Subscription to Multiple Stream Originators

draft-zhou-netconf-multi-stream-originators

Tianran Zhou

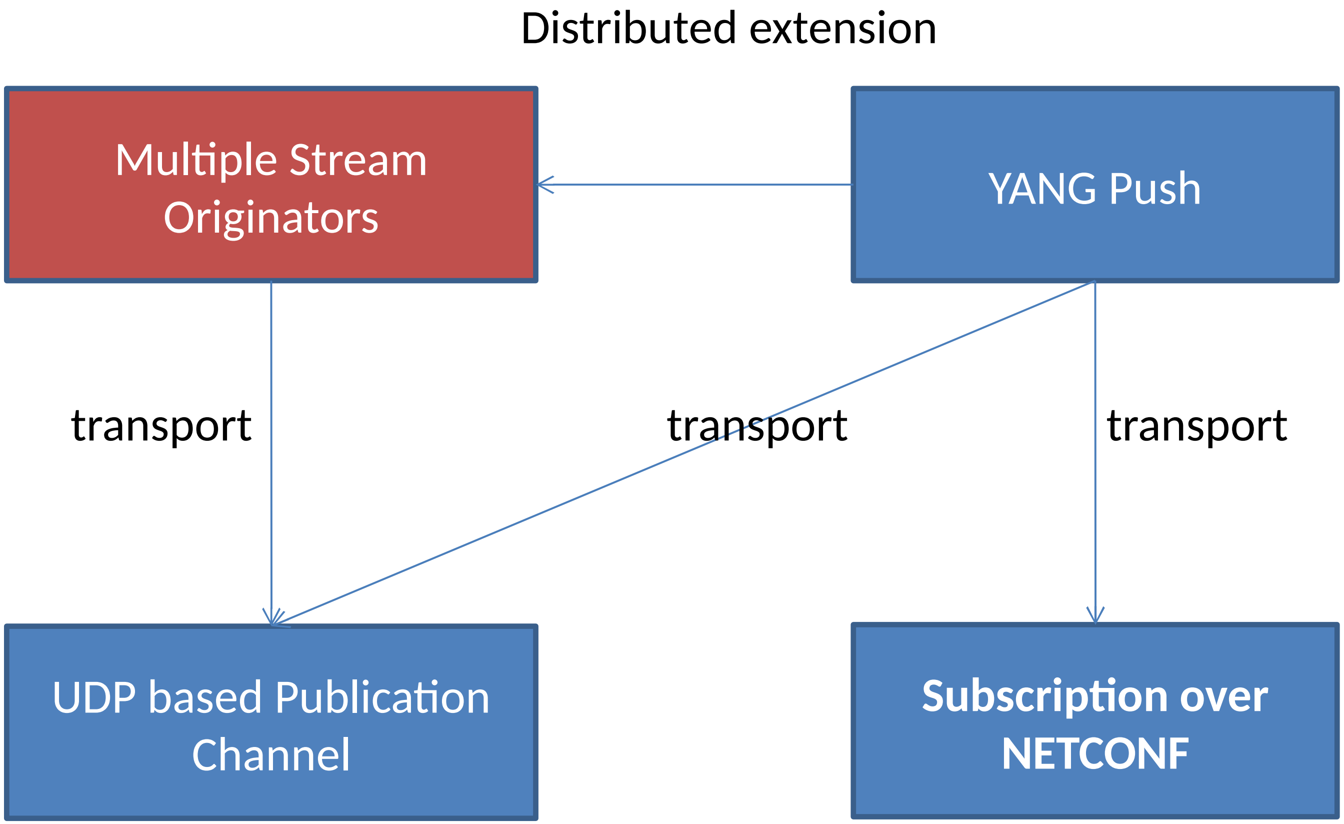
Guangying Zheng

Eric Voit

Alexander Clemm

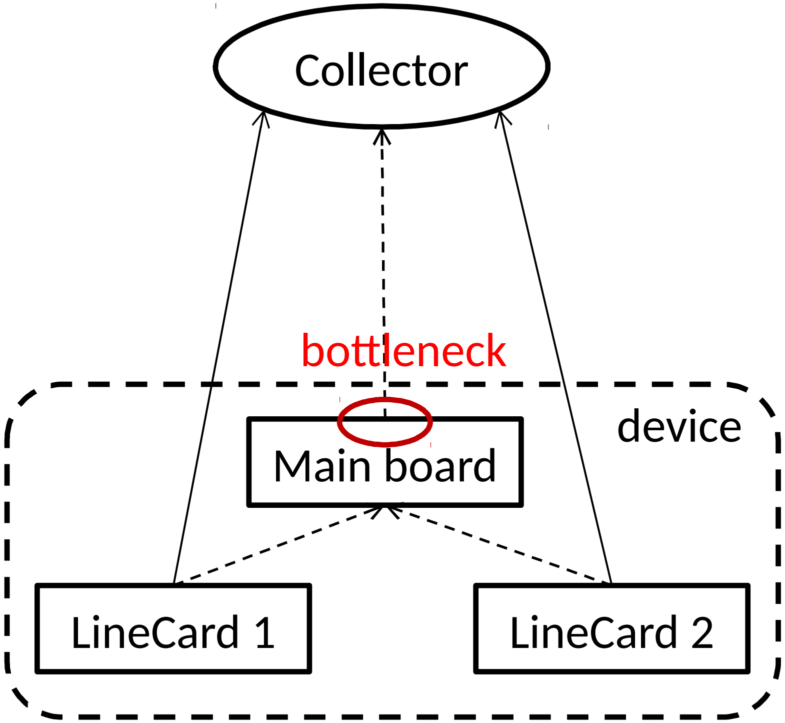
Andy Bierman

Draft Dependencies


Distributed extension
Multiple Stream Originators
YANG Push
transport
transport
transport
UDP based Publication Channel
Subscription over NETCONF

