UDP based Publication Channel for Streaming Telemetry

draft-ietf-netconf-udp-pub-channel-03

Guangying Zheng
Tianran Zhou
Alexander Clemm
Overview

• UDP vs TCP
  – Data collector will suffer a lot of TCP connections from many line cards equipped on different devices.
  – As no connection state needs to be maintained, UDP encapsulation can be easily implemented by hardware which will further improve the performance.
  – Because of the lightweight UDP encapsulation, higher frequency and better transit performance can be achieved, which is important for streaming telemetry.

• A new UDP based Publication Channel
  – Facilitate the distributed data collection mechanism (e.g., directly push data from line cards of many devices to a collector)
  – Support multiple encoding (including Binary)
  – Adapting to SN/YANG-Push
  – Enable options for extensibility
Transport Mechanisms

One way transport from publisher to the collector, adapting to SN/YANG-Push scheme

a) Call Flow For Dynamic Subscription

b) Call Flow For Configured Subscription
Transport Overview

- **DTLS**: provide reusable security and authentication functions over UDP
- **Message Header**: some important information before de-serializing the notification.
  - Encoding method: GPB, CBOR, JSON, XML
  - Message generator ID
  - Sequence number
  - Fragmentation
  - Options for extensibility
- **Notification**: include a notification header, as defined in draft-ietf-netconf-notification-messages-03
  - Encoded with the content.
Data Format of the UPC Message Header

Version=0
Flag (1 byte)=bit 0 (reliability), bit 1 (fragmentation)
Encoding Type(4 bits)= 0(GPB), 1(CBOR), 2(JSON), 3(XML)
Length(2bytes)
Message-Generator-ID: is a 32-bit identifier of the process which created the notification message.
Message ID: is generated continuously by the message generator.
Next

• Augment the SN model to add the UPC configurations.
• Security considerations.
• Any other to consider or address?
Thank you