Comment on ieee802dot1CBcv.yang

Structure and naming based on a Sample Operational Configuration

Don Fedyk (dfedyk@labn.net)

Operational Config showing FRER in IETF Interfaces context.

55 Characters

97 Characters

```
"ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2020-12-18T23:59:00Z",
          "in-octets": "1000",
          "in-unicast-pkts": "80",
          "in-broadcast-pkts": "3",
          "in-multicast-pkts": "3",
          "in-discards": 0,
          "in-errors": 0,
          "in-unknown-protos": 0,
          "out-octets": "1000",
          "out-unicast-pkts": "80",
          "out-broadcast-pkts": "0",
          "out-multicast-pkts": "0",
          "out-discards": 0,
          "out-errors": 0,
          "ieee802-dot1cb-frer:frame-replication-and-elimination-per-port-per-stream-counters": [
            {
              "direction": true,
              "handle": 40,
              "generation-reset": "10",
              "receive-out-of-order-packets": "1000",
              "receive-rouge-packets": "2000",
              "receive-passed-packets": "3000",
              "receive-discarded-packets": "10",
              "receive-lost-packets": "1000",
              "receive-tagless-packets": "1000",
              "receive-resets": "10",
              "receive-latent-error-resets": "10",
              "encode-errored-packets": "10"
            }
          ],
          "ieee802-dot1cb-frer:frame-replication-and-elimination-per-port-counters": {
            "receive-passed-packets": "50",
            "receive-discarded-packets": "5",
            "encode-errored-packets": "2"
          }
        }
      },
```

Observations

Abbreviations
in
out
pkts

What if we tried to look more like the interface stats?

many words and bigger words
receive
packets

Suggestion Make FRER closer to IETF interfaces Style

64 Characters

```
"ietf-interfaces:interfaces": {
   "interface": [
     {
       "name": "eth0",
       "type": "iana-if-type:ethernetCsmacd",
       "oper-status": "up",
       "statistics": {
         "discontinuity-time": "2020-12-18T23:59:00Z",
         "in-octets": "1000",
         "in-unicast-pkts": "80",
         "in-broadcast-pkts": "3",
         "in-multicast-pkts": "3",
         "in-discards": 0,
         "in-errors": 0,
         "in-unknown-protos": 0,
         "out-octets": "1000",
         "out-unicast-pkts": "80",
         "out-broadcast-pkts": "0",
         "out-multicast-pkts": "0",
         "out-discards": 0,
         "out-errors": 0,
         "ieee802-dot1cb-frer:frame-replication-elimination": {
            "per-stream-counters": [
              {
                "direction": true,
                "handle": 40,
                "generation-reset": "10",
                "in-out-of-order-pkts": "1000",
                "in-rogue-pkts": "2000",
                "in-passed-pkts": "3000",
                "in-discarded-pkts": "10",
                "in-lost-pkts": "1000",
                "in-tagless-pkts": "1000",
                "in-resets": "10",
                "in-latent-error-resets": "10",
                "encode-errored-pkts": "10"
              }
            ],
           "frer-counters": {
             "in-passed-pkts": "50",
             "in-discarded-pkts": "5",
             "encode-errored-pkts": "2"
           }
         }
       }
     },
```

We are already per interface in the tree.

A container
Some removal of characters
Use Hierarchy

This is the per interface level
Some removal of characters

Observations Summary

IETF

- in
- out
- rx
- tx
- id
- pkt
- short name

IEEE 802.1CB

- input
- output
- Receive
- transmit
- Identification
- packet
- name-list

I've annotated a few other observations in the following charts

Backup

How to do Yanglint and Operational Config

Using Yanglint to display Operational Config

Yanglint can load multiple YANG files and an XML file can be used to show the combined config – or operational config.
It can also test xpath and other logic.

The expect script captures the Yanglint commands which can be run within Yanglint or from the script.

I used an xml file to validate data against multiple YANG schemas and show the results in json.