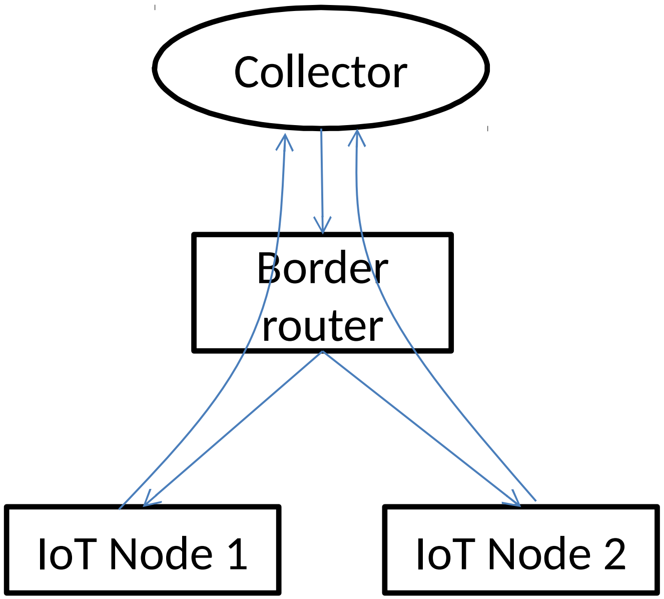
Use Case 1

- Large amount of data collection from devices with main board and line cards.
- Existing solution consider only one push server reside in the main board.
  - Result in performance bottleneck when data are forwarded to the main board and converged to one consolidated stream.
- Request for **distributed data collection mechanism** which can directly push data from line cards to a collector.



