NETCONF WG
60th IETF

San Diego, CA USA

August 5, 2004

NETCONF WG Details

- Mailing List
  - » Discussion: netconf@ops.ietf.org
  - » Subscribe: netconf-request@ops.ietf.org
    - – ‘subscribe’ in the message body
  - » Archive: http://ops.ietf.org/lists/netconf/
- WG Chairs
  - » Simon Leinen <simon@switch.ch>
  - » Andy Bierman <abierman@cisco.com>
- WG Charter Page
  - » http://www.ietf.org/html.charters/netconf-charter.html
- WG Home Page
  - » http://www.ops.ietf.org/netconf/
- WG Issues List
  - » http://www.nextbeacon.com/netconf/

NETCONF Drafts

- WG Internet Drafts:
  - » NETCONF Configuration Protocol
    - – draft-ietf-netconf-prot-03.txt
  - » BEEP Application Protocol Mapping for NETCONF
    - – draft-ietf-netconf-beep-01.txt
  - » NETCONF Over SOAP
    - – draft-ietf-netconf-soap-02.txt
  - » Using the NETCONF Configuration Protocol over Secure Shell (SSH)
    - – draft-ietf-netconf-ssh-01.txt

NETCONF WG Agenda

- Report on NETMOD BOF (Sharon Chisholm, 15 min)
  - » BOF results summary
  - » Impact on NETCONF protocol documents
    (e.g., move netconf-state data model to a different document)
- Security Issues (Wes Hardaker, 15 min)
- Discussion of Major Open Issues (30 min)
  - » Retrieval filtering
  - » Rollback
  - » Default operation
- Discussion of WG Documents (50 min)
  - » NETCONF Configuration Protocol
  - » BEEP Application Protocol Mapping for NETCONF
  - » NETCONF Over SOAP
  - » Using the NETCONF Configuration Protocol over Secure Shell (SSH)
- Next Steps (10 min)
  - » Need to finish up the last bits, and start WG Last Calls ASAP

Retrieval Filtering

- Open Issue:
  - » Need to select a mandatory-to-implement mechanism to request a subset of a data model with <get> and <get-config> operations
- 2 proposals under consideration
  - » Subtree filtering
    - – Current method in the spec (examples only), documented in email from <abierman@cisco.com> on 5/28/04
  - » Xpath subset
    - – Provides almost equivalent functionality to subtree filtering, documented in email from <j.schoenwaelder@iu-bremen.de> on 6/5/04

Comparing Subtree vs Xpath subset

- Sub-tree filtering
  - » Pros:
    - – Fully specified and not intended to be extended
      - Use full Xpath for more functionality
  - » Cons
    - – Creates a NETCONF-specific XML filtering mechanism
- XPath subset
  - » Pros:
    - – Subset of a well-implemented standard
    - – Developers will not need to know 2 filtering mechanisms if full Xpath is implemented on some devices
  - » Cons
    - – Creates a NETCONF-specific subset of Xpath
    - – Vendors may not limit their implementations to the NETCONF subset, but rather pick and choose features (from full Xpath) to add to the subset

Retrieval Filtering Decision Points

- Support full Xpath as an option
  - » Full Xpath MAY be implemented, indicated by the #xpath capability
    - – <filter> element needs to support full Xpath somehow
      - E.g., <filter filter-type="xpath">
- Choose mandatory-to-implement filter
  - » A) none
  - » B) subtree
  - » C) Xpath subset

Rollback Overview

- WG agreed to address rollback and retrieval filtering at IETF #59
  - » Needed to undo edits or commits to the <running> configuration to support multi-device configuration updates
  - » Proposal sent by <abierman@cisco.com> on 7/16/04 to add:
    - – #rollback capability
    - – <checkpoint> operation
    - – <rollback> operation
    - – <discard-checkpoint> operation
    - – <rollback-depth> data model element

Rollback Definition (1/5)

- Design
  - » Per session ring buffer of implementation dependent “restore information”
  - » Restores entire configuration to the previous state; not a per-session restore operation
    - – Locking must be used properly for concurrent configuration changes
  - » Restore data is not accessible with any protocol operations
    - – Implementation-specific format, not a configuration database
- #rollback capability
  - » Indicates the <checkpoint>, <discard-checkpoint>, and <rollback> operations are supported. The <rollback-depth> parameter value in the netconf-state data model must be equal to a value greater than zero.

Rollback Definition (2/5)

- <checkpoint> operation
  - » This protocol operation causes the agent to take whatever steps required to capture the current state of the <running> configuration, so a subsequent <rollback> operation will cause the <running> configuration to revert to this state. This operation has no parameters.
  - » If the per-session ring buffer is full, then the data for the oldest restore operation is deleted and restore data for the current state of the <running> configuration is saved
  - » It is possible for a <checkpoint> to fail due to insufficient resources

Rollback Definition (3/5)

- <rollback> operation
  - » This protocol operation causes the agent to take whatever steps required to revert the <running> configuration to the state captured with the most recent <checkpoint> data stored for the session.
    - – The checkpoint data is removed from the session's rollback ring buffer after this operation is completed.
    - – The agent SHOULD revert the configuration in a manner that causes the least amount of disruption to the running network.
    - – A “restart required” warning may be returned if complete restoration of operational state requires a HW or SW restart
      - Need to define elements to express the details of the restart, for the <error-info> parameter of <rpc-error>
    - – Operation can fail if ring buffer is empty or insufficient resources

Rollback Definition (4/5)

