

YANG-push status overview

INSA de Lyon

Alex Huang Feng (alex.huang-feng@insa-lyon.fr)

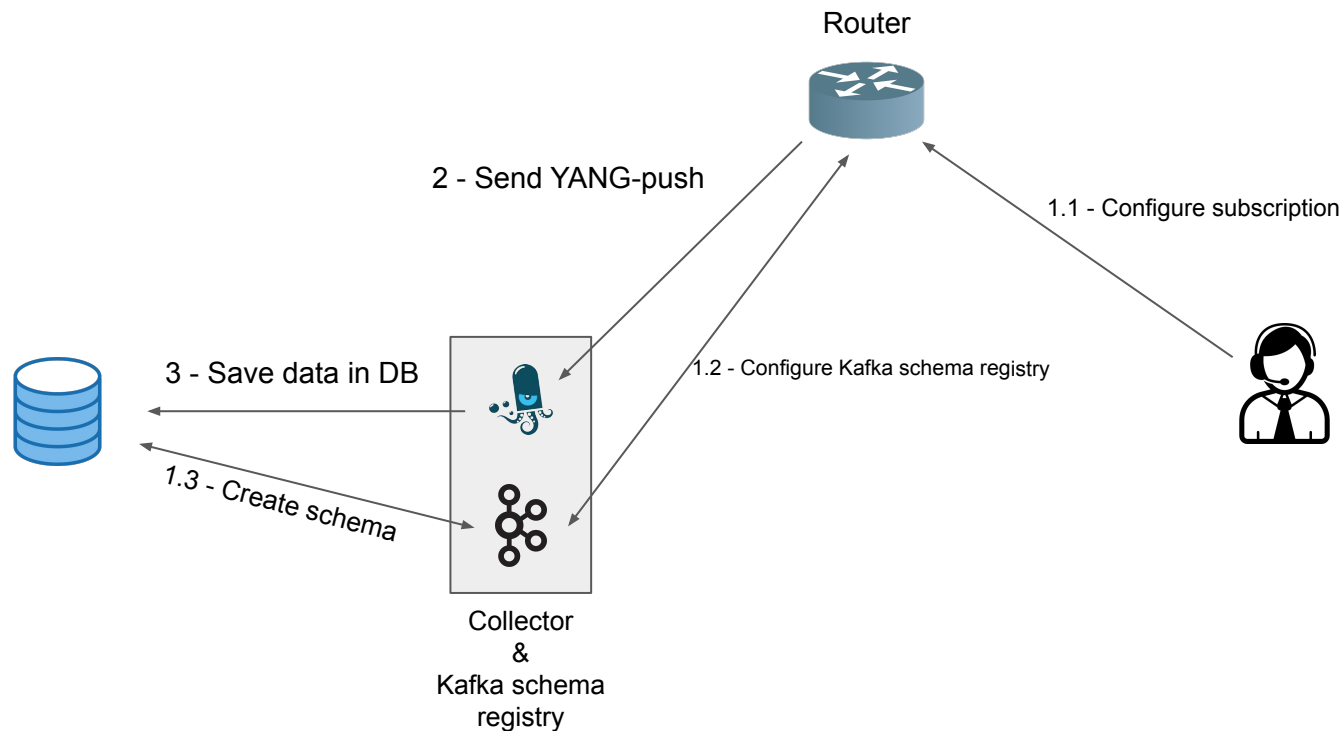
Pierre Francois (pierre.francois@insa-lyon.fr)

Goals

- Architecture overview
- Describe how to configure YANG Push
 - Management plane: Netconf RPCs
 - Dataplane: YANG Push messages
- Configuration variants
 - Datastore
 - Stream
 - On change vs. periodic updates
- Testing: Features of Scapy emulator
- Discuss possible gaps in the architecture/RFCs/Drafts

Architecture overview

Topology for Configured Subscriptions RFC8639/RFC8641

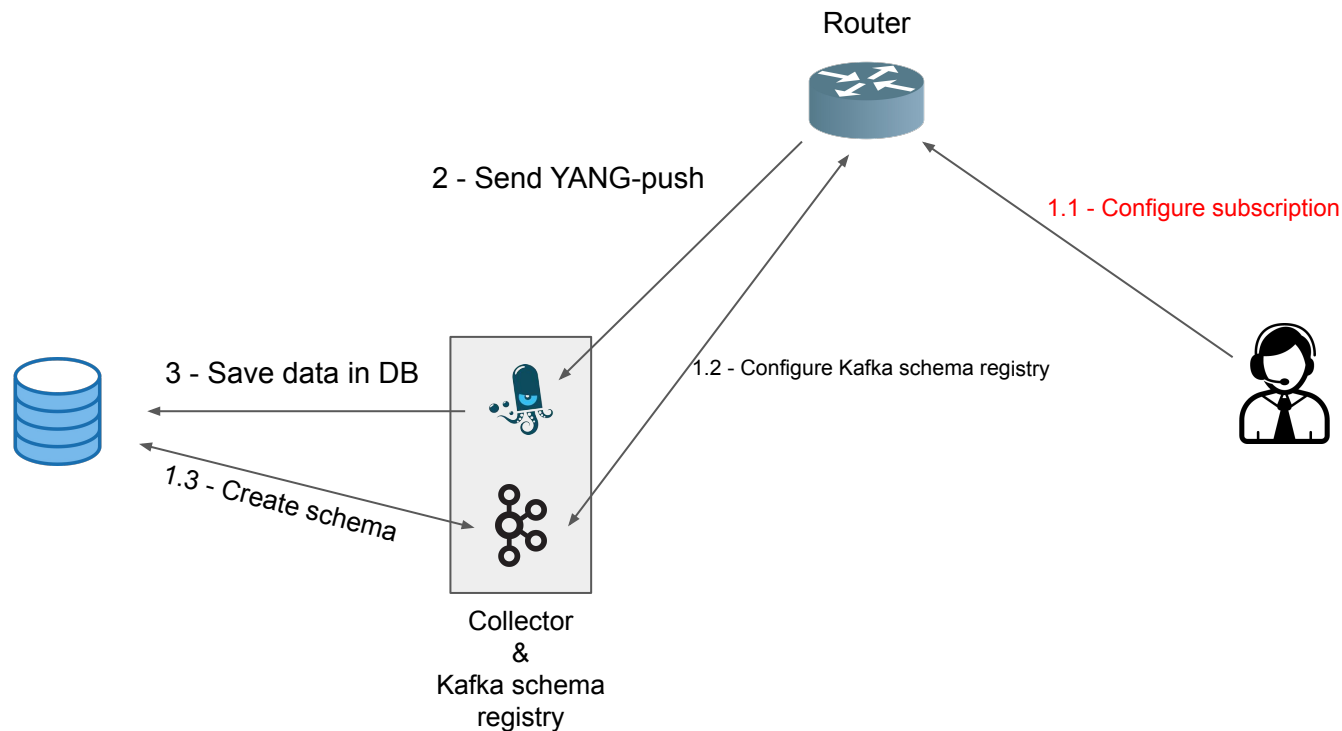


Architecture overview

Configure subscription

YANG Push messages

Topology for Configured Subscriptions RFC8639/RFC8641



Stream and datastore

- [RFC8639] A configuration performed to obtain a flow of events from the publisher
- NIC temperatures
- Netconf events [RFC8639]

```

module: ietf-subscribed-notifications
+--ro streams
|   +--ro stream* [name]
|   |   +--ro name                string
|   |   +--ro description?        string
|   |   +--ro replay-support?     empty {replay}?
|   |   +--ro replay-log-creation-time ietf-yang-types:date-and-time {replay}?
|   |   +--ro replay-log-aged-time? ietf-yang-types:date-and-time {replay}?

```

```

+--:(stream)
|   +--rw (stream-filter)?
|   |   +--:(by-reference)
|   |   |   +--rw stream-filter-name                stream-filter-ref
|   |   +--:(within-subscription)
|   |   +--rw (filter-spec)?
|   |   |   +--:(stream-subtree-filter)
|   |   |   |   +--rw stream-subtree-filter?        <anydata> {subtree}?
|   |   +--:(stream-xpath-filter)
|   |   |   +--rw stream-xpath-filter?              ietf-yang-types:xpath1.0 {xpath}?
+--rw stream                                          stream-ref
+--ro replay-start-time?                             ietf-yang-types:date-and-time {replay}?
+--rw configured-replay?                             empty {configured,replay}?

```

- [RFC8641] A configuration performed to obtain a flow of updates happening on a datastore [RFC8342] of the publisher
- An update to a config
<startup>; <candidate>; <running>;
<operational>; <intended>

```

+--:(ietf-yang-push:datastore)
+--rw ietf-yang-push:datastore          identityref
+--rw (ietf-yang-push:selection-filter)?
+--:(ietf-yang-push:by-reference)
| +--rw ietf-yang-push:selection-filter-ref  selection-filter-ref
+--:(ietf-yang-push:within-subscription)
+--rw (ietf-yang-push:filter-spec)?
+--:(ietf-yang-push:datastore-subtree-filter)
| +--rw ietf-yang-push:datastore-subtree-filter?  <anydata> {sn:subtree}?
+--:(ietf-yang-push:datastore-xpath-filter)
+--rw ietf-yang-push:datastore-xpath-filter?  ietf-yang-types:xpath1.0 {sn:xpath}?