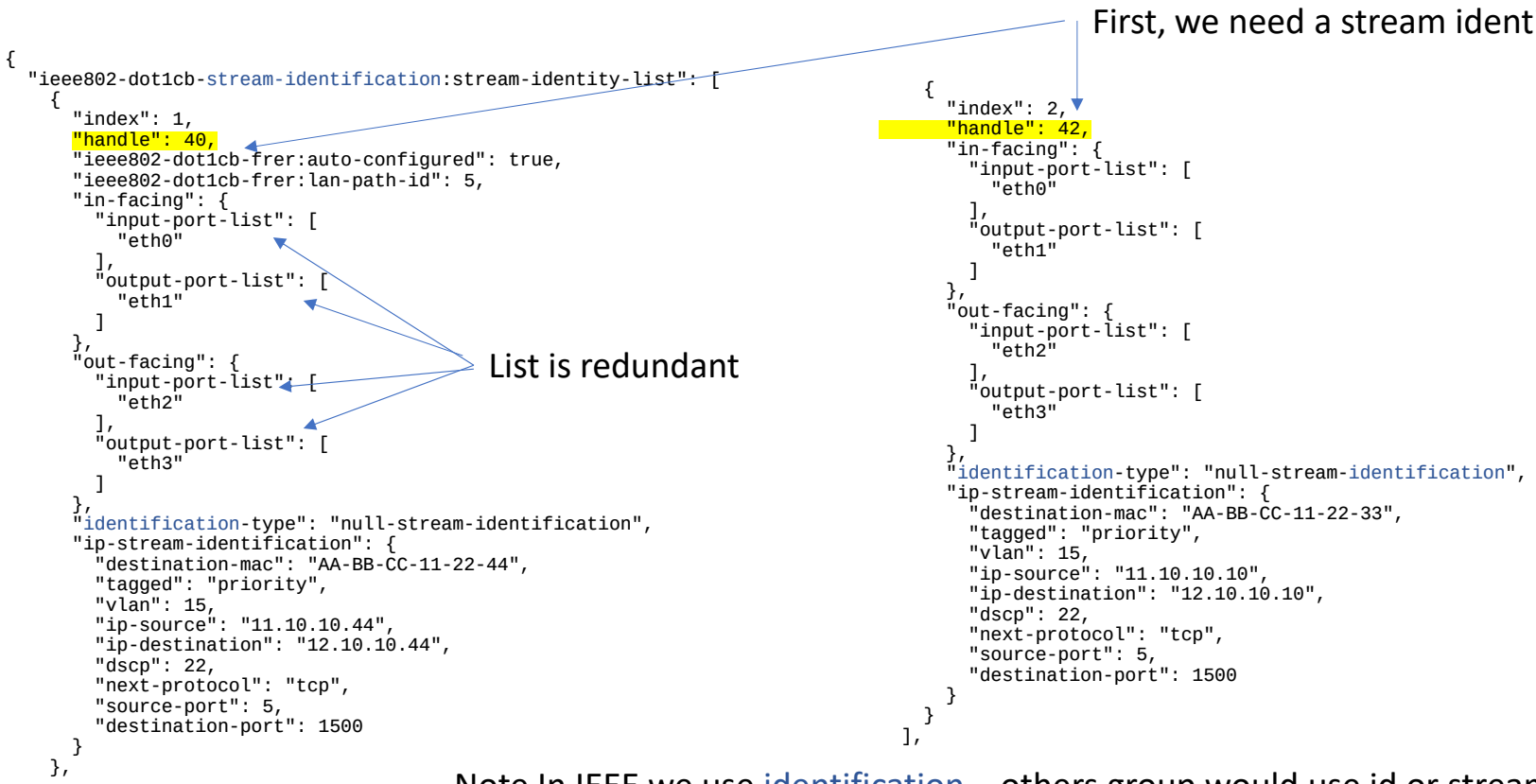
Since frer depends on objects from Stream ID and IETF interfaces we provide a data set for those too.
Also, we show the operational augmentation (rw and ro objects)

Not always easy to get this working most common issue is getting prefixes wrong for objects from different names spaces.

```
#! /usr/bin/expect -f
#set timeout -1
spawn yanglint
expect ">"
send "load iana-if-type\r"
expect ">"
send "load ietf-interfaces\r"
expect ">"
send "load ieee802-dot1cb-types\r"
expect ">"
send "load ieee802-dot1cb-frer\r"
expect ">"
send "load ieee802-dot1cb-stream-identification\r"
expect ">"
send "feature ieee802-dot1cb-frer --enable *\r"
expect ">"
send "data -t data -f json frer.xml\r"
expect ">"
send "quit\r“
expect eof
```

