NETCONF WG
58th IETF
Minneapolis, MN
November 10, 2003
November 12, 2003

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/

NETCONF Drafts

- WG Internet Drafts:
  - » NETCONF Configuration Protocol
    - – draft-ietf-netconf-prot-01.txt
  - » BEEP Application Protocol Mapping for NETCONF
    - – draft-ietf-netconf-beep-00.txt
  - » NETCONF Over SOAP
    - – draft-ietf-netconf-soap-00.txt
  - » Using the NETCONF Configuration Protocol over Secure Shell (SSH)
    - – draft-ietf-netconf-ssh-00.txt
- Additional Internet Drafts:
  - » XML Network Management Interface
    - – draft-weijing-netconf-interface-01.txt
  - » SYMPLE Scripting Protocol and architecture for seamless management of XML based mobile devices and SNMP based devices
    - – draft-adwankar-netconf-symple-00.txt

NETCONF WG Agenda (1/3)

- Application Mapping Issues (80 min)
  - » Presentations of Application Mapping documents
    - – NETCONF over BEEP
    - – NETCONF over SSH
    - – NETCONF over SOAP
      - Changes since last draft
      - Use cases
      - Implementation Issues
  - » Channel related issues
  - » Criteria for selecting a mandatory application mapping

- Operational Environment and Security Issues (40 min)
  - » Configuration databases
  - » Configuration files
  - » Running configuration
  - » Authorization model impact on protocol and op-env

NETCONF WG Agenda (2/3)

- Protocol Operations Issues (60 min)
  - » Presentation of NETCONF Protocol document
    - – Changes since last draft
    - – Open issues
  - » Edit-config operations
  - » Transactions
  - » Notifications
  - » Actions
  - » High-level RPCs
  - » Error handling
  - » XSD design
  - » Initial netconf-state XSD
- Data Modelling Issues (30 min)
  - » Presentation on data modelling impact on the protocol
  - » Discuss plan for starting standard data model work

NETCONF WG Agenda (3/3)

- Next Steps (15 min)
  - » Finishing the protocol draft
  - » Finalizing the set of application mappings
  - » Selecting a mandatory application mapping
- Alternate Approaches (if time permits) (15 min)
  - » Presentation on the SYMPLE Scripting Protocol

Application Mappings (1/2)

- NETCONF over BEEP issues
  - » sec. 2.2) why would a single <rpc> (MSG) which in turn causes a single <rpc-reply> result in multiple RPY messages?
  - » why is reliable syslog (3195) the only assumed notification data format? Any format should be possible on the notification channel
- NETCONF over SSH issues
  - » sec 4) why is reliable syslog (3195) the only assumed notification data format? Any format should be possible on the notification channel
  - » How does the high-level NETCONF application code know that some proto-ops (or management channel features) are not available?
  - » Should an end-of-message directive <?eom?> be used to provide message framing?

Application Mappings (2/2)

- NETCONF over SOAP
  - » NETCONF has no proxy; text related to proxy should be removed
  - » Consider SOAP over BEEP (better for NETCONF than HTTP)
    - – Is the SOAP community going to adopt this mapping soon?
  - » Is the SOAP usage defined in netconf-soap-00 reasonable or will existing tools expect more features to be used?
- Should the protocol be tailored (optimized) for each application mapping or should it be kept the same for each application mapping?

Application Mapping Comparison

Feature	BEEP	SOAP	SSH
Agent initiates connection	**Y**	**N**	**Y**
Multiple Channels per conn.	**Y**	**N**	**N**
Supports <rpc-progress>	**Y**	**Y***	**Y***
Supports <rpc-abort>	**Y**	**Y***	**Y***
Supports notifications	**Y**	**Y***	**Y****

* Requires multiple transport connections

** Notifications are mixed with responses

Selecting a Mandatory Mapping

- Should we choose by:
  - » Easiest to implement?
  - » Supports the most features?
  - » Has best applications tools support?
  - » Desired by the most operators?
  - » Desired by the most developers?
  - » Has the best transition from CLI support?
- Not easy to pick a clear winner!

