

YANG Notifications

Scott Mansfield (Ericsson)

yangsters meeting

Introduction

- RFC 7950: The YANG 1.1 Data Modeling Language
 - Notification Definition: <https://datatracker.ietf.org/doc/html/rfc7950#section-4.2.10>
- RFC 8639: Subscription to YANG Notifications (the new way)
 - <https://datatracker.ietf.org/doc/html/rfc8639>
 - Along with [RFC 8640](#) and [RFC 8641](#)
- RFC 5277: NETCONF Event Notifications (the “old” way)
 - <https://datatracker.ietf.org/doc/html/rfc5277>

What a notification looks like in YANG

- ieee802-dot1q-cfm-alarm.yang has a notification
- Done as an augment so if you don't want to support notifications, don't use the yang file.

```
/* -----  
 * Augmentations of objects defined in generic CFM YANG module  
 * (ieee802-dot1q-cfm) to define CFM alarms.  
 * -----  
 */  
augment "/dot1q-cfm:cfm/dot1q-cfm:maintenance-group/dot1q-cfm:mep" {  
  when  
    "dot1q-cfm:mep-id";  
  description  
    "Augment the MEP object defined in the ieee802-dot1q-cfm (CFM MEP)  
    YANG module with CFM alarm specific attributes.";  
  notification mep-fault-alarm {  
    description  
      "To alert the Manager to the existence of a fault in a monitored MA  
      by issuing a Fault Alarm.";  
    reference  
      "12.14.7.7.2c of IEEE Std 802.1Q-2021";  
    leaf mep-priority-defect {  
      type leafref {  
        path  
          '..'+  
          '/../'+  
          '/dot1q-cfm:continuity-check'+  
          '/dot1q-cfm:highest-priority-defect';  
      }  
    }  
    config false;  
    mandatory true;  
    description  
      "The highest priority defect that has been present since the MEPs  
      Fault Notification Generator state machine was last in the  
      FNG_RESET state.";  
  }  
}
```

Why use notifications?

- Setup automatic notification when something changes
- No need to “poll” for a value
- Lots of information on requirements and rationale here:
 - [RFC 7923](#): Requirements for Subscription to YANG Datastores

Don's Question/Advice...

We have a concept of max-peers when using group addresses. We want to inform the user if there are a lot of members in a group because we support fragmentation and each destination for each source that does fragmentation requires two reassembly buffers. Mick had suggested a configured leaf that sets max-peers and a leaf that shows if max-peers is exceeded. See diagram.

I coded the YANG for the whole model but for max-peers-exceeded I said I think that a notification object should be used versus just an operational object.

We have a max-peers and a leaf-list.

```
augment /if:interfaces/if:interface:
```

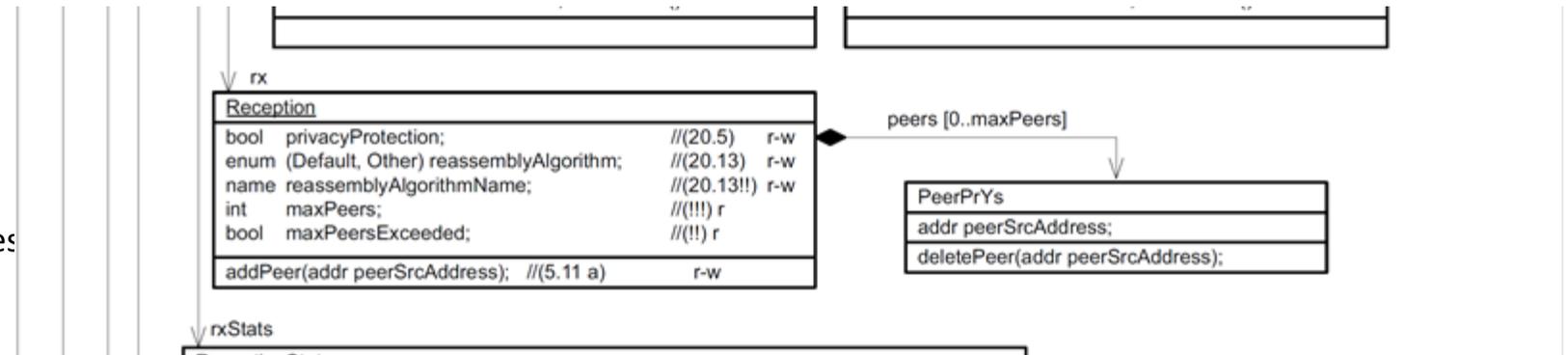
```
  +--rw pry {macsec-priv}?
```

```
    +--rw reception
```

```
      | +--rw max-peers?      uint8
```

```
      | +--rw peer-src-address* ieee:mac-address
```

```
      | +--rw fragment-ordering? identityref
```



So that is the simple question.

The harder question we have not resolved is it is not really the max peers that is an issue but the max peers that are doing fragmentation. We assume fragmentation by default. But we would like to indicate when the group is large. (It is rather an unlikely case because MAC privacy would not be too efficient for most tasks in a large multicast group.) And practically we have another issue in that each source is typically sending continuous stream of privacy frames that are of fixed size so our group cannot be too large or we will have hardly any bandwidth per source. There are cases though where you have a group of listeners that have low bandwidth to the source.

So the simple question is how would we code Max-peers-exceeded in YANG (But we have to figure out how practical the situation is.)