YANG based Config for MAC Privacy 802.1AEdk Granularity of Privacy Configuration

Don Fedyk (dfedyk@labn.net)

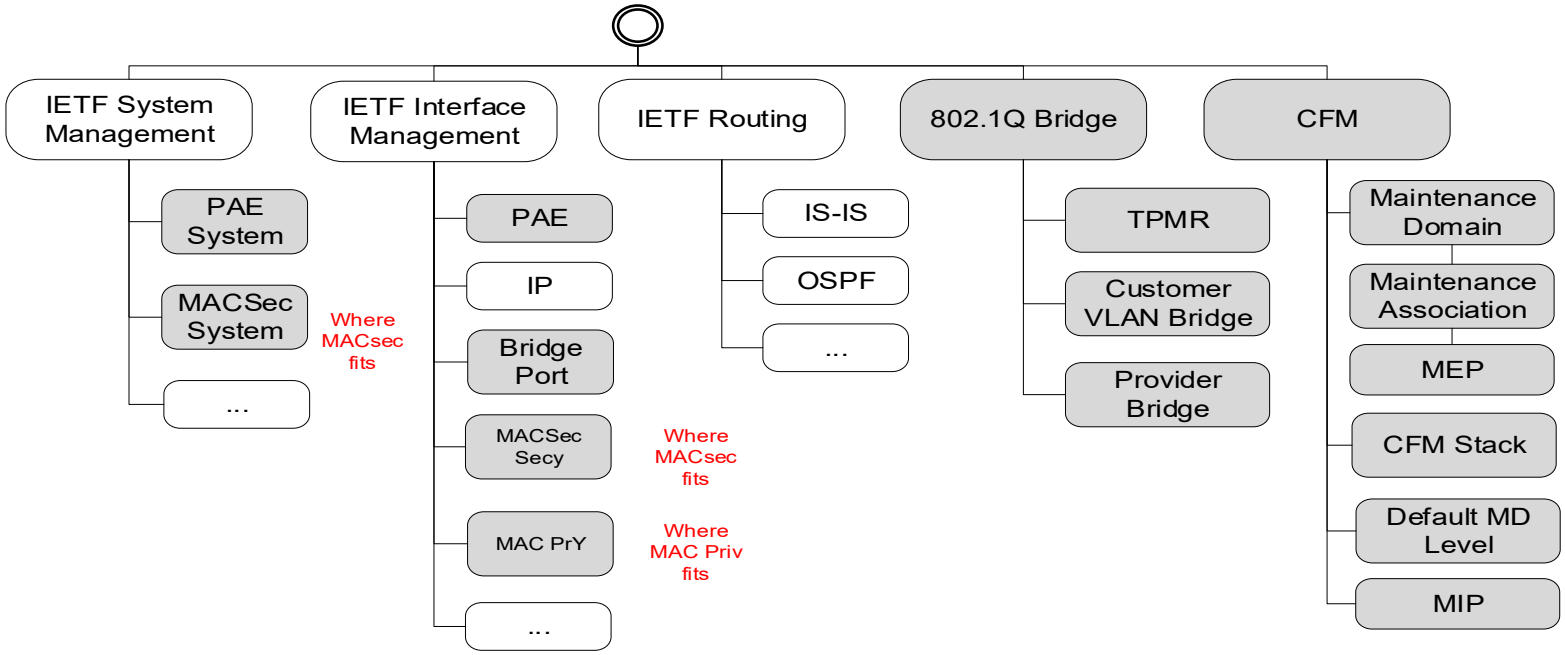
Outline

- Proto Config for MAC Privacy
- Considering Control of MAC privacy

Forward

- This presentation is for a discussion on detailed config.
- It may contain errors/omission and should be consider a work in progress.
- An updated version the presentation will be posted after discussion to correct it, but it will remain a work in progress.

