A Broker-based Pub/Sub System for NDN

Haksuh Kim, Sung-Hyuk Byun, Namseok Ko, and Sun-Me Kim

Electronics and Telecommunications Research Institute (ETRI)
{tiple, shbyun, nsko, kimsunme}@etri.re.kr
Dec. 1, 2020
Motivations

▸ A few Pub/Sub mechanisms for current NDN
  – Partial-sync mode of Psync
  – Periodic requests with long-lived Interests

▸ Not scalable and limited especially in low-performance IoT producers

▸ Not flexible as in IP-based mechanisms: no wildcard topic matches

▸ We need a flexible and scalable pub/sub architecture for NDN
Design Directions

▸ To cope with the issues on low-performance producers
  – Broker-based approach

▸ To support scalability
  – Multiple brokers

▸ To support flexibility
  – MQTT-like wildcard Topic matches
    • Single-level wildcard (+), e.g., /prefix/etri/7d/+/temp
    • Multi-level wildcard (#), e.g., /prefix/etri/7d/#
    • Multiple wildcards can be supported in a Topic
  ※ Topics are defined by subscribers
Multiple brokers (or Rendezvous Nodes)
  - Do brokering of publishers and subscribers
  - Store published data for limited performance publishers
    ※ Note that data can be stored in devices themselves and other external repositories
  - Manage published data (names) based on DHT
Architecture Overview

Logical separation of Topic and Data management

**Topic Tree**

<table>
<thead>
<tr>
<th>Topic Name</th>
<th>RN</th>
</tr>
</thead>
<tbody>
<tr>
<td>/topic-2</td>
<td>RN-1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Data Management**

<table>
<thead>
<tr>
<th>Topic Name</th>
<th>RN</th>
</tr>
</thead>
<tbody>
<tr>
<td>/rn-1</td>
<td>RN-1</td>
</tr>
<tr>
<td>/rn-2</td>
<td>RN-2</td>
</tr>
<tr>
<td>/rn-3</td>
<td>RN-3</td>
</tr>
<tr>
<td>/rn-4</td>
<td>RN-4</td>
</tr>
</tbody>
</table>

**Pub-1**

**Pub-2**

**Pub-3**

**Sub-1**
Architecture Overview

- Logical separation of Topic and Data management
  - A Data RN is an RN for data management and a Topic RN is an RN for topic management
  - Pub/sub service has a **service prefix**, `rn` and each RN has its own **node prefix**, `rn-x`

- Naming scheme
  - Data
    - `/<data stream name>/[sequence-number]`: e.g., `/etri/7d/room385/temp/1`
  - Command
    - `/<service prefix>/<command>/<data name>`: e.g., `/rn/PA/etri/7d/room385/temp`
Protocol Messages (Commands)

▸ Publish Procedure
  – Publish Advertisement (PA): Advertise the name of a data stream to publish
  – Publish Unadvertisement (PU): Revoke the publish of a data stream
  – Publish Data (PD): publish a data

▸ Subscribe Procedure
  – Subscribe Topic Subscription (ST): Subscribe to a topic (request a topic manifest)
    * Topic manifest: a list of data RN holding subscribed data streams
  – Subscribe Manifest Request (SM): Request a data manifest (to a specific Data RN)
    * Data manifest: data names for a data stream
  – Subscribe Data Request (SD): Request a data (to a specific Data RN)
Publish Advertisement

Producer (Pub-2)  
Publish Adv. (I:/rn/PA/topic-1/a [PS])

Data-RN (RN-2)  
Find Topic-RN
DHT: hash(topic-1) → RN-4

Topic-RN (RN-4)  
Publish Adv. (I:/RN-4/PA/topic-1/a [RN-2])

Insert Topic Tree /topic-1/a : RN-2

Res. Publish Adv. (D:/rn/PA/topic-1/a [OK])

Res. Publish Adv. (D:/RN-4/PA/topic-1/a [OK])
Publish Unadvertisement

Producer (Pub-2)

