Current State of IEEE 802.15.4/6LoWPAN Stack inside the Linux Kernel

FOSDEM 2014

Alexander Aring
Pengutronix
<aar@pengutronix.de>
Topics of Discussion

- Project history
- Introduction
- Linux implementation
- Future work
- Demo
Project history

• Project started in 2008
  • Project name „linux-zigbee“
  • ZigBee is an alternative to 6LoWPAN
  • License issues: ZigBee stack into kernelspace
  • Possible not released userspace ZigBee stack?

• Now: Project aims to implement 6LoWPAN
  • Open standard
  • Based on IEEE 802.15.4 networks
  • Additional 6LoWPAN upper layer protocols
  • Mainline since year 2009
Introduction
IEEE 802.15.4 and 6LoWPAN

- MAC-Layer: IEEE 802.15.4
  - Low-Rate Wireless Personal Area Networks

- 6LoWPAN
  - IPv6 over Low power Wireless Personal Area Networks
  - RFC4944 - Transmission of IPv6 Packets
  - RFC6282 - IPv6 Header Compression

- Areas of Applications
  - Sensor networks
  - Home and industrial automation

- Related work: ContikiOS
  - Most used 6LoWPAN stack implementation
  - Small stack implementation
Linux implementation
IPv6 - Architecture

Application layer
Socket layer
Transport layer
Packet layer
Ethernet interface
Linux implementation

6LoWPAN - Architecture

- Application layer
- Socket layer
- Transport layer
- Packet layer
- lowpan interface
- 6LoWPAN adaptation layer
- IEEE 802.15.4 interface
Linux implementation
6LoWPAN adaptation layer

- Compression of the 40 bytes IPv6 header
  - Version, traffic class, flow-label, hop-limit
  - Addresses (link-local, multicast)
  - We can remove the payload length
  - Smallest 6LoWPAN header: 3 bytes

- Compression of transport header
  - For example UDP has normally 8 bytes
  - Special port ranges and removing of checksum
  - Smallest UDP 6LoWPAN header: 5 bytes

- 6LoWPAN fragmentation
  - 127 (IEEE 802.15.4) to 1280 (IPv6) MTU
Linux implementation
Experienced issues

• Started with kernel version 3.8

• Tried to ping another 6LoWPAN node
  • Worked with non link-local addresses only
  • Fragmented 6LoWPAN packets did work in a Linux to Linux communication only
  • Got race conditions while fragmentation

• Run an UDP application
  • Random null pointer dereferences occurred
  • Didn't work on UDP 6LoWPAN port ranges
Linux implementation

Fixed issues

- IPHC (IPv6 Header Compression)
  - Address compression/uncompression
    - Did never work correctly
    - Reimplement necessary functions
  - UDP compression/uncompression
    - Byte ordering issues
    - Wrong pointer arithmetic (Null pointer problem)
    - Reverse source/destination port ordering

- Static IEEE 802.15.4 header size value
  - IEEE 802.15.4 header has a dynamic size
  - Size determined by flow control field
  - Value used in fragmentation for reconstruction
Linux implementation

Known existing issues

6LoWPAN Fragmentation

- Isn't RFC compliant
- Still having race conditions
- Issues with ACK handling on MAC layer
  - No Data Sequence Number increment
    - ACKs do not work correctly

→ There are patches for a solution
  - Which is RFC compliant
  - No race conditions
    - Idea: Implement it like IPv6 fragmentation
  - Put the increment of DSN on the right place
Linux Implementation
What we have done now?

• Before
  • Ping to a contiki device wasn't possible
  • Suddenly Linux kernel crashed

• Now
  • Use of link-local addresses works
  • Connection to a contiki device works
  • IPHC and fragmentation is RFC complaint

• Bluetooth 6LoWPAN
  • Share IPv6 header compression format
  • Improving 6LoWPAN implementation
Future Work

6LoWPAN upper layer protocols

- RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks
  - Route-over: ICMPv6
  - Prototype implementation: SimpleRPL by Tony Cheneau
    - Limited functionality
    - Has lot of dependencies: python, zeromq, ...

- Neighbor Discovery Optimization for 6LoWPAN
  - Optimization for non-multicast MAC-Layer
  - Need some great idea to implement it
  - Problem: Possible handling in 6LoWPAN adaptation layer?

- CoAP for Userspace (Constrained Application Protocol)
  - HTTP for sensor networks but UDP based
  - Tested libcoap successful under Linux
Thanks!

Project Website:
http://sourceforge.net/projects/linux-zigbee/

Mailing list:
linux-zigbee-devel@lists.sourceforge.net

Special Thanks to:
• Werner Almesberger
• Tony Cheneau
• Alan Ott