Channels (1/2)

- Should a mapping be allowed to leave out protocol features?
  - » How important is rpc-progress? rpc-abort can be approximated by closing the session, but no workaround for rpc-progress.
- Should multiple transport connections per session be used to implement multiple channels or should protocols that cannot support channels implement less protocol features?
- [Protocol, sec. 2.4] Channel definitions
  - » Should we modularize the definition of channels?
    - – Define conceptual channels in the protocol
    - – Define channel implementation details in the application mapping

Channels (2/2)

- Management channel operations
  - » [Protocol, sec. 3.4] <rpc-abort-reply> sent immediately
    - – Is this sent when the abort is accomplished or when the <rpc-abort> is received?
  - » [Protocol, sec. 3.8] <rpc-progress>
    - – Should the manager be expected to handle 'extra' <rpc-progress> messages after the corresponding <rpc-reply> is completely received from the agent?
- Multiple Operations channels
  - » [Protocol, sec. 3.9] on any given operations channel..
    - – No mention of multiple operations channels given anywhere else in the document

Sessions

- Should netconf support multiple transport connections per session?
  - » <session-id> already exists; any other support needed?
  - » NETCONF over SOAP draft uses this design
- Should <session-id> be returned in session startup somehow?
- Is a special <edit-session> operation needed?
- [SSH issues, sec 5] Close session
  - » Use session-id == zero to indicate kill the current session
- [Protocol, sec. 5.8] <kill-session>
  - » Says an error is returned if the current session is killed. This conflicts with NETCONF over SSH spec.
  - » Should allow session-id == 0 to kill the current session.
  - » Should say how the session-id to kill is obtained (from a lock error or netconf-state data)

Capabilities

- [Protocol, sec. 7] Requirements
  - » Says MAY implement; some are MUST implement, such as base and 1 of manager/agent
- capabilities representation, choice of:
  - » URI
  - » URI + version
  - » Naming authority + capability name + version
- Version ID
  - » URI form has version ID before category:
    - – http://ietf.org/netconf/1.0/base (Style A)
    - – http://ietf.org/netconf/base/1.0 (Style B)
  - » Style A is not optimal for URIs that identify a data model. Want to version each component separately..
  - » Style B may not be optimal for netconf protocol capabilities because it may be simpler over time to increment the version for the entire protocol each time it is changed.

Capabilities for NETCONF v1.0

Name	Description
manager	NETCONF peer acting in manager role
agent	NETCONF peer acting in agent role
writable-running	<edit-config> and <copy-config> can be applied to the <running> configuration
candidate	Protocol operations can be applied to the <candidate> configuration
validate	<validate> can be applied to configuration databases
startup	Protocol operations can be applied to the <startup> configuration
notification	NETCONF peer can support a notification channel
url	Configurations can be identified by a (possibly remote) URL target

Missing Capabilities for NETCONF v1.0

Name	Description
user-db	User-created configuration databases are supported
user-file	User-created configuration files are supported
xpath	XPath content filtering is supported
rollback	Rollback of <running> configuration is supported
channels	All NETCONF channels are supported in some manner. Either multiple channels per connection or multiple connections per session.

Locks

- DoS attack possible if global lock allows users to lock more of the config dB than they have write access. Choose one of:
  - » Only users with all-access can lock the dB
  - » Only grant lock for areas that write-access is allowed
  - » Support partial locks
  - » Simply document the problem in the security section
- Is there a need for the <steal-lock> operation?
  - » Attacker can open a session and quickly grab a lock; kill-session followed by lock may not be fast enough to stop the attack, so steal-lock is needed.
  - » Need to steal the lock and kill the session in one operation or the session will not know the lock was stolen
- Should multiple occurrences of the <target> parameter be allowed to acquire multiple locks at once.
- Target parameter says it is optional; should say default <running>
- Negative response says session-id of lock owner will be returned; how will actually be done (specific element)
- What error response is given if a <lock> fails because a non-NETCONF entity holds the lock?

Locks on <candidate> Configuration

- [Protocol, sec. 7.5.5.1] <candidate> capability
  - » User cannot acquire lock if any other user has made changes to the candidate. Can this cause a DoS attack?
  - » How does session A know that session B invoked <discard-changes> and wiped out the change-set that session A was building?