Data-RN (RN-2)

Topic-RN (RN-4)

Publish Unadv. (l:/rn/PU/topic-1/a [PS])

Find Topic-RN
DHT : hash(topic-1) → RN-4

Publish Unadv. (l:/RN-4/PU/topic-1/a [RN-2])

Delete Topic Tree

Res. Publish Unadv. (D:/rn/PU/topic-1/a [OK])

Res. Publish Unadv. (D:/RN-4/PU/topic-1/a [OK])
Publish Data

Data are stored and data manifest files for the topics are updated in each RN
Publish Data (small data)

Producer (Pub-1) → Data-RN (RN-1) → Topic-RN (RN-1)

Data Publish (I:/rn/PD/topic-2/1 [val])

Store Data - /topic-2/1 : val
Update Manifest

Res. Data Publish (D:/rn/PD/topic-2/1)
Publish Data (large data)

Producer (Pub-1) → Data-RN (RN-2) → Topic-RN (RN-4)

- Data Publish (I:/rn/PD/topic-1/a/1 [PS])
- Request Data (I:/topic-1/a/1)
- Send Data (D:/topic-1/a/1)
- Store Data - /topic-1/a/1 : val
- Update Manifest
Topic Subscription

**Topic Tree**

<table>
<thead>
<tr>
<th>Topic Name</th>
<th>RN</th>
</tr>
</thead>
<tbody>
<tr>
<td>/topic-2</td>
<td>RN-1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Data management**

**Topic Subs.**

/bridge/rn/ST/topic-1/#

**Data Manifest**

Data

**Topic Manifest**

/topic-1/a : RN-2
/topic-1/b : RN-4
Topic Subscription

- Consumer (Sub-1)
- Data-RN (RN-3)
- Topic-RN (RN-4)

- Find Topic-RN
  DHT: hash(topic-1) \(\rightarrow\) RN-4

- Topic Subs. (I:/rn/ST/topic-1/#)
- Topic Subs. (I:/RN-4/ST/topic-1/#)

- Res. Topic Subs. (D:/rn/ST/topic-1/# [C])
- Res. Topic Subs. (D:/RN-4/ST/topic-1/# [C])

- Topic Manifest
  /topic-1/a : RN-2
  /topic-1/b : RN-4
Data Manifest Request

- **Consumer (Sub-1)**
  - Data Mani. Req. (I:/RN-2/SM/topic-1/a)
  - Res. Data Mani. Req. (D:/RN-2/SM/topic-1/a [C])

- **Data-RN (RN-2)**
  - Data Manifest
  - Data name : /topic-1/a
  - Startseq:1
  - Lastseq:3

- **Data-RN (RN-4)**
  - Data Mani. Req. (I:/RN-4/SM/topic-1/b)
  - Res. Data Mani. Req. (D:/RN-2/SM/topic-1/b [C])
  - Data Manifest
  - Data name : /topic-1/b
  - Startseq:1
  - Lastseq:5
Data Request

Topic Tree

<table>
<thead>
<tr>
<th>Topic Name</th>
<th>RN</th>
</tr>
</thead>
<tbody>
<tr>
<td>/topic-2</td>
<td>RN-1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Topic management

Data management

Data Req. /rn-2/SD/topic-1/a/1

Data Req. /rn-5/SD/topic-1/b/1

Pub-1

Pub-2

Pub-3

Sub-1
Data Request

Consumer (Sub-1) ---- Data-RN (RN-2) ---- Data-RN (RN-4)

Data Req. (I:/RN-2/SD/topic-1/a/3)

Res. Data Req. (D:/RN-2/SD/topic-1/a/3 [val])

Data name: /topic-1/a/3
Val: 1000

Data Req. (I:/RN-4/SD/topic-1/b/4)

Res. Data Req. (D:/RN-4/SD/topic-1/b/4 [val])
Demo
Summary

- Broker-based Pub/Sub for NDN
- Scalable and flexible than existing mechanisms
- Plan to release our code as open source software soon
Thank You!