CSE 291: Wireless and Communication in the Internet of Things
IEEE 802.15.4

Pat Pannuto, UC San Diego
ppannuto@ucsd.edu
Today’s Goals
[n.b. don’t expect to finish the slides today; probably through PHY]

• Introduction to 802.15.4

• Overview of physical layer details

• Exploration of link layer
  – Network topologies
  – Communication structure
  – Access control
  – Packet structure
References

- 802.15.4 Specification [2006]
  - “Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)”

Other helpful references:
- Paper introducing the 802.15.4 draft
- NXP 802.15.4 Stack User Guide
- 2005 presentation on 802.15.4
Outline

- Overview
- Physical Layer
- Link Layer
- Packet Structure
Comparison of networks

Data Throughput

Range
Comparison of networks

Data Throughput vs. Range

- WiFi
- Cellular
- Bluetooth
- BLE
- 802.15.4
- LPWANs
Comparison of networks

There are some missing qualities here. Why be closer to the origin?

Data Throughput

- WiFi
- Bluetooth
- BLE
- 802.15.4
- Cellular
- LPWANs

Range
Comparison of networks

- **Bluetooth**
  - Data: Lower Power & Lower Cost
  - Range: LPWANs

- **WiFi**
  - Data: Higher Power & Higher Cost

- **Cellular**
  - Data: Higher Power & Higher Cost

- **BLE**
  - Data: Lower Power & Lower Cost
  - Range: LPWANs

- **802.15.4**
  - Data: Lower Power & Lower Cost

- **LPWANs**
IEEE 802....

- Anyone heard “Eight-Oh-Two Dot”?
  - Where?
  - What is it?
IEEE 802

- Network standards for variable-sized packets
  - Ethernet
  - WiFi
  - WPANs

- E.g. **not** networks that send periodic constant-sized packets

- Specify PHY Layer and Link Layer \([\text{MAC} + \text{LLC}]\)
IEEE 802.15

<table>
<thead>
<tr>
<th>IEEE 802.15</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.15</td>
<td>Wireless PAN</td>
<td>Active</td>
</tr>
<tr>
<td>IEEE 802.15.1</td>
<td>Bluetooth certification</td>
<td>Disbanded</td>
</tr>
<tr>
<td>IEEE 802.15.2</td>
<td>IEEE 802.15 and IEEE 802.11 coexistence</td>
<td>Hibernating[4]</td>
</tr>
<tr>
<td>IEEE 802.15.3</td>
<td>High-Rate wireless PAN (e.g., UWB, etc.)</td>
<td>?</td>
</tr>
<tr>
<td>IEEE 802.15.4</td>
<td>Low-Rate wireless PAN (e.g., ZigBee, WirelessHART, MiWi, etc.)</td>
<td>Active</td>
</tr>
<tr>
<td>IEEE 802.15.5</td>
<td>Mesh networking for WPAN</td>
<td>?</td>
</tr>
<tr>
<td>IEEE 802.15.6</td>
<td>Body area network</td>
<td>Active</td>
</tr>
<tr>
<td>IEEE 802.15.7</td>
<td>Visible light communications</td>
<td>?</td>
</tr>
</tbody>
</table>

- **Wireless Personal-Area Networks (WPAN)**
  - All the things within the workspace of a person
  - Conceptually smaller domain than the Local Area Network
  - Realistically about the same thing as a LAN (or really a WLAN)

- **Formerly included a Bluetooth spec**
  - Bluetooth SIG took over governance
802.15.4 (LR-WPANs) Overview
“Low-Rate Wireless Personal Area Networks”

• Goals
  – “The IEEE 802.15 TG4 was chartered to investigate a low data rate solution with
    multi-month to multi-year battery life and very low complexity.” [TG4]

• Applications
  – “Potential applications are sensors, interactive toys, smart badges, remote controls,
    and home automation.” [TG4]
  – Ultimately home automation, industrial control/monitoring, vehicular sensing,
    agriculture; really most M2M sensor applications you might imagine

• Other contemporary technologies
  – WiFi 802.11b and Bluetooth Classic
    • Too complex in specification and overachieving in capability
IEEE 802.15.4