Instance Diagram for MACSec and MAC Privacy

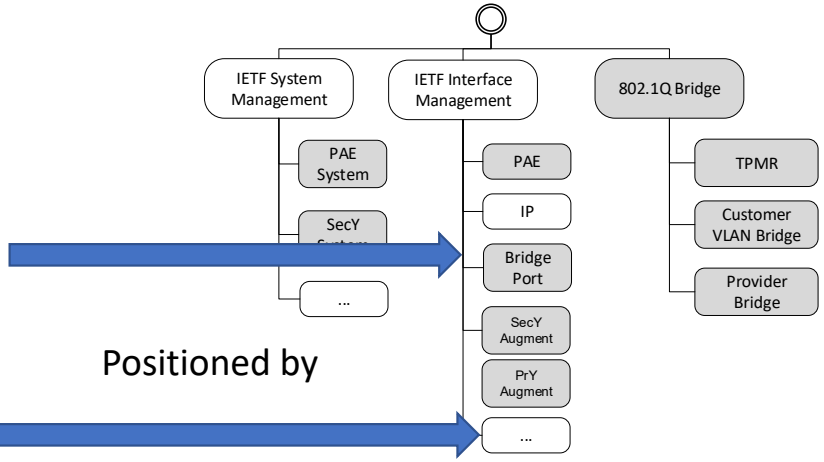


MACsec and MAC Privacy

- Mainly 2 New Modules
- ieee802-dot1ae
  - ieee802-dot1ae-types
- ieee802-dot1ae-pry

Depends on:

  - ietf-yang-types
  - ietf-inet-types
  - iana-if-type
- ieee802-dot1q-bridge
  - ieee802-dot1q-types
- ieee802-dot1x
  - ieee802-dot1x-types
- ietf-interfaces
- ietf-system
- ieee802-types



MAC Privacy Sample Config

```
<qb:bridges
  xmlns:qb="urn:ieee:std:802.1Q:yang:ieee802-dot1q-bridge"
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:sy="urn:ietf:params:xml:ns:yang:ietf-system"
  xmlns:in="urn:ieee:std:802.1Q:ieee802-dot1q-types"
  xmlns:it="urn:ieee:std:802.1Q:ieee802-types"
  xmlns:ae="urn:ieee:std:802.1AE:yang:ieee802-dot1ae"
  xmlns:py="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-pry"
  xmlns:at="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-types"
  xmlns:dx="urn:ieee:std:802.1X:yang:ieee802-dot1x"
  xmlns:xt="urn:ieee:std:802.1X:yang:ieee802-dot1x-types"
  xmlns:yt="urn:ietf:params:xml:ns:yang:ietf-yang-types"
  xmlns:in="urn:ietf:params:xml:ns:yang:ietf-inet-types">
  <qb:bridge>
    <qb:name>bridge1</qb:name>
    <qb:address>10-10-10-10-10-10</qb:address>
    <qb:bridge-type>qb:customer-vlan-bridge</qb:bridge-type>
    <qb:component>
      <qb:name>cv1</qb:name>
      <qb:id>1</qb:id>
      <qb:type>qb:c-vlan-component</qb:type>
      <qb:bridge-vlan>
       <qb:vlan>
          <qb:vid>2</qb:vid>
          <qb:name>vlan2</qb:name>
       </qb:vlan>
      </qb:bridge-vlan>
    </qb:component>
    <qb:component>
      <qb:name>cv2</qb:name>
      <qb:id>2</qb:id>
      <qb:type>qb:c-vlan-component</qb:type>
      <qb:bridge-vlan>
       <qb:vlan>
          <qb:vid>2</qb:vid>
          <qb:name>vlan2</qb:name>
       </qb:vlan>
      </qb:bridge-vlan>
    </qb:component>
  </qb:bridge>
</qb:bridges>
```