```

Configure Subscription with RFC8639: Stream



config tree are wrapped in <edit-config> rpc

(1.1)

```
<?xml version='1.0' encoding='UTF-8'?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
    <subscription>
      <id>6666</id>
      <stream-subtree-filter>some-subtree-filter</stream-subtree-filter>
      <stream>some-stream</stream>
      <transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
      <encoding>encode-json</encoding>
      <receivers>
        <receiver>
          <name>subscription-specific-receiver-def</name>
          <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">global-udp-notif-receiver-def</receiver-instance-ref>
        </receiver>
      </receivers>
    </subscription>
  <receiver-instances xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">
    <receiver-instance>
      <name>global-udp-notif-receiver-def</name>
      <udp-notif-receiver xmlns="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">
        <address>192.0.5.1</address>
        <port>12345</port>
        <enable-segmentation>false</enable-segmentation>
        <max-segment-size/>
      </udp-notif-receiver>
    </receiver-instance>
  </receiver-instances>
</subscriptions>
</config>
```


Configure Subscription with RFC8641: Datastore



config tree are wrapped in <edit-config> rpc

(1-bis.1)

```
<?xml version='1.0' encoding='UTF-8'?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
    <subscription>
      <id>6666</id>
      <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">ds:operational</datastore>
      <datastore-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">/some/path</datastore-xpath-filter>
      <transport xmlns="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
      <encoding>encode-json</encoding>
      <receivers>
        <receiver>
          <name>subscription-specific-receiver-def</name>
          <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">global-udp-notif-receiver-def</receiver-instance-ref>
        </receiver>
      </receivers>
    </subscription>
  <receiver-instances xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">
    <receiver-instance>
      <name>global-udp-notif-receiver-def</name>
      <udp-notif-receiver xmlns="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">
        <address>192.0.5.1</address>
        <port>12345</port>
        <enable-segmentation>false</enable-segmentation>
        <max-segment-size/>
      </udp-notif-receiver>
    </receiver-instance>
  </receiver-instances>
</subscriptions>
</config>
```

Configure Subscription RFC8639/RFC8641: RPC reply



(1.2)

QUESTION

- Name of the stream == YANG module name?
- The datastore name is associated to a YANG module name by default? How can we find this relation?

ANSWER

Thomas: we get the YANG module from the XPath

```
<rpc-reply message-id="101"  
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">  
  <ok/>  
</rpc-reply>
```

After 1.2, there is still no reference to the YANG module name

Getting the YANG module from XPath

- From RFC8641 Section 3.6

*A subscription **must** specify both the selection filters and the datastore[...].*

[...]

*A publisher **MUST** support **at least one** type of selection filter.*

- YANG Types
 - subtree: anydata
 - xpath: yang:xpath1.0

3.6. Defining the Selection with a Datastore

A subscription must specify both the selection filters and the datastore against which these selection filters will be applied. This information is used to choose and subsequently push data from the publisher's datastore to the receivers.

Only a single selection filter can be applied to a subscription at a time. An RPC request proposing a new selection filter replaces any existing filter. The following selection filter types are included in the YANG-Push data model and may be applied against a datastore:

- o subtree: A subtree selection filter identifies one or more datastore subtrees. When specified, update records will only come from the datastore nodes of selected datastore subtree(s). The syntax and semantics correspond to those specified in [\[RFC6241\], Section 6](#).
- o xpath: An "xpath" selection filter is an XPath expression that returns a node set. (XPath is a query language for selecting nodes in an XML document; see [\[XPath\]](#) for details.) When specified, updates will only come from the selected datastore nodes.

These filters are intended to be used as selectors that define which objects are within the scope of a subscription. A publisher **MUST** support at least one type of selection filter.

We need to assume that the namespace/YANG module name is specified in the Xpath.

XPath in XML

- /example:root
- /example:root/actors
- /example:root/foo:singers

```
<root xmlns:yp="urn:ietf-yang-push" xmlns="urn:example">
  <actors>
    <actor id="1">Christian Bale</actor>
    <actor id="2">Liam Neeson</actor>
    <actor id="3">Michael Caine</actor>
  </actors>
  <foo:singers>
    <foo:singer id="4">Tom Waits</foo:singer>
    <foo:singer id="5">B.B. King</foo:singer>
    <foo:singer id="6">Ray Charles</foo:singer>
  </foo:singers>
</root>
```

“example” is the YANG module name bound to the namespace
“urn:example”

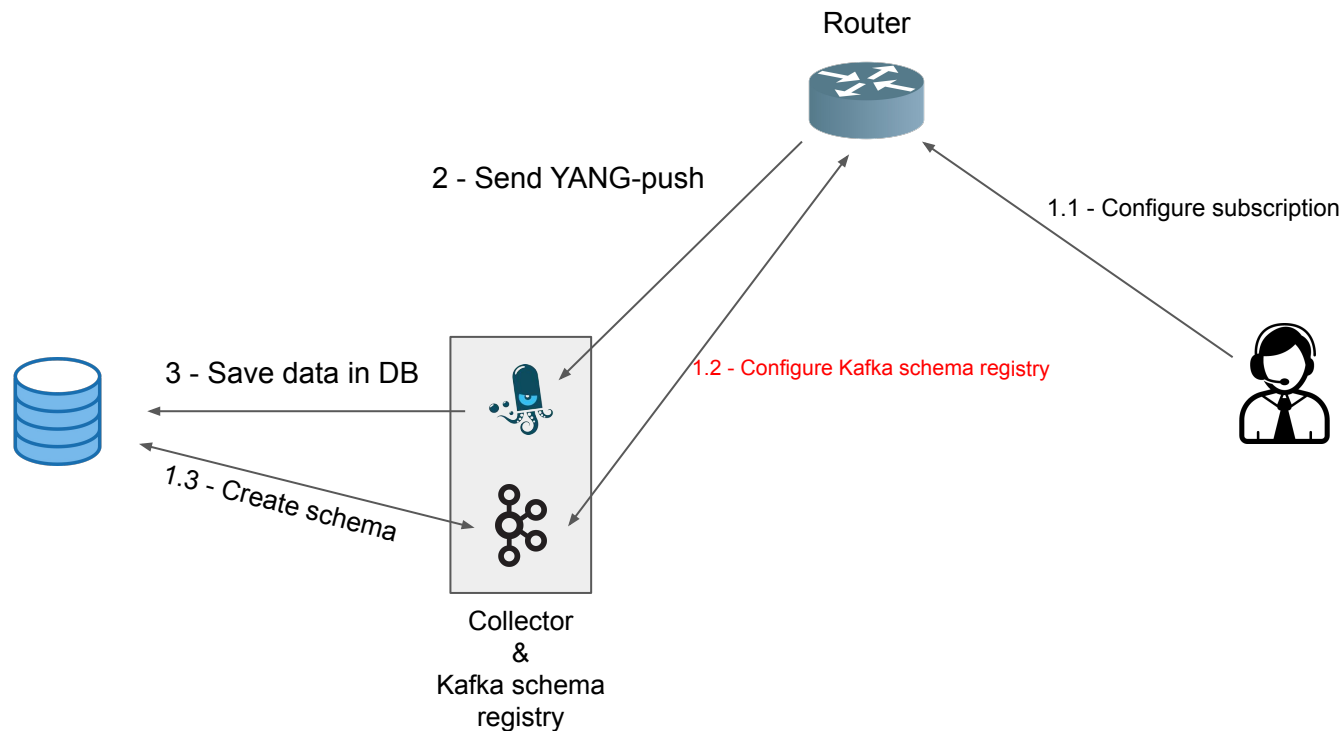
Unwritten rule:

- *An XPath has always the root yang module name*

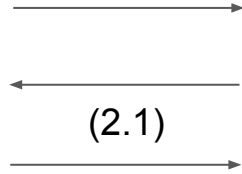
Current drafts

- draft-tgraf-netconf-yang-notifications-versioning
 - Adds “revision” and “revision-label” to the subscription
- draft-tgraf-netconf-notif-sequencing
 - Adds “sysName” and “sequenceNumber”
- draft-tgraf-yang-push-observation-time
 - Adds “start-observation-time” to push-update
 - Adds “changed-observation-time” to push-change-update

Topology for Configured Subscriptions RFC8639/RFC8641



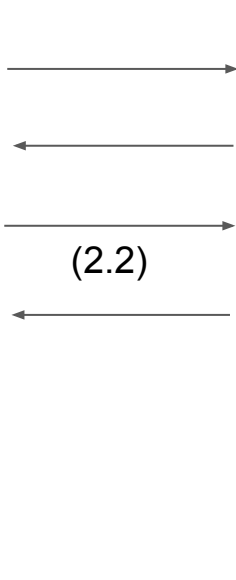
Configure Subscription RFC8639/RFC8641 - Obtaining available schemas



