Adaptive Subscription to YANG Notification

draft-wang-netconf-adaptive-subscription-09

Qin Wu (Huawei)

Wei Song (Huawei)

Peng Liu (CMCC)

Qiufang Ma (Huawei) Presenter

Wei Wang (China Telecom)

Recap

- Motivation and Goal
  - A high frequency data collection leads to more resource consumption while low frequency data collection is insufficient for fault localization.
  - To address challenge in massive data collection and processing and look for balance between expensive data management cost and data fidelity for troubleshooting.
- Main idea
  - To install subscription policy built on top of YANG-PUSH mechanism and allow servers switch to different update intervals based on network condition changes.
- Document status
  - first proposed in March 2020 and presented several times in NETCONF session and received a lot of comments
    - Thanks Andy Bierman, Kent Watsen, Rob Wilton, Thomas Graf, Michael Richardson, Alex Clemm, Mahesh, etc
  - The WG adoption call was initiated in 10 January 2022 and we got a lot of supports
    - Thanks Peng Liu, Wei Wang, Zheng Qiang Li, Chongfeng Xie, Hongwei Li, Yongbo Liu, Aijun Wang, Zhixiong Niu, Kun Xie, Ren Gang, Chunshan Xiong
    - However we also received concerns from Andy Bierman and Per Andersson
      - Use of watermark, evaluation of xpath-external-eval, stateless xpath, xpath instance information, conflicts in multiple adaptive-period entries