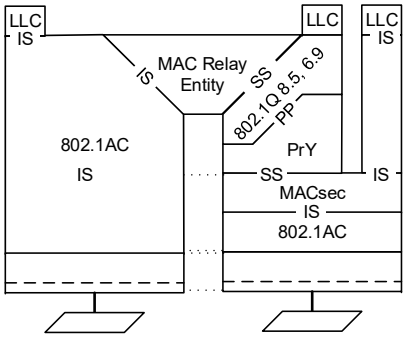
```
<if:interfaces
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:sy="urn:ietf:params:xml:ns:yang:ietf-system"
  xmlns:qb="urn:ieee:std:802.1Q:yang:ieee802-dot1q-bridge"
  xmlns:in="urn:ieee:std:802.1Q:ieee802-dot1q-types"
  xmlns:it="urn:ieee:std:802.1Q:ieee802-types"
  xmlns:ae="urn:ieee:std:802.1AE:yang:ieee802-dot1ae"
  xmlns:py="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-pry"
  xmlns:at="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-types"
  xmlns:dx="urn:ieee:std:802.1X:yang:ieee802-dot1x"
  xmlns:xt="urn:ieee:std:802.1X:yang:ieee802-dot1x-types"
  xmlns:yt="urn:ietf:params:xml:ns:yang:ietf-yang-types"
  xmlns:in="urn:ietf:params:xml:ns:yang:ietf-inet-types"
  xmlns:ia="urn:ietf:params:xml:ns:yang:iana-if-type">
    <if:interface>
      <if:name>eth0</if:name>
      <if:type>ia:bridge</if:type>
      <qb:bridge-port>
        <qb:bridge-name>bridge1</qb:bridge-name>
        <qb:component-name>cv1</qb:component-name>
        <qb:port-type>qb:c-vlan-bridge-port</qb:port-type>
    </qb:bridge-port>
    </if:interface>
    <if:interface>
      <if:name>eth1</if:name>
      <if:type>ia:ethernetCsmacd</if:type>
      <qb:bridge-port>
        <qb:bridge-name>bridge1</qb:bridge-name>
        <qb:component-name>cv2</qb:component-name>
        <qb:port-type>qb:c-vlan-bridge-port</qb:port-type>
      </qb:bridge-port>
    <ae:secy>
      <ae:controlled-port-number>1</ae:controlled-port-number>
      <ae:verification>
       <ae:validate-frames>strict</ae:validate-frames>
       <ae:replay-protect>true</ae:replay-protect>
      </ae:verification>
      <ae:generation>
       <ae:max-transmit-channels>16</ae:max-transmit-channels>
       <ae:max-transmit-keys>16</ae:max-transmit-keys>
       <ae:protect-frames>true</ae:protect-frames>
       <ae:always-include-sci>true</ae:always-include-sci>
       <ae:use-es>true</ae:use-es>
       <ae:use-scb>true</ae:use-scb>
```

```
<ae:user-priority-tc>
        <ae:user-priority>0</ae:user-priority>
        <ae:traffic-class>0</ae:traffic-class>
        <ae:access-class-de0>0</ae:access-class-de0>
        <ae:access-class-de1>0</ae:access-class-de1>
      </ae:user-priority-tc>
      <ae:user-priority-tc>
        <ae:user-priority>1</ae:user-priority>
        <ae:traffic-class>1</ae:traffic-class>
        <ae:access-class-de0>1</ae:access-class-de0>
        <ae:access-class-de1>1</ae:access-class-de1>
      </ae:user-priority-tc>
      <ae:user-priority-tc>
        <ae:user-priority>2</ae:user-priority>
        <ae:traffic-class>2</ae:traffic-class>
        <ae:access-class-de0>2</ae:access-class-de0>
        <ae:access-class-de1>2</ae:access-class-de1>
      </ae:user-priority-tc>
      <ae:user-priority-tc>
        <ae:user-priority>3</ae:user-priority>
        <ae:traffic-class>3</ae:traffic-class>
        <ae:access-class-de0>3</ae:access-class-de0>
        <ae:access-class-de1>3</ae:access-class-de1>
      </ae:user-priority-tc>
     </ae:generation>
   </ae:secy>
```

