YANG Data Model for Network Topology

draft-clemm-netmod-yang-network-topo-01.txt

Alexander Clemm, alex@cisco.com
Hariharan Ananthakrishnan, hanantha@juniper.net
Jan Medved, jmedved@cisco.com
Tony Tkacik, ttkacik@cisco.com
Robert Varga, robert.varga@pantheon.sk
Nitin Bahadur, nitinb@juniper.net
Purpose

• YANG Data Model for Network Topologies
• Generic topology model, extensions for specific topologies
  – L3 Unicast IGP, OSPF, IS-IS as part of this draft
  – Can be extended for other topologies
• Applications
  – Data nodes capture and reconcile their understanding of network topology, propagate topology info
  – Network controllers represent controller network topology (e.g. Open Daylight)
• Ask: Adopt as WG item
  – Presented in Berlin
  – Positive feedback received so far
Recap: Data model structure

YANG modules

Traffic engineering data definitions/groupings, used in multiple topologies

YANG module: Generic topology model

Other topologies (future)

Model allows for multiple levels of refinement

Additional topologies can be added at any refinement level, represented through additional YANG modules
Recap: Data model structure (contd.)

- Links connect nodes, are terminated by termination points
- Topologies can refer to underlay topologies
- Links can refer to underlay links
- Nodes can refer to underlay nodes
- Unidirectional, point-to-point links represent non-PTP through hierarchies of nodes, links

Information model underlying base “network topology” YANG module
Recap: Data model structure (contd.)

- Derive Layer 3 Unicast IGP topology object classes
- Integrity rules ensure links, nodes, topology of matching type
Recap: Data model structure (contd.)

- Topology
  - Node
    - TP
    - Prefix
    - L3 IGP node
    - L3 IGP link
  - Link
    - L3 Unicast IGP Topology
    - OSPF topo
    - ISIS topo
    - **Pattern recurses**
Recap: YANG structure

module: network-topology
  ---rw network-topology
  ---rw topology [topology-id]
    ---rw topology-id topology-id
    ---ro server-provided? boolean // ro flag, to be discussed
  ---rw topology-types
  ---rw underlay-topology [topology-ref]
    |   ---rw topology-ref topology-ref
  ---rw node [node-id]
    |   ---rw node-id node-id
    |   |   ---rw supporting-node [node-ref]
    |   |   |   ---rw node-ref node-ref
    |   ---rw termination-point [tp-id]
    |     ---rw tp-id tp-id
    |     ---ro tp-ref* tp-ref
  ---rw link [link-id]
    ---rw link-id link-id
    ---rw source
      |   ---rw source-node node-ref
      |   ---rw source-tp? tp-ref
    ---rw destination
      |   ---rw dest-node node-ref
      |   ---rw dest-tp? tp-ref
    ---rw supporting-link [link-ref]
      ---rw link-ref link-ref
Recap: YANG structure (contd.)

module: network-topology
  +++-rw network-topology
    +++-rw topology [topology-id]
      +++-rw topology-types
        |   +++-rw l3t:13-unicast-igp-topology?
      +++-rw node [node-id]
        |   +++-rw termination-point [tp-id]
          |   |   +++-rw l3t:igp-termination-point-attributes
          |   |     +++-rw (termination-point-type)?
          |   |     |     +++:(ip)
          |   |     |     |     +++-rw l3t:ip-address* inet:ip-address
          |   |     |     |     +++-:(unnumbered)
          |   |     |     +++-rw l3t:unnumbered-id? uint32
          |   |   +++-rw l3t:igp-node-attributes
          |   |     +++-rw l3t:name? inet:domain-name
          |   |     +++-rw l3t:flag* flag-type
          |   |     +++-rw l3t:router-id* inet:ip-address
          |   |     +++-rw l3t:prefix [prefix]
          |   |     |     +++-rw l3t:prefix inet:ip-prefix
          |   |     +++-rw l3t:metric? uint32
          |   |     +++-rw l3t:flag* flag-type
        |   +++-rw link [link-id]
          |   +++-rw l3t:igp-link-attributes
          |        +++-rw l3t:name? string
          |        +++-rw l3t:flag* flag-type
          |        +++-rw l3t:metric? uint32
        +++-rw l3t:igp-topology-attributes
          |   +++-rw l3t:name? string
          |   +++-rw l3t:flag* flag-type
Open issues

• Read-write or read-only
  – Mixed hierarchies conceivable, e.g. configurable overlay topology versus discovered L2/underlay
  – Alternative 1 (current option): read-write with "flag"
  – Alternative 2:
    • read-only
    • additional configuration “sand box” mirrored by main topology
Questions?