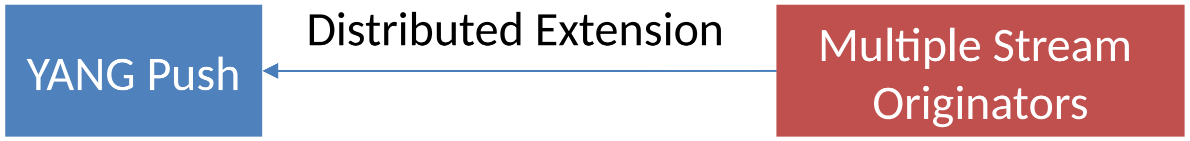
Subscription to Multiple Stream Originators

draft-zhou-netconf-multi-stream-originators

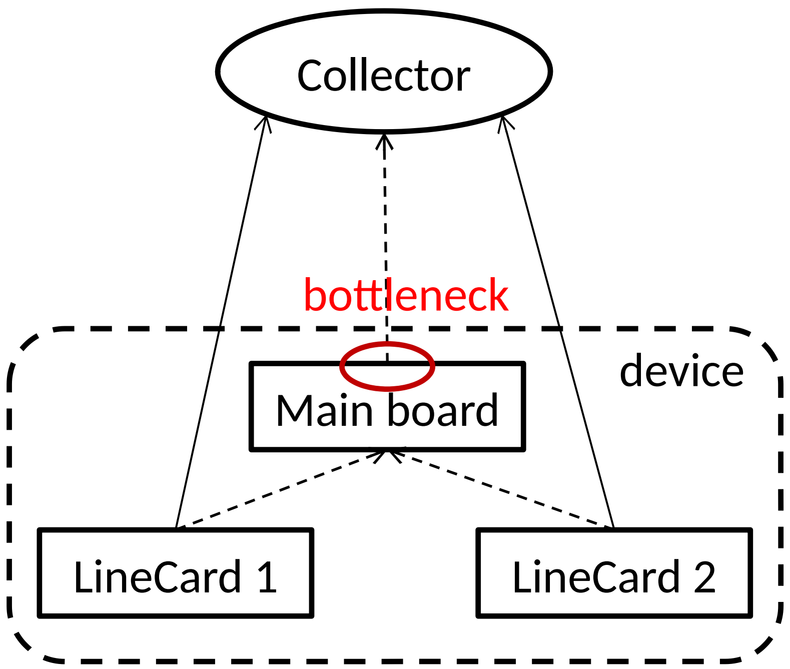
Tianran Zhou
Guangying Zheng
Eric Voit
Alexander Clemm
Andy Bierman

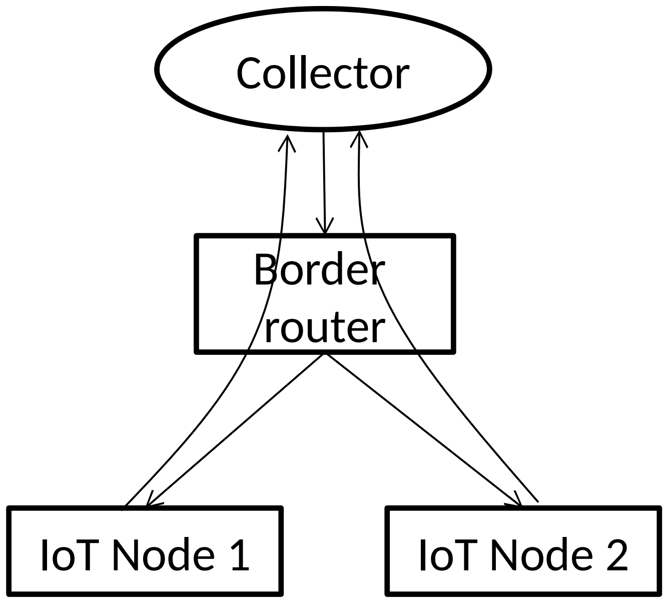
Introduction

- **Distributed data export** mechanism that allows multiple data streams to be managed using **a single subscription**.
- Transport independent