Adding MAC Privacy is similar

```
<py:pry>
        <py:mac-privacy>enabled</py:mac-privacy>
        <py:user-priority-to-pry>
         <py:user-priority>0</py:user-priority>
         <py:privacy-type>py:none</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>1</py:user-priority>
         <py:privacy-type>py:frame-a</py:privacy-type>
        <py:user-priority-to-pry>
         <py:user-priority>2</py:user-priority>
         <py:privacy-type>py:express-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>3</py:user-priority>
         <py:privacy-type>py:express-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>4</py:user-priority>
         <py:privacy-type>py:standard-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>5</py:user-priority>
         <py:privacy-type>py:standard-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>6</py:user-priority>
         <py:privacy-type>py:standard-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>7</py:user-priority>
         <py:privacy-type>py:standard-channel</py:privacy-type>
       </py:user-priority-to-pry>
       <py:privacy-channel>
         <py:pc>py:standard-channel</py:pc>
         <py:max-per-second-bitrate>10000000000</py:max-per-second-bitrate>
         <py:max-mppdu-size>1500</py:max-mppdu-size>
         <py:mppdu-priority>3</py:mppdu-priority>
       </py:privacy-channel>
      </py:pry>
      <dx:pae>
        <dx:pae-system>pae1</dx:pae-system>
      </pae>
    </if:interface>
</if:interfaces>
```

```
<sy:system
  xmlns:sy="urn:ietf:params:xml:ns:yang:ietf-system"
  xmlns:yt="urn:ietf:params:xml:ns:yang:ietf-yang-types"
  xmlns:it="urn:ieee:std:802.1Q:ieee802-types"
  xmlns:xt="urn:ieee:std:802.1X:yang:ieee802-dot1x-types"
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:ia="urn:ietf:params:xml:ns:yang:iana-if-type"
  xmlns:dx="urn:ieee:std:802.1X:yang:ieee802-dot1x">
  <sy:contact>test</sy:contact>
  <dx:pae-system>
     <dx:name>pae1</dx:name>
     <dx:system-access-control>enabled</dx:system-access-control>
  </dx:pae-system>
</sy:system>
```



Yanglint JSON output for a VLAN Bridge with MACsec

```
> data  -t config   -f json basic-vlan-bridge-with-pry.xml     {
{
  "ieee802-dot1q-bridge:bridges": {
    "bridge": [
      {
        "name": "bridge1",
        "address": "10-10-10-10-10-10",
        "bridge-type": "customer-vlan-bridge",
        "component": [
          {
            "name": "cv1",
            "id": 1,
            "type": "c-vlan-component"
          },
          {
            "name": "cv2",
            "id": 2,
            "type": "c-vlan-component"
          }
        ]
      }
    ]
  },
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:bridge",
        "ieee802-dot1q-bridge:bridge-port": {
          "bridge-name": "bridge1",
          "component-name": "cv1",
          "port-type": "c-vlan-bridge-port",
          "pvid": 1
        },
        "ieee802-dot1x:pae": {

        }
      },
```

Note Abbreviated
TC table is 8 priorities 4 are shown etc

```
"name": "eth1",
"type": "iana-if-type:ethernetCsmacd",
"ieee802-dot1q-bridge:bridge-port": {
  "bridge-name": "bridge1",
  "component-name": "cv2",
  "port-type": "c-vlan-bridge-port",
  "pvid": 1
},
"ieee802-dot1ae:secy": {
  "controlled-port-number": 1,
  "verification": {
    "validate-frames": "strict",
    "replay-protect": true
  },
  "generation": {
    "max-transmit-channels": 16,
    "max-transmit-keys": 16,
    "protect-frames": true,
    "always-include-sci": true,
    "use-es": true,
    "use-scb": true,
    "user-priority-tc": [
      {
        "user-priority": 0,
        "traffic-class": 0,
        "access-class-de0": 0,
        "access-class-de1": 0
      },
      {
        "user-priority": 1,
        "traffic-class": 1,
        "access-class-de0": 1,
        "access-class-de1": 1
      },
      {
        "user-priority": 2,
        "traffic-class": 2,
        "access-class-de0": 2,
        "access-class-de1": 2
      },
```

```
{
            "user-priority": 3,
            "traffic-class": 3,
            "access-class-de0": 3,
            "access-class-de1": 3
          }
        ]
      }
    },
    "ieee802-dot1ae-pry:pry": {
      "mac-privacy": "enabled",
      "user-priority-to-pry": [
        {
          "user-priority": 0,
          "privacy-type": "none"
        },
        {
          "user-priority": 1,
          "privacy-type": "frame-a"
        },
        {
          "user-priority": 2,
          "privacy-type": "express-channel"
        },
        {
          "user-priority": 3,
          "privacy-type": "express-channel"
        },
        {
          "user-priority": 4,
          "privacy-type": "default-channel"
        },
        {
          "user-priority": 5,
          "privacy-type": "default-channel"
        },
        {
          "user-priority": 6,
          "privacy-type": "default-channel"
        },
        {
          "user-priority": 7,
          "privacy-type": “default-channel"
        }
      ],
```

