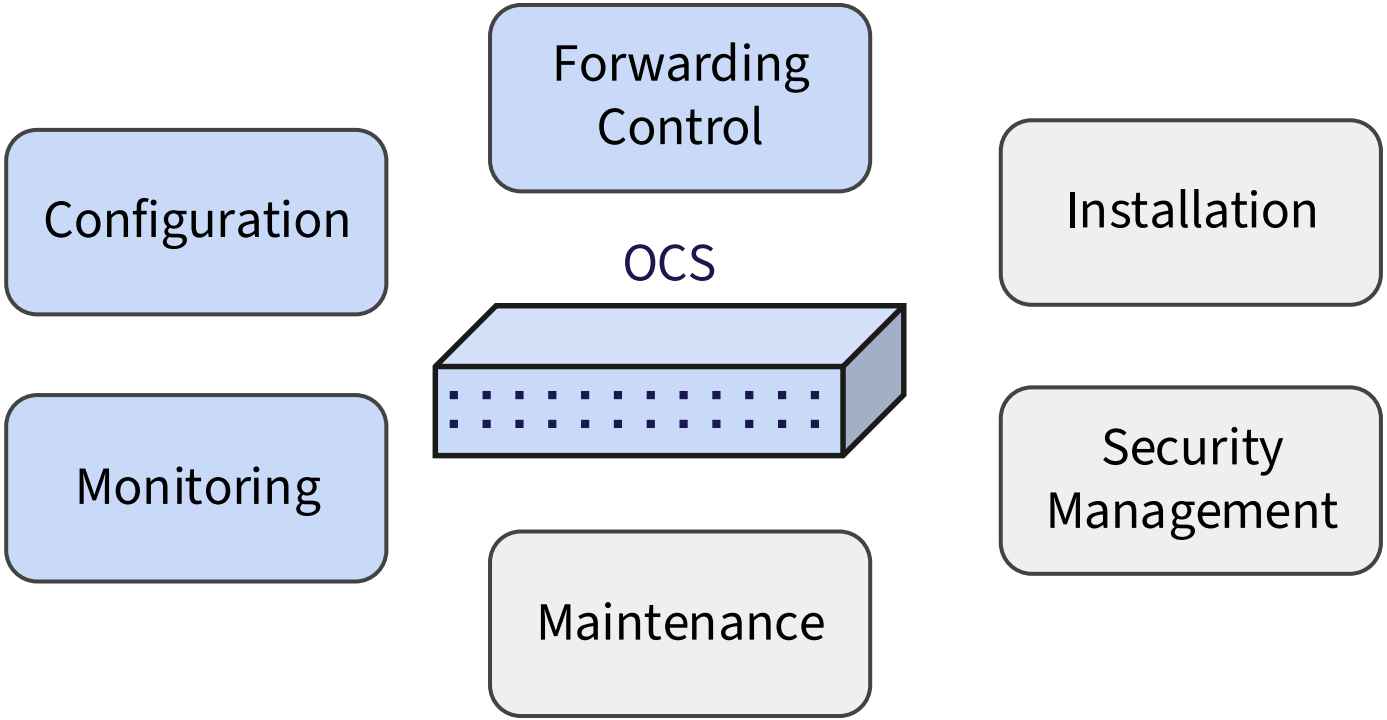
Cost-effective

- No transceivers
- Lower power consumption (less than 5% compared to traditional electric packet switches)