- Low-Rate Wireless PAN
  - 250 kbps, ~100 m range
  - Radio hardware available with low-power and low-cost

- Specification: 2003
    - Mostly various added capabilities such as extra PHY layers
    - Also define optional security, scheduling, and larger frame sizes

- We’ll mostly work off of the 2006 version
  - Thread is based on 2006 version
  - Zigbee is based on the original 2003 version
  - Roughly 200 pages of meaningful specification (100 of appendices)
    - Compare to 3000 pages of Bluetooth/BLE
Outline

- Overview
- Physical Layer
- Link Layer
- Packet Structure
802.15.4 Physical Layers

- Multiple options of physical layers are supported
  - We’ll focus on 2.4 GHz (2400 MHz)

<table>
<thead>
<tr>
<th>PHY (MHz)</th>
<th>Frequency band (MHz)</th>
<th>Spreading parameters</th>
<th>Data parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Chip rate (kchip/s)</td>
<td>Modulation</td>
</tr>
<tr>
<td>868/915</td>
<td>868–868.6</td>
<td>300</td>
<td>BPSK</td>
</tr>
<tr>
<td></td>
<td>902–928</td>
<td>600</td>
<td>BPSK</td>
</tr>
<tr>
<td>868/915</td>
<td>868–868.6</td>
<td>400</td>
<td>ASK</td>
</tr>
<tr>
<td>(optional)</td>
<td>902–928</td>
<td>1600</td>
<td>ASK</td>
</tr>
<tr>
<td>868/915</td>
<td>868–868.6</td>
<td>400</td>
<td>O-QPSK</td>
</tr>
<tr>
<td>(optional)</td>
<td>902–928</td>
<td>1000</td>
<td>O-QPSK</td>
</tr>
<tr>
<td>2450</td>
<td>2400–2483.5</td>
<td>2000</td>
<td>O-QPSK</td>
</tr>
</tbody>
</table>
Physical Layer

- **O-QPSK modulation**
  - Offset Quadrature Phase-Shift Keying
  - Twice the data rate of BPSK for same BER
  - Cost: most complicated design of receivers
    - Which is pretty minimal with all the transistors we’ve got
    - Plus the ability to reuse previous designs
  - 4 bits per symbol

- **Symbols versus bits**
  - A symbol is the unit of data transfer for a modulated signal
    - Does not necessarily correspond 1:1 with bits
  - The rate of symbols per second is a baudrate

- **802.15.4 bit rate at 2.4 GHz:** 2000 chips/s, which is 250 kbps, which is 62.5 kBaud
802.15.4 Modulation (@2.4 GHz $f_c$)
O-QPSK with half-sine shaping is MSK!
802.15.4 Modulation (@2.4 GHz $f_c$)

O-QPSK with half-sine shaping is MSK!

Input bit stream

Each bit of the PN code is called a chip

Each chip encodes half a sine wave

Chips alternate in-phase and quadrature

Quadrature component is offset $\pi/2$

Table:

<table>
<thead>
<tr>
<th>Bit Sequence</th>
<th>PN Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>11011001100001110101001000101110</td>
</tr>
<tr>
<td>0001</td>
<td>1101101100111000011101010010001011</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1111</td>
<td>11001010111100001110111101100010</td>
</tr>
</tbody>
</table>

Broken into 4-bit symbols

Each symbol maps to a 32-bit pseudo-noise code (PN-code) or sometimes pseudo-random sequence

I and Q half-sines are baseband, which are mixed with the carrier

I and Q carriers are combined to create the final on-air signal

Signal is MSK, which is a special, optimal case of FSK!
**Final detail:**

This shows a $f_b :: f_c$ ratio of 1 :: 10 so you can see the impact on the carrier. In reality, it’s closer to 1 :: 1200 (2,000 chips / s :: 2,400,000 Hz).
The magic of I and Q channels are that we get two dimensions

- This is called a “constellation diagram”
  - We’ll talk about these more with cellular
Constellation Diagrams give ‘at-a-glance’ understanding of modulation schemes

- What would the constellation diagram of FSK look like?
  - And what does that tell us about how the two modulation schemes compare?
Constellation Diagrams give ‘at-a-glance’ understanding of modulation schemes

- What would the constellation diagram of FSK look like?
  - And what does that tell us about how the two modulation schemes compare?