Two Use Cases





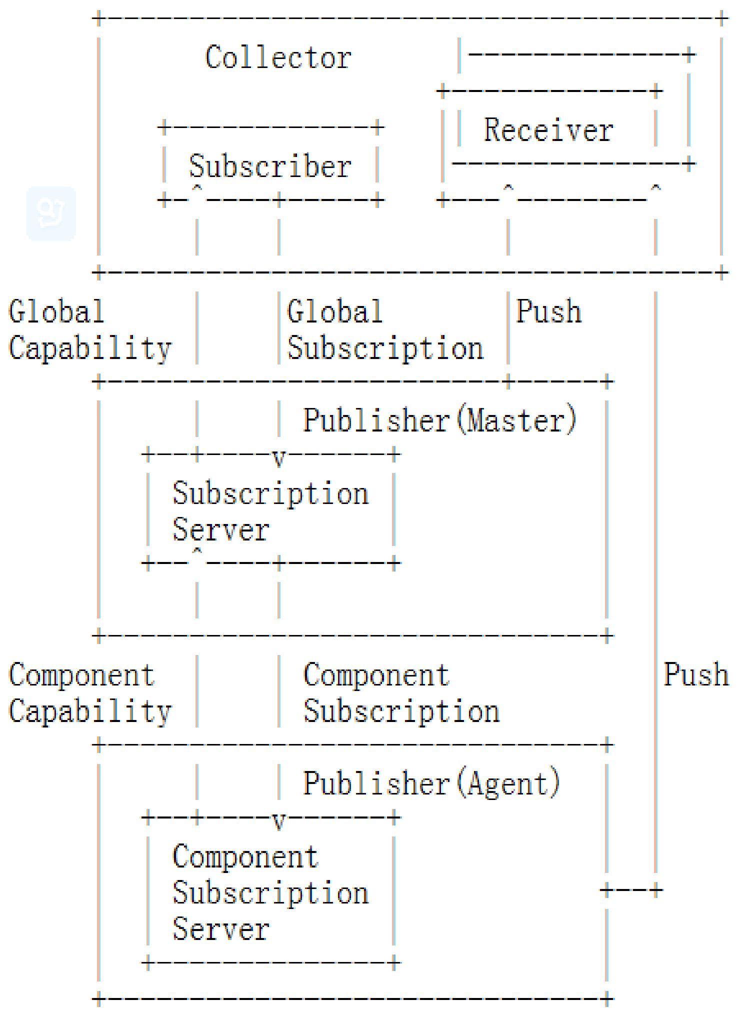
- directly push data from line cards to a collector.

- The border router does not assemble data as a broker.

Discussion: Is it helpful to include the IoT use case in the draft?
This case requires node management protocols in addition to push mechanism.

