UDP-based Transport for Configured Subscriptions

draft-ietf-netconf-udp-notif-01

G. Zheng, Huawei
T. Zhou, Huawei
T. Graf, Swisscom
P. Francois, INSA-Lyon
P. Lucente, NTT

draft-ietf-netconf-udp-notif-01
Agenda

- Quick reminder
- -00 vs. -01, considering comments received on the ML
- Discuss

draft-ietf-netconf-udp-notif-01
Reminder

Objective

- Publication of **massive amounts** of networking device data
- **LC's to directly send out data,** need for low performance impact

Configured subscriptions

- To be used in conjunction with "Subscription to Distributed Notifications" draft-ietf-netconf-distributed-notif-01

draft-ietf-netconf-udp-notif-01
Diff 1/3 Terminology and Applicability

Terminology

- Fragmentation option ➔ Segmentation option (Transport)
- Generator-ID ➔ Observation-Domain-ID (consistency with netconf-distributed-notif-01)

Applicability section

- Reworked content to align with RFC8085 (UDP Usage Guidelines)
- Congestion control, Message size, Lack of reliability

draft-ietf-netconf-udp-notif-01
Diff 2/3 UDP Notif message Header (proposal)

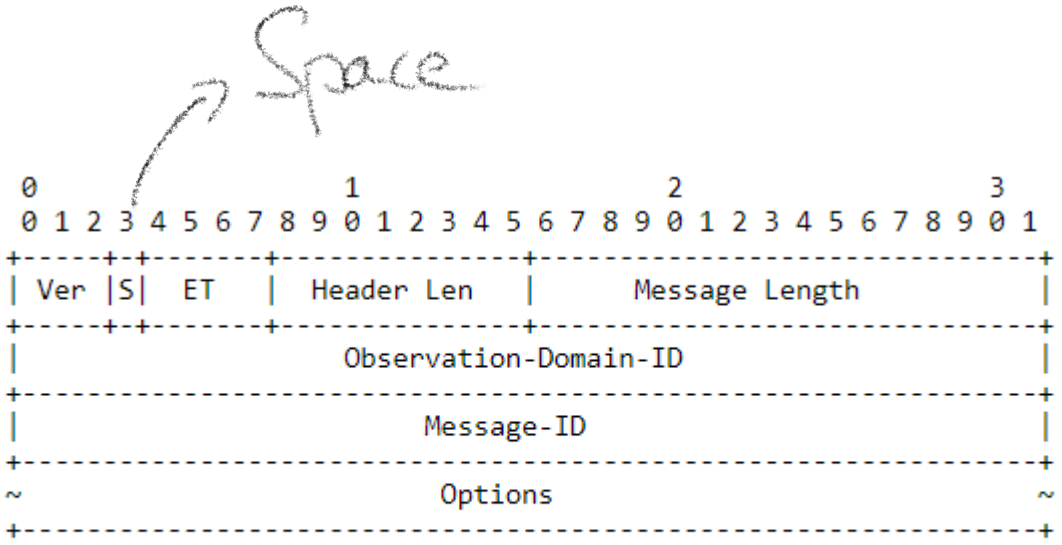

Figure 2: UDP-Notif Message Header Format

Comments Andy/Mahesh

> Obs domain and message ID Needed
>
> Segmentation + QoL

> Version set to 1 : OK.

> How to deal with private space: Discuss

Version: ~~4~~ 3 bits

S: Space flag, 1 bit

- 0: Standard ET space
  - 0 : CBOR
  - 1 : JSON
  - 2 : XML
- 1: Private ET space
  - E.g. GPB

draft-ietf-netconf-udp-notif-01
Diff 3/3 Segmentation Option Header

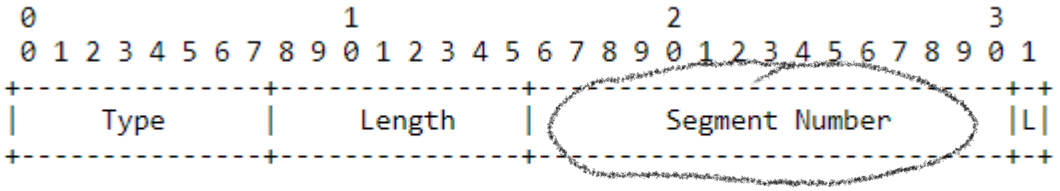

Figure 4: Segmentation Option Format

Segment number
16 bits (vs. 32)

*An implementation of this specification **MUST NOT** rely on IP fragmentation by default to carry large messages. An implementation of this specification MUST either restrict the size of individual messages carried over this protocol or support the segmentation option.*

TODO: Simplify: SHOULD NOT do fragmentation
TODO: SeqNum vs. L bit relationship (Andy/Mahesh)

draft-ietf-netconf-udp-notif-01
Discuss

Implementation by Huawei tested within Swisscom labs

Implementation status

- Producer and collector implementation projects now follow -01
  Scapy based collector validation
  VRP → golang collector → Kafka producer following -01 in the lab
  VRP → C collector : Segmentation support and integration with pmacct remaining

Next steps

- Draft DTLS
- Proposed changes based on comments
- More?

Thanks !