Status Report for IEEE 802.15.4 and 6LoWPAN in Linux

Embedded Linux Conference San Jose 2015

Stefan Schmidt
Samsung Open Source Group
s.schmidt@samsung.com
Agenda

- Introduction
- Project History
- Mainline Status
- Current Work Areas
- Future Work
Introduction
IEEE 802.15.4 / LoWPAN

- IEEE standard
- Low-Rate Wireless Personal Area Networks
- Specifies the physical and the MAC layer
- Simple addressing but no routing
- Star and Peer-to-Peer topologies supported
- Mesh topologies need some layers on top of these
- Applications are small battery powered devices like sensors
6LoWPAN

• A series of IETF specifications
• IPv6 over LoWPAN
• RFC4944: IPv6 Convergence Layer
• RFC6282: IPv6 Header Compression
• Updates and extensions in other RFC's
  (see resources slide at the end)
Motivation and Use Cases

- Battery powered sensors might not run Linux but choose a smaller OS
- Main powered appliances might run Linux already and would benefit from native 6LoWPAN support
- Border Routers / Gateways are likely to run Linux
- IEEE 802.15.4 chips could easily be integrated in WiFi accesspoints or routers which already run Linux
Project History
Early Days

• Started in 2008 as linux-zigbee project on Sourceforge
• Mainly driven by Siemens AG
• Kernel code as well as lowpan-tools userspace configuration utilities
ZigBee Relations

• The name itself was very misleading
• The code only implemented the IEEE 802.15.4 layers and no ZigBee protocols or profiles at all
• ZigBee licensing seems incompatible with the GPL so no ZigBee support for the Kernel
Mainlining

• The first steps of mainlining moved the core parts of the sourceforge repo over around 2012
• Main Siemens developers withdraw over time
• Community slowly took over
Under New Management

• New project name to avoid confusion: linux-wpan
• New maintainer: Alexander Aring, Pengutronix
• Mailinglist moved to vger like most other Kernel lists
• Patches are now handled on the list and picked up through the Bluetooth tree
• http://wpan.cakelab.org, releases, docs
Mainline Status
Overview

- ieee802154 handles the MAC layer and drivers (wpan0 interface)
- 6LoWPAN sits on top of the wpan devices and acts as convergence layer to be used by the normal IPv6 kernel stack (lowpan0 interface)
- 6LoWPAN transparently handles the fragmentation and defragmentation between the different MTU's (127 vs 1280) as well as compressions

Source: Alexander Aring
Current Mainline Status

- Basic ieee802154 layer with drivers for various chips (at86rfxxx, mrf24j40, cc2520)
- 6LoWPAN implementation
- IP Header Compression
- Connection between Linux devices works
- Connection to Contiki devices works
6LoWPAN Compressions

- Fragmentation handling is only one part
- A IPv6 header alone is 40 bytes which means it would use almost 1/3 of a ieee802154 frame
- Re-use of the 64 bit wpan address and various other compressions brings the smallest header down to 3 bytes
- On top there are compression modes for UDP and others
Bluetooth LE Relationship

- IETF specification for IPv6 over Bluetooth LE
- Still in draft phase (draft-ietf-6lo-btle)
- Common code is thus shared between the wpan and Bluetooth subsystems
Current Work Areas
Overview

• Main work areas right now:
  – New netlink framework nl80215 (Major part done)
  – ieee802154 cryptography layer on top of nl802154
  – Improvements in frame parsing and creation
New Netlink Interface - Kernel

- nl802154 is the netlink interface between Kernel and userspace
- Used for configuration (PAN ID, short address, etc)
- Inspired by nl80211 from the wireless developers
- Aligning these two should help to make the code easier to understand for already established hackers as well as newcomers
New Netlink Interface - Userspace

- Available since Kernel version 3.19
- Needs a new userspace tool: iwpan
- Also inspired and aligned with iw from the wireless community
- Old netlink interface still available but considered deprecated
MAC Layer Cryptography

• The IEEE802.15.4 specification defines AES 128 bit cryptography to encrypt and or authenticate the transmitted data
• 8 different security policies are defined (AES-CBC-MAC and AES-CCM in various length)
• Almost all transceivers implement this in hardware
• While the Kernel will handle the interface to the hardware nl802154 needs to be extended to handle AES key setting, etc
Next Header Compression

- RFC6282
- 6 LoWPAN Next Header Compression (NHC)
- Describes various compression formats
- Kernel framework allows for different modules to handle one compression and decompression format each
- Mix and match different modules/formats
- Only UDP NHC is implemented right now
Future Work
IEEE 802.15.4

• Implement missing parts of the spec
  – Coordinator support in MAC layer and wpan-tools
  – Scan for available PANs
  – Expose more MAC functionality through nl802154

• Improve existing drivers and add support for new chips
6LoWPAN / NHC

- Run time configuration of NHC (Handled by loading and unloading modules right now)
- Implement more NHC modules for other compression schemes
Miscellaneous

• Routing Protocol for Low-Power and Lossy Networks (RFC6550)
  – SimpleRPL, unstrung, linux-rpl as current implementations
• Neighbor Discovery Optimization for 6LoWPAN (RFC6775)
• Test with more high level protocols on to (CoAP, MQTT, etc)
Related work

• ContikiOS implements 6LoWPAN as well (Kernel implementation origins from it)
• Threads uses parts of 6LoWPAN for their protocol
• 6LoWPAN over powerline
Resources

- RFC4919: 6LoWPAN Problem Statement
- RFC4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks
- RFC6282: Compression Format for IPv6 Datagrams
- RFC6550: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks
- RFC6775: Neighbor Discovery Optimization for 6LoWPAN
- RFC7400: 6LoWPAN-GHC: Generic Header Compression for 6LoWPAN
Thank you.
We are hiring.

jobs@osg.samsung.com