NETCONF Interoperability Testing

Ha Manh Tran Iyad Tumar Jürgen Schönwälder

Jacobs University Bremen

July 1, 2009

NETCONF Interoperability Testing

- Describe a NETCONF interoperability testing plan that is used to test whether NETCONF protocol implementations meet the NETCONF protocol specification in RFC 4741.

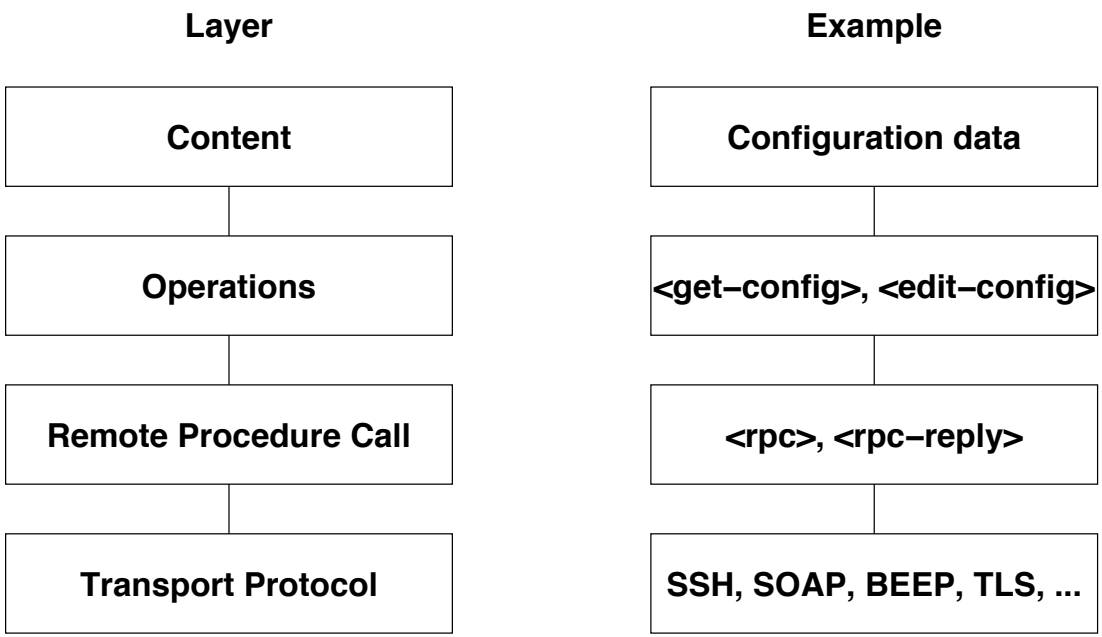
- The test plan particularly focuses on testing the correctness of NETCONF messages and operations (not to measure the performance of NETCONF implementations).

Outline

Outline

NETCONF Overview

- The NETCONF protocol provides mechanisms to install, manipulate, and delete the configuration of network devices.
- It uses an Extensible Markup Language based data encoding on top of a simple Remote Procedure Call layer.



NETCONF protocol layers

NETCONF Overview

NETCONF protocol operations (arguments in brackets are optional)

Operation	**Arguments**
get-config	source [filter]
edit-config	target [default-operation] [test-option] [error-option] config
copy-config	target source
delete-config	target
lock	target
unlock	target
get	[filter]
close-session	
kill-session	session-id
discard-changes	
validate	source
commit	[confirmed confirm-timeout]
create-subscription	[stream] [filter] [start] [stop]

Outline

Systems Under Test

- Cisco 1802 integrated services routers.
- Juniper J6300 Routers.
- Tail-f ConfD software for configuration management.
- EnSuite software for configuration management.

Systems Under Test

System	Platform	SSH Support
Juniper	JUNOS ver. 9.0	ver. 1.5/2.0
Tail-f	ConfD ver. 2.5.2	ver. 2.0
Cisco	IOS ver. 12.4	ver. 2.0
EnSuite	YencaP ver. 2.1.11	ver. 2.0

Systems Under Test

NETCONF capabilities supported by the systems under test

Capability	Juniper	Tail-f	Cisco	EnSuite
:base	√	√	√	√
:writable-running		√	√	√
:candidate	√	√		√
:confirmed-commit	√	√		
:rollback-on-error		√		
:validate	√	√		
:startup			√	√
:url	√	√	√	√
:xpath		√		√

Outline

Test Plan

- We divided our test plan into five test suites.

Test suites and current number of test cases

Test Suite	No. Test Cases
GENERAL	19
GET	11
GET-CONFIG	16
EDIT-CONFIG	15
VACM	26

- A test suite is a collection of test cases that are intended to be used to test and verify whether the systems under test meet the NETCONF protocol specification contained in RFC 4741 and RFC 4742.