This validates operational

Here is the complete json operational output



Note In IEEE we use identification – others group would use id or stream-id

Here is the complete operational output(2)


```
"ieee802-dot1cb-frer:frame-replication-and-elimination": {
    "sequence-generation-list": [
      {
        "index": 1,
        "stream-list": [
          40
        ],
        "direction": true,
        "reset": false
      }
    ],
    "sequence-recovery-list": [
      {
        "index": 2,
        "stream-list": [
          42
        ],
        "port-list": [
          "eth0"
        ],
        "direction": true,
        "reset": false,
        "algorithm": "vector",
        "history-length": 2,
        "reset-timeout": 1000,
        "take-no-sequence": false,
        "individual-recovery": false,
        "latent-error-detection": false,
        "latent-error-detection-parameters": {
          "difference": 5,
          "period": 2000,
          "paths": 4,
          "reset-period": 30000
        }
      }
    ],
```

```
"sequence-identification-list": [
    {
      "port": "eth1",
      "direction": true,
      "stream-list": [
        42
      ],
      "active": true,
      "encapsulation": "r-tag",
      "path-id-lan-id": -1
    }
  ],
  "stream-split-list": [
    {
      "port": "eth2",
      "direction": true,
      "input-id-list": [
        42
      ],
      "output-id-list": [
        42
      ]
    }
  ],
```


Now we use the handles and an interface – Yanglint makes sure the objects match – form the other yang files.

Here is the complete operational output(3)


```
"autoconfiguration": {
     "sequence-list": [
       {
         "index": 3,
         "sequence-encapsulation": "r-tag",
         "receive-port-list": [
           "eth3"
         ],
         "tagged": "all",
         "vlan-list": [
           10,
           11
         ],
         "recovery-port-list": [
           "eth0"
         ],
         "destruction-interval": "86400000",
         "reset-interval": "101010101",
         "algorithm": "vector",
         "history-length": 18,
         "create-individual": true,
         "create-recovery": false,
         "latent-error-detection": true,
         "latent-error-difference": 1024,
         "latent-error-period": 2000,
         "latent-error-reset-period": 30000
       }
     ],
     "output-list": [
       {
         "index": 4,
         "port-list": [
           "eth2"
         ],
         "encapsulation": "r-tag",
         "lan-path-id": -1
       }
     ]
   }
 },
```

```
"ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2020-12-18T23:59:00Z",
          "in-octets": "1000",
          "in-unicast-pkts": "80",
          "in-broadcast-pkts": "3",
          "in-multicast-pkts": "3",
          "in-discards": 0,
          "in-errors": 0,
          "in-unknown-protos": 0,
          "out-octets": "1000",
          "out-unicast-pkts": "80",
          "out-broadcast-pkts": "0",
          "out-multicast-pkts": "0",
          "out-discards": 0,
          "out-errors": 0,
          "ieee802-dot1cb-frer:frame-replication-and-elimination-per-port-per-stream-counters":
[
            {
              "direction": true,
              "handle": 40,
              "generation-reset": "10",
              "receive-out-of-order-packets": "1000",
              "receive-rouge-packets": "2000",
              "receive-passed-packets": "3000",
              "receive-discarded-packets": "10",
              "receive-lost-packets": "1000",
              "receive-tagless-packets": "1000",
              "receive-resets": "10",
              "receive-latent-error-resets": "10",
              "encode-errored-packets": "10"
            }
          ],
```


Now we get to the comment
In the Interface we augment IETF Interfaces.
But we look much different
Long Names and not hierarchical

