Network Configuration Protocol (NETCONF)
Proxy

draft-wangzheng-netconf-proxy-00

Zitao Wang
Guangying Zheng
Motivation & Scenarios

• This document proposes a NETCONF Proxy mechanism.
  – The proxy can acts as an intermediary between manager and target device.
  – The client can set up a NETCONF session to a target through a NETCONF Proxy.
  – The mechanism allows the client to subsequently direct NETCONF requests to the server, to receive responses, and to subscribe to notifications from the server.
• It would be useful when a client does not have direct network access to a target host.
Motivation & Scenarios: VNF Scenario

- EMS is connected to Netconf Proxy through public network;
- Netconf Proxy is connected to all VNFs through private network;
- Netconf Proxy act as a pass-through for VNFs;
- EMS perform configuration of all VNFs through Netconf Proxy.
- Support for aggregated managed VNFs,
- Can reduce the management resources (like IP resources, bandwidth, etc) of large-scale management activities.
Motivation & Scenarios: Cloud Center Scenarios

- Netconf Proxy located in the gateway network element (GNE);
- Cloud Center EMS is connected to Netconf Proxy through Netconf over SSH;
- The transport protocol between GNE and N-GNEs is some private protocol;
- The N-GNEs are not IP devices, therefore, the EMS cannot be aware of the address of N-GNEs,
- and the N-GNEs are not supported NAT.
1. Sending a `<hello>` message each other, capability exchange.

```
<hello>
</hello>
```

2. The client sends a `<get>` RPC to proxy to retrieve the "target-list" of the proxy.

```
<get>
  <target-list>
</get>
```

3. The proxy responds with a `<get-reply>` RPC containing "target-list" attributes.

```
<reply target-list:
  targetID=A1,
  protocol,
  Authen...>
```

Example for target-list
```
<target-list>
  <target-id>A1</target-id>
  <protocol>foo</protocol>
  <authentication>foo</authentication>
</target-list>
```

4. The client constructs a `<hello>` message according to the received "target-list".

```
<hello targetID="A1">
```

This `<hello>` message SHOULD contain at least a "target-id" attribute.

5. The proxy receives the `<hello>` message and checks the value of "target-id" attribute.

```
authenticate & connect
```

If the target can be found, then the proxy initiates a connection to corresponding target.

```
<hello targetID="A1">
```

And then proxy forwards the `<hello>` message, which received from client, to corresponding target.

6. The target receives the `<hello>` message and then responds a `<hello>` message containing a list of capabilities.

```
<rpc messageID="101"
  target="A1"
  xmlns="urn:xxxx">  
  <get-config>
```

7. Subsequently, the client can direct NETCONF requests to the target.
Next Steps

• The authors appreciate thoughts, feedback, and text on the content of the documents.
• And then prepare another version.