Discovering available YANGs

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <netconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
        <schemas/>
      </netconf-state>
    </filter>
  </get>
</rpc>
```

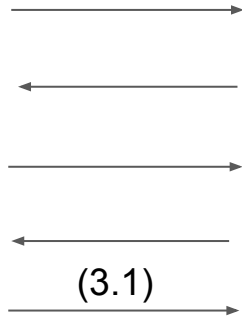
Configure Subscription RFC8639/RFC8641 - Obtaining available schemas



Discovering available YANGs

```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <netconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
      <schemas>
        <schema>
          <identifier>foo</identifier>
          <version>2020-10-10</version>
          <format>yang</format>
          <namespace>http://example.com/foo</namespace>
          <location>
            http://example.com/schema/foo@2020-10-10.yang
          </location>
          <location>NETCONF</location>
        </schema>
        <schema>
          <identifier>bar-types</identifier>
          <version>2008-06-01</version>
          <format>yang</format>
          <namespace>http://example.com/bar</namespace>
          <location>
            http://example.com/schema/bar-types@2008-06-01.yang
          </location>
          <location>NETCONF</location>
        </schema>
      </schemas>
    </netconf-state>
  </data>
</rpc-reply>
```

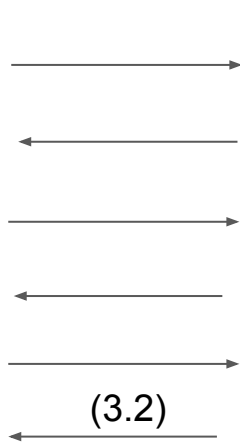

Configure Subscription RFC8639/RFC8641 - Get schema



```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-schema xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
    <identifier>foo</identifier>
    <version>1.0</version>
    <format>yang</format>
  </get-schema>
</rpc>
```

Configure Subscription RFC8639/RFC8641

Using the schema

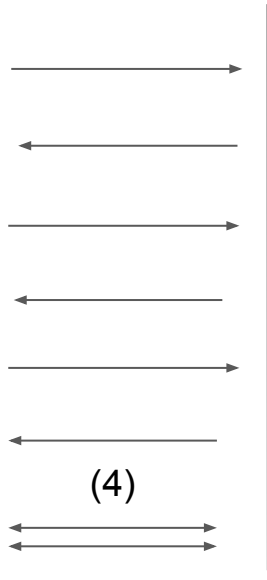


```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
    module foo {
      //default format (yang) returned
      //bar version 2020-10-10 yang module
      //contents here ...
    }
  </data>
</rpc-reply>
```

*Once got the YANG module and the Xpath
→ Get recursively the imported YANG modules*

Configure Subscription RFC8639/RFC8641

Recursively get YANGS



Get all imported YANGs

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-schema xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
    <identifier>foo</identifier>
    <version>1.0</version>
    <format>yang</format>
  </get-schema>
</rpc>
```

```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
    module foo {
      //default format (yang) returned
      //bar version 2020-10-10 yang module
      //contents here ...
    }
  </data>
</rpc-reply>
```

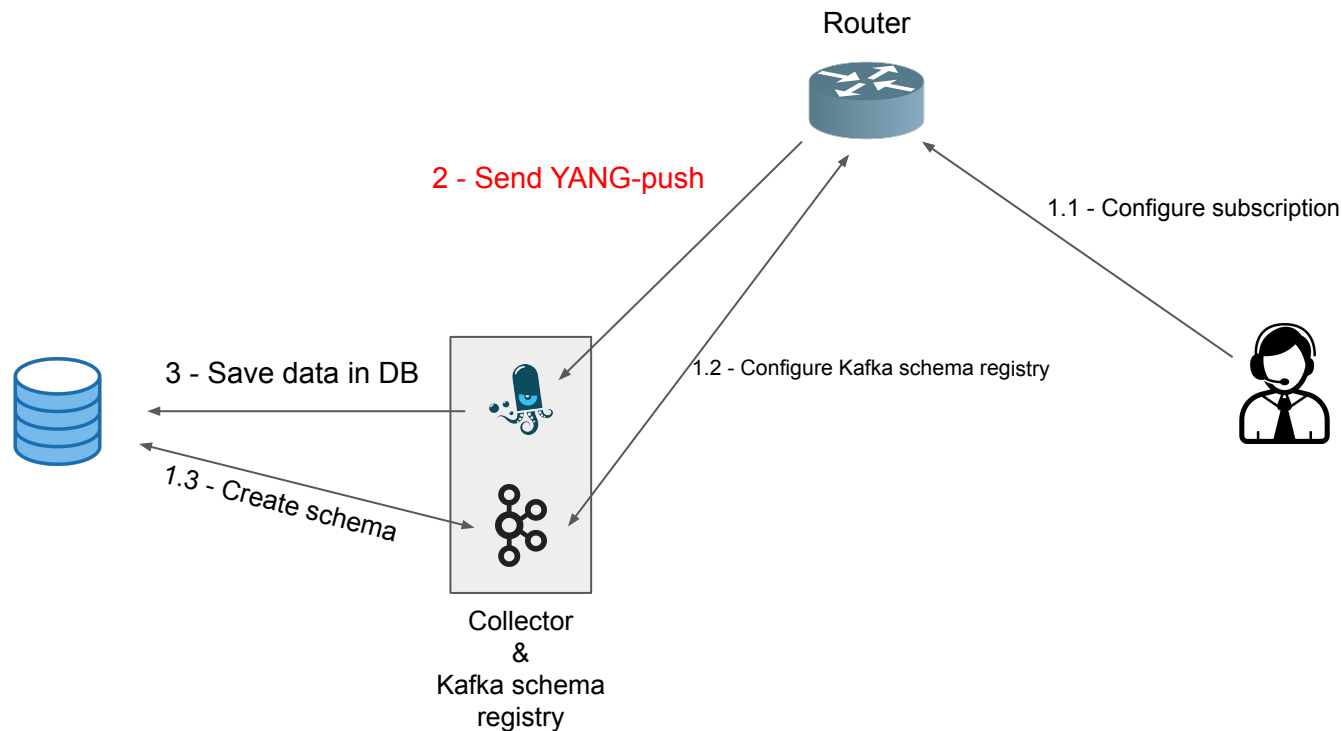
Once got all the YANG module and the Xpath
→ Push all these YANG modules to the Schema registry
→ Reconstruct YANG module with all the notification header and everything compiled for the consumer/producer

