Hierarchical Storage for Building Management System

Zhehao Wang[1], Jiayi Meng[2], Jeff Burke[3]
[123]UCLA REMAP

Introduction

Building management system (BMS) is a sensor data acquisition system which automatically manages a building's heating, ventilation and air conditioning, and other systems.
An NDN based BMS leverages the architecture's advantages in hierarchical data naming and name-based routing and forwarding, in-network caching, and inherent security support, and may overcome the challenges IP faced, namely complexity of network addressing and configuration, reliance on middleware, and a lack of security [1].
This summer's work focuses on the data aggregation and signing/verification in NDN BMS, and updates the previous work by Wentao.

Design Objectives

- Hierarchical storage and aggregation

Objective: Design a hierarchical storage approach and a stream-based approach to calculaIng aggregates, distribuIng processing and taking advantage of local storage.

- BMS data signing and verification

Design Overview

Figure 1 illustrates the namespace of the BMS application, and Figure 2 illustrates an example name of a piece of BMS data.

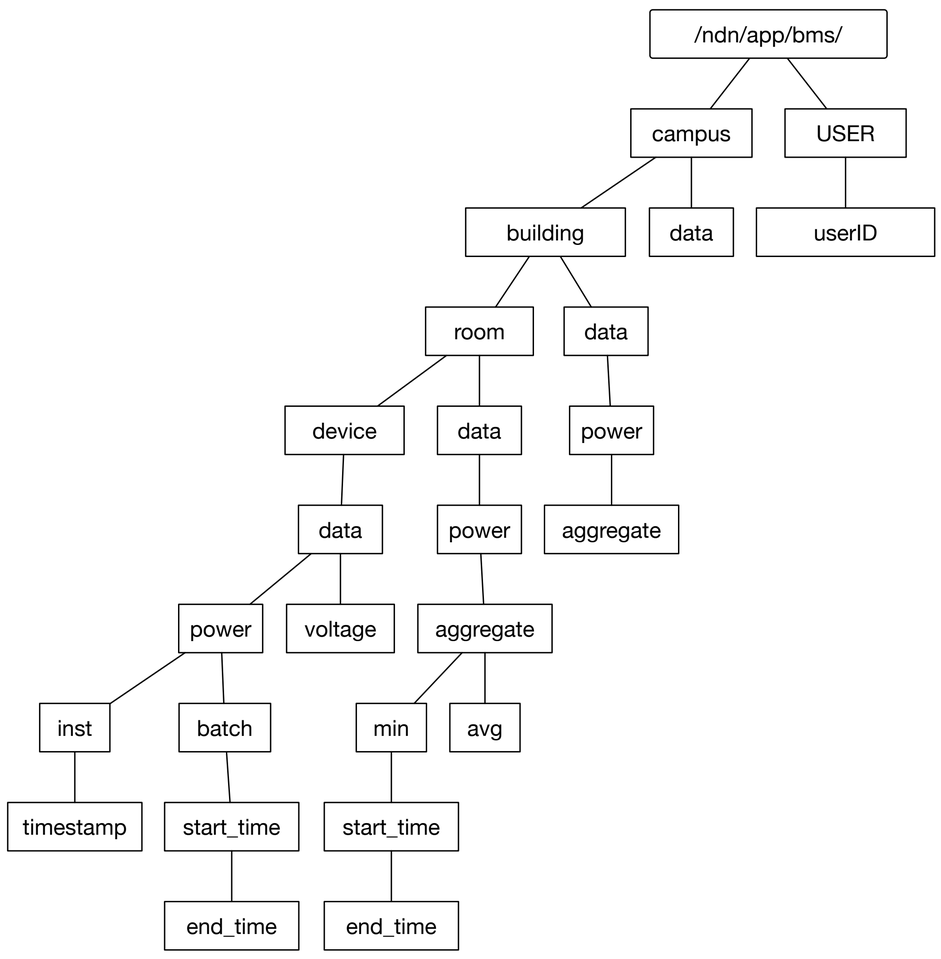

Figure 1: BMS namespace

- Physical location branch represents the hierarchical structure of BMS data, organized in Campus - Building - Room - Device
- User branch records the list of BMS user identities, to be used by GBE later

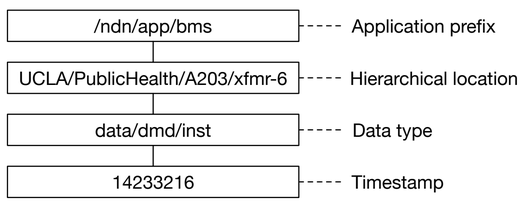

Figure 2: Example BMS data name

Data Aggregation

- Leaf nodes publish aggregated data at fixed time window.
- Non-leaf aggregate the data after all children respond, and publish data with the same time window.
- Long lived interests for fixed time window moves data across layers.

