Competition: Wireless-Transparent Sensing Platform

Chun-Hao Liao, Theerat Sakdejayont, Makoto Suzuki, Yoshiaki Narusue, and Hiroyuki Morikawa (Team 07)

School of Engineering, The University of Tokyo
Wireless-Transparent Sensing Platform

- Full paper presented in EWSN 2017 [1]
  - TDMA-based protocol adopting the concurrent transmission flooding (CTF)
    - The analysis in PHY layer [2] showed why IEEE 802.15.4 can survive synchronized packet collision.
  - Flexible timeslot management
    - On-the-fly scheduling
    - Service-driven scheduling

Concurrent Transmission Flooding (CTF)

**Timing diagram**
- **Initiator**: Transmits (Tx)
- **1st Relays**: Receives (Rx) and Transmits (Tx)
- **2nd Relays**: Receives (Rx) and Transmits (Tx)

**Data**
- Initiator

Diagram shows the flow of data and transmission between nodes.
Given multi-source multi-destination traffic pattern and harsh RF environments, several refinements have been made on WTSP including:

1) A dedicated service that realizes UDP-like one-way streaming dissemination
2) Progressive forwarding
3) Source redundancy
Challenge: Poor reliability even with end-to-end retransmission due to the harsh RF jamming

- Contention-based slot scheduling requires ACK to ensure end-to-end reliability.
- In dense traffic and harsh RF environments, this two-way communication may lead to severe network congestion.

One-way communication, Fixed scheduling
UDP-like One-way Streaming (2)

**Improvement:** Simple but efficient one-way dissemination without ACK

Pre-assigned slot scheduling

---

**Diagram:**

- SRC 1, SRC 2, SRC 3, REL 1, SRC 4, SRC 5, REL 2, SRC 6, SRC 7, REL 3, SRC 8, SRC 9, REL 4, SRC 10, REL 5

---

**Diagram Details:**

- Time axis labeled at bottom right:
  - DES 3
  - DES 5
  - DES 7
  - DES 8

---

**Diagram Legend:**

- SrC: Source Count
- REL: Release
- DES: Destination
Progressive Forwarding (1)

Challenge: “Volatile relaying”

- A packet from an initiator hardly gets through the harsh RF jamming in one flooding.

- Poor synchronization
- Poor end-to-end connectivity

Assisting nodes
**Improvement**: Slot-owning nodes help synchronization and relaying information in progressive manner.

### Progressive synchronization

1. “You can sync to me” flag set by a slot-owning node → *Quick sync.*
2. Re-bootstrapping if not directly received from the network head for too long → *Unity of the network*

### Progressive forwarding nodes
Source Redundancy (1)

**Challenge:** Inefficiency of one-to-one packet format

- *Large* packet overhead, but *tiny* GPIO information

For example,

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEN</td>
<td>1 byte</td>
</tr>
<tr>
<td>Service header</td>
<td>1 byte</td>
</tr>
<tr>
<td>Slot No.</td>
<td>1 byte</td>
</tr>
<tr>
<td>Initiator node ID</td>
<td>1 byte</td>
</tr>
<tr>
<td>Relay count</td>
<td>1 byte</td>
</tr>
<tr>
<td>CRC</td>
<td>2 bytes</td>
</tr>
</tbody>
</table>

**VS**

1-bit GPIO
**Source Redundancy (2)**

**Improvement:** Transmitting information from multiple sources in a packet

- Highly-efficient packet format:
  - 1-bit for each GPIO and 2-bit age information for each source

```
<table>
<thead>
<tr>
<th>LEN</th>
<th>Service Header</th>
<th>Slot No.</th>
<th>Init. node ID</th>
<th>...</th>
<th>Relay Cnt.</th>
<th>16-bit CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>32</td>
<td>64</td>
<td>72</td>
</tr>
</tbody>
</table>
```

“Sync” flag
Conclusion

- UDP-like one-way streaming as a dedicated service of WTSP
  - Pre-assigned slot scheduling
- Progressive forwarding
  - Quick synchronization, but still maintaining the unity
  - Arbitrary nodes can be assigned to be forwarding nodes.
- Source Redundancy
  - Efficient packet format
    1-bit for each GPIO, 2-bit age information for each source
- Others: channel hopping
Thank you.