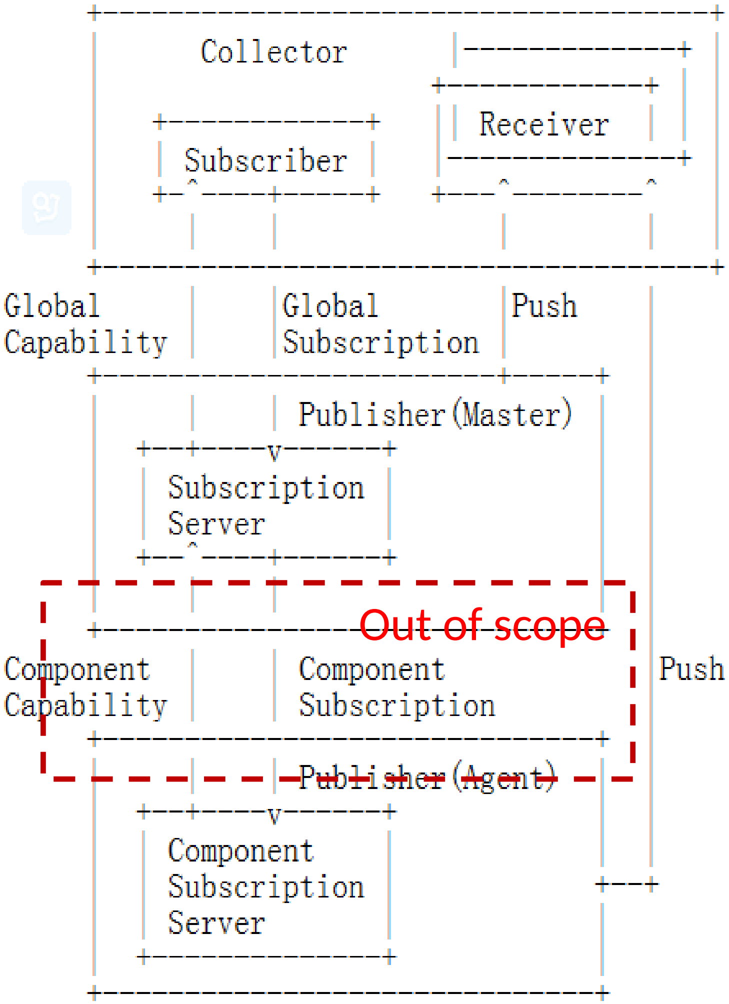
Use Case 2

- Collector cannot subscribe/access data directly from IoT nodes.
  - subscribe data from border router
  - border router distribute the subscription to Nodes.
  - IoT Nodes stream data to the collector through BR.
  - Collector assembles the subscription data.



Solution Overview

- Publisher
  - Two roles: master and agent
- Interactions between the Master and Agents
  - Agents need to have a registration or announcement handshake with the Master, so the Master is aware of them and of life-cycle events.
  - Contracts are needed between the Master and each Agent on the Component Capability, and the format for streaming data structure.
  - The Master relays the component subscriptions to the Agents.
  - The Agents indicate status of Component Subscriptions to the Master. The status of the overall subscription is maintained by the Master.



Subscription Decomposition

- The Master
  - expose the Global Capability that can be served by multiple Publishers;
  - disassemble the Global Subscription to multiple Component Subscriptions, and distribute them to the corresponding telemetry sources;
  - notify on changes when portions of subscription moving between different Agents over time.
- The Master may need a data structure, typically a Resource-Location Table, to keep track of the mapping between the resource and the corresponding location of the Subscription Server which commits to serve the data.

Publication Composition

- The Receiver recognizes data records associated with one subscription according the **Subscription ID**.
- The Receiver assembles data generated at the same time period based on the **recording time** consisted in each data record.
- Receiver need to know the **number of Component Subscriptions** which the Global Subscription is decomposed to.
  - **Propose to add a list of Publisher ID**
  - The response of the "establish-subscription" and "modify-subscription“
  - The "subscription-started" and "subscription-modified" notification

Subscription State Change Notifications

- In addition to sending event records to receivers, the Master MUST also send subscription state change notifications when events related to subscription management have occurred.
- All the subscription state change notifications MUST be delivered by the **Master Publication Channel** which is the session between the Master Publisher and the Receiver.
- When the subscription decomposition result changed, the "subscription-modified" notification MUST be sent to indicate the new list of Publishers.

Next

- Add the YANG Model to show how to extend t he existing Subscribed Notification Model.
- Any other issues need to consider for this distr ibuted extension of the YANG-Push work?
- Ask for the WG Adoption.

Thank you