Building Common Interfaces



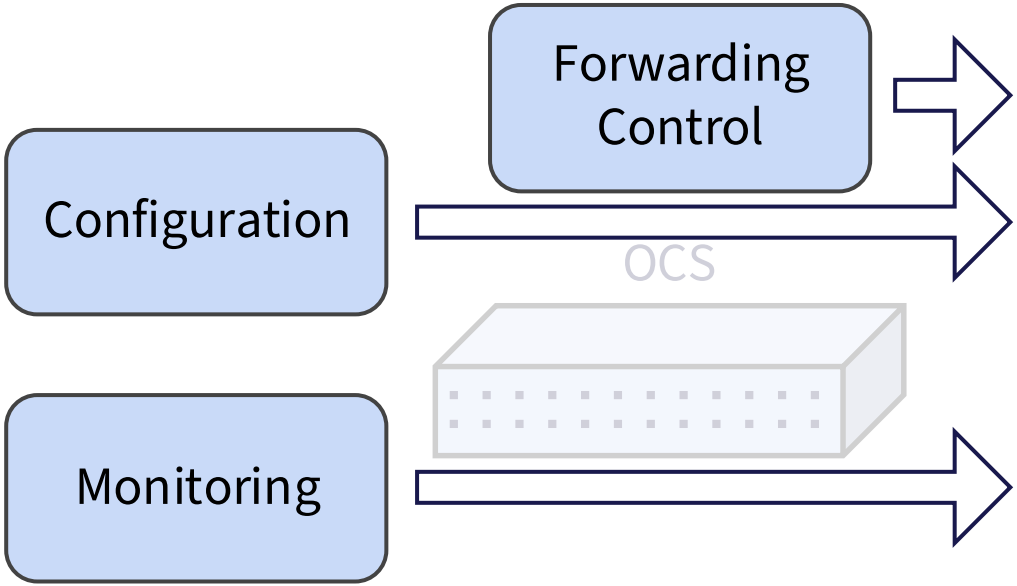
Building Common Interfaces



Modified for OCS

Reused As-Is

Building Common Interfaces



OpenConfig YANG Model

```
components/
    component/
        state/
            oper-status
             empty

interfaces/
    interface/
        state/

qos/
    state/

system/
    state/
```

Why OpenConfig YANG?

- **Leverage** existing support for non-OCS-specific telemetry and configuration
- Easier **onboarding** for operators
- Existing **tools** and **support** for vendors
- Established **rules** and **guidelines**
- Supports **configuration** and **streaming telemetry**

OpenConfig YANG Model

```
components/
    component/
        state/
            oper-status
             empty

interfaces/
    interface/
        state/

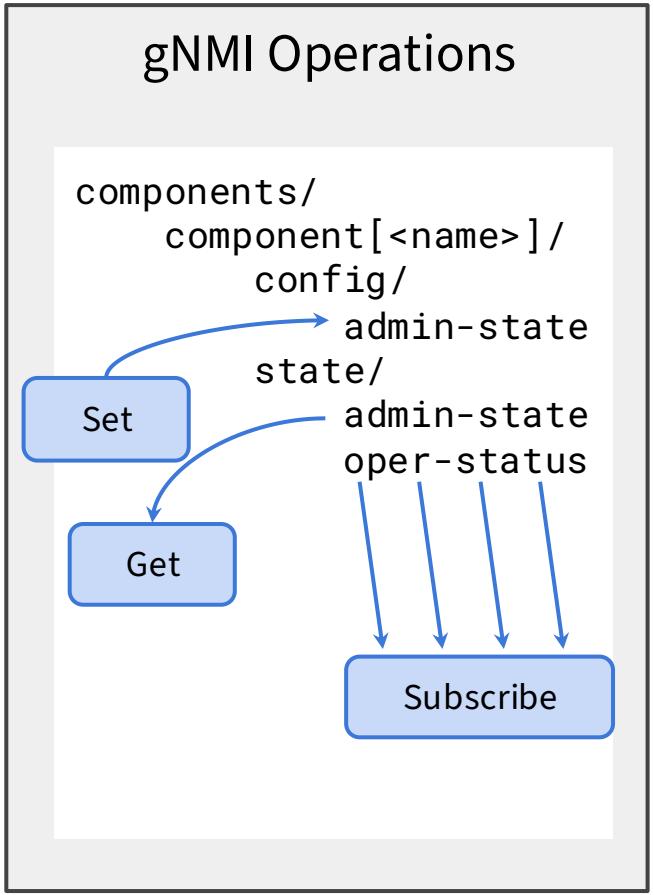
qos/
    state/

system/
    state/
```

Execution: gNMI

gRPC Network Management Interface

- **Model-driven** API that natively supports YANG-modeled schemas.
- **Efficient RPC Operations**: Get, Set, Subscribe
- Combines **config** and **monitoring** (state) into one API