Architecture overview

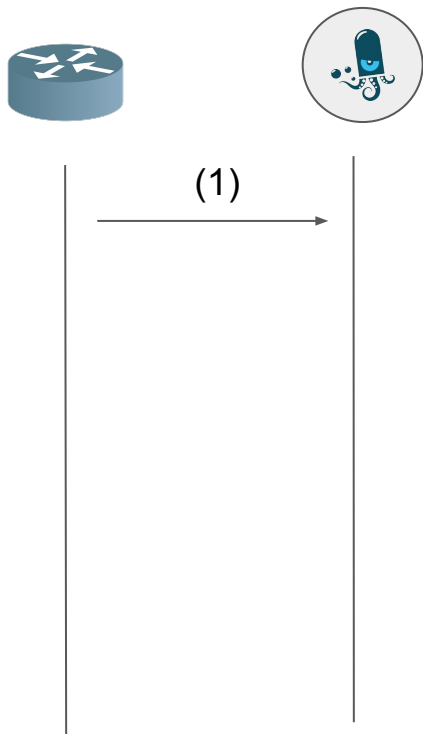
Configure subscription

YANG Push messages

Topology for Configured Subscriptions RFC8639/RFC8641

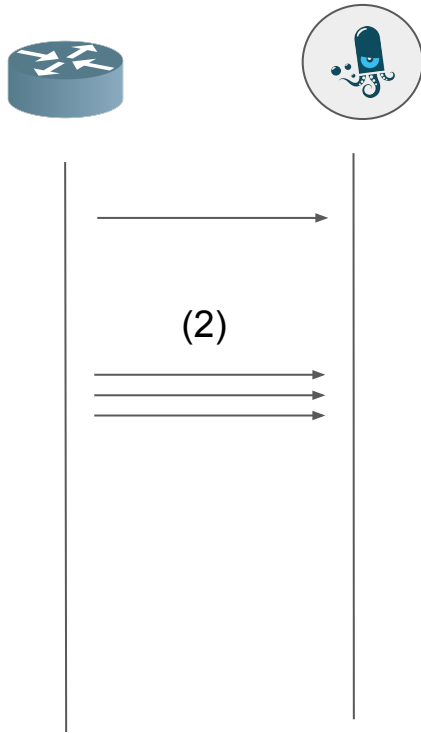


Sending YANG-push to the collector: Subscription started



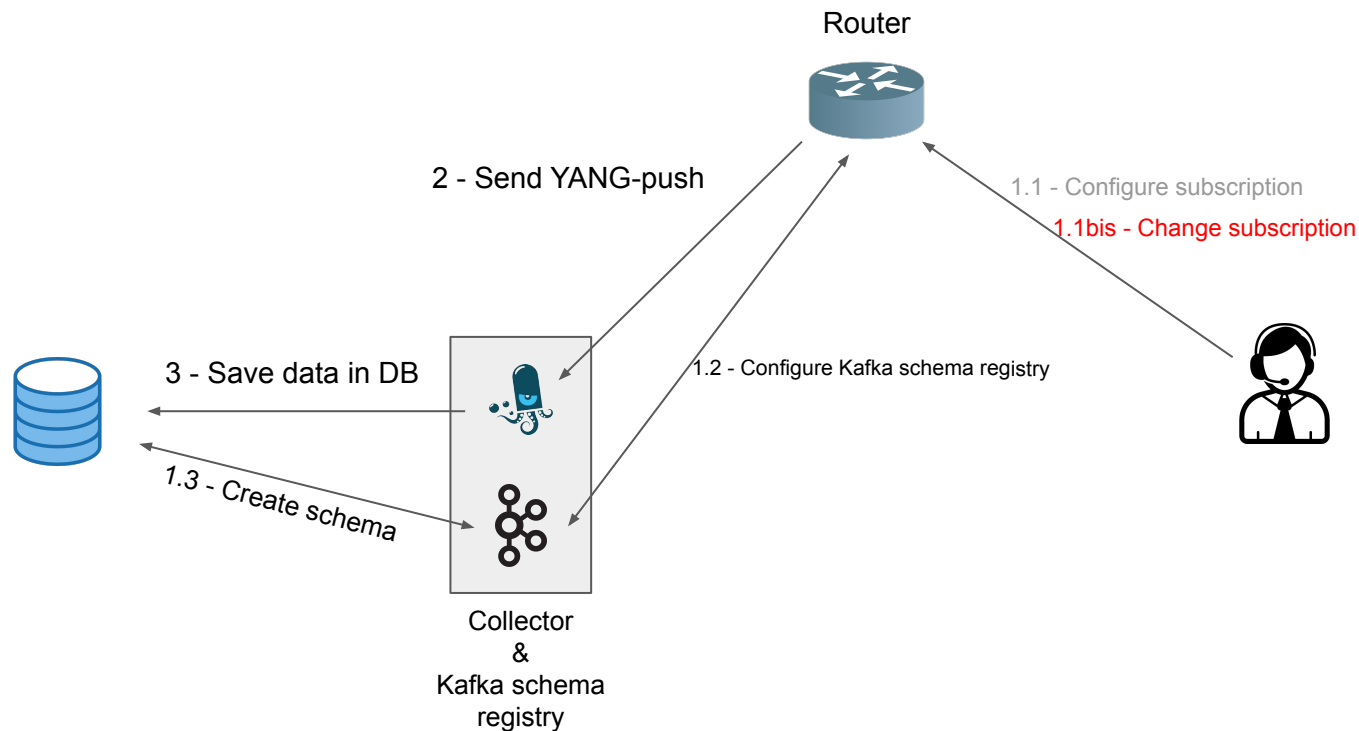
```
{
  "ietf-notification:notification": {
    "eventTime": "2023-03-25T08:30:11.22Z",
    "ietf-subscribed-notification:subscription-started": {
      "id": 1011,
      "target": {
        "ietf-yang-push:datastore": "ds:operational"
      },
      "transport": "ietf-udp-notif-transport:udp-notif",
      "encoding": "encode-json",
      "ietf-yang-push:periodic": {
        "ietf-yang-push:period": 100
      }
    }
  }
}
```

2 - Sending YANG-push to the collector: Updates



```
{
  "ietf-notification:notification": {
    "eventTime": "2023-03-25T08:30:11.22Z",
    "ietf-yang-push:push-update": {
      "id": 6666,
      "datastore-contents": {
        "ietf-interfaces:interfaces": [
          {
            "interface": {
              "name": "eth0",
              "oper-status": "up"
            }
          }
        ]
      }
    }
  }
}
```

Topology for Configured Subscriptions RFC8639/RFC8641



Change Subscription RFC8639/RFC8641 - Router



Example: Change encoding

(1-bis.1)

```
<?xml version='1.0' encoding='UTF-8'?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
    <subscription>
      <id>6666</id>
      <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">ds:operational</datastore>
      <datastore-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">/some/path</datastore-xpath-filter>
      <transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
      <encoding>encode-xml</encoding>
      <receivers>
        <receiver>
          <name>subscription-specific-receiver-def</name>
          <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">global-udp-notif-receiver-def</receiver-instance-ref>
        </receiver>
      </receivers>
    </subscription>
    <receiver-instances xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">
      <receiver-instance>
        <name>global-udp-notif-receiver-def</name>
        <udp-notif-receiver xmlns="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">
          <address>192.0.5.1</address>
          <port>12345</port>
          <enable-segmentation>false</enable-segmentation>
          <max-segment-size/>
        </udp-notif-receiver>
      </receiver-instance>
    </receiver-instances>
  </subscriptions>
</config>
```

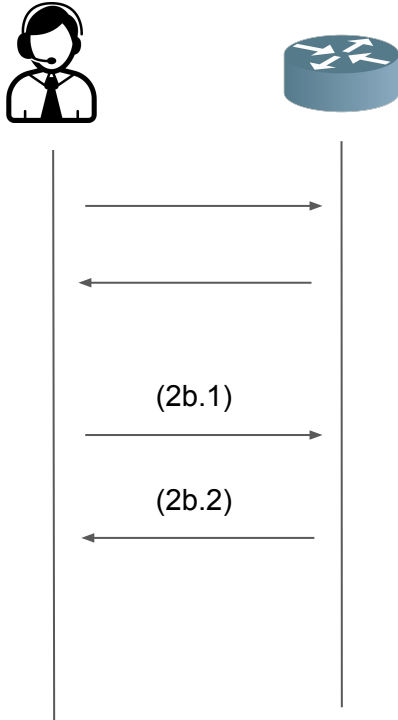
Change Subscription RFC8639/RFC8641 - Router



(1.2)

```
<rpc-reply message-id="101"  
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">  
  <ok/>  
</rpc-reply>
```

Change Subscription RFC8639/RFC8641 - Router

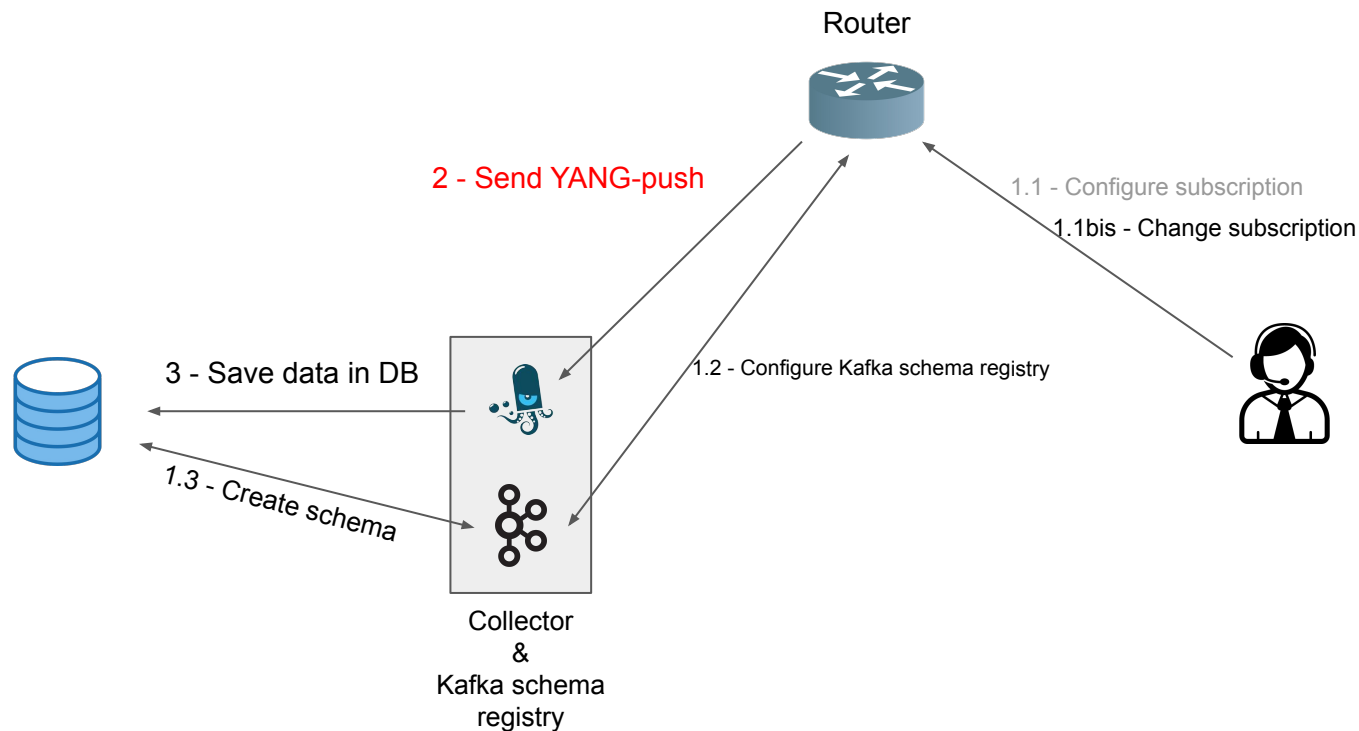


```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-schema xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
    <identifier>foo</identifier>
    <version>1.0</version>
    <format>yang</format>
  </get-schema>
</rpc>
```

