DESIGN OF SCALABLE DIRECTORY SERVICE FOR FUTURE IoT APPLICATIONS

Ved P. Kafle, Yusuke Fukushima, Pedro Martinez-Julia, and Hiroaki Harai
National Institute of Information and Communications Technology (NICT)

Bangkok, Thailand
14-16 November 2016
Outline

- Background and motivation
- Proposed IoT directory service components
- Design approach
- Security and privacy protection
- Record caching, replication, lookup, update process
- Resource adjustment
- Conclusion
Background and motivation

• IoT – driving next generation of ICT
• Billions of new devices getting connected
• M2M communication being commonplace
  – Autonomic, fast, secure communication between devices

• Requiring a scalable, low latency directory service
  – Scalable to hold billions of records (info about devices)
  – Lookup latency of few milliseconds
  – Update latency of few seconds
Related work limitations

• Domain Name System (DNS)
  – Can’t store heterogeneous types of names
  – Response time not guaranteed
  – No fast update support
• Auspice [5]
  – Not considering record privacy and access control
• MDHT [6]
  – Not considering frequent updates
• CoDoNS [7]
  – Not considering updates and privacy
• X.500 [8]
  – Not considering minimization of response time
Proposed IoT directory service components

- Resource controller (RC)
- Authorized public directory
- Private directory
- Record registration
- Record owners - IoT devices (e.g. red cars)
- On-demand record caches
- Replication
- Record update
- Record caching
- Record registration
- Resource provisioning
- Load statistics
- Lookup
- IoT application clients (e.g. yellow cars)

Legend:
- Components in red font
- On-demand allocated resource
- Process in black font with arrows
Design approach

• **IoT service-wise directory service**
  – E.g. automated/safe driving and smart-grid control applications will have two different directory services

• **On-demand trackable record caching/replication**
  – Caching for low latency lookup by reducing distance
  – Trackable caches for quick update of dynamic records

• **QoS guarantee by dynamic resource provisioning**
  – In the event of fluctuating workload and network status

• **Leveraging established component technologies**
  – SDN; NFV; VM creation, migration
Record replication process

• Replication to avoid bottleneck, maintain performance
  – Despite increasing volume of records (initially, up to 1Billion)
  – Despite increasing workload
Record lookup process

- Record looked up from the nearest replicas provided with enough resource
  - Fast response (<= 10ms)
Security and privacy protection scheme

- Responding only to queries that pass thru’ both security and privacy checks:
  1. Message security check
  2. Record privacy check
Record update process

- Local replicas updated first, then public directory
- Neighboring caches updated either by the anchor cache (2.2) or by public directory (2.4)
Resources adjustment for performance

- Computation, storage, and networking resources dynamically adjusted based on workload and performance

<table>
<thead>
<tr>
<th>Type</th>
<th>Amount</th>
<th>Place</th>
</tr>
</thead>
</table>

Resource Controller (RC)

- Load statistics
- Performance prediction

IoT Directory Service Provider

- Records storage & replication
- Lookup (scheduling) Update

Resource allocation

Load statistics and performance metrics

Bangkok, Thailand, 14-16 November 2016
ITU Kaleidoscope 2016 - ICTs for a Sustainable World
Conclusion

• Presented design of IoT directory service to store information (records) about huge number of IoT devices and provide fast lookup and dynamic update
• Envisioning IoT applications requiring info about IoT devices within milliseconds (for secure, real-time communication)

• Future work
  – Modeling of required resources to guarantee performance
  – Implementation with NoSQL database (to store and lookup heterogeneous names), NFV and SDN tools
  – Standardization of research outcome in ITU-T