YANG Patch Media Type

draft-ietf-netconf-yang-patch-03
NETCONF WG
IETF #92 Dallas, TX, USA

Andy Bierman <andy@yumaworks.com>
Martin Björklund <mbj@tail-f.com>
Kent Watsen <kwatsen@juniper.net>

v0.1

Agenda

- YANG Patch-03 Update highlights
- Review resolution proposals for open issues
  - https://github.com/netconf-wg/yang-patch/issues
- Next Steps

Issues resolved in draft-03

- #1 Structure of edit value
  - MUST be a container with 1 child node matching target
- #2 parsing QNames in value parameter
  - dead: considered implementation detail

Edit List vs. <config> Blob

- NETCONF defines a datastore patch using <edit-config>;
  Pros for edit list:
  - <config> is poorly specified wrt/ corner cases
    - Nested operations and duplicate sub-trees
    - Must provide complete resource representation in order to move an ordered-by user list or leaf-list
    - Attributes (operation, insert, value, key) are XML specific
      - JSON attributes are very inefficient to implement
      - Binary encodings do not support scoped meta-data so <config> is not protocol neutral
  - <config> is not resource-oriented
    - always patching the datastore root
  - Error information may not identify the edit step that was rejected by the server
    - new yang-patch-status has additional info that may help identify correctable errors

Edit List vs. <config> Blob (2)

- NETCONF defines a datastore patch using <edit-config>
  - Cons for edit list:
    - Two ways to do the same thing, so not needed
    - Ordered edit list is new and requires significant development resources
    - Not clear what edit ordering really means since only the result of all the edits is subject to YANG validation

Issues from Jürgen

- What is the reason for not defining a NETCONF RPC operation?
  - Intended for use with RESTCONF, not NETCONF
  - NETCONF-EX has <edit2> operation; WG does not seem interested in changing NETCONF
  - Sec 2.1 mentions how to define a NETCONF operation but does not define it

Issues from Jürgen (2)

- Why is it useful to have absolute and relative target paths?
  - To specify the resource being patched
- Would it not be simpler if target path would always be absolute that the root would always be a datastore?
  - PATCH requires the resource being patched to exist
  - If-Match edit-collision detection is more granular if the data resource is specified instead of the datastore

Issues from Jürgen (3)

- Why is there an automatic commit to startup?
  - Unified datastore requires the server to copy to NV-storage if a manual step would be required for NETCONF
  - There is no access to any particular datastore; they are just implementation details in RESTCONF
  - The server will persist the edit because the WG did not agree on any client controls for NV-storage

Issues from Jürgen (4)

- Why do we use RESTCONF specific types in the YANG module?
  - RESTCONF uses a target URL
- Why not be protocol agnostic by using instance-identifier instead of target-resource-offset?
  - It is more complicated and more verbose

Issues from Jürgen (5)

- How can the error objects use RESTCONF specific definitions if we claim that this would also work with NETCONF?
  - We should remove any mention of NETCONF beyond the minimum to integrate RESTCONF
  - The NETCONF protocol is not being updated by YANG Patch – this should be more clear in the next draft

Issues from Jürgen (6)

- Why is the comment restricted to 1024 characters? Seems like an arbitrary restriction.
  - Agreed. Text can be added that says any size comment may be attempted, but the server can return a 'too-big' error-tag at any time, so the client might consider this issue when sending this parameter

Issues from Jürgen (7)

- Where is it detailed how an implementation has to validate a patch edit?
  - The YANG validation only applies to the result of all individual edits
  - Text about “MUST be validated by the server to be a well-formed message” will be removed.
    - Any implementation that alters the configuration for incomplete or malformed messages is non-compliant, so no need to mention it

Issues from Jürgen (8)

- The usage of anyxml seems under-specified. It is non-interoperable using the JSON encoding document
  - The JSON draft will be cited in the next version
  - If the WG cannot agree on how anyxml is handled in JSON then JSON support should be removed
  - Mixed mode XML is not forbidden for the 'value' node, if the target resource (or sub-resource) is anyxml

Summary

- Need to decide if WG is throwing out YANG Patch
  - Could start over on a different PATCH method
  - Could ignore the multi-resource-edit problem and let vendors define their own PATCH media types
  - Could finish this draft using an edit list approach