

NETCONF Extension for Handling big <rpc-reply> *(draft-liu-netconf-fragmentation-01)*

Bing Liu (Ed.) , Guangying Zheng

IETF91 @Honolulu, Nov 2014

Reminder

- Problem: response data might be very big when
 - the NMS is retrieving a large data store (e.g. routes in a core router)
 - data synchronizing between the NETCONF Client and Server
 - etc.
- Discussion in last meeting
 - There was a consensus that this problem need to be solved
 - Will consider to add this problem to the charter (after finishing current charter)

Solution Candidates (1/3)

- NETCONF <get-block> Extension
 - The original solution proposed in the draft
 - a fragmentation mechanism in NETCONF layer
- Basic Approaches
 - <get-block> capability negotiation
 - NETCONF Server encapsulates the response data into multiple <rpc-reply> messages (fragments) . Each fragment is limited in a certain amount of size (e.g. 30K)
 - NETCONF server sends the first fragment and waits the <get-block> instruction from the NETCONF Client to send the next fragment

Solution Candidates (2/3)

- **Linked Replies** (proposed by Juergen Schoenwaelder)
 - to change or augment NETCONF at some point in time such that an `<rpc>` can lead to a sequence of `<rpc-reply>` with a suitable cancel mechanism.

Here is what a new client might do if it wants to use linked replies:

- `<rpc message-id="101" link-id="123" xmlns="..."> </rpc>`

The server can either simply send an `rpc-reply` or it starts sending linked replies, e.g.:

- `<rpc-reply message-id="101" next-message-id="102" link-id="123" xmlns="..."> </rpc-reply>`
- `<rpc-reply message-id="102" next-message-id="103" link-id="123" xmlns="..."> </rpc-reply>`
- `<rpc-reply message-id="103" link-id="123" xmlns="..."> </rpc-reply>`

Solution Candidates (3/3)

- Subtree Iteration (proposed by Andy Bierman)
 - An "iterator" approach allows a list resource to be retrieved in chunks. An RPC function could be added to do the iteration operation.

```
<rpc>
  <get-list>
    <target>/if:interfaces/if:interface</target>
  </get-list>
</rpc>
```

```
<rpc-reply>
  <data>
    <interfaces>
      <interface> .... first entry </interface>
      ...
      <interface> .... 25th entry </interface>
    </interfaces>
  </data>
</rpc-reply>
```

```
...
<rpc>
  <get-list>
    <target>/if:interfaces/if:interface</target>
    <start>25</start>
  </get-list>
</rpc>
```

```
<rpc-reply>
  <data>
    <interfaces>
      <interface> .... 26th entry </interface>
      ...
      <interface> .... 50th entry </interface>
    </interfaces>
  </data>
</rpc-reply>
```

Next Step

- Any other solution proposals?
- Analysis of solution candidates
 - Pros/Cons

Comment?
Thank you!

leo.liubing@huawei.com
zhengguangying@huawei.com