OCS Basic Operations

Forwarding Control

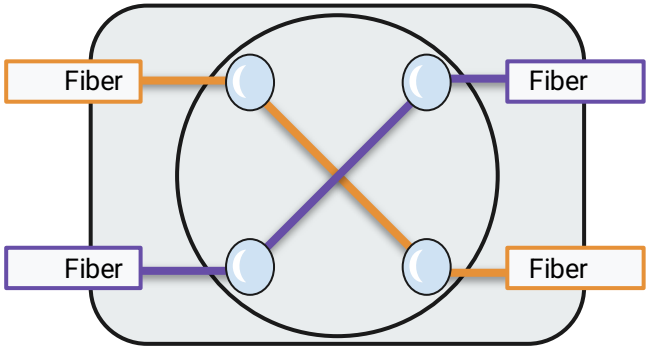
- Establish port-port connections

Configuration

- Configure connection restoration behavior
- Discover device capabilities

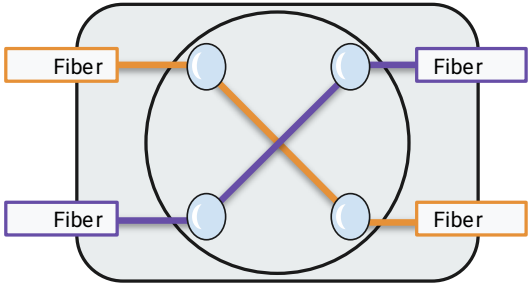
Monitoring

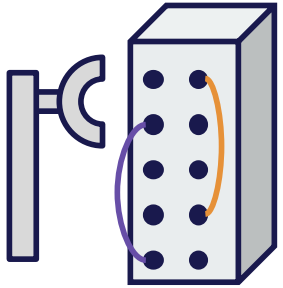
- Monitor connection status
- Monitor port health



OCS Basic Operations - Variations

- Establish port-port connections
  - Any-Any vs. Some-Some (E.g. A-B)
- Monitor connection status
  - Fast switching (Beam steering, Si Photonics)
  - Slow technologies (Robotic)
- Monitor port health
  - Actively maintained links (Beam steering, Si Photonics)
  - Passively maintained links (Robotic)
- Configure connection restoration behavior
  - After reboot: Break, Recover, Maintain





Niche Features (Not fully covered)

- Multicast
- Unidirectional

What's New

- New Top-Level Module for OCS configuration and advertisement

```
Module: ocp-oc-optical-switch
  +--rw oc-optical-switch
     +--rw config
     |  +-- rw bidirectional                      boolean
     |  +-- rw cold-boot-recovery                 enumeration {MAINTAIN, REMAKE, REMOVE}
     +--ro state
        +-- ro bidirectional                      boolean
        +-- ro cold-boot-recovery                 enumeration {MAINTAIN, REMAKE, REMOVE}
        +-- ro cold-boot-recovery-capability*     enumeration {MAINTAIN, REMAKE, REMOVE}
        +-- ro port-groups* [name]
               +-- ro name                        string
               +-- ro direction                   enumeration {INPUT, OUTPUT, ANY}
               +-- ro connection-groups* [peer-name]
                   +-- ro peer-name               string
```

What's New

- New ocs-port component

```
augment .../oc-platform:component:
 +--rw ocp-ocs-port
    +--rw config
    |  +--rw enabled       boolean
    |  +--rw alias         string
    |  +--rw description   string
    +--ro state
       +--ro enabled       boolean
       +--ro alias         string
       +--ro description   string
       +--ro group         leafref
                             -> /ocp-oc-optical-switch:optical-switch/state/port-groups/name
       +--ro index         uint32
       +--ro status        enumeration {OFF, BLOCKED, FAILED, TUNING, TUNED, UNKNOWN}
       +--ro input-optical-power-dbm     uint32
       +--ro output-optical-power-dbm    uint32
       +--ro connection    [see callout]
        +--ro counters     [see callout]
```

```
+--ro connection
   +--ro peer                           leafref
   +--ro connected                      boolean
   +--ro counters
      +--ro correctable-error-count     counter64
      +--ro total-error-count           counter64
```

```
+--ro counters
   +--ro correctable-error-count        counter64
   +--ro total-error-count              counter64
```