- The total number of test cases is 87.

Test Plan

- Each test case contains three parts:
  - Pre-configuration.
  - Main test.
  - Post-configuration.
- Organization of test cases into test suites is not directly following the vertical layering model.
- To reduce the overhead of the pre-configuration and post-configuration parts during the execution of the test suite on the systems under test.

Test Suites

1. GENERAL: It includes test cases for individual operations such as `lock`, `unlock`, `close-session`, `kill-session`, `discard-changes`, `validate`, and `commit`.
2. GET: This suite aims to test the filter mechanism of the `get` operations.
3. GET-CONFIG: This suite aims to test the filter mechanism of the `get-config` operations.
4. EDIT-CONFIG: Involves tests modifying the configuration data in the datastore.
5. VACM suite verifying the NETCONF protocol operations against the VACM data model.

Test Suites

- The `edit-config` operation test cases support the `create`, `replace`, `merge` and `delete` operation attributes.
- This suite includes test cases for:
  - `delete-config` operations.
  - `copy-config` operations.
  - `edit-config` operations.
- Several test cases in this suite are data model specific due to the lack of a common data model, thus we need to implement several tests in different ways.

Outline

Test Tool (NIT)

- We have implemented a tool called NIT (NETCONF Interoperability Testing tool) to automatically execute the test suites against a system under test.

- NIT tool basically performs the following operations:
  1. connecting to a system under test using the SSH.
  2. verifying the initial `hello` message.
  3. executing test cases by:
     - sending a test request and receiving a response.
     - verifying both the request and the response following the criteria defined by RFC 4741.
  4. reporting the failure or the success of each test.

Outline

Preliminary Observations

Test result summary organized by the systems under test

System	Success	Failure	Irrelevant
A	47.2%	14.9%	37.9%
B	82.8%	9.2%	8.0%
C	17.3%	10.3%	72.4%
D	17.3%	21.8%	60.9%

- Success column: Indicates the percentage of passed test cases.
- Failure column: Indicates the percentage of failed test cases.
- Irrelevant column: Indicates the percentage of test cases that cannot be applied to a specific system due to either system configuration or implementation issues.

Preliminary Observations

System	Success	Failure	Irrelevant
A	47.2%	14.9%	37.9%
B	82.8%	9.2%	8.0%
C	17.3%	10.3%	72.4%
D	17.3%	21.8%	60.9%

- Systems A and B comply reasonably well with the RFCs.
- System A fails 14.9% (format of request and response messages, filter mechanism of the `get` operation).
- Systems A and B have very few problems with the filter mechanism of the `get-config` and `edit-config` operations.
- Systems C and D perform poorer (format of requests and responses, filter mechanism of the `get` operation).

Preliminary Observations

Test result summary organized by the test suites

Test Suite	Success	Failure	Irrelevant
GENERAL	73.6%	13.2%	13.2%
GET	29.5%	52.3%	18.2%
GET-CONFIG	48.4%	14.1%	37.5%
EDIT-CONFIG	38.3%	1.7%	60%
VACM	19.2%	5.8%	75%

- We did manually re-check the failed test cases in order to erase bugs in the test scripts.
- Several test cases reflect our interpretation of RFC 4741 and there might not be full agreement with our interpretation and thus the numeric results in the previous tables should be taken with a grain of salt.

Observations about RFC4741

- There are several things where the RFC is either somewhat ambiguous or totally silent.
- The RFC should provide more detailed descriptions for error situations.
- It might be necessary to better constrain the currently open ended format of request and response messages since they for example allow arbitrary values for attributes.
- The RFC should be updated with clearer examples.

Observations about RFC4741

- Some particular issues are listed below:
  - The RFC ignores the XML declaration for requests and responses.

```
<?xml version$="1.0" encoding="UTF-8"?>
```

  - The examples in RFC 4741 often omit namespace declarations for request and response messages. It would help interoperability if the examples would contain namespace declarations where necessary.

  - RFC 4741 allows arbitrary strings for the `message-id` attribute. These arbitrary strings terminate the session often without an error indication or return strange results.

Outline

Conclusions

- This work aims at observing the compliance of NETCONF implementations with RFC 4741.
- We have used the NIT tool to test four different NETCONF implementations.
- Our preliminary observations indicate that the number of failed test cases is relatively high for some systems.
- We have also noted some inconsistencies in RFC 4741.

Future Work

- This work still requires several improvements:
  - Increased the number of test cases.
  - It would be nice to reduce the dependency of the test cases on different data models.
  - The NIT tool should be improved to better support more complicated test cases that involve multiple NETCONF sessions.
  - It would be valuable to repeat the tests with a larger number of different NETCONF implementations.

THANK YOU

Questions?