NDN-IoT Framework and Example Application “Flow”

Zhehao Wang, Eitan Mendelowitz, Zoe Sandoval, Jeff Burke

Mar 23, 2017
Introduction – NDN-IoT & Flow

• **NDN-IoT framework**
  – Libraries in JavaScript, Python, C# and C++
  – Implement naming, trust and bootstrap, discovery and application-level pub/sub functionalities NDN team’s IoTDI ’16 invited paper [1]

• **"Flow" application:**
  – Home IoT game application built using the framework
  – Cloud-independent, secure (authentication and authorization)
  – Its design and implementation discussed in NDN teams’ IoTDI ’17 invited paper (to appear)

## Flow: components

<table>
<thead>
<tr>
<th>Component</th>
<th>Device</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenPTrack (opt)</td>
<td>Linux workstation machine</td>
<td>Produces time-series data (location of multiple tracked persons) at 30Hz</td>
</tr>
<tr>
<td>Mobile website</td>
<td>Mobile phone</td>
<td>Sends environment control commands and consumes environment status</td>
</tr>
<tr>
<td>Virtual camera control</td>
<td>RFduino</td>
<td>Produces NDN data from gyroscope reading. Packetize, and send to helper (RaspberryPi) via bluetooth</td>
</tr>
<tr>
<td>Controller/Gateway</td>
<td>RaspberryPi2</td>
<td>Serves as the trust anchor (runs NDN-pi controller)</td>
</tr>
<tr>
<td>Unity3D game engine</td>
<td>OSX machine</td>
<td>Consumes opt, phones, and Arduino data; and renders the virtual environment</td>
</tr>
</tbody>
</table>
Flow: app message exchanges

1A. Unity asks for a "track_hint"
1B. Names when asking for a track location

2A. Mobile phone sends command Interest and initiates "track matching" of phone's ID to Track ID provided by OPT
2B. Command Interest to control the dropping of images in Unity

3A. Unity asks for gyroscope data, produced by RaspberryPi helper on behalf of RFduinos connected via Bluetooth LE

Please refer to next page for diagram key
Flow: app message exchange keys

Diagram Key for NDN Interest-Data Exchange

1A.

**Interest**: /home/flow1/opt1/<run_id>/hints/, exclude: <last_received_timestamp>
**Data**: Interest name + <timestamp>, content: {“id”: 45, “seq#”: 312}

1B.

**Interest**: /home/flow1/opt1/<run_id>/tracks/<track_id>/<seq#>
**Data**: Interest name, content: {“x”: 1.0, “y”: 0.9, “z”: -0.3}

2A.

**Command interest**: /home/flow1/unity1/<action:match, id:alice_phone>
**Data**: Interest name, content: {“status”: “200”, “data”: “<html>track 45, show links</html>“}

2B.

**Command Interest**: /home/flow1/unity1/<action:link_click, id:alice_phone, link:img_3>
**Data**: interest name, content: {“status”: “200”}

3A.

**Interest**: /home/flow1/gyros/gyro1/, exclude: <last_received_timestamp>
**Data**: Interest name, content: {“p”: 0.3, “y”: 0.5, “r”: 0.1}
Links and documentation

• **Code repository**
  – NDN-IoT framework
    • Functionality overview
    • Interface description
  – Flow application

• **Technical guide** (installation and troubleshooting)

• **Demo poster**

• **Application screen recording**