A Scalable Cloud Storage for Sensor Networks

Gérôme Bovet\textsuperscript{1, 2} – Gautier Briard\textsuperscript{3} – Jean Hennebert\textsuperscript{2}

\textsuperscript{1}Telecom ParisTech France
\textsuperscript{2}University of Applied Sciences Western Switzerland
\textsuperscript{3}University of Belfort-Montbéliard France

gerome.bovet@hefr.ch
Introduction

A Scalable Cloud Storage for Sensor Networks
Introduction

- Autonomous and self-sufficient networks
- Training phase requires access to historical data
- Limit network traffic and augment data availability where it is used
Zones

- **Build zones according to the building structure**
  - Do the training where data is produced

```
root: home.com
children: floor1, floor2
```

```
root: floor1.home.com
children: living
```

```
root: floor2.home.com
children: -
```

```
root: living.floor1.home.com
children: -
```
Zones - Splitting

\[
C(Z_1, Z_2) = Ci(Z_1) + Ci(Z_2) + Ce(Z_1, Z_2)
\]

\[
Ci(Z) - \min(C(Z_1, Z_2)) > \epsilon_s
\]

*Ci*: Total number of hops between producers and storage peers

*Ce*: Inter-zone costs depending on the filling rate of the child zone
Zones - Merging

\[ C_p(Z) = (C_i(Z_1) + C_i(Z_2)) \times \ln(D(Z_1) + D(Z_2)) \]
\[ C(Z_1, Z_2) - C_p(Z) > \epsilon_m \]

\( D \): Depth of a zone’s internal locations tree
Entities - Storage Peers

- Act as small databases for time-series

- Offer a RESTful interface hiding the internal storage technology (DBMS)
  - Creating/deleting datasets
  - Inserting/reading data

- Data insertion realized through multicast notifications of producers
Entities - Storage Coordinators

- Are managing a specific zone
- Decide whenever their zone should be split or merged
- Organize the repartition of data between storage peers
- Offer a RESTful interface for retrieving stored data on peers
  - Select the appropriate peers
  - Assemble the data
Interfaces

- **General management interface (common multicast)**
  - Used for discovering zones

- **Zone interface (zone specific multicast)**
  - Inter-zone communications
  - Data storage announcement
  - Data retrieval

- **Server interface (unicast)**
  - Synchronization between master and slave storage coordinators
Storing sensor data

- Ensure replication on several databases
  - Unicast is inefficient
Storing sensor data

- CoAP observe and groupcomm are mutually exclusive
- Improvements of CoAP
  - Dynamically create multicast endpoints (token and multicast address)
  - Activate observation (multicast address)
  - Send notifications to multicast address
- No reliability possible
CoAP multicast observation

- Combine observation and group communication

- Membership creation (from CoAP groupcomm)
  
  POST /{+gp}
  
  { "n": "All-Devices.floor1.west.bldg6.example.com",
  "a": "[ff15::4200:f7fe:ed37:abcd]:4567",
  "t":1234}

- Observation
  
  - Use same token as for membership creation (t)
  - Add option Mcast-obsv containing the multicast address (a)
Usage - Zone discovery

Mcast GET /home/floor1/living

Contact my zone over ff15::4200:f7fe:ed37:abcd

root: home.com
children: floor1, floor2

root: floor1.home.com
children: living

root: floor2.home.com
children: -

root: living.floor1.home.com
children: -
Usage - Data storage announcement

{  "url": "temp.living.floor1.home.com/celsius",  "max-age": 90,  "unit": "day"}
Usage - Data retrieval

```
Mcast GET home/floor1/kitchen/temp/celsius?
from=2014-02-13&to=2014-02-18

[{
  "timestamp": "2014-02-13 13:54:01",
  "value": 21.7
},
...
]```
Evaluation

- **3 rooms over 2 floors as testbed**
  - 21 KNX and EnOcean sensors producing data
  - 8 Flyport OpenPicus as storage peers
  - 4 Raspberry Pies as storage coordinators

![](Scalability_Test_with_Concurrent_Clients.png)

Each request returning **49** entries
Average **186 ms** pro request
Contributions

- **Location-based zone management**
  - Store data where it is needed
  - Efficient communications

- **Entirely anonymous communications with multicast**
  - Ensure availability
  - Insensitive to disturbances (storage coordinators)

- **CoAP adaptations for multicast notifications**
  - Becoming a draft?
Conclusion

- **Extensible-cloud like storage**
  - Add devices for extending the capacity
  - Adapted for machine learning applications

- **Self adaptation according to required storage space**

- **Easy to use RESTful interfaces for retrieving time-series**

- **CoAP multicast notifications very useful**
  - Need for standardization
Questions