openconfig-network-instance
Overview and Motivation.

- Network instance models a generic virtual forwarding table on a device.
- Each device has at least one network instance - which represents the global forwarding table.
- **Motivation**: Ensure that we can model mixed-mode entities (e.g., L2VSI with a numbered IP interface within it - rVPLS).
Basic Anatomy of a Network Instance.

- **interfaces**: reference to interfaces that are associated with this instance (currently basic, needs subints).
- **connection-points**: logical groupings of local or remote interfaces which make up a service interface (e.g., PWE redundancy).
- **tables**: The individual forwarding tables (RIBs or FIBs) which exist within the network instance.
- **protocols**: The protocol instances which run under the network instance - e.g., BGP or static routes.

**Inter-instance-policies**: controls distribution of entries between instances.
Tables - Two Models.

- All protocols install their entries into a single table (per-address-family).
- ‘Leaking’ happens by default between tables (an export policy in BGP can match ‘install-protocol-eq IS-IS’ when exporting from ‘ipv4-unicast’ table).

- Each protocol installs entries into its own (per-address-family) table.
- Leaking does not happen by default - an export policy in BGP matching ‘install-protocol-eq’ IS-IS does not match any entries.
- Requires explicit connection of tables.
Table Configuration - Per-AF Table.

- Each protocol specifies its ‘target-table’ to install entries into, on a per-address-family basis.
- For an implementation with a single table per-AFI, this specifies the same table as the target for all protocols.

NB: target-table needs to be converted to a list in current YANG.
Table Configuration - Per-Protocol Per-AF Tables.

- For an implementation with per-protocol, per-AFI tables, then a separate table is defined (in tables/) per protocol, and the target-table specified as such for each protocol.

NB: target-table needs to be converted to a list in current YANG.
Leaking Between Tables.

- **IPv4 Unicast**: Allow use of labelled next-hops for IPv4 unicast routes.
- **IPv4 Labelled Unicast**: Allow use of labelled next-hops for IPv4 unicast routes.
- **BGP IPv4 Unicast**: Allow BGP to export static IPv4 routes.
- **Static IPv4 Unicast**: Allow BGP to export static IPv4 routes.

```
"tables": {
  "table": {
    "ipv4-unicast": { ... },
    "ipv4-labelled-unicast": { ... }
  },
  "table-connections": {
    "table-connection": {
      "ipv4-labelled-unicast ipv4-unicast": {
        "src": "ipv4-labelled-unicast",
        "dst": "ipv4-unicast",
        "apply-policy": { ... }
      }
    }
  }
},
"tables": {
  "table": {
    "bgp-ipv4-unicast": { ... },
    "static-ipv4-unicast": { ... }
  },
  "table-connections": {
    "table-connection": {
      "static-ipv4-unicast bgp-ipv4-unicast": {
        "src": "static-ipv4-unicast",
        "dst": "bgp-ipv4-unicast",
        "apply-policy": { ... }
      }
    }
  }
}
```
network-instance Model TODO:

- ietf-bgp/openconfig-bgp can directly augment network-instances/network-instance/protocols/protocol today - no changes to model required (with described table changes).

- Integrate openconfig-mpls into this model - again, should be a case of augmenting protocols/protocol.

- Integrate RIB and LFIB model - determine how these interact with the ‘tables’ construct.

-FIXME:
  - target-table specified on a per-AF basis.
  - augment to interfaces/interface and interfaces/interface/subinterfaces/subinterface to reference network-instance to configure instance ownership - with state leaf-list of leafrefs within the network instance.
  - Add additional L2 configuration options (MTU, control-word, signalling-protocol, FAT-PW...