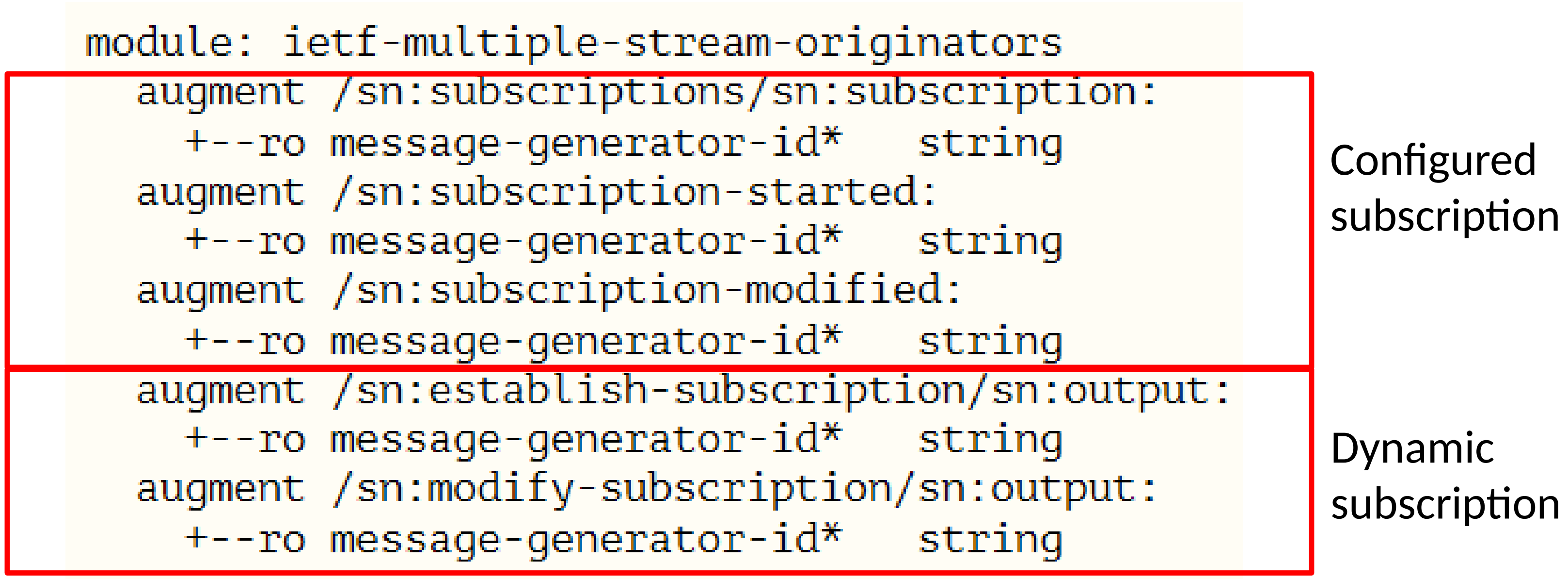
Solution Overview

- Collector
  - Subscriber
  - Receiver
- Distributed Publisher
  - Master with the Subscription server
  - Agent with the Component subscription server
- Mechanism
  - Subscription decomposition
  - Publication composition
  - Subscription State Change Notifications



Extensions for Publication Composition

- Receiver need to know the **number of Component Subscri ptions** which the Global Subscription is decomposed to.
  - **Propose to add a list of Publisher ID**

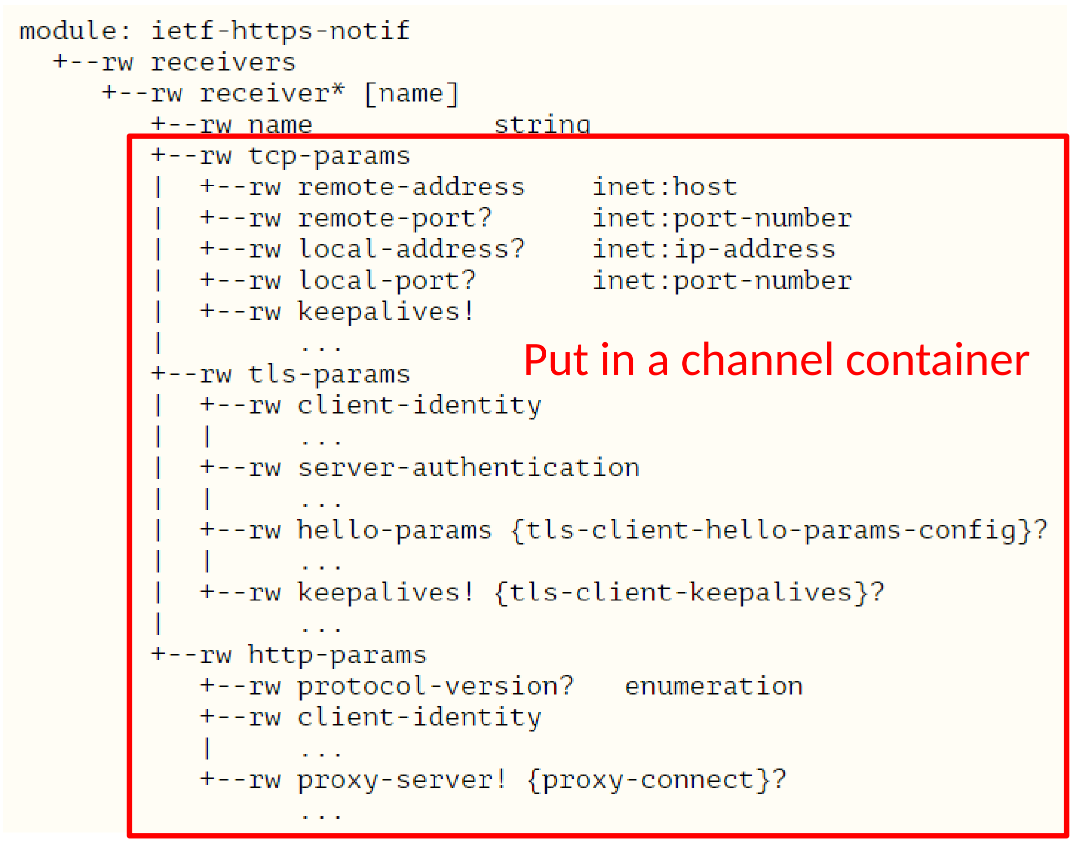


```
module: ietf-multiple-stream-originators
  augment /sn:subscriptions/sn:subscription:
    +--ro message-generator-id*   string
  augment /sn:subscription-started:
    +--ro message-generator-id*   string
  augment /sn:subscription-modified:
    +--ro message-generator-id*   string
  augment /sn:establish-subscription/sn:output:
    +--ro message-generator-id*   string
  augment /sn:modify-subscription/sn:output:
    +--ro message-generator-id*   string
```

