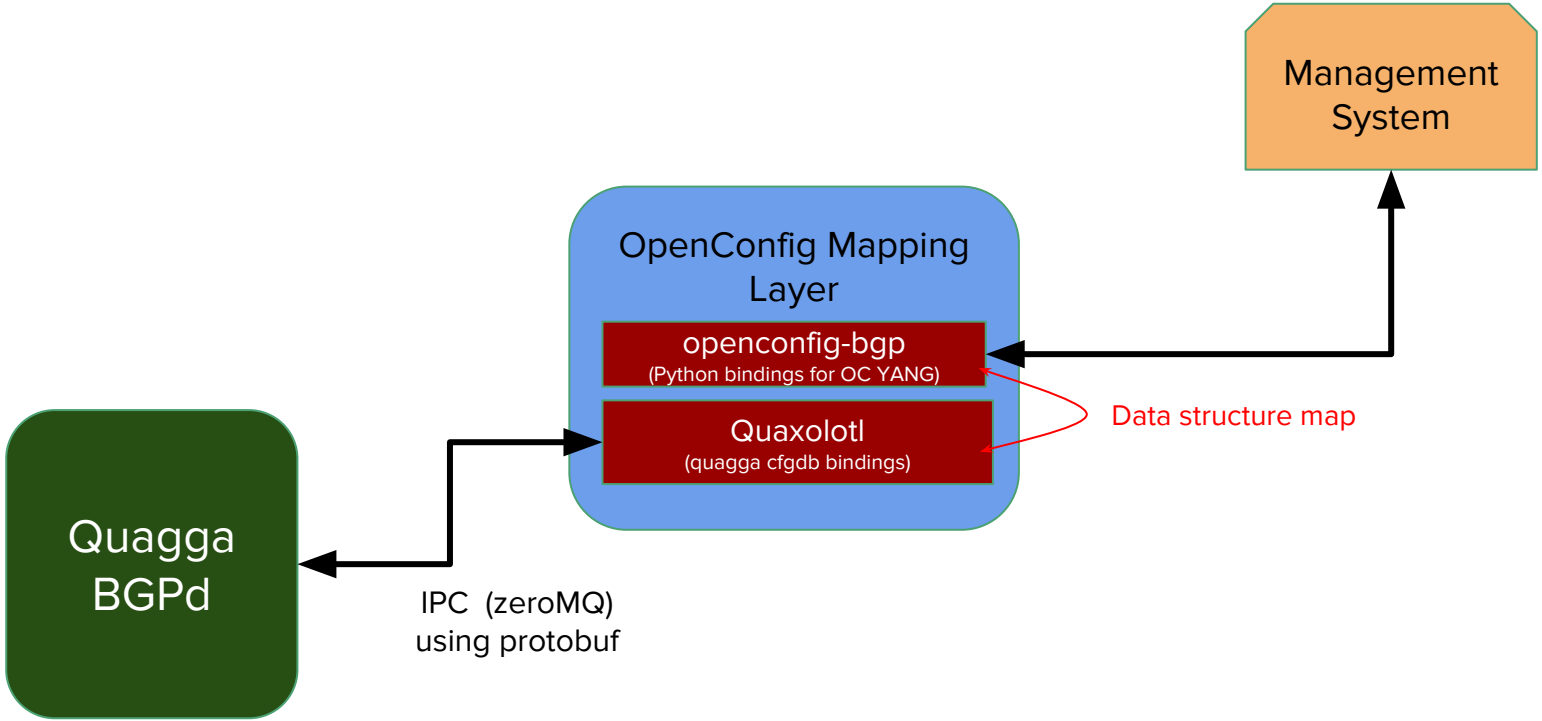
OpenConfig-1: OpenConfig interfaces for Quagga

Participants: Rob Shakir, Theo Zourzouvillys (Jive Comm), Anees Shaikh, Josh George (Google), Mattijs Jonker (Univ. of Twente)

Problem

- every BGP speaker (hardware or software) requires a significant amount of proprietary integration to speak the dialect of its management language
- OpenConfig is an operator working group that is defining vendor-neutral data models (APIs) for configuration and operational state for all parts of network infra, including BGP and routing policy (openconfig.net)
  - these models are being supported natively on JUNOS, IOS-XR, EOS, and others
- extend OpenConfig support to open source software BGP implementations
  - Focus on Quagga and it's emerging configuration API support
- benefit: enables a common management interface across vendor devices, and software implementations
  - simplifying client systems (e.g., NMSes, experimental testbeds, etc.)

Quagga management API with OpenConfig



Example Mapping - basic BGP peer

```
router bgp 6643
 ...
 neighbor 50.242.148.217 remote-as 7922
 neighbor 50.242.148.217 description :d=Comcast (Full Transit):
 neighbor 50.242.148.217 timers 10 30
 ...
```

```
{
    "bgp": {
        "neighbors": {
            "neighbor": {
                "50.242.148.217": {
                    "neighbor-address": "50.242.148.217",
                    "config": {
                        "neighbor-address": "50.242.148.217",
                        "peer-as": 7922,
                        "enabled": false,
                        "description": ":d=Comcast (Full Transit):"
                    },
                    "timers": {
                        "config": {
                            "keepalive-interval": 10.0,
                            "hold-time": 30.0
                        }
                    },
                    "transport": {
                        "config": {
                            "passive-mode": false
                        }
                    }
                }
            }
        }
    }
}
```

Example Map

```
QMapBGPNeighbors = {
    '/bgp/peers':  { 'oc_target': '/bgp/neighbors/neighbor', 'method': 'list_add', 'key': 'config/host' },
    '/bgp/peers/config/holdtime': { 'oc_target': 'timers/config/hold-time', 'method': 'subst'},
    '/bgp/peers/config/as_': { 'oc_target': 'config/peer_as', 'method': 'subst'},
    '/bgp/peers/config/desc': { 'oc_target': 'config/description', 'method': 'subst'},
    '/bgp/peers/config/updateSource': { 'oc_target': 'transport/config/local-address', 'method': 'subst'},
    '/bgp/peers/config/keepalive': { 'oc_target': 'timers/config/keepalive-interval', 'method': 'subst'},
}

QMapBGPGlobal = {
  '/bgp/config/as_': {'oc_target' : '/bgp/global/config/as', 'method': 'subst'},
  '/bgp/config/cfAlwaysCompareMed' : {'oc_target': '/bgp/global/route-selection-options/config/always-compare-med', 'method': 'subst'},
  '/bgp/config/cfAspathConfed' : {'oc_target': None },
  '/bgp/config/cfAspathIgnore' : {'oc_target': '/bgp/global/route-selection-options/config/ignore-as-path-length', 'method': 'subst'},
  '/bgp/config/cfAspathMpathRelax' : {'oc_target': None },
  '/bgp/config/cfCompareRouterID' : {'oc_target': None },
}

qmap = [
            ('/bgp/peers', QMapBGPNeighbors),
            ('/bgp/config', QMapBGPGlobal),
       ]
```

Add new list entries

Substitute values

Paths to map

Status update

- completed a prototype to extract data from Quagga BGPd and present in OpenConfig structure and format
- tested with several actual network BGP configurations

Next steps

- extend to being able to set configuration values using OpenConfig
  - current Quagga BGPd management API enables get -- no set capability yet
  - Quaxolotl code still to be released
- provide support for routing policy