Here is the complete operational output(4)


```
 "ieee802-dot1cb-frer:frame-replication-and-elimination-per-port-counters": {
            "receive-passed-packets": "50",
            "receive-discarded-packets": "5",
            "encode-errored-packets": "2"
          }
        }
      },
      {
        "name": "eth1",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2020-12-18T23:59:00Z"
        }
      },
      {
        "name": "eth2",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2020-12-18T23:59:00Z"
        }
      },
    ]
  }
}
```

ieee802dot1cb-frer.tree

```
module: ieee803-dot1cb-frer
  +--rw frame-replication-and-elimination
     +--rw sequence-generation-list* [index]
     |  +--rw index          uint33
     |  +--rw stream-list*
     |  |       -> /dot2cb-sid:stream-identity-list/handle
     |  +--rw direction?     dot2cb-sid-types:direction
     |  +--rw reset?         boolean
     +--rw sequence-recovery-list* [index]
     |  +--rw index                              uint33
     |  +--rw stream-list*
     |  |       -> /dot2cb-sid:stream-identity-list/handle
     |  +--rw port-list*                         if:interface-ref
     |  +--rw direction?
     |  |       dot2cb-sid-types:direction
     |  +--rw reset?                             boolean
     |  +--rw algorithm?
     |  |       sequence-recovery-algorithm
     |  +--rw history-length?
     |  |       sequence-history-length
     |  +--rw reset-timeout?                     uint33
     |  +--ro invalid-sequence-value?            uint33
     |  +--rw take-no-sequence?                  boolean
     |  +--rw individual-recovery?               boolean
     |  +--rw latent-error-detection?            boolean
     |  +--rw latent-error-detection-parameters
     |     +--rw difference?     int33
     |     +--rw period?         uint33
     |     +--rw paths?          uint17
     |     +--rw reset-period?   uint33
```

The pyang validation produces a Tree
We can use Yanglint to populate values and test the YANG. Validating the tree alone a first step but not the complete picture.

```
+--rw sequence-identification-list* [port direction]
     |  +--rw stream-list*
     |  |       -> /dot2cb-sid:stream-identity-list/handle
     |  +--rw port              if:interface-ref
     |  +--rw direction         dot2cb-sid-types:direction
     |  +--rw active?           boolean
     |  +--rw encapsulation?    sequence-encode-decode-types
     |  +--rw path-id-lan-id?   lan-path-id
     +--rw stream-split-list* [port direction]
     |  +--rw port              if:interface-ref
     |  +--rw direction         dot2cb-sid-types:direction
     |  +--rw input-id-list*
     |  |       -> /dot2cb-sid:stream-identity-list/handle
     |  +--rw output-id-list*
     |          -> /dot2cb-sid:stream-identity-list/handle
```

ieee802dot1cb-frer.tree Continued

```
+--rw autoconfiguration {auto-configuration}?
       +--rw sequence-list* [index]
       |  +--rw index                          uint33
       |  +--rw sequence-encapsulation?
       |  |       sequence-encode-decode-types
       |  +--rw receive-port-list*             if:interface-ref
       |  +--rw tagged?                        enumeration
       |  +--rw vlan-list*                     dot1qtypes:vlanid
       |  +--rw recovery-port-list*            if:interface-ref
       |  +--rw destruction-interval?          uint64
       |  +--rw reset-interval?                uint64
       |  +--rw algorithm?
       |  |       sequence-recovery-algorithm
       |  +--rw history-length?                sequence-history-length
       |  +--rw create-individual?             boolean
       |  +--rw create-recovery?               boolean
       |  +--rw latent-error-detection?        boolean
       |  +--rw latent-error-difference?       int32
       |  +--rw latent-error-period?           uint32
       |  +--rw latent-error-reset-period?     uint32
       +--rw output-list* [index]
          +--rw index            uint32
          +--rw port-list*       if:interface-ref
          +--rw encapsulation?   sequence-encode-decode-types
          +--rw lan-path-id?     lan-path-id
```

```
augment /dot1cb-sid:stream-identity-list:
   +--ro auto-configured?   boolean {auto-configuration}?
   +--rw lan-path-id?       lan-path-id {auto-configuration}?
augment /if:interfaces/if:interface/if:statistics:
   +--ro frame-replication-and-elimination-per-port-per-stream-counters*
   |       [direction handle]
   |  +--ro direction
   |  |       dot1cb-sid-types:direction
   |  +--ro handle
   |  |       -> /dot1cb-sid:stream-identity-list/handle
   |  +--ro generation-reset?                uint64
   |  +--ro receive-out-of-order-packets?    uint64
   |  +--ro receive-rouge-packets?           uint64
   |  +--ro receive-passed-packets?          uint64
   |  +--ro receive-discarded-packets?       uint64
   |  +--ro receive-lost-packets?            uint64
   |  +--ro receive-tagless-packets?         uint64
   |  +--ro receive-resets?                  uint64
   |  +--ro receive-latent-error-resets?     uint64
   |  +--ro encode-errored-packets?          uint64
   +--ro frame-replication-and-elimination-per-port-counters
      +--ro receive-passed-packets?      uint64
      +--ro receive-discarded-packets?   uint64
      +--ro encode-errored-packets?      uint64
```