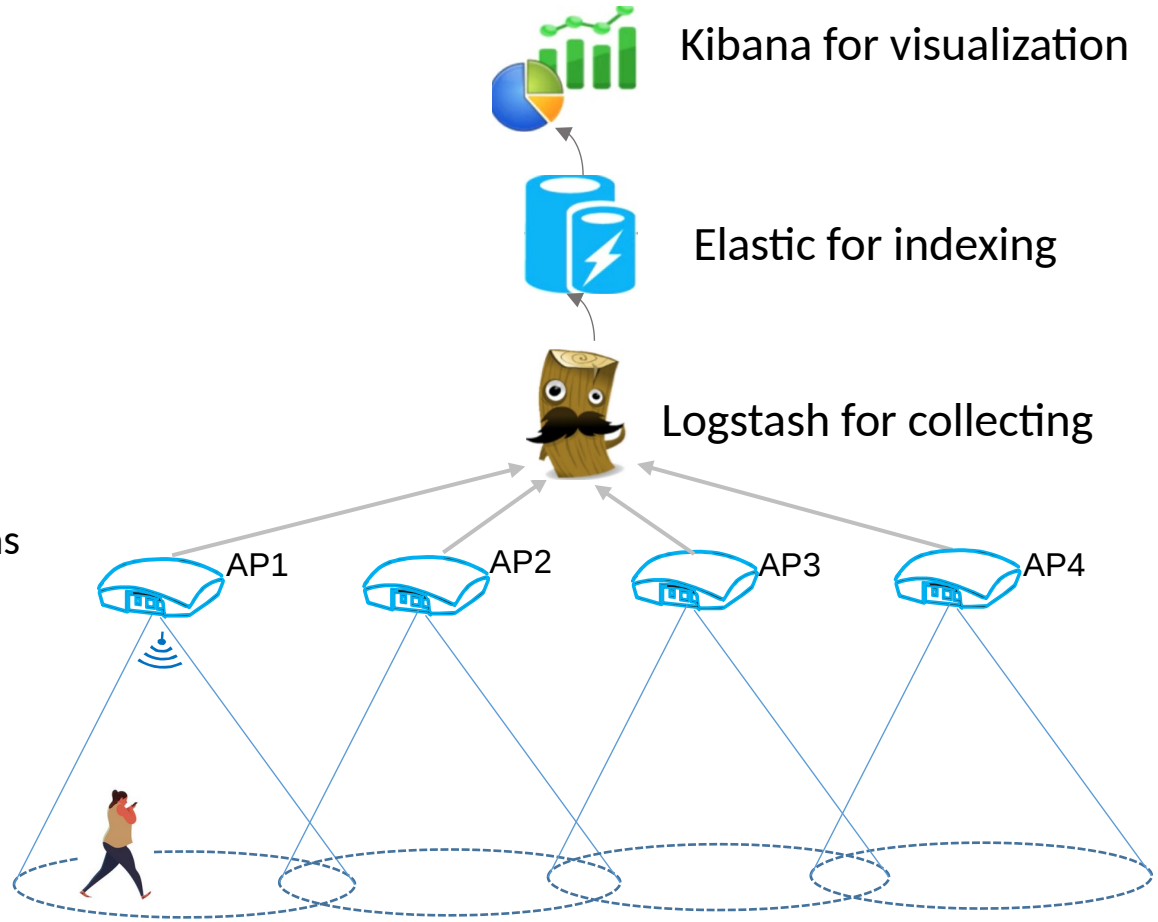
Since IETF112

- Document Changes since -07
  - Define three **new RPC errors** to report when adaptive subscription unsupported, an Xpath syntax defined in “xpath-external-eval” unsupported, and **multiple XPath criteria conflict**, respectively.
  - **Remove the "watermark"** parameter.
  - Add **clarification about how to evaluate the XPath expression** defined in "xpath-external-eval".
  - Add **clarification about how to compare** a targeted data object **in a specific list entry**.
  - Add **clarification about how often** does the server check if the period should change.
  - Add a new example showing how the RPC error being returned by a publisher.
  - The usage examples fixed in the Appendix.
  - Grammatical errors correction(missing articles, plurality mismatches, etc).
- An IETF113 Hackathon project is proposed to provide implementation results on performance evaluation.
  - Monitor KPI changes at different frequency of data collection (high frequency, low frequency, adaptive frequency)
  - Evaluate the performance of adaptive subscription (e.g., telemetry data volume)

Hackathon—Test Environment setup

- gRPC-based telemetry to collect data from Access Points in our campus;
  - The following data collection methods are evaluated：
    - A high-frequency periodic telemetry
    - A low-frequency periodic telemetry
    - An adaptive-frequency telemetry
  - For each data collection method, two cases are evaluated:
    - One is to report the rssi values so as to detect real-time WIFI roaming across different APs(Access points).
    - The other is to stream the bytes sent from the AP uplink so as to detect the possible uplink congestion
- ELK is used to collect, analyze, filter and visualize data.
  - The acronym for three open source tools: Elasticsearch, Logstash, and Kibana
  - The Huawei plugin for ELK to collect and process information from Huawei devices has been developed and open sourced

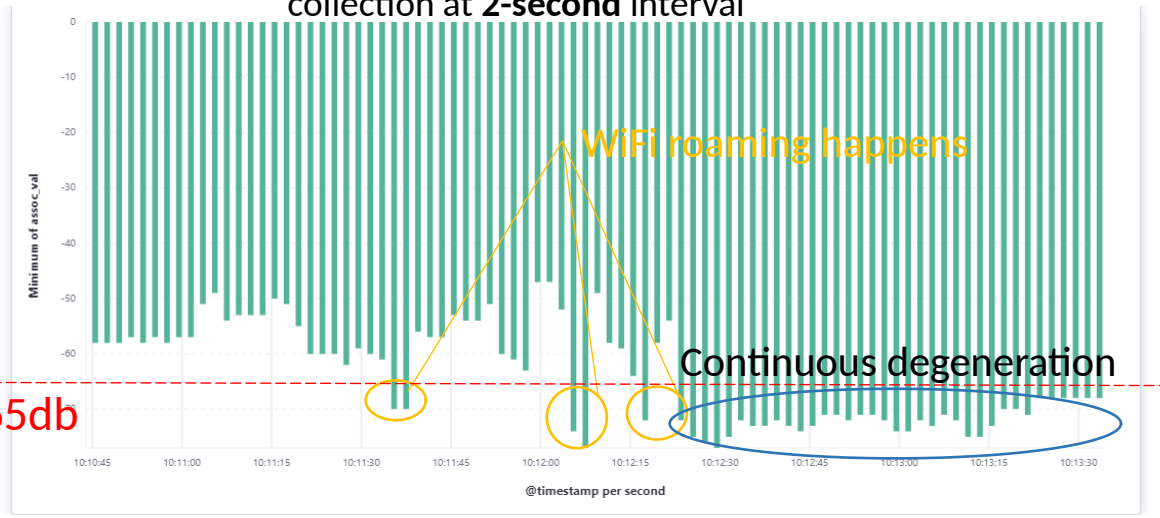
Code: https://github.com/HuaweiDatacomm/elk-huawei-plugin