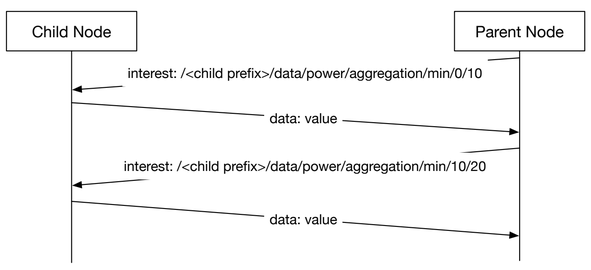

Figure 3: BMS move aggregation sequence

Trust Schema and Bootstrapping

- BMS data should be verified hierarchically
- The certificates of BMS children node should be signed by their parent nodes in the tree
- Campus certificate is the root of trust for BMS data and user. (not so in namespace)

```
rule {
  id "BMS data"
  for data
  filter {
    type name
    name /ndn/app/bms
    relation is-prefix-of
  }
  checker {
    type hierarchical
    sig-type rsa-sha256
  }
}
```

```
rule {
  id "BMS nodes"
  for data
  filter {
    type name
    regex ^<ndn><app><bms>[^<USER>]([^<KEY>]+)<KEY><><ID-CERT><>$
  }
  checker {
    type customized
    sig-type rsa-sha256
    key-locator {
      k-regex ^<ndn><app><bms>([^<KEY>]*)<KEY><><ID-CERT><>$
      k-expand \\1
      h-relation is-prefix-of
      p-regex ^<ndn><app><bms>([^<KEY>]+)<KEY><><ID-CERT><>$
      p-expand \\1
    }
  }
}
```

- Bootstrap:

Each BMS children node should obtain a valid certificate from the parent when joining the network for the first time. Figure 4 illustrates this process.

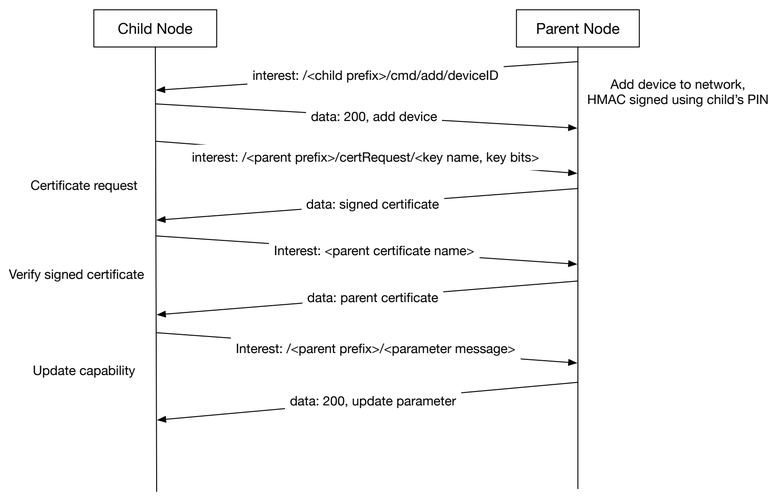

Figure 4: BMS add child sequence

Demo Implementation

As illustrated in Figure 5, we are running BMS aggregation nodes in mini-ndn, with connection to the NDN testbed and BMS data publisher gateway node. An in-browser visualization interface, as demonstrated in Figure 6 is being developed.

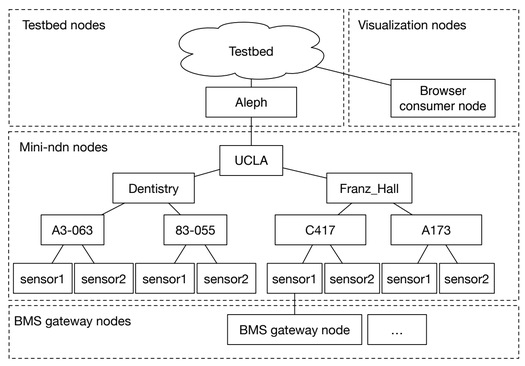

Figure 5: BMS deployment structure

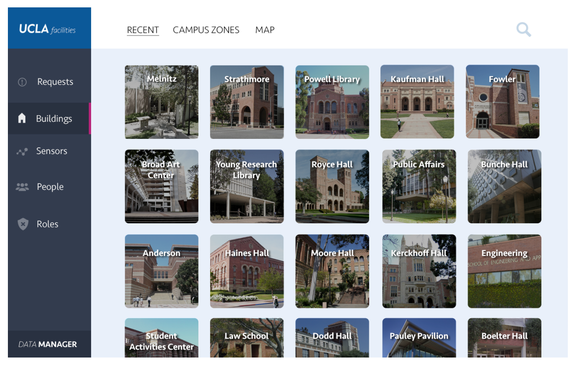

Figure 6: BMS UI demo

Future Work

- Access control / GBE in BMS
- In-browser consumer interface
- Namespace update? Mentioned earlier about some unresolved issues?

References

[1] Wentao Shang, Qiuhan Ding, A. Marianantoni, J. Burke, and Lixia Zhang. Securing building management systems using named data networking. *Network, IEEE*, 28(3):50–56, May 2014.

Contact Information

- Email: zhehao@remap.ucla.edu