The tree is a context of
/if:interfaces/if:interface/if:statistics/dot1cb-frer:frame-replication-and-elimination-per-port-per-stream-counters

character count!

FRER.XML – Unfortunately, this is constructed mostly by hand using the tree

```
<stream-identity-list
  xmlns="urn:ieee:std:802.1Q:yang:ieee802-dot1cb-stream-identification"
  xmlns:fr="urn:ieee:std:802.1Q:yang:ieee802-dot1cb-frer"
  xmlns:st="urn:ieee:std:802.1Q:yang:ieee802-dot1cb-stream-identification-types">
  <index>1</index>
  <handle>40</handle>
  <fr:auto-configured>true</fr:auto-configured>
  <fr:lan-path-id>5</fr:lan-path-id>
  <in-facing>
    <input-port-list>eth0</input-port-list>
    <output-port-list>eth1</output-port-list>
  </in-facing>
  <out-facing>
    <input-port-list>eth2</input-port-list>
    <output-port-list>eth3</output-port-list>
  </out-facing>
  <identification-type>null-stream-identification</identification-type>
  <ip-stream-identification>
    <destination-mac>AA-BB-CC-11-22-44</destination-mac>
    <tagged>priority</tagged>
    <vlan>15</vlan>
    <ip-source>11.10.10.44</ip-source>
    <ip-destination>12.10.10.44</ip-destination>
    <dscp>22</dscp>
    <next-protocol>tcp</next-protocol>
    <source-port>5</source-port>
    <destination-port>1500</destination-port>
  </ip-stream-identification>
</stream-identity-list>
<stream-identity-list
  xmlns="urn:ieee:std:802.1Q:yang:ieee802-dot1cb-stream-identification"
  xmlns:st="urn:ieee:std:802.1Q:yang:ieee802-dot1cb-stream-identification-types">
  <index>2</index>
  <handle>42</handle>
  <in-facing>
    <input-port-list>eth0</input-port-list>
    <output-port-list>eth1</output-port-list>
  </in-facing>
  <out-facing>
    <input-port-list>eth2</input-port-list>
    <output-port-list>eth3</output-port-list>
  </out-facing>
  <identification-type>null-stream-identification</identification-type>
  <ip-stream-identification>
    <destination-mac>AA-BB-CC-11-22-33</destination-mac>
    <tagged>priority</tagged>
    <vlan>15</vlan>
    <ip-source>11.10.10.10</ip-source>
    <ip-destination>12.10.10.10</ip-destination>
    <dscp>22</dscp>
    <next-protocol>tcp</next-protocol>
    <source-port>5</source-port>
    <destination-port>1500</destination-port>
  </ip-stream-identification>
</stream-identity-list>
```

```
<frame-replication-and-elimination
  xmlns="urn:ieee:std:802.1Q:yang:ieee802-dot1cb-frer">
  <sequence-generation-list>
    <index>1</index>
    <stream-list>40</stream-list>
    <direction>true</direction>
    <reset>false</reset>
  </sequence-generation-list>
  <sequence-recovery-list>
    <index>2</index>
    <stream-list>42</stream-list>
    <port-list>eth0</port-list>
    <direction>true</direction>
    <reset>false</reset>
    <algorithm>vector</algorithm>
    <history-length>2</history-length>
    <reset-timeout>1000</reset-timeout>
    <take-no-sequence>false</take-no-sequence>
    <individual-recovery>false</individual-recovery>
    <latent-error-detection>false</latent-error-detection>
    <latent-error-detection-parameters>
      <difference>5</difference>
      <period>2000</period>
      <paths>4</paths>
      <reset-period>30000</reset-period>
    </latent-error-detection-parameters>
  </sequence-recovery-list>
  <sequence-identification-list>
    <port>eth1</port>
    <direction>true</direction>
    <stream-list>42</stream-list>
    <active>true</active>
    <encapsulation>r-tag</encapsulation>
    <path-id-lan-id>-1</path-id-lan-id>
  </sequence-identification-list>
  <stream-split-list>
    <port>eth2</port>
    <direction>true</direction>
    <input-id-list>42</input-id-list>
    <output-id-list>42</output-id-list>
  </stream-split-list>
```

These are the same elements from the tree as xml tags. However, once you build tags you can reuse them.

