Network Topology Models
draft-ietf-i2rs-yang-network-topo-01.txt*
draft-ietf-i2rs-yang-l3-topo-00.txt^

IETF 94, 2-Nov-2015, Yokohama, Japan

Alexander Clemm, Jan Medved (Cisco)*^

Robert Varga, Tony Tkacik (Pantheon)*^

Nitin Bahadur (Bracket Computing)*^

Hari Ananthakrishnan (Packet Design)*^

Xufeng Liu (Ericsson)^

Igor Bryskin, Aihua Guo (Adva Optical)^

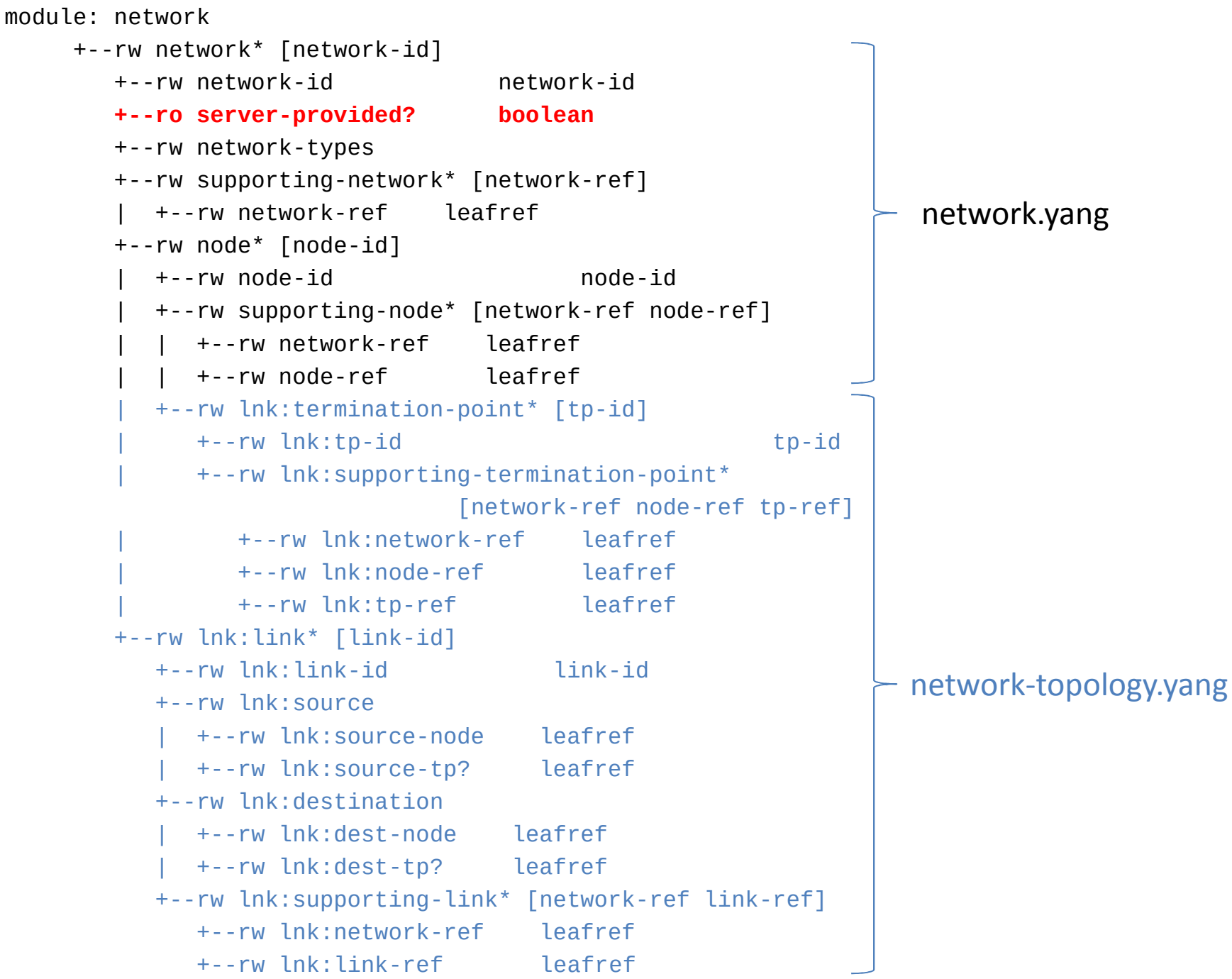
Pavan Beeram (Juniper)^

Updates from WGLC

- Comments received from Juergen Schoenwaelder
  - Mail from 10/1 – number of nits – accepted
- Open topic: dealing with configurable vs. read-only topology
  - Topologies can be layered
  - Topology/nodes/links/TPs can be layered on top of supporting topology/nodes/links/TPs
  - Specifically, topologies could be configured on top of topologies that are server-provided
- Current solution
  - Server-provided object indicates source of network topology

```
description
"Indicates whether the information concerning this
 particular network is populated by the server
 (server-provided true, the general case for network
 information discovered from the server),
 or whether it is configured by a client
 (server-provided true, possible e.g. for
 service overlays managed through a controller).";
```

  - Same model is used whether network topology is discovered or whether it is configured by a client application
  - Server-provided topologies are in effect configured by server-internal app; attempts to configure by non-server should in effect be rejected


module: network
+--rw network* [network-id]
+--rw network-id network-id
+--ro server-provided? boolean
+--rw network-types
+--rw supporting-network* [network-ref]
| +--rw network-ref leafref
+--rw node* [node-id]
| +--rw node-id node-id
| +--rw supporting-node* [network-ref node-ref]
| | +--rw network-ref leafref
| | +--rw node-ref leafref
| +--rw lnk:termination-point* [tp-id]
| +--rw lnk:tp-id tp-id
| +--rw lnk:supporting-termination-point*
[network-ref node-ref tp-ref]
| +--rw lnk:network-ref leafref
| +--rw lnk:node-ref leafref
| +--rw lnk:tp-ref leafref
+--rw lnk:link* [link-id]
+--rw lnk:link-id link-id
+--rw lnk:source
| +--rw lnk:source-node leafref
| +--rw lnk:source-tp? leafref
+--rw lnk:destination
| +--rw lnk:dest-node leafref
| +--rw lnk:dest-tp? leafref
+--rw lnk:supporting-link* [network-ref link-ref]
+--rw lnk:network-ref leafref
+--rw lnk:link-ref leafref
network.yang
network-topology.yang

Issues raised

- Separate config true and config false information
  → Move “server-provided” object into a separate tree for state
- What happens to overlay in wake of underlay changes
  - Can maintain integrity by building topologies from bottom to top, tearing down from top to bottom
