Supporting Internet of Things applications in Information-centric networking

Jenda Brands & Olaf Elzinga
What is ICN?

- Individual apps
  - browser chat...
  - File Stream...
- Security
- Content chunks
- Strategy
  - IP UDP P2P BCast...
  - copper fiber radio...

- Individual links
  - IP packets
  - ethernet PPP...
  - CSMA async sonet...
  - copper fiber radio...

- Every node
  - TCP UDP...
  - SMTP HTTP RTP...
  - email WWW phone...
ICN projects

- content centric networking (CCN)
  - 2007 Palo Alto Research Center (PARC)
  - 2009 first software implementation
- named data networking (NDN)
  - 2010 based of CCN
  - Funded by National Science Foundation
  - Forked in 2013
Research question

- How does content centric networking (CCN) compare to named data networking (NDN) with regards to IoT applications?
The basics

- Two packet types:
  - Interest
  - Data
Data structures

- **Forwarding Information Base (FIB)**
  - Used for Interest forwarding

- **Pending Interest Table (PIT)**
  - Used for Data forwarding

- **Content Store (CS)**
  - Cache for Data packets received
Naming

- Flatnames and Hierarchical naming
- Naming in CCN must be an exact match
- NDN allows ‘overmatching’
  - Interest for: /nl/uva/os3/serverroom
  - Will match: /nl/uva/os3
Forwarding

- **CCN**
  - Interest
    - Check HopLimit
    - Check ContentStore
    - Add PIT entry

- **NDN**
  - Interest
    - Check HopLimit
    - Add PIT entry
    - Check ContentStore
Routing

- Early research stage
- NDN implementation:
  - Named-data LinkState Routing (NLSR)
## Differences

<table>
<thead>
<tr>
<th>Type</th>
<th>CCNx 1.x</th>
<th>NDN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Naming</strong></td>
<td>Must match exactly</td>
<td>Overmatching possible</td>
</tr>
<tr>
<td><strong>Packet format</strong></td>
<td>Fixed-size header</td>
<td>Variable-size header</td>
</tr>
<tr>
<td><strong>Forwarding</strong></td>
<td>First ContentStore check, then PIT Entry</td>
<td>First PIT Entry, then ContentStore check</td>
</tr>
<tr>
<td><strong>Single name multiple objects</strong></td>
<td>Likely to cause issues</td>
<td>Likely to cause issues</td>
</tr>
<tr>
<td><strong>Routing</strong></td>
<td>No routing implemented yet</td>
<td>Link state and static</td>
</tr>
<tr>
<td><strong>Push-based possibilities</strong></td>
<td>No overmatching</td>
<td>Overmatching</td>
</tr>
<tr>
<td><strong>Data in Interest</strong></td>
<td>Not possible</td>
<td>Not possible</td>
</tr>
</tbody>
</table>
Use case (1/2)

- Cars on highway
- Speed & Temperature from car sensors
- Access points where cars connect with
- Moving objects
Use case (2/2)

- Requirements/constraints
  - Network connection
  - Naming
  - Routing
  - Push based communication
  - Data aggregation
Proof of Concept (1/4)

- Only for NDN
- Python (PyNDN)
Proof of concept (2/4)

1. Create static route → Add to Polling list

Polling process

2.A. Receive Data → Place in CS

2.B. Interest time-out → Remove from Polling list → Remove static route
Proof of concept (3/4)

3
Send Interest

3
Send DATA

AP

Application

3
Check FIB matching data

3
Collect all matching FIB entries

4
Replace old data & update time stamp

4
Send one single Data message
Proof of concept (4/4)

- Array with data that is sent back
- Sector with speed from all cars

```json
{
  "s39": {
    "speed": ["165", "127", "125", "129", "128", "127"]
  },
  "s38": {
    "speed": ["165", "133", "127", "131", "145"]
  }
}
```
Conclusion

- Both very similar
- Use case specific
  - NDN turned out better for our use case
- Push communication not possible (by design)
Future Work

- Feasibility of sending data inside an Interest packet
- Influence of forwarding with constrained devices
Thanks for your attention