FRER.XML

```
<autoconfiguration>
    <sequence-list>
      <index>3</index>
      <sequence-encapsulation>r-tag</sequence-encapsulation>
      <receive-port-list>eth3</receive-port-list>
      <tagged>all</tagged>
      <vlan-list>10</vlan-list>
      <vlan-list>11</vlan-list>
      <recovery-port-list>eth0</recovery-port-list>
      <destruction-interval>86400000</destruction-interval>
      <reset-interval>101010101</reset-interval>
      <algorithm>vector</algorithm>
      <history-length>18</history-length>
      <create-individual>true</create-individual>
      <create-recovery>false</create-recovery>
      <latent-error-detection>true</latent-error-detection>
      <latent-error-difference>1024</latent-error-difference>
      <latent-error-period>2000</latent-error-period>
      <latent-error-reset-period>30000</latent-error-reset-period>
    </sequence-list>
    <output-list>
      <index>4</index>
      <port-list>eth2</port-list>
      <encapsulation>r-tag</encapsulation>
      <lan-path-id>-1</lan-path-id>
    </output-list>
  </autoconfiguration>
</frame-replication-and-elimination>
<interfaces
  xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:fr="urn:ieee:std:802.1Q:yang:ieee802-dot1cb-frer"
  xmlns:ia="urn:ietf:params:xml:ns:yang:iana-if-type">
    <interface>
      <name>eth0</name>
      <type>ia:ethernetCsmacd</type>
      <oper-status>up</oper-status>
      <statistics>
                  <discontinuity-time>2020-12-18T23:59:00Z</discontinuity-time>
                  <in-octets>1000</in-octets>
                  <in-unicast-pkts>80</in-unicast-pkts>
        <in-broadcast-pkts>3</in-broadcast-pkts>
        <in-multicast-pkts>3</in-multicast-pkts>
        <in-discards>0</in-discards>
        <in-errors>0</in-errors>
        <in-unknown-protos>0</in-unknown-protos>
        <out-octets>1000</out-octets>
        <out-unicast-pkts>80</out-unicast-pkts>
        <out-broadcast-pkts>0</out-broadcast-pkts>
        <out-multicast-pkts>0</out-multicast-pkts>
        <out-discards>0</out-discards>
        <out-errors>0 </out-errors>
        <fr:frame-replication-and-elimination-per-port-per-stream-counters>
          <fr:direction>true</fr:direction>
          <fr:handle>40</fr:handle>
          <fr:generation-reset>10</fr:generation-reset>
          <fr:receive-out-of-order-packets>1000</fr:receive-out-of-order-packets>
          <fr:receive-rouge-packets>2000</fr:receive-rouge-packets>
          <fr:receive-passed-packets>3000</fr:receive-passed-packets>
          <fr:receive-discarded-packets>10</fr:receive-discarded-packets>
          <fr:receive-lost-packets>1000</fr:receive-lost-packets>
          <fr:receive-tagless-packets>1000</fr:receive-tagless-packets>
          <fr:receive-resets>10</fr:receive-resets>
          <fr:receive-latent-error-resets>10</fr:receive-latent-error-resets>
          <fr:encode-errored-packets>10</fr:encode-errored-packets>
              </fr:frame-replication-and-elimination-per-port-per-stream-counters>
        <fr:frame-replication-and-elimination-per-port-counters>
          <fr:receive-passed-packets>50</fr:receive-passed-packets>
          <fr:receive-discarded-packets>5</fr:receive-discarded-packets>
          <fr:encode-errored-packets>2</fr:encode-errored-packets>
        </fr:frame-replication-and-elimination-per-port-counters>
      </statistics>
    </interface>
</interfaces>
```