NETCONF <get2> Operation

draft-bierman-netconf-get2-02
IETF 85, November 2012

Andy Bierman
November 7, 2012

Problem Statement

- NETCONF retrieval operations are not efficient enough to scale to large datastore sizes
- New mechanisms are needed to improve client and server efficiency

P1:Wrong Datatype Returned

- The <get> operation returns all data, including all the config=true nodes that are returned by <get-config>.
  - A mechanism is needed to retrieve just operational data that does not duplicate the configuration datastore contents

P2:No last-modified Filtering

- A NETCONF application which periodically polls datastore contents must retrieve all the relevant content and compare it to a stored version.
  - A time-stamp and time-filter mechanism is needed so the client can specify that data should only be returned if it has been modified since a specified date and time. The client must be able to retrieve the last-modified time for nodes which support it.

P3:No Instance Discovery

- The client needs to provide explicit subtree filters for all the list entries it wishes to discover. There is no simple mechanism to retrieve just instance naming information.
  - A simple mechanism is needed to retrieve only the key values that identify a particular list instance
  - The client needs be able to discover nested list instances, which subtree filters cannot support

P4: No Subtree depth control

- A client can select which subtrees to retrieve but cannot efficiently control the depth of descendant nodes to return. E.g., a client cannot discover if a P-container exists by retrieving just the container node.
  - A mechanism is needed to control the number of nested layers within a subtree that should be returned

P5:Filter not extensible

- The NETCONF <filter> parameter is defined in a way that does not allow YANG augment-stmt to extend the filter types. Proper YANG statements are not used to define the filter.
  - A mechanism is needed that will allow new content filter mechanisms to be defined which do not require existing mechanisms to be republished.

P6:No data source metadata

- There are no standard mechanisms for a client to determine the data source of operational data. E.g., the current time of day may be derived from NTP or a CLI operation
  - A mechanism is needed to allow a client to discover the data source for a particular operational data node

Solution Proposal

- Add a new operation <get2> to augment or eventually replace the <get> and <get-config> operations
  - keep existing functionality
  - add new functionality

Time Filters

- time-filter: retrieval nodes only if changed since the specified date-and-time
  - server MUST maintain a time-stamp for the entire running datastore
  - server MAY maintain a time-stamp for particular data nodes
  - client can discover which nodes have timestamps

Depth Filters

- depth-filter: request specific number of nest levels, starting from the 'selection' node
  - depth=1 returns just the requested node
  - depth=2 returns the requested node and its child nodes

Operational Datastore

- New source datastore called 'operational' defined for <get2>, which contains:
  - all config=false nodes
  - any ancestor nodes of config=false nodes
  - any list keys for ancestor-or-self axis config=false nodes

Data Sources

- YANG identity to identify data source and 5 initial data sources:
  - server
  - running
  - operation
  - ntp
  - dns
- Data model specific data sources will be defined as needed

<get2> Parameters

- source: datastore to retrieve from
- filter-spec: content filter
- keys-only: discover instances
- if-modified-since: time filter
- depth: depth filter
- with-defaults: defaults retrieval
- with-timestamps: last change time retrieval
- with-data-sources: data source retrieval