Obligatory EE Disclaimer

Many FSK frontends are implemented via IQ modulation internally...
Why do we map symbols to chips?

- We took the 4 bits we want to send...
  ... and sent 32 bits instead??

- Why?
Direct Sequence Spread Spectrum (DSSS)

- Increases the **signal** bandwidth of a transmission beyond **information** bandwidth
  - Send sequences of chips, which are a translation of one symbol to a pattern of many bits
  - Chips are transmitted much faster than symbols, essentially increasing the data rate

- Enables better interference avoidance
  - Received bits are correlated against codes to see which is most likely
  - 802.15.4 tolerates 13-15 bit flips (almost half!)
DSSS example

- Data sent is 101
  - Code is longer than data, so we replicate bits
  - Data is recoverable, even with noise

https://circuitcellar.com/research-design-hub/dsss-in-a-nutshell/
802.15.4 RF channels

- 27 channels across three bands
- 5 MHz channel separation at 2.4 GHz
  - Compare to 2 MHz for BLE
  - (or to 1 MHz for BT Classic)
Regional bands

- Different RF bands have different regional availability

- Also have different rules
  - 915 MHz: 400 ms dwell time
  - 868 MHz: 1% duty cycle
Signal strength

• Transmit power
  – Typical: 0 dBm (remember: 1 mW)

• Receiver sensitivity
  – nRF52840 802.15.4: -100 dBm
    • Compare to BLE sensitivity of -95 dBm
  – Minimum acceptable per-spec: -85 dBm
  – Circa-2006 radios (CC2420): -95 dBm

• Which has longer range, 802.15.4 or BLE? Why?
Signal strength

• Transmit power
  – Typical: 0 dBm (remember: 1 mW)

• Receiver sensitivity
  – nRF52840 802.15.4: -100 dBm
    • Compare to BLE sensitivity of -95 dBm
  – Minimum acceptable: -85 dBm
  – Circa-2006 radios (CC2420): -95 dBm

• Which has longer range, 802.15.4 or BLE? Why?
  – 802.15.4 with +5 dBm more margin; lower bit rate plays into this
Outline

• Overview

• Physical Layer

• Link Layer

• Packet Structure
802.15.4 network topologies

- Only specifies PHY and MAC, but has use cases in mind
Star and Tree topologies

- **PAN Coordinator**
  - Receives and relays all messages
  - Most capable and power-intensive

- **Coordinators (a.k.a. Routers)**
  - Control “clusters”
  - Receives and relays to its children
  - Communicates up to parent coordinator

- **End Devices**
  - Only communicate with single parent coordinator
  - Least capable and power intensive
Mesh networks

• Most devices are capable of communicating with multiple neighbors

• What are advantages of mesh?

• What are disadvantages of mesh?
**Mesh networks**

- Most devices are capable of communicating with multiple neighbors

- **What are advantages of mesh?**
  - Devices can communicate over longer distances
  - Device failures less likely to collapse the entire network

- **What are disadvantages of mesh?**
  - Some nodes have to spend more energy communicating
  - Network protocol becomes more complicated to manage routing
Reminder: CSMA/CA
Carrier Sense Multiple Access with Collision Avoidance

1. First, wait a random amount (collision avoidance part)
2. Then, listen for a duration and determine if anyone is transmitting (carrier sense part)
   - If idle, you can transmit
   - If busy, repeat step 1 (often increasing maximum wait time)

- Can be combined with notion of slotting
  - Synchronize to slots (smaller than transmit times)
  - Wait for a number of slots
  - Listen for idle slots
Modes of operation

- **Beacon-enabled PAN**
  - Slotted CSMA/CA
  - Structured communication patterns
  - Optionally with some TDMA scheduled slots

- **Non-beacon-enabled PAN**
  - Unslotted CSMA/CA
  - No particular structure for communication
    - Could be defined by other specifications, like Thread or Zigbee
Beacon-enabled superframe structure

- Beacons occur periodically [15 ms – 245 seconds]
  - Devices must listen to each beacon