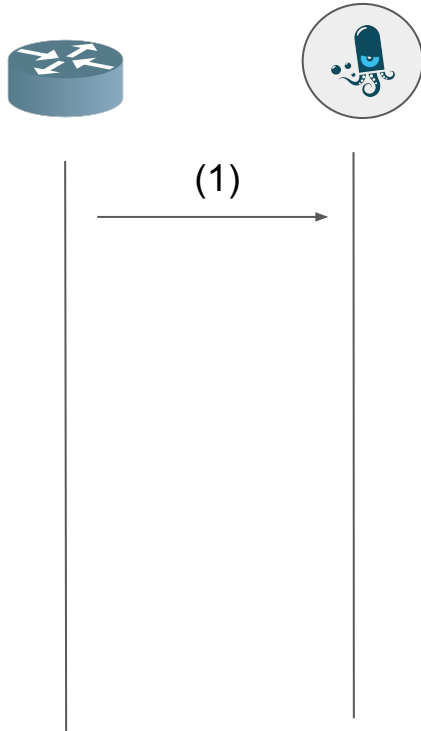
```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
    module foo {
      //default format (yang) returned
      //bar version 2020-10-10 yang module
      //contents here ...
    }
  </data>
</rpc-reply>
```

Pushing new YANG module to Schema registry

Topology for Configured Subscriptions RFC8639/RFC8641



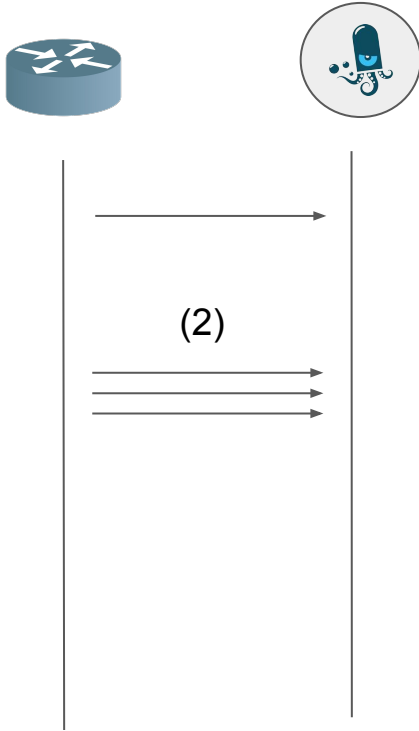
2b - Sending YANG-push to collector



```
{  
  "ietf-notification:notification": {  
    "eventTime": "2023-03-25T08:30:11.22Z",  
    "ietf-subscribed-notification:subscription-modified": {  
      "id": 6666,  
      "target": {  
        "ietf-yang-push:datastore": "ds:operational"  
      },  
      "transport": "ietf-udp-notif-transport:udp-notif",  
      "encoding": "encode-json",  
      "ietf-yang-push:periodic": {  
        "ietf-yang-push:period": 100  
      }  
    }  
  }  
}
```

This first message is in json or in xml?

2 - Sending YANG-push to collector



```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2022-09-02T10:59:55.32Z</eventTime>
  <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <id>6666</id>
    <datastore-contents>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>eth0</name>
          <oper-status>up</oper-status>
        </interface>
      </interfaces>
    </datastore-contents>
  </push-update>
</notification>
```

Scapy mock generator

Implementation status

Planned updates

Scapy: Status

- Classic flow
 - Subscription started
 - PUSH Updates
 - Subscription terminated
- Transport:
 - UDP-notif (-08): <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-udp-notif-08>
 - UDP pub channel (-05): <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-udp-pub-channel-05>
- Encoding
 - JSON and XML supported
 - CBOR on the radar

Scapy: Planned

- A subscription is changed to a new (backward compatible) YANG module
 - Subscription started
 - PUSH Updates
 - Subscription modified
 - PUSH Updates
 - Subscription terminated
- Support new drafts in YANG-push notifications:
 - draft-ahuang-netconf-notif-yang
 - draft-tgraf-netconf-notif-sequencing
 - draft-tgraf-yang-push-observation-time
 - draft-tgraf-netconf-yang-notifications-versioning
 - draft-ietf-netconf-distributed-notif
- Transport:
 - UDP-notif (-09): will be published soon
 - UDP pub channel (-05): <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-udp-pub-channel-05>
- Issue:
 - **How to know from the configuration to which YANG module we are subscribing to?**

Scapy mock generator

Draft integrated in the project

JSON supported messages

XML supported messages

Supported RFC/draft in UDP-notif scapy mock generator

- draft-ahuang-netconf-notif-yang
- draft-tgraf-netconf-notif-sequencing
- draft-tgraf-yang-push-observation-time
- draft-tgraf-netconf-yang-notifications-versioning
- draft-ietf-netconf-distributed-notif
- RFC8639 - Subscribed Notifications
- RFC8641 - YANG-push
- RFC7223 - A YANG Data Model for Interface Management