Extensions for Configured Subscription

```
+--rw subscription* [id]
      +--rw id
      |     subscription-id
      ...
      +--rw transport?
      |     {configured}?
      +--rw encoding?
      ...
      +--rw receivers
        +--rw receiver* [name]
          +--rw name
          |...
```

A list of channel configurations



```
module: ietf-https-notif
  +--rw receivers
     +--rw receiver* [name]
        +--rw name                   string
        +--rw tcp-params
        |  +--rw remote-address     inet:host
        |  +--rw remote-port?       inet:port-number
        |  +--rw local-address?     inet:ip-address
        |  +--rw local-port?        inet:port-number
        |  +--rw keepalives!
        |        ...
        +--rw tls-params
        |  +--rw client-identity
        |  |     ...
        |  +--rw server-authentication
        |  |     ...
        |  +--rw hello-params {tls-client-hello-params-config}?
        |  |     ...
        |  +--rw keepalives! {tls-client-keepalives}?
        |        ...
        +--rw http-params
           +--rw protocol-version?   enumeration
           +--rw client-identity
           |     ...
           +--rw proxy-server! {proxy-connect}?
                 ...
```

Put in a channel container

draft-mahesh-netconf-https-notif

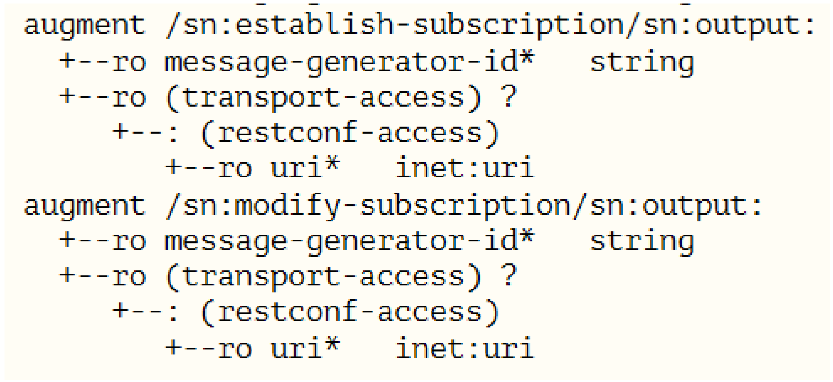
All the potential channels are preconfigured. Actual publication channels are selected based on the subscription decomposition result.

Extensions for Dynamic Subscription

- Several transport options:
  - The line-card runs on server mode(A) or client mode(B)?
  - The connections are dynamically set up(C) or pre-configured(D)?

Option 1: Generalize
draft-ietf-netconf-restconf-notif

Mode: A+C
RPC return the resource access information
Receiver get data from the linecards



```
augment /sn:establish-subscription/sn:output:
  +--ro message-generator-id*   string
  +--ro (transport-access) ?
     +--: (restconf-access)
        +--ro uri*   inet:uri
augment /sn:modify-subscription/sn:output:
  +--ro message-generator-id*   string
  +--ro (transport-access) ?
     +--: (restconf-access)
        +--ro uri*   inet:uri
```

Option 2: consistent with
configured subscription

Mode: B+D
All the channels are preconfigured.
Just push when subscription request
accepted.

Next

- Improve examples. What kind of example is ex pected?
- Any other issues need to consider for this distr ibuted extension of the YANG-Push work?

Thank you