- Contention Access Period
  - Slotted CSMA/CA synchronized by beacon start time

- Inactive Period
  - No communication occurring. Assumes sleepy devices
Guaranteed Time Slots (GTS)

- PAN Coordinator may create a Contention Free Period with Guaranteed Time Slots
  - TDMA schedule assigned to specific devices
  - Slots eat up part of the Contention Access Period
  - No CSMA/CA within a slot
Handling tree-based topologies

- All coordinators listen to beacon from PAN coordinator
  - And can participate in that contention period

- Send their own beacons to child devices during inactive period
  - Children participate in that contention period
Non-beacon-enabled PAN

- Same idea, just no beacons
  - Which removes synchronization benefit (and slotted CSMA/CA)
  - Also removes beacon listening cost
    - Devices only need to check for activity before transmitting
  - Still need an algorithm to determine when it should receive data
    - All the time is a huge energy drain
    - Algorithms can get complicated here
    - Does BLE mechanism of listen-after-send apply?
Non-beacon-enabled PAN

- Same idea, just no beacons
  - Which removes synchronization benefit (and slotted CSMA/CA)
  - Also removes beacon listening cost
    - Devices only need to check for activity before transmitting
  - Still need an algorithm to determine when it should receive data
    - All the time is a huge energy drain
    - Algorithms can get complicated here
    - **Does BLE mechanism of listen-after-send apply?**
      - Only if sending to a high-power device, not among equals
Receiving messages

1. Listen during entire contention period
   - Can receive direct messages from any other device
   - Can immediately respond to messages as well

2. Request messages from Coordinator
   - Make all communication go through Coordinator
   - Send a request-for-data packet to coordinator to get information
   - Coordinator can include list of devices with pending data in beacon

• More complicated listening algorithms are possible
Clear Channel Assessment (CCA)

- The “listen” part of CSMA/CA
- Variety of implementations are acceptable
  1. Energy above threshold
     - Energy for 8 symbol durations above threshold (RSSI)
  2. Carrier sense
     - Valid 802.15.4 carrier signal
  3. Energy AND/OR Carrier
Slotted CSMA/CA operation

- Have data to send
- Wait for next backoff slot (synchronized from beacon)
- Wait for 0-7 backoff slots (slot is 20 symbol durations: 320 us)
- Listen for two empty slots
  - Idle: Transmit
  - Occupied: wait 0-15 backoff slots and repeat
    - Next time: 0-31 backoff slots and repeat
    - Next time: 0-31 backoff slots and repeat (upper limit configurable)
    - Next time: 0-31 backoff slots and repeat
    - Next time: 0-31 backoff slots and repeat
    - Timeout
Unslotted CSMA/CA operation

- Have data to send
- Wait for next backoff slot (synchronized from beacon)
- Wait for 0-7 backoff slots (slot is 20 symbol durations: 320 us)
- Listen for two empty slots
  - Idle: Transmit
  - Occupied: wait 0-15 backoff slots and repeat
    - Next time: 0-31 backoff slots and repeat
    - Next time: 0-31 backoff slots and repeat (upper limit configurable)
    - Next time: 0-31 backoff slots and repeat
    - Next time: 0-31 backoff slots and repeat
    - Timeout
Break + Question

• What are benefits/costs of using or not using beacons?
Break + Question

• What are benefits/costs of using or not using beacons?

  – Beacons
    • Enable energy savings by designating period with radios off
    • Enable structured communication like Guaranteed Slots
    • Require some central coordinator within range of all devices
    • Tradeoff in inactive period:
      – communication latency vs beacon-listening costs

  – No beacons
    • Enable all devices to be identical (no coordinator needed)
    • Require custom communication scheme
      – Could be better or worse for various qualities... (always-on radios?)
Outline

- Overview
- Physical Layer
- Link Layer
- Packet Structure
Base packet format

- **Synchronization**
  - Preamble: four bytes of zeros
  - Start-of-Packet: 0xA7

- **PHY Header**
  - One field: length 0-127
  - Why still 8 bits?
Base packet format

- **Synchronization**
  - Preamble: four bytes of zeros
  - Start-of-Packet: 0xA7
- **PHY Header**
  - One field: length 0-127
  - Why still 8 bits? Because computers depend on bytes
MAC frame format

- **Frame control**
  - **Header**
  - **Sequence number**
    - 8-bit monotonically increasing
  - **Addressing fields**
    - PAN and addresses
    - Varies based on frame type
- **Frame payload**
  - Depends on frame type
- **Frame check sequence**
  - 16-bit CRC
Frame control

- Frame type
  - Type of payload included
- Security enabled
  - Packet is encrypted
  - (extra 0-14 byte header)
- Frame pending
  - Fragmented packet

- Acknowledgement required
- PAN ID compression
  - No PAN ID if intra-network
- Addressing modes
  - Which fields to expect

Why no length field?