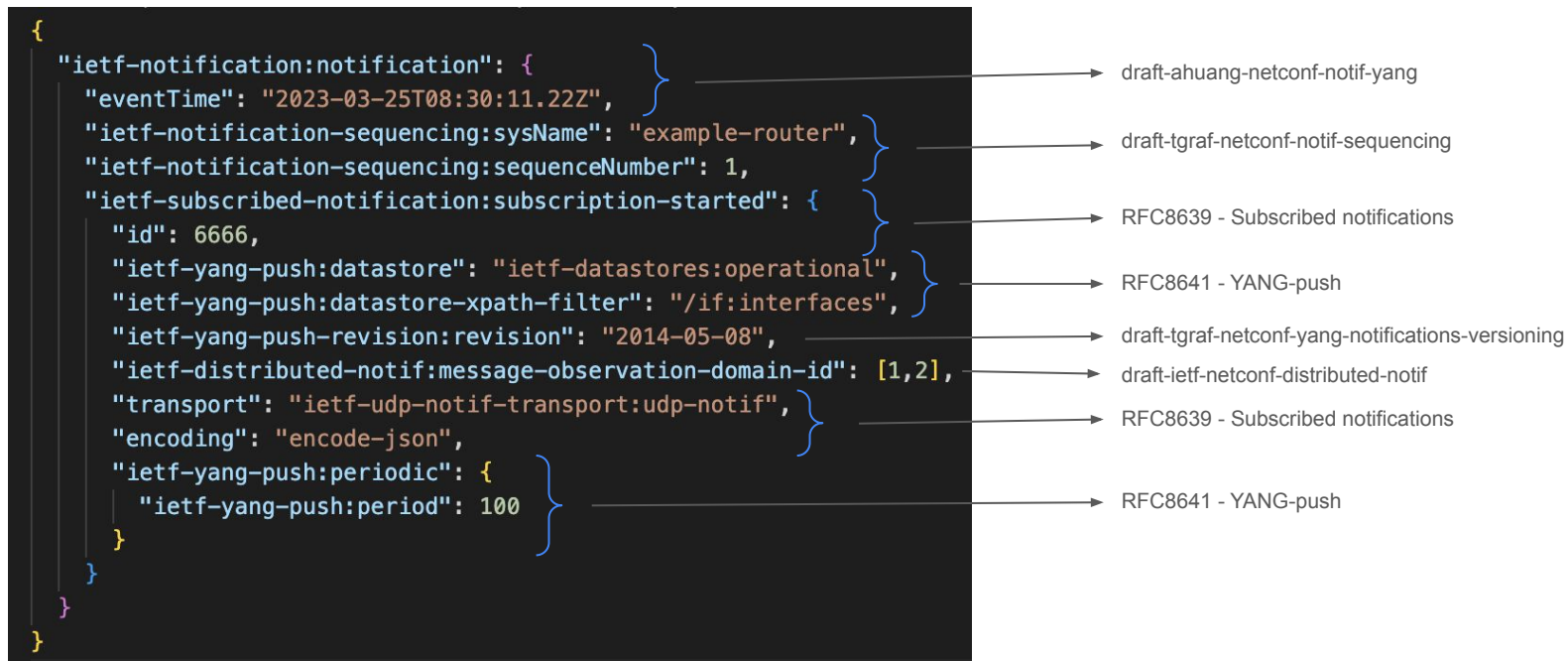
Scapy mock generator

Draft integrated in the project

JSON supported messages

XML supported messages

JSON messages : <subscription-started>



JSON messages : <push-update>

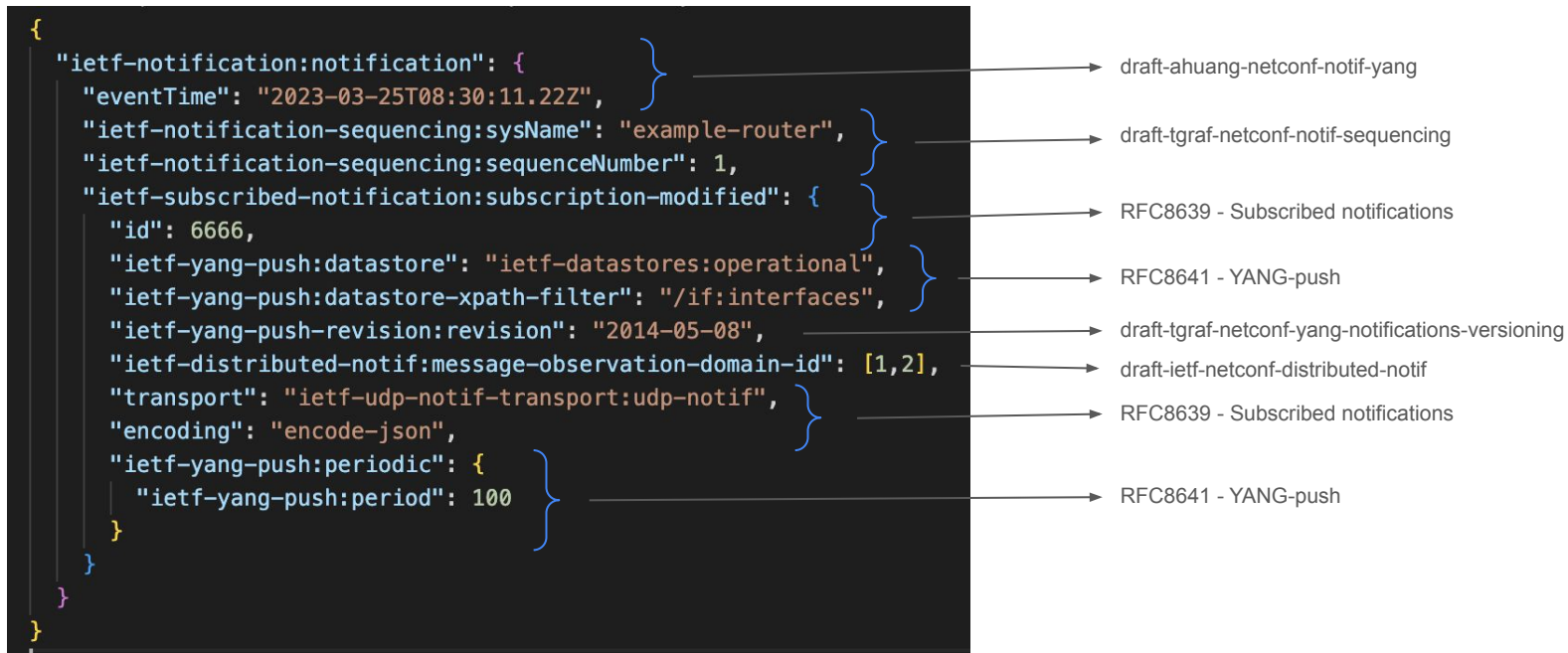
```

{
  "ietf-notification:notification": {
    "eventTime": "2023-03-25T08:30:11.22Z",
    "ietf-notification-sequencing:sysName": "example-router",
    "ietf-notification-sequencing:sequenceNumber": 1,
    "ietf-yang-push:push-update": {
      "id": 6666,
      "ietf-yang-push-netobs-timestamping:observation-time": "2023-03-25T08:30:11.22Z",
      "datastore-contents": {
        "ietf-interfaces:interfaces": [
          {
            "interface": {
              "name": "eth0",
              "oper-status": "up"
            }
          }
        ]
      }
    }
  }
}

```

- draft-ahuang-netconf-notif-yang
- draft-tgraf-netconf-notif-sequencing
- RFC8639 - Subscribed notifications
- draft-tgraf-yang-push-observation-time
- RFC8641 - YANG-push
- Monitored Yang: ietf-interfaces.yang

JSON messages : <subscription-modified>



JSON messages : updated <push-update>

```
{
  "ietf-notification:notification": {
    "eventTime": "2023-03-25T08:30:11.22Z",
    "ietf-notification-sequencing:sysName": "example-router",
    "ietf-notification-sequencing:sequenceNumber": 1,
    "ietf-yang-push:push-update": {
      "id": 6666,
      "ietf-yang-push-netobs-timestamping:observation-time": "2023-03-25T08:30:11.22Z",
      "datastore-contents": {
        "ietf-interfaces:interfaces": [
          {
            "interface": {
              "name": "eth0",
              "oper-status": "up",
              "ietf-mock-interfaces:packet-count": 150
            }
          }
        ]
      }
    }
  }
}
```

draft-ahuang-netconf-notif-yang

draft-tgraf-netconf-notif-sequencing

RFC8639 - Subscribed notifications

draft-tgraf-yang-push-observation-time

RFC8641 - YANG-push

Monitored Yang: ietf-interfaces-02.yang

Scapy mock generator

Draft integrated in the project

JSON supported messages

XML supported messages

XML messages : <subscription-started>

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2022-09-02T10:59:55.32Z</eventTime>
  <sysName xmlns="urn:ietf:params:xml:ns:yang:ietf-notification-sequencing">example-router</sysName>
  <sequenceNumber xmlns="urn:ietf:params:xml:ns:yang:ietf-notification-sequencing">1</sequenceNumber>
  <subscription-started xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notification">
    <id>6666</id>
    <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push"
      xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">ds:operational</datastore>
    <datastore-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push"
      xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">/if:interfaces</datastore-xpath-filter>
    <revision xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push-revision">2014-05-08</revision>
    <message-observation-domain-id xmlns="urn:ietf:params:xml:ns:yang:ietf-distributed-notif">1</message-observation-domain-id>
    <message-observation-domain-id xmlns="urn:ietf:params:xml:ns:yang:ietf-distributed-notif">2</message-observation-domain-id>
    <transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
    <encoding>encode-xml</encoding>
    <periodic xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
      <period>100</period>
    </periodic>
  </subscription-started>
</notification>
```

draft-ahuang-netconf-notif-yang

draft-tgraf-netconf-notif-sequencing

RFC8639 - Subscribed notifications

RFC8641 - YANG-push

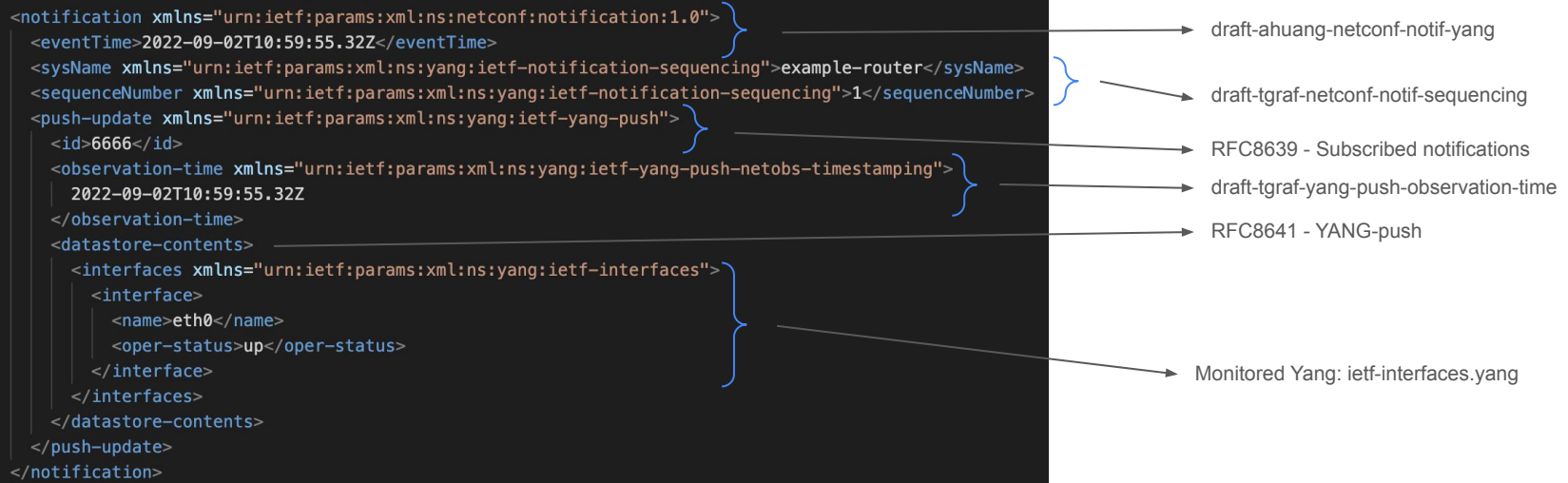
draft-tgraf-netconf-yang-notifications-versioning

draft-ietf-netconf-distributed-notif

RFC8639 - Subscribed notifications

RFC8641 - YANG-push

XML messages : <push-update>



XML messages : <subscription-modified>



XML messages : updated <push-update>

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2022-09-02T10:59:55.32Z</eventTime>
  <sysName xmlns="urn:ietf:params:xml:ns:yang:ietf-notification-sequencing">example-router</sysName>
  <sequenceNumber xmlns="urn:ietf:params:xml:ns:yang:ietf-notification-sequencing">1</sequenceNumber>
  <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <id>6666</id>
    <observation-time xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push-netobs-timestamping">
      2022-09-02T10:59:55.32Z
    </observation-time>
    <datastore-contents>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>eth0</name>
          <oper-status>up</oper-status>
          <packet-count xmlns="urn:ietf:params:xml:ns:yang:ietf-mock-interfaces">150</packet-count>
        </interface>
      </interfaces>
    </datastore-contents>
  </push-update>
</notification>
```