Network Scenario

Hackathon—RSSI signals data streaming

Continuous high-frequency data collection at **2-second** interval



Adaptive-frequency; **condition evaluated by the subscriber**

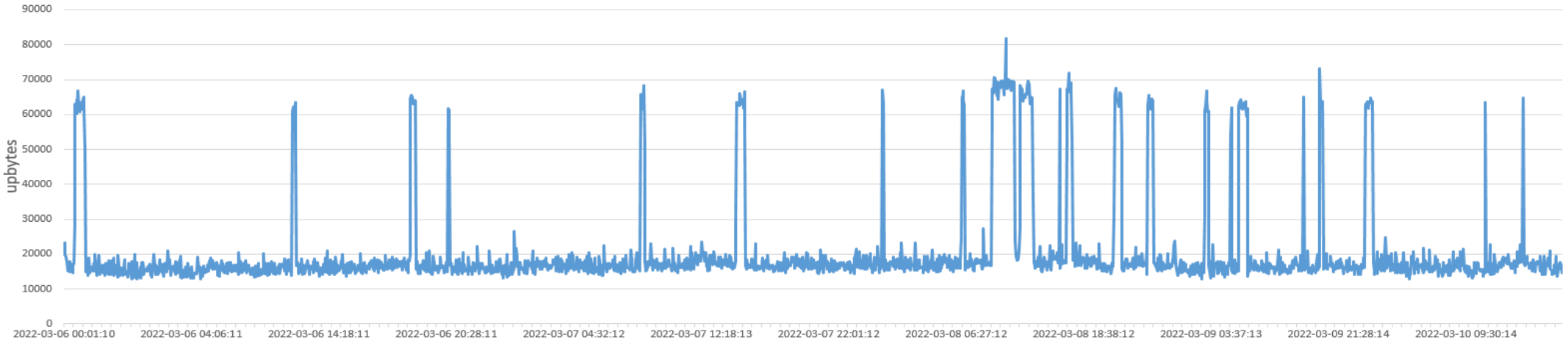
Continuous low-frequency data collection at **30-second** interval

Streaming data at a fixed period.

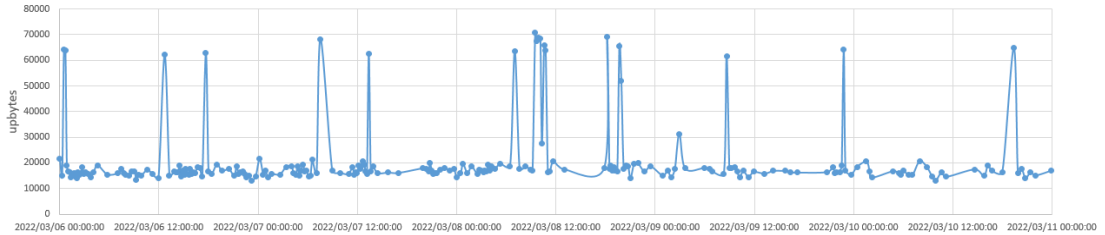
Adaptive-frequency; **condition evaluated by the server**

Period set to be every 2 seconds if the rssi value < -65dB;
If the rssi value >= -65dB, switch to 30 seconds period value.