- <discard-checkpoint> operation
  - » This protocol operation causes the agent to remove a specified number (or all) of configurations previously saved with the <checkpoint> operation.
  - » Parameter: <count> : NonNegativeInteger
    - – The number of saved configurations to discard. The value zero indicates that all saved rollback configurations for this session should be discarded. If this value is greater than the current number of saved rollback configurations, then a BAD_ELEMENT error will be returned, with the <bad-element> value set to "count".
    - – This parameter is optional. The default value of one is used if this parameter is not present.
  - » All restore information for a session is discarded when that session is terminated

Rollback Definition (5/5)

- <rollback-depth> data model element
  - » A read-only parameter to indicate the maximum number of saved rollback configurations the agent will support for a single NETCONF session. If the #rollback capability is not supported, then the value zero MUST be returned. A positive integer value indicates the maximum depth of the rollback ring-buffer that the agent supports for any NETCONF session.
  - » Open Issues
    - – Do we need to allow <rollback-depth> to be a different value, depending on the access rights of the user? I.e., a per-session parameter instead of a per-device parameter?
    - – Do we need to encode the rollback-depth value in the #rollback capability, e.g.,:
      - urn:ietf:params:xml:ns:netconf:base:1.0#rollback?depth=4

Rollback Decision Points

- Should the rollback feature be added to the protocol operation set?
- If so, should the proposed text (abierman email) be used as the starting point for the operations in the next version of the protocol draft?
  - » Should the <rollback-depth> be per-session instead of a device-wide constant?
  - » Should the #rollback URI encode the rollback-depth?

Default Operation

- Problem statement:
  - » Current draft does not properly support scoped edit-config operations (i.e., prevent inadvertent (or unauthorized) data re-creation)
  - » The default operation for edit-config is merge, which is applied at each node of the specified <config> parameter.
    - – Agent will process the <config> tree top-down, and apply the appropriate operation (via internal API).
  - » There is no way to indicate that no operation should be applied at a particular node

Default Operation Can’t Always Be Merge

- Modify a node in the data model without touching any of its ancestors
  - » Current draft:
    - –
      ```
      <edit-config>
        <config xmlns="example-uri" xmlns:xc="netconf-uri">
          <users>
            <user>
              <name>barney</name>
              <type xc:operation="replace">superuser</type>
            </user>
          </users>
        </config>
      </edit-config>
      ```
    - – This operation is intended to only modify the ‘type’ element, but the agent will attempt to create the 'barney' entry if it doesn't already exist, since the operation in effect at the ‘user’ node is merge.

Default Operation Is Sometimes ‘none’

- Need a new edit-config parameter called ‘default-op’
  - » Proposed by <abierman@cisco.com> on 7/15/04
  - » 'default-op': enum (none, merge, replace)
    - – The edit-config operation that will assumed if the operation attribute is not present for a particular data model object.
    - – Default value: merge
  - » Example
    - –

```
<edit-config>
    <default-op>none</default-op>
    <config xmlns="example-uri" xmlns:xc="netconf-uri">
    <users>
      <user>
        <name>barney</name>
        <type xc:operation="replace">superuser</type>
      </user>
    </users>
  </config>
</edit-config>
```

    - – This operation will only modify the ‘type’ element, and the operation will fail if its ancestor nodes do not already exist.

Default Operation Is Sometimes ‘replace’

- Useful for loading previously saved configuration data as an opaque XML block
  - » 1) copy-config is used to retrieve a configuration
  - » 2) edit-config is used to overwrite a configuration (e.g., bring up a failover device)
    - – Set the <default-op> to ‘replace’
    - – Place the entire retrieved configuration block into the <config> parameter without inspection or modification
- Currently need to find all the top-level elements in the data model and add *xmlns* statements (for NETCONF base) and *xc:operation="replace"* attributes to them.

Default Operation Decision Points

- Should the default-op parameter be added to the edit-config operation?
- If so, should the proposed text (abierman email) be used as the starting point for the operations in the next version of the protocol draft?

Default Configuration Target

- Problem statement
  - » Protocol draft says the default for <lock> and <unlock> operation changes from <running> to <candidate> if #candidate capability is supported
  - » This cannot be properly expressed in the XSD
  - » This presumes the implementation allows only edits to the <candidate> if #candidate is supported, but an agent could conceivably allow writes to both <candidate> and <running>
- Proposal
  - » Don’t have any default for the <lock> and <unlock> operations

Application Mapping General Issues

- Titles should be consistent; every one is a different style.
  - We need to decide which style is best:
    - BEEP Application Protocol Mapping for NETCONF
    - NETCONF Over SOAP
    - Using the NETCONF Configuration Protocol over Secure Shell (SSH)
- The following normative sections should exist in all documents, and be consistent in content:
  - NETCONF Session Establishment
  - NETCONF Capabilities Exchange
  - NETCONF Session Usage
  - NETCONF Session Teardown
    - In the session tear-down section, each document must explain what to do when <close-session> or <kill-session> operations are invoked

NETCONF Over SOAP Issues

- Problem Statement:
  - » Document seems too specific to SOAP over HTTP
    - – HTTP is not really an appropriate transport for NETCONF
  - » No support for SOAP over BEEP
    - – This standard binding is more useful to NETCONF application developers than SOAP over HTTP, although not widely deployed
  - » Issues wrt/ HTTP caching should be mentioned in sec 2.4
- Proposal:
  - » Separate SOAP-generic and SOAP-over-HTTP text as much as possible
  - » Add text to support for SOAP over BEEP (RFC 3288)
  - » Update section 2.4 with text on potential impact on NETCONF sessions due to HTTP caching