draft-ahuang-netconf-notif-yang

draft-tgraf-netconf-notif-sequencing

RFC8639 - Subscribed notifications

draft-tgraf-yang-push-observation-time

RFC8641 - YANG-push

Monitored Yang: ietf-interfaces-02.yang

How to track a YANG module

Using a periodic subscription to <subscriptions> container



(1.1)



```
<rpc message-id="101">
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <edit-config>
      <target>
        <running/>
      </target>
      <config>
        <subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
          <subscription>
            <id>2222</id>
            <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">ds:operational</datastore>
            <datastore-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push"
              xmlns:s="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"/>sn:subscriptions</datastore-xpath-filter>
            <transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
            <encoding>encode-xml</encoding>
            <receivers>
              <receiver>
                <name>subscription-specific-receiver-def</name>
                <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">global-udp-notif-receiver-def</receiver-instance-ref>
              </receiver>
            </receivers>
            <periodic xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
              <period>30000</period>
            </periodic>
          </subscription>
          <subscription>
            <id>6666</id>
            <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">ds:operational</datastore>
            <datastore-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push"
              xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>if:interfaces</datastore-xpath-filter>
            <transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
            <encoding>encode-json</encoding>
            <receivers>
              <receiver>
                <name>subscription-specific-receiver-def</name>
                <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">global-udp-notif-receiver-def</receiver-instance-ref>
              </receiver>
            </receivers>
            <periodic xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
              <period>6000</period>
            </periodic>
          </subscription>
          <receiver-instances xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">
            <receiver-instance>
              <name>global-udp-notif-receiver-def</name>
              <udp-notif-receiver xmlns="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">
                <address>192.0.5.1</address>
                <port>12345</port>
                <enable-segmentation>false</enable-segmentation>
                <max-segment-size/>
              </udp-notif-receiver>
            </receiver-instance>
          </receiver-instances>
        </subscriptions>
      </config>
    </edit-config>
  </rpc>
```

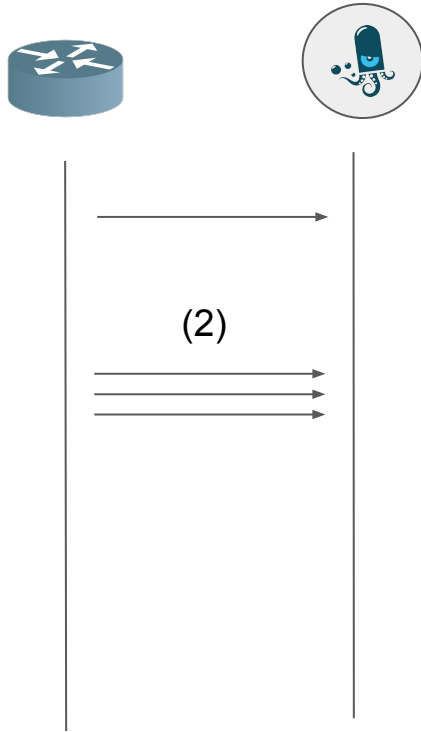
Using a on-change subscription to <subscriptions> container



(1.1)

```
<rpc Message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
        <subscription>
          <id-2222/>
          <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">ds:operational</datastore>
          <datastore-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push"
            xmlns:sn="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"/>sn:subscriptions</datastore-xpath-filter>
          <transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
          <encoding encode="xml">encoding</encoding>
          <receivers>
            <receiver>
              <name>subscription-specific-receiver-def</name>
              <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">global-udp-notif-receiver-def</receiver-instance-ref>
            </receiver>
          </receivers>
          <on-change xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
            <dampening-period>0</dampening-period>
            <sync-on-start>true</sync-on-start>
            <excluded-change>
              <!-- # entries: 0.. -->
            </excluded-change>
          </on-change>
        </subscription>
        <subscription>
          <id-6666/>
          <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">ds:operational</datastore>
          <datastore-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push"
            xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>if:interfaces</datastore-xpath-filter>
          <transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
          <encoding encode="json">encoding</encoding>
          <receivers>
            <receiver>
              <name>subscription-specific-receiver-def</name>
              <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">global-udp-notif-receiver-def</receiver-instance-ref>
            </receiver>
          </receivers>
          <periodic xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
            <period>6000</period>
          </periodic>
        </subscription>
        <receiver-instances xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">
          <receiver-instance>
            <name>global-udp-notif-receiver-def</name>
            <udp-notif-receiver xmlns="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">
              <address>192.0.5.1</address>
              <port>12345</port>
              <enable-segmentation>false</enable-segmentation>
              <max-segment-size>
            </udp-notif-receiver>
          </receiver-instance>
        </receiver-instances>
      </subscriptions>
    </config>
  </edit-config>
</rpc>
```


2 - Sending Telemetry YANG-push to collector



```
{
  "ietf-notification:notification": {
    "eventTime": "2023-03-25T08:30:11.22Z",
    "ietf-yang-push:push-update": {
      "id": 6666,
      "datastore-contents": {
        "ietf-interfaces:interfaces": [
          {
            "interface": {
              "name": "eth0",
              "oper-status": "up"
            }
          }
        ]
      }
    }
  }
}
```

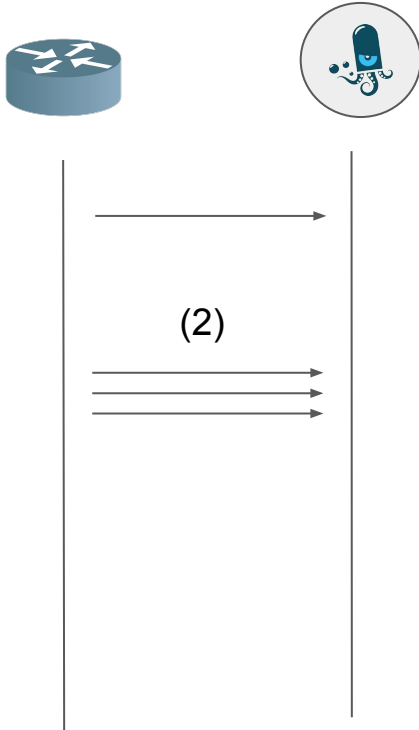
2 - Sending Subscription change periodically



(2)

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2022-09-02T10:59:55.32Z</eventTime>
  <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <id>2222</id>
    <datastore-contents>
      <subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
        <subscription>
          <id>2222</id>
          <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">ds:operational</datastore>
          <datastore-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push"
            xmlns:s="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"/>s:subscriptions</datastore-xpath-filter>
          <transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
          <encoding>encode-xml</encoding>
          <receivers>
            <receiver>
              <name>subscription-specific-receiver-def</name>
              <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">global-udp-notif-receiver-def</receiver-instance-ref>
            </receiver>
          </receivers>
          <periodic xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
            <period>30000</period>
          </periodic>
        </subscription>
        <subscription>
          <id>6666</id>
          <datastore xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">ds:operational</datastore>
          <datastore-xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push"
            xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>if:interfaces</datastore-xpath-filter>
          <transport xmlns:unt="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">unt:udp-notif</transport>
          <encoding>encode-json</encoding>
          <receivers>
            <receiver>
              <name>subscription-specific-receiver-def</name>
              <receiver-instance-ref xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">global-udp-notif-receiver-def</receiver-instance-ref>
            </receiver>
          </receivers>
          <periodic xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
            <period>6000</period>
          </periodic>
        </subscription>
        <receiver-instances xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notif-receivers">
          <receiver-instance>
            <name>global-udp-notif-receiver-def</name>
            <udp-notif-receiver xmlns="urn:ietf:params:xml:ns:yang:ietf-udp-notif-transport">
              <address>192.0.5.1</address>
              <port>12345</port>
              <enable-segmentation>false</enable-segmentation>
              <max-segment-size>
            </udp-notif-receiver>
          </receiver-instance>
        </receiver-instances>
      </subscriptions>
    </datastore-contents>
  </push-update>
</notification>
```

2 - Sending Subscription change on-change