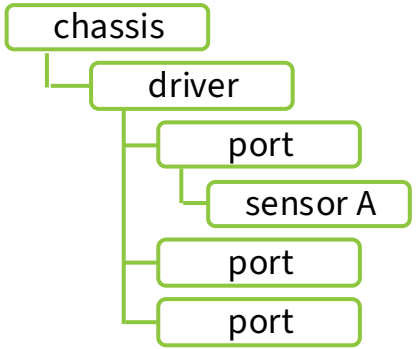
What's New

- Connection Management

```
Module: oc-optical-switch-connections
  +--rw connections
     +--ro last-change-time           yang:date-and-time
     +--rw port-connection* [name]
        +--ro name                          leafref -> ../config/name
        +--rw config
        |  +--rw name                    string    // arbitrary name selected by controller
        |  +--rw input-port-name         leafref -> /oc-platform:components/component/name
        |  +--rw output-port-name      leafref -> /oc-platform:components/component/name
        +--ro state
           +--ro name                  string
           +--ro input-port-name         leafref -> /oc-platform:components/component/name
           +--ro output-port-name        leafref -> /oc-platform:components/component/name
           +--ro status                {CONNECTED, CONNECTING, DELETING, FAILED, INVALID}
           +--ro status-reason         string
           +--ro last-updated          yang:date-and-time
```

Schema Benefits

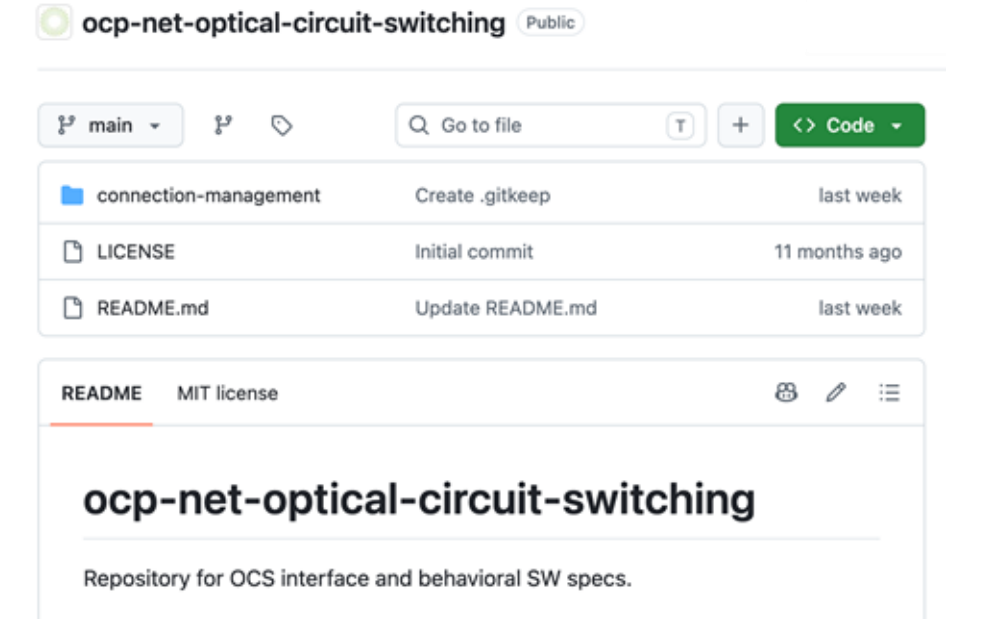
- Free integration with extendable component hierarchies
- Simultaneous visibility to:
  - Current active status [ocp-ocs-port] and
  - Intent [optical-switch-connections]
- Leverage YANG modeling language constraints to enforce valid connection sets from gNMI front-end
  - Under investigation



OCS Interface Spec v0.8

- Draft presented to OCS working group in early September
- Spec is available on GitHub

https://github.com/opencomputeproject/ocp-net-optical-circuit-switching



Core Contributors

- Alvaro Tellez (iPronics)
- Brian O'Connor (Google)
- Iwan Kartawira (iPronics)
- Marwan Abdul-Nour (Lumentum)
- Michael DeMerchant (Lumentum)
- Michael Enrico (Huber+Suhner)
- Ramiro Voicu (Telescent)
- Richard Yu (Google)

Thanks for your contributions!

Call to Action

- Establish a common API that covers most OCS capabilities
- Join the **OCP OCS working group**
  - Project Site: https://www.opencompute.org/projects/optical-circuit-switching
  - Project Wiki: https://www.opencompute.org/wiki/Networking/Optical-Circuit-Switching
  - Project Mailing List: https://ocp-all.groups.io/g/ocp-optical-circuit-switching
  - Project Meetings: **Wednesday, 9am PT**
  - OCS Software Models: https://github.com/opencomputeproject/ocp-net-optical-circuit-switching
- YANG / OpenConfig experts wanted!

Thank You!