Configuration databases

- Full set of protocol operations are supported
- [Protocol, sec. 4.1]
  - » No mention of user-named configuration databases or files

Configuration files

- Limited set of protocol operations allowed
  - » copy-config
  - » delete-config
  - » lock
  - » unlock
- User named databases
  - » How to tell if this is a config dB or a config file
  - » Need capabilities for named-db and named-file
    - – NETCONF draft is supposed to support this feature
  - » Is a <create-config> operation needed for named configurations?
  - » Can <copy-config> be used to create a named configuration?
  - » Need a data model to determine the list of named configurations and their attributes

<running> Configuration

- Complete retrieval requirement
  - » Some concern that it is too much of a burden to return the complete running configuration in one get-config operation (too complicated, too much memory required)
  - » Some concern that equivalent of SNMP context needs to be supported so data model extensions like the Bridge MIB (context == VLAN ID) can be supported
    - – Data model extensions are easier to achieve with XML; e.g., return VLAN ID as an attribute in the data model

<candidate> Configuration

- [Protocol, sec. 7.5] Candidate
  - » This doesn't say that the candidate config is global and shared by all sessions.
  - » Should we allow for per-session candidate configs?
    - – If these are just named configs then why have <candidate> at all? Why is a global candidate so special and so different from a per-session candidate?
- [Protocol, sec. 7.5.4.1] <commit>
  - » 'confirmed' parameter is problematic
    - – What happens if session is terminated before the 2nd <commit> is received?
    - – This provides an implicit rollback. The protocol should have an explicit rollback that works the same whether #candidate or #writable-running is supported
- <discard-changes>
  - » Says content 'automatic' is allowed for the <discard-changes> operation. This is not actually documented.

High Level RPC Functions

- Access control impact
  - » Treat as separate low-level data elements:
    - – If user is permitted to call add_bgp_neighbor then it doesn't matter if underlying data model elements are accessible by that user. Leave as an administrative issue.

  OR:

  - » Device must make sure user is permitted to access every underlying data model element used by a high level function.
    - – This may be difficult to enforce.

Notifications

- Should the protocol specify the data format (e.g. RFC 3195 syslog)? This seems to be a data model issue. The protocol should be data model neutral for notifications as well as RPC requests.
  - » Identify notification encoding with a namespace URI on the <notification> element
- [Protocol, sec. 7.8]
  - » No support for multiple notification channels.
  - » Open/close operations need to convey a channel #
  - » The <notification> element is not documented anywhere
- [Protocol, sec. 7.8.4.1] <open-notifications>
  - » 'matching' parameter (for filtering) is not documented at all. This should be removed or fully documented.

Text vs. XML Format

- Propose that the text/XML parameter be removed (only format is XML) and instead use a <text> element in the data models.
  - » This allows a namespace to be specified, which is useful to identify the text syntax and version:
    - – <t:text xmlns:t="http://example.com/CLI/1.3">
      command_1
      command_2
      </t:text>
- Need to understand how element sub-tree filtering and access control are affected by text mode
- <edit-config> should have the format option
  - » want a consistent solution for all operations

Protocol Operations (1/2)

- <get-config>
  - » Element filtering introduced by example. This needs to be completely defined in a previous section.
- <edit-config>
  - » No operation for 'add'
  - » No <error-option> for rollback-on-error
  - » No format parameter for text or XML encoding
- <copy-config>
  - » Says if <format> is omitted then use source format, but in XML, omitted means use the default, which is XML
- <get-state>
  - » Decided to have <get-all> instead

Protocol Operations (2/2)

- Additional Issues
  - » Can the target <startup> be deleted or just emptied of commands?
  - » Do we need a special designation for the 'last-known-good‘ configuration (remote or local)?
  - » Can any operations besides <copy-config> apply to remote configurations?
  - » XPath support
    - – Add now as an optional capability or leave out of v1.0?
  - » Error codes
    - – Need to define initial set, decide how they will be maintained
    - – Need to finalize all fields in the <error-reply> element
    - – Need details on including multiple <error-reply> elements per <rpc-reply>

Multi-device Transactions

- Distributed unit of work identifier
  - » It may be useful to add an additional parameter to RPC requests and replies to represent a distributed unit of work identifier. This would be provided by the application and ignored by the device (simply returned in the <rpc-reply>).
- Other features needed to better support multi-device configuration changes?

Specifying a Data Model Subset

- subset by name
  - » Table row(s)
  - » Column(s)
  - » Instances (of rows and/or columns)
- Subset by parameter value
  - » Similar to SQL SELECT
- Retrieval options
  - » Selected subset
  - » Instance identifiers of selected subset
  - » N nest levels of selected subset
  - » Select all children of specific start point
  - » Select all siblings (and their children of a specific start point

Data Naming Issues

- Does choice of data naming actually impact the protocol?
  - » Using a child element as the instance identifier impacts subset retrieval, since the search results desired are all sibling sub-trees of the instance ID element.
  - » Does the parent element need to be returned as well to provide a container (and semantic context)?