```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2022-09-02T10:59:55.32Z</eventTime>
  <push-change-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <id>2222</id>
    <datastore-changes>
      <yang-patch>
        <patch-id>patch_54</patch-id>
        <comment>Changing encoding to JSON and increasing the period to 10 minutes</comment>
        <edit>
          <edit-id>id_change_1</edit-id>
          <operation>merge</operation>
          <target xmlns:sn="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">/sn:subscriptions/subscription[id=2222]</target>
          <value>
            <encoding>encode-json</encoding>
            <periodic xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
              <period>60000</period>
            </periodic>
          </value>
        </edit>
      </yang-patch>
    </datastore-changes>
  </push-change-update>
</notification>
```

What messages to implement in Scapy (periodic)?

1. <subscription-started> for monitored YANG module (ietf-interfaces.yang)
2. <subscription-started> for monitored YANG-push subscription
3. N messages <push-update> of “ietf-interfaces.yang” messages
4. M messages <push-update> of YANG-push subscription

(simulating here the ietf-interfaces.yang version change)

5. <subscription-modified> for monitored YANG module (ietf-interfaces.yang)
6. N messages <push-update> of new “ietf-interfaces.yang” messages

Assuming we know a reference to the YANG module during the subscription

What messages to implement in Scapy (on-change)?

1. <subscription-started> for monitored YANG module (ietf-interfaces.yang)
2. <subscription-started> for monitored YANG-push subscription
3. N messages <push-update> of “ietf-interfaces.yang” messages

(simulating here the ietf-interfaces.yang version change)

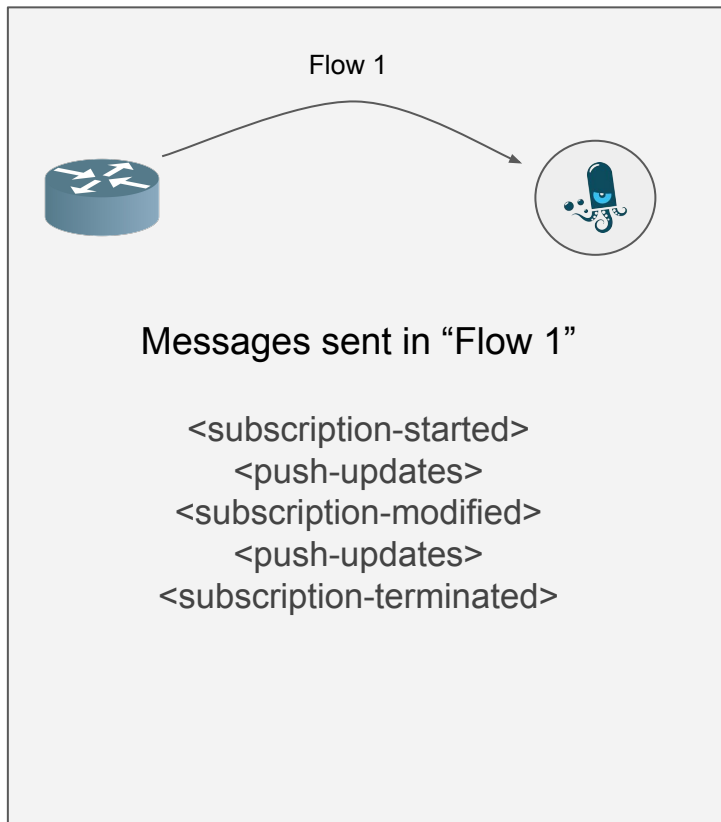
4. <push-change-update> for YANG-push subscription
5. <subscription-modified> for monitored YANG module (ietf-interfaces.yang)
6. N messages <push-update> of new “ietf-interfaces.yang” messages

Assuming we know a reference to the YANG module during the subscription

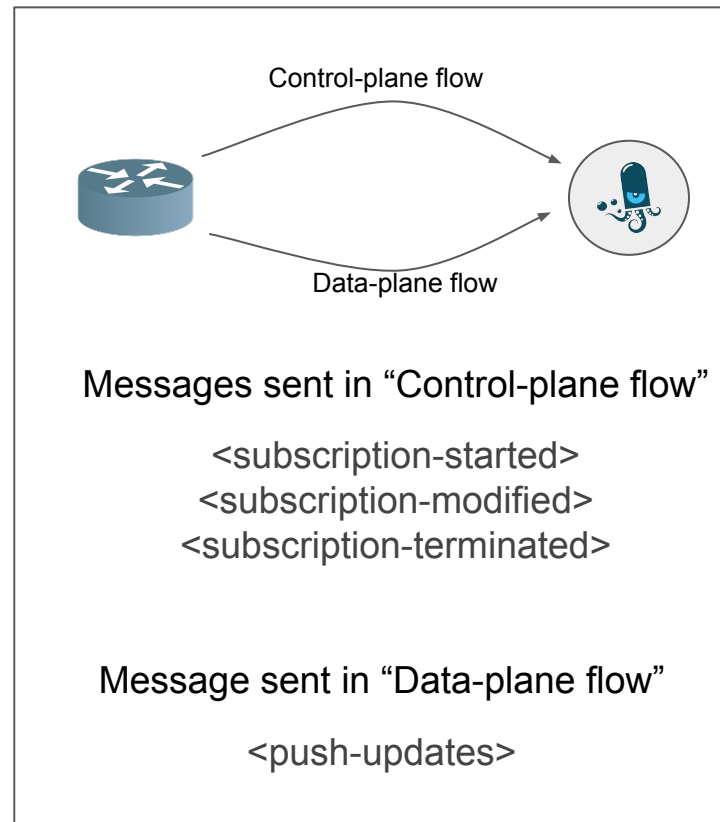
YANG push control-plane messages

- Are <subscription-started> and <subscriptn-modified> sent in the same dataplane flow?

YANG-push flows



VS



Reading RFC8639 (1)

Questions:

1. Section 2.7 (Subscription State Change Notifications)

- “[...] *they (subscription state change notifications) are inserted [...] into the sequence of notification messages sent to a particular receiver.*”
- I understand that these notifications are part of the same event notification flow.
- There is no way to separate these flows in the config
 - Though, **dependencies** between subscriptions are allowed (Section 2.3 QoS)

[2.7.](#) Subscription State Change Notifications

In addition to sending event records to receivers, a publisher **MUST** also send subscription state change notifications when events related to subscription management have occurred.

Subscription state change notifications are unlike other notifications in that they are never included in any event stream. Instead, they are inserted (as defined in this section) into the sequence of notification messages sent to a particular receiver. Subscription state change notifications cannot be dropped or filtered out, they cannot be stored in replay buffers, and they are delivered only to impacted receivers of a subscription. The identification of subscription state change notifications is easy to separate from other notification messages through the use of the YANG extension "subscription-state-notif". This extension tags a notification as a subscription state change notification.

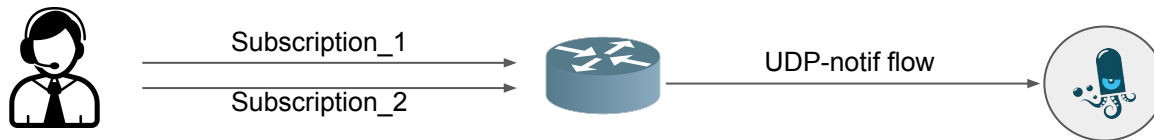
The complete set of subscription state change notifications is described in the following subsections.

Reading RFC8639 (2)

Questions:

2. Section 2.5 (Configured Subscription)

- “[...] *Multiple configured subscriptions MUST be supportable over a single transport session.*”
- This means multiple subscriptions can send messages over the same notification flow?
- Use case: 2 subscriptions sending to the same transport UDP-notif



Agree this is not an issue?

Distributed notifications

Draft

ietf-netconf-distributed-notif

What is the usecase of the draft?

Adding observation domain id to YANG-push messages

- Adds the list of observation domain ID to the to the subscription container and YANG-push notifications.
 - It references **draft-ietf-netconf-notification-messages** for the *observation-domain-id* definition
 - Type: string
 - The observation-domain-id in UDP-notif is an integer of 32 bits.
-
- Q1: Should this YANG module definition changed to an integer?
 - Q2: What it is the use of this draft?
 - Q3: Should this be added to scapy?

9. YANG Tree

```
module: ietf-distributed-notif
  augment /sn:subscriptions/sn:subscription:
    +--ro message-observation-domain-id*  string
  augment /sn:subscription-started:
    +--ro message-observation-domain-id*  string
  augment /sn:subscription-modified:
    +--ro message-observation-domain-id*  string
  augment /sn:establish-subscription/sn:output:
    +--ro message-observation-domain-id*  string
```

Gaps

Something to be done?

References

- YANG-push
 - RFC8639: Subscription to YANG Notifications
 - RFC8641: Subscription to YANG Notifications for Datastore Updates
- Definition of a Datastore:
 - RFC 8342: Network Management Datastore Architecture (NMDA)
- Encoding:
 - RFC7950: The YANG 1.1 Data Modeling Language (for XML)
 - RFC7951: JSON Encoding of Data Modeled with YANG
 - RFC9254: Encoding of Data Modeled with YANG in CBOR
- YANG-push RPCs and notifications in XML:
 - <https://github.com/network-analytics/udp-notif-scapy/tree/feature/add-yang-push/src/resources/xml> (will be merged)
- YANG-push notification in JSON:
 - <https://github.com/network-analytics/udp-notif-scapy/tree/feature/add-yang-push/src/resources/json/notifications> (will be merged)

Raised issues

- Architecture: 1.2 configure kafka schema registry is between the router and kafka instead of the operator.
- `<get-schema>` is recursive getting all the dependent YANG modules
 - Represent also that we need to create a yang module from it having all the notification headers
- Can we have a subscription subscribed to multiple xpath?
- Ask Netconf WG about the xml namespace, how in YANG module name is encoded in the xpath
- `<subscription-modified>`: Thomas says this message is not in the same flow as `<push-update>`; if it is, the subscription in the subscription container should send the notifications first in a modification in a subscription
- Distributed-notif draft should be added somewhere in the slide deck

Remarks

- One stream is a set of datastore changes
- If we want to change a subscription, it is better to terminate the subscription and create a new one
- No opinion in “on-change” subscription yet
- Scapy implementation: think about not only changing the version, but changing the xpath we are subscribed to