  - Changing underlay can lead to referential integrity issues
  - Requires auto-updating the overlay as needed (e.g. by removing references to underlay concurrently)
  - Alternatively, might consider introducing state indicating whether layering integrity compromised (and notifications when this occurs)
- How to deal with server-provided semantics – suggested alternatives
  - Leave as is – topology data is configuration data, populated by client or server-based app
  - Split model into network state (operational) and network configuration
  - Don’t support configuration of topology
  - Add metadata annotation

Treat all topology as “regular” config

- Servers can have embedded app that populates topology
- Issues with not differentiating between apps
  - Other clients could access server-provided topology as well
    - Server provided app could “revert back”
    - Malicious clients could “lock out” server provided app
- Possible extension: add metadata
  - Tag server-provided data as such
  - Configuration and locking attempts by other clients will be rejected
  - Metadata does affect behavior of config data in this case

Split model into network state and network config

- Network state tracks network configuration where it is provided
- Issues with model split
  - Configuration data in YANG cannot refer operational data in YANG
    - Layering of configured overlays on top of server-provided networks?
  - Possible solution: replace leafrefs to references with names
    - This punts the problem to the user; model no longer ensures integrity
    - Reminiscent of MIB descriptions
    - Distinguish topology/node/link/tp-"config"-ref and topology/nod/link/tp-state-ref
  - Various integrity rules of the model can no longer be supported
    - E.g. underlay nodes/links need to be part of a supporting network
    - List keys cannot support choices of data nodes (either state of or config)
  - More complex model augmentation
    - Two subtrees to augment (cannot augment groupings themselves)
  - Concern: split results in higher complexity and reduced accuracy, vs. best practices purity
- Note: model split by itself does not address problem of underlay changes (applications still required to deal with updating overlay)

Leave current solution in place

- Accept semantics of server-provided topology
  - Leverage YANG semantics provided by YANG framework to validate configuration of overlays, instead of punting this responsibility to applications
  - Simpler model structure, augmentation
  - Involves acceptance to not treat topology configuration as configuration like any other
- Possible modifications
  - Move server-provided leaf into separate state branch
  - Add documentation re: referential integrity in event of underlay changes