To be added a “None” type

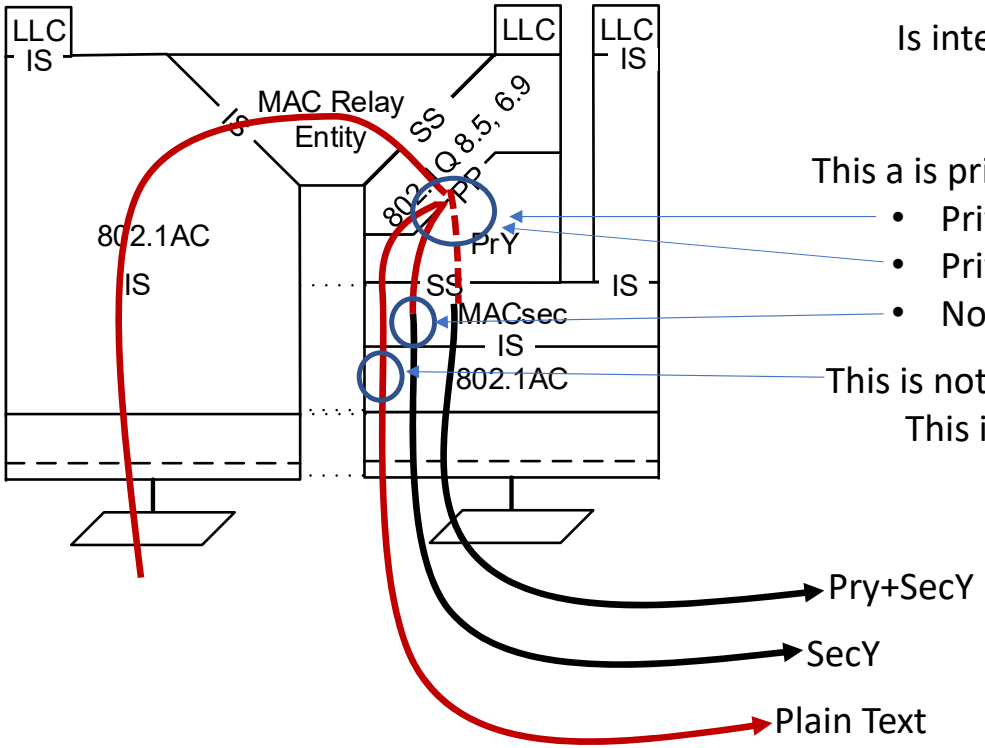
Yanglint JSON output for a VLAN Bridge with MACsec


```
        "privacy-channel": [
          {
            "pc": "default-channel",
            "max-per-second-bitrate": "10000000000",
            "max-mppdu-size": 1500,
            "mppdu-priority": 3
          }
        ]
        "privacy-frame": [
          {
            "pf": "frame-a",
            "max-mppdu-size": 1500,
            "mppdu-priority": 6
          }
        ]
      },
      "ieee802-dot1x:pae": {
        "pae-system": "pae1"
      }
    }
  ]
 },
 "ietf-system:system": {
   "contact": "test",
   "ieee802-dot1x:pae-system": {
     "name": "pae1",
     "system-access-control": "enabled"
   }
 }
}
>
```


Note this table only shows configured items – Defaults are used for other priorities

Note Uncontrolled port and Controlled port have no configurable options.
They do not show up in a configuration view like this.
(Currently stats are read write but that is an error)

How traffic can pass through the Mac Privacy Shim?