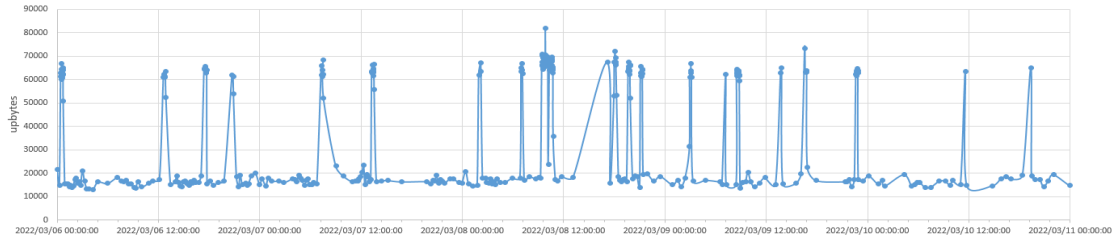
Hackathon—upbytes statistics streaming



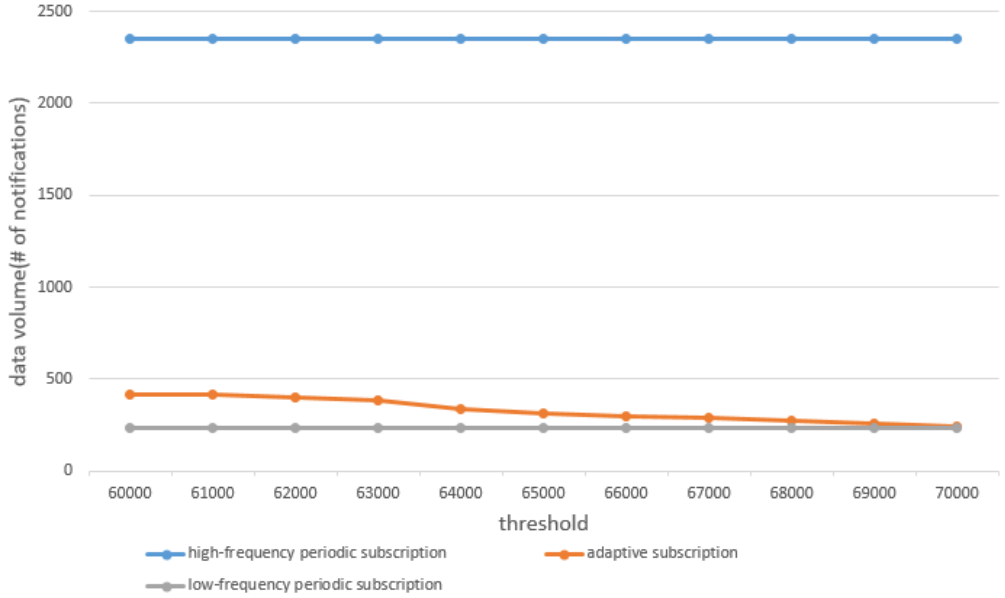
High-frequency periodic subscription at 1-min interval within about 5 days



Low-frequency periodic subscription at 10-min interval



If the upbytes < 60000 bytes, stream data at 10-min interval;
if the upbytes >= 60000 bytes, stream data at 1-min interval



Received number of notifications for different collection methods

Issues clarification

- YANG-PUSH supports adaptive subscription already
  - It's true that the subscriber can monitor the targeted data object change and modify the period for an existing subscription, but
    - More communication/computation/storage resource are consumed
      - From telemetry data collection
      - To modification operations sending
    - Sending modification request to switch from low frequency to high frequency streaming will experience service discontinuity
      - slow to response event and some data are lost since it can only polls at that low frequency.
      - It is late to instruct the device to report the telemetry data since insufficient data for troubleshooting has already occurred.
    - When tens of thousands of network devices need to be managed, frequent modifications are prone to errors.

- How often does the server check if the period should change?
  - The targeted object can be evaluated in the returned node set at the end of each high-frequency streaming update period.
  - To reduce the frequency of evaluation, the server can choose to check targeted object change at every multiple (e.g.,2 or 3) update periods.

Issues clarification (cont.)

- XPath instance information
  - Q: “How to select no more than one entry if there are list instances to handle?”
  - Xpath can be used to identify a particular instance, e.g., to represent a comparison for a leaf in a list entry:

    /if:interfaces/if:interface[if:name="eth0"]/if:in-errors>1000

- XPath is not stateful, the problem statement seems to be limited to absolute values of specific leaf or leaf-list. Usually at least a rate (e.g., two values retrieved at a known time interval) is needed.
  - XPath 1.0 supports + operator to perform addition, - operator to perform subtraction, div operator to perform division, etc.
  - a rate (e.g., two values retrieved at a known time interval) can be represented using XPath with various numeric operators such as

    /bookstore/book[title="learning XML"]/price div /bookstore/book[title="Learning XML"]/numbers
    - In this case, two data objects are required for XPath evaluation

    /bookstore/book[title="learning XML"]/price div 100
    - In this case, one data object is sufficient for XPath evaluation
  - The current draft focuses on smart filter case, e.g., monitored data object exceeds a specific threshold
    - In this case, a single data object is sufficient
  - Another design consideration, is to define a data node which has already performed numeric operation or hide mathematic operation, e.g., a data node with average value as output

Comments, Questions, Concerns?