<table>
<thead>
<tr>
<th>Octets:2</th>
<th>1</th>
<th>0/2</th>
<th>0/2/8</th>
<th>0/2</th>
<th>0/2/8</th>
<th>variable</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame control</td>
<td>Sequence number</td>
<td>Destination PAN identifier</td>
<td>Destination address</td>
<td>Source PAN identifier</td>
<td>Source address</td>
<td>Frame payload</td>
<td>Frame check sequence</td>
</tr>
</tbody>
</table>

**Addressing fields**

<table>
<thead>
<tr>
<th>Bits: 0-2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7-9</th>
<th>10-11</th>
<th>12-13</th>
<th>14-15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame type</td>
<td>Security enabled</td>
<td>Frame pending</td>
<td>Ack. Req.</td>
<td>PAN ID compression</td>
<td>Reserved</td>
<td>Dest. addressing mode</td>
<td>Frame version</td>
<td>Source addressing mode</td>
</tr>
</tbody>
</table>

- **Frame type**
  - Type of payload included
- **Security enabled**
  - Packet is encrypted
  - (extra 0-14 byte header)
- **Frame pending**
  - Fragmented packet
- **Acknowledgement required**
- **PAN ID compression**
  - No PAN ID if intra-network
- **Addressing modes**
  - Which fields to expect

Why no length field?

Already in prior header
Frame types - Beacon

- **Beacon**
  - Information about the communication structure of this network
  - Sent in response to requests from scanning devices
  - Sent periodically at start of Superframes (if in use)
    - Sent without CSMA/CA

- **MAC Header**
  - Source address only, broadcast to everyone

- **Packet contents**
  - Superframe details, including Guaranteed Time Slots (if any)
  - Pending addresses lists devices for which Coordinator has data
Frame types - Data

- **Data**
  - Data from higher-layer protocols

- **MAC Header**
  - Source and/or Destination addresses as necessary

- **Packet Contents**
  - Whatever bytes are desired (122 bytes – address sizes)
  - May be fragmented across packets
Frame types – MAC Command

- **MAC Command**
  - Various commands for supporting link layer
    - Join/leave network
    - Change coordinator within network
    - Request data from coordinator
    - Request Guaranteed Time Slot

- **MAC Header**
  - Source and/or Destination addresses as necessary
Frame types - Acknowledgement

• Acknowledgement
  – Acknowledges a Data or MAC Command packet
  – Not beacons or other acknowledgements
    • What happens if acknowledgement isn’t received?

• MAC Header
  – Repeats Sequence Number of acknowledged packet
  – No Source or Destination addresses

• Sent $T_{IFS}$ after the packet it is acknowledging (immediately)
Frame types - Acknowledgement

- **Acknowledgement**
  - Acknowledges a Data or MAC Command packet
  - Not beacons or other acknowledgements
    - **What happens if acknowledgement isn’t received?**
      - Packet will be transmitted again

- **MAC Header**
  - Repeats Sequence Number of acknowledged packet
  - No Source or Destination addresses

- **Sent** $T_{IFS}$ **after the packet it is acknowledging (immediately)**
Analysis: maximum goodput

• Assume best possible case for data transmission
  – 122 Bytes per packet
    • At 250 kbps -> 3.904 ms
  – Plus Inter-frame spacing of 40 symbols
    • At 62.5 kBaud -> 0.640 ms

  – 122 Bytes / 4.544 ms -> 214 kbps
    • Compare to BLE advertisements: 9.92 kbps
    • Compare to BLE connections: 520 kbps
Next time: More 802.15.4