Is interface granular enough?

This a is priority table-based map

- Privacy Channel or
- Privacy Frame or
- No Privacy

This is not explicitly supported by the YANG

This is the uncontrolled port but there is no YANG control

Missing two cases?

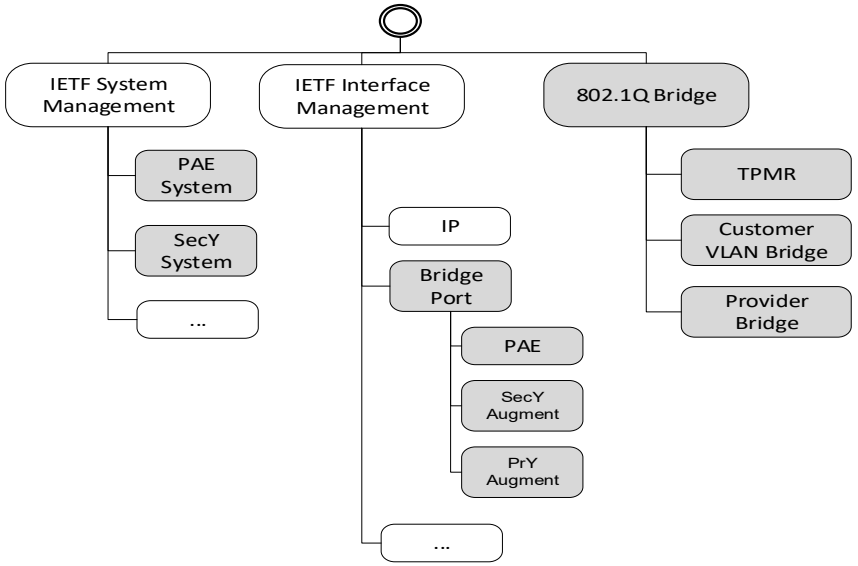
Non-Priority based Traffic

- Mapped to MACsec only (streams)
- Bypass Plain text

Do We have this Right?

- Currently following the precedent set by 802.1X:
- Bridge-port, pae, secy and pry are all at the level of an interface.
- Shouldn’t pae, secy and pry be tied to a bridge-port – cases where there are multiple bridge ports on an interface?
- Alternatively we create multiple virtual interfaces

Maybe we should have this?



This would scope traffic to the bridge port. But still no way to indicate Controlled or uncontrolled port.

Stream/ Scheduler is under a bridge port component.


```
{
 "ieee802-dot1q-bridge:bridges": {
   "bridge": [
     {
       "name": "bridge1",
       "address": "10-10-10-10-10-10",
       "bridge-type": "customer-vlan-bridge",
       "component": [
         {
           "name": "cv1",
           "id": 1,
           "type": "c-vlan-component",
           "ieee802-dot1q-stream-filters-gates:stream-filters": {
             "stream-filter-instance-table": [
               {
                 "stream-filter-instance-id": 1,
                 "wildcard": [null],
                 "priority-spec": "three",
                 "max-sdu-size": 0,
                 "stream-blocked-due-to-oversize-frame-enabled": true,
                 "stream-blocked-due-to-oversize-frame": true,
                 "stream-gate-ref": 1,
                 "ieee802-dot1q-psfp:flow-meter-instance-id": 1,
                 "ieee802-dot1q-psfp:flow-meter-enable": true
               }
             ]
           },
           "ieee802-dot1q-stream-filters-gates:stream-gates": {
             "stream-gate-instance-table": [
               {
                 "stream-gate-instance-id": 1,
                 "gate-enable": true,
                 "admin-gate-states": "open",
                 "ieee802-dot1q-psfp:oper-ipv": "null",
                 "ieee802-dot1q-psfp:admin-control-list": {
                   "gate-control-entry": [
                     {
                       "index": 1,
                       "operation-name": "set-gate-and-ipv",
                       "time-interval-value": 1000,
                       "gate-state-value": "open",
                       "ipv-spec": "null",
                       "interval-octet-max": 1000
                     },
                     {
                       "index": 2,
                       "operation-name": "set-gate-and-ipv",
                       "time-interval-value": 1000,
                       "gate-state-value": "open",
                       "ipv-spec": "null",
                       "interval-octet-max": 1000
                     }
                   ]
                 },
                 "ieee802-dot1q-psfp:admin-cycle-time": {
                   "numerator": 10,
                   "denominator": 100
                 },
                 "ieee802-dot1q-psfp:admin-cycle-time-extension": 100,
                 "ieee802-dot1q-psfp:admin-base-time": {
                   "seconds": "1",
                   "nanoseconds": 1000
                 },
                 "ieee802-dot1q-psfp:config-change": false,
                 "ieee802-dot1q-psfp:gate-closed-due-to-invalid-rx-enable": true,
                 "ieee802-dot1q-psfp:gate-closed-due-to-invalid-rx": false,
                 "ieee802-dot1q-psfp:gate-closed-due-octets-exceeded-enable": false,
                 "ieee802-dot1q-psfp:gate-closed-due-octets-exceeded": false
               }
             ],
             "ieee802-dot1q-psfp:supported-list-max": 100,
             "ieee802-dot1q-psfp:supported-cycle-max": {
               "numerator": 100,
               "denominator": 1000
             },
             "ieee802-dot1q-psfp:supported-interval-max": 1000
           },
```

