

CB-2017-Rev All-0 MAC and IP address Parameters

For Further Discussion

Comments Towards CB-Rev D0.1

9.1.5.1 tsnCpelpIdDestMac

Specifies the destination_address parameter that identifies a packet in an EISS indication primitive.
An address of all 0 indicates that the MAC destination address is to be ignored on packets received from lower layers.

Comment:

In section "9.1.5.1 tsnCpelpIdDestMac" there is now a wildcard for the MAC destination address available with all 0s - Is that conformant with the YANG type mac-address in <https://github.com/YangModels/yang/blob/main/standard/ieee/published/802/ieee802-types.yang>, as used in 12.6.1.2, as it references IEEE Std 802?

9.1.5.4 tsnCpelpIdIpSource

Specifies the IPv4 (RFC 791) or IPv6 (RFC 2460) source address parameter that must be matched to identify packets coming up from lower layers. An address of all 0 indicates that the IP source address is to be ignored on packets received from lower layers.

Comment:

I think the type inet:ip-address from <https://github.com/YangModels/yang/blob/main/standard/ietf/RFC/ietf-inet-types%402013-07-15.yang> as used in 12.6.2.2 (page 196) does not allow the all 0's as a "wildcard", as it references RFC791.

Proposed Remedy

Please change the data-type.

If we had strictly stuck to the comments and their proposed remedies, I think the dispositions should have been “reject”, since the types used in the YANG models (and hopefully the MIB definitions) allow all-0 values for both address types (I thought ipv4’s didn’t but I was obviously wrong).

- destination-mac in Cl. 12.6.2.2 is of type ieee:mac-address

```
typedef mac-address {  
    type string {  
        pattern "[0-9a-fA-F]{2}(-[0-9a-fA-F]{2}){5}";  
    }  
}
```

...

- ip-source in Cl. 12.6.2.2 is of type inet:ip-address

```
typedef ip-address {  
    type union {  
        type inet:ipv4-address;  
        type inet:ipv6-address;  
    }  
}
```

...

```
typedef ipv4-address {  
    type string {  
        pattern  
            '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'  
            + '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])'  
            + '(%[\p{N}\p{L}]+)?';  
    }  
}
```

...

[ipv6_address pattern not included]

Proposed Remedy - Continued

- Now, about the interpretation of the text of clauses 9.1.5.1 and 9.1.5.4, it just indicates that IP stream identification functions that include an all-0 tsnCpelpIdDestMac and/or an all-0 tsnCpelpIdIpSource parameter(s) use all of their other parameters to check for a match.
- ... meaning that it is not possible to individually identify streams, which packets include an all-0 destination MAC and/or IP source. Do we have to be able to explicitly identify these streams ?
- If yes, we'd need to define some kind of "don't care" for these parameters: this was the reason for the introduction of the "wildcard" in the dispositions implemented in Cl. 9.1.5.1 and 9.1.5.4 of CB-2017-Rev D0.2 (and related YANG leaves in cl. 12.6.2.2).
- A question: is it possible to have such parameter definitions and their translation into YANG (and SMI) ?
- e.g., for the MAC address: a union of mac-address and a pattern like "***_**_**_**_**_**_" ? (not sure about that ...)