
Jakob Schärer

Dr. Zhongliang Zhao, Prof. Dr. Torsten Braun

University of Bern (Switzerland)

04.11.2018 Shenzhen 2018
Content

> Introduction
  — Routing with RPL
  — SDN based routing

> Routing Protocol: DTARP

> Evaluation

> Conclusion
Wireless Sensor Networks (WSNs)

- **Sensor Nodes**
  - Limited Resources (Energy, computational capability, etc.)
  - Most energy used for radio

- **Lossy ad-hoc networks**

- **Sink** collects sensor readings

- Well researched routing protocols for *node-to-sink* traffic (e.g. RPL)
Wireless Sensor and Actuator Networks (WSANs)

- WSN are evolving to (WSANs) and Actuators become more important

- Sink-to-node and node-to-node becomes important
Routing with RPL

- Tree based routing
- Node-to-node possible
- Packets are sent along the tree
Problems with RPL

> Tree based routing leads to:
  — Congestion at root node
  — Nodes close to root need much more energy

> Nodes closer to the root run first out of battery
  → network fails
Software Defined Networking

> Separate control logic from network device

> Smart control plane
  — Global network information

> Fast data plane
  — Switches with Flowtables

> SDN-Wise ([http://sdn-wise.dieei.unict.it](http://sdn-wise.dieei.unict.it))
  — SDN framework for WSNs
Routing with SDN-Wise (RSSI)

- Shortest path routing
- Congestion at central nodes
- Node betweenness centrality

\[ c_B(v) = \frac{1}{n(n-1)} \sum_{s,t \in V} \frac{\sigma(s, t | v)}{\sigma(s, t)} \]

- \( n \) is the number of nodes
- \( \sigma(s, t) \) is the number of shortest paths between \( s \) and \( t \)
- \( \sigma(s, t | v) \) is the number of shortest paths between \( s \) and \( t \) over node \( v \)
Problem of SDN-Wise (RSSI)

- Central nodes handle much more traffic with shortest path
  - More energy is used
  - Run first out of battery

> Goal
   — Even distribution of the traffic
   — Increase network lifetime

> Solution
   — Define cost function
DTARP – Cost function

\[
\text{cost}(e) = \begin{cases} 
1 + \alpha & \text{if } \text{rs} \text{si} < \text{threshold} \\
\alpha + \beta \cdot \text{tf}(e) + \frac{1}{2} (1 - \beta)(c_B(v_1) + c_B(v_2)) & \text{else} 
\end{cases}
\]

Where \( e \) is a link between node \( v_1 \) and node \( v_2 \), and \( c_B(v) \) is the node betweenness centrality

\[
\text{tf}(e) = \frac{1}{2} \frac{\text{traffic}(v_1) + \text{traffic}(v_2)}{\sum_{u \in V} \text{traffic}(u)}
\]
Performance Measurements

> Simulation to analyze specific scenarios
  — Proof of concept

> Real-world test and comparison to RPL with SDN-Wise (RSSI)
Simulation - Topology
## Simulation - Results

### Initial route

<table>
<thead>
<tr>
<th>Time</th>
<th>ID</th>
<th>Event</th>
<th>Node</th>
<th>Message ID</th>
<th>Source</th>
<th>Destination</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>01:11.208</td>
<td>3</td>
<td>TXU</td>
<td>3</td>
<td>message_id: 3.4</td>
<td>3</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>01:11.290</td>
<td>6</td>
<td>RXU</td>
<td>6</td>
<td>message_id: 3.4</td>
<td>3</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>01:11.630</td>
<td>5</td>
<td>RXU</td>
<td>5</td>
<td>message_id: 3.4</td>
<td>3</td>
<td>1</td>
<td>98</td>
</tr>
<tr>
<td>01:11.671</td>
<td>1</td>
<td>RXU</td>
<td>1</td>
<td>message_id: 3.4</td>
<td>3</td>
<td>1</td>
<td>97</td>
</tr>
<tr>
<td>01:21.208</td>
<td>3</td>
<td>TXU</td>
<td>3</td>
<td>message_id: 3.5</td>
<td>3</td>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>01:31.208</td>
<td>3</td>
<td>TXU</td>
<td>3</td>
<td>message_id: 3.6</td>
<td>3</td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

### Stable route

<table>
<thead>
<tr>
<th>Time</th>
<th>ID</th>
<th>Event</th>
<th>Node</th>
<th>Message ID</th>
<th>Source</th>
<th>Destination</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>01:41.211</td>
<td>3</td>
<td>TXU</td>
<td>3</td>
<td>message_id: 3.7</td>
<td>3</td>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>01:41.415</td>
<td>6</td>
<td>RXU</td>
<td>6</td>
<td>message_id: 3.7</td>
<td>3</td>
<td>4</td>
<td>99</td>
</tr>
<tr>
<td>01:41.631</td>
<td>5</td>
<td>RXU</td>
<td>5</td>
<td>message_id: 3.7</td>
<td>3</td>
<td>4</td>
<td>98</td>
</tr>
<tr>
<td>01:41.763</td>
<td>4</td>
<td>RXU</td>
<td>4</td>
<td>message_id: 3.7</td>
<td>3</td>
<td>4</td>
<td>97</td>
</tr>
<tr>
<td>01:51.208</td>
<td>3</td>
<td>TXU</td>
<td>3</td>
<td>message_id: 3.8</td>
<td>3</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>01:51.400</td>
<td>2</td>
<td>RXU</td>
<td>2</td>
<td>message_id: 3.8</td>
<td>3</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>01:51.423</td>
<td>1</td>
<td>RXU</td>
<td>1</td>
<td>message_id: 3.8</td>
<td>3</td>
<td>1</td>
<td>98</td>
</tr>
</tbody>
</table>
Benchmark - SDNWisebed

> Realworld testbed
  — SDN based
  — 40 TelosB nodes
  — Resource reservation
  — Program sensor nodes
  — remotely reset nodes
  — collecting and storing experimental data
Benchmark - Scenario

> Each node sends to one distinct other node (permutation)

> Compare different protocols
   — RPL
   — SDNWise with RSSI routing
   — DTARP
Benchmark – Traffic Distribution

\[ tf(v) = \frac{\text{traffic}(v)}{\sum_{u \in V} \text{traffic}(u)}, \quad v \in V = \text{nodes} \]
Benchmark – Packet Loss Rate (PLR)

\[ PLR = 1 - \frac{\text{# received data packets}}{\text{# sent data packets}} \]
Benchmark – Average Hop Count (AHC)

\[ AHC = \frac{\sum_{p \in P} \text{hops\_at\_dst}(p)}{\#P} \]
Conclusion

- Node-to-node communication benefits from SDN in WSNs
  — Less congestion

- DTARP reduces congestion in the network and increases network lifetime
Future Work

- Security in SDN-controlled WSNs
- Improvement of route updates
- Mobility of sensor nodes
Questions?
Literature