Stream is under a bridge port.


```
"ieee802-dot1q-psfp:flow-meters": {
            "flow-meter-instance-table": [
              {
                "flow-meter-instance-id": 1,
                "committed-information-rate": "1000000000",
                "committed-burst-size": 10000,
                "excess-information-rate": "10000000",
                "excess-burst-size": 100000,
                "coupling-flag": "one",
                "color-mode": "color-aware",
                "drop-on-yellow": true,
                "mark-all-frames-red-enable": false,
                "mark-all-frames-red": false
              }
            ],
            "max-flow-meter-instances": 1000
          }
        }
      ]
    }
  ]
},
```

```
"ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:bridge",
        "ieee802-dot1q-bridge:bridge-port": {
          "bridge-name": "bridge1",
          "component-name": "cv1",
          "port-type": "c-vlan-bridge-port",
          "pvid": 1,
          "ieee802-dot1q-sched:gate-parameter-table": {
            "queue-max-sdu-table": [
              {
                "traffic-class": 0,
                "queue-max-sdu": 100
              }
            ],
            "gate-enabled": true,
            "admin-gate-states": 255,
            "admin-control-list": {
              "gate-control-entry": [
                {
                  "index": 1,
                  "operation-name": "set-gate-states",
                  "time-interval-value": 100,
                  "gate-states-value": 255
                }
              ]
            },
            "admin-cycle-time": {
              "numerator": 10,
              "denominator": 1000
            },
            "admin-cycle-time-extension": 999,
            "admin-base-time": {
              "seconds": "1",
              "nanoseconds": 1000
            },
            "config-change": true,
            "supported-list-max": 100,
            "supported-cycle-max": {
              "numerator": 10,
              "denominator": 1000
            },
            "supported-interval-max": 100
          }
        }
      },
```

Back up

- Yanglint Validation

MACsec and MAC Privacy YANG Some lessons learned

- Instance Model – Where the YANG trees lives
- YANG Models – What to configure and what to display
  - Our bridge Model is a large superset that supports many permutations.
  - The model contains a lot of detail.
  - The tree provides a useful summary (a slice of the instance model)
- Validation
  - Pyang – validates a single model
  - Various other tools
- Instance Configuration – IEEE is in general only beginning to look at this
  - Yuma123
  - Confd (free version)
  - Yanglint (Used by IETF)

Validation versus Instance configuration

- Validation
  - YANG syntax is correct
  - YANG xpath is syntactically correct x=y (but x may be apples and y may be oranges)
  - The whole set of permutations it the tree file or the xml decription.
- Instance configuration
  - Config values are tested reference pointers are checked
  - YANG syntax is correct and multiple modules that are not related can exist side by side
  - x = y and x is the set of apple types and y is a type of apple (Macintosh but not iphone!).
  - A slice of valid configuration references links are tested