Bluetooth Mesh

Robin Heydon
Senior Director, Technology, Qualcomm
Two Models

Advertising Based
- scan for information from advertisers
- advertisers send data periodically
- don’t know who will get it or what they may do with that

Connection Based
- connect to another devices
- perform queries on that device (read / write)
- setup notifications when values change
Connection Based

useful when...
– the two devices are always next to each other
– only need to send data to one other device
– devices are bonded (securely connected once)

not useful when...
– many devices that need the same information
– devices may be out of range
– devices may only ever talk with another device once
HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE’S
USE CASES. YEAH!

[Soon:]

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.
Start with a problem that needs solving

“I want to control all my lights from my phone”
“at the same time”
“they should all change instantly”
“yes, I want it secure”
“yes, it should always work”
“no, an extra box costs too much and is complex”
“err, I have a big house, a very big house”
“I can’t do that with any standard out there today”
Derive Requirements

“I want to control all my lights from my phone”

– turn them on/off
– change their colour
– dim lights
– change colour / dimming over a period of time
Derive Requirements

“at the same time”
  – implies a broadcast mechanism

“they should all change instantly”
  – disallows connect / send / disconnect / connect to next light

“yes, I want it secure”
  – authentication
  – encryption
  – authorization
  – protect against attacks
Derive Requirements

“yes, it should always work”
  – must be robust against interference

“no, an extra box costs too much and is complex”
  – must use technology that is already in phones
  – implies either Cellular, Bluetooth or Wi-Fi

“err, I have a big house, a very big house”
  – big houses have lots of rooms
  – lots of rooms have lots of walls
  – walls have up to 12 dB of attenuation
Mesh is really Multi-Hop

Assume:

- transmit at 10 dB (maximum)
- receive at -90 dB (typical minimum sensitivity)
- \( n \) walls

implies approximately 100 dB “range”
assume pathloss model of \( pl = 40 + 25 \log(d) \)

100 dB implies \(~250\) metre range
More walls means less distance

<table>
<thead>
<tr>
<th>Number of Walls</th>
<th>Resultant Signal (dB)</th>
<th>Resultant Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>251</td>
</tr>
<tr>
<td>1</td>
<td>88</td>
<td>83</td>
</tr>
<tr>
<td>2</td>
<td>76</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>9.1</td>
</tr>
<tr>
<td>4</td>
<td>52</td>
<td>3.0</td>
</tr>
</tbody>
</table>
10 dB Tx uses lots of power

<table>
<thead>
<tr>
<th>Number of Walls</th>
<th>Resultant Signal (dB)</th>
<th>Resultant Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>85</td>
<td>63</td>
</tr>
<tr>
<td>1</td>
<td>78</td>
<td>21</td>
</tr>
<tr>
<td>2</td>
<td>66</td>
<td>6.9</td>
</tr>
<tr>
<td>3</td>
<td>54</td>
<td>2.3</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>0.76</td>
</tr>
</tbody>
</table>
Derived Requirements

Use Bluetooth low energy
  – because it is in phones

Use a “Broadcast” channel to send data
  – because we want “quick” and “simple”

This implies we are advertising...
## Protocols on Advertising Channels

### Generic Advertising Packets (Length Tag Value)

<table>
<thead>
<tr>
<th>Len</th>
<th>Tag</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 to n</td>
<td>0 to m</td>
</tr>
</tbody>
</table>

### iBeacon

Apple have multiple protocols inside their tag format 0x02 = iBeacon

<table>
<thead>
<tr>
<th>Len</th>
<th>Vendor</th>
<th>Apple</th>
<th>0x02</th>
<th>Len</th>
<th>UUID</th>
<th>Major</th>
<th>Minor</th>
<th>Tx Power</th>
</tr>
</thead>
</table>
## Protocols on Advertising Channels

### Generic Advertising Packets (Length Tag Value)

<table>
<thead>
<tr>
<th>Len</th>
<th>Tag</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 to n</td>
<td>0 to m</td>
</tr>
</tbody>
</table>

### iBeacon

<table>
<thead>
<tr>
<th>Len</th>
<th>Vendor</th>
<th>Apple</th>
<th>0x02</th>
<th>Len</th>
<th>UUID</th>
<th>Major</th>
<th>Minor</th>
<th>Tx Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

They have a second length !!!
Protocols on Advertising Channels

Generic Advertising Packets (Length Tag Value)

<table>
<thead>
<tr>
<th>1</th>
<th>1 to n</th>
<th>0 to m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Len</td>
<td>Tag</td>
<td>Value</td>
</tr>
</tbody>
</table>

iBeacon

each iBeacon has a unique UUID

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>16</th>
<th>2</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Len</td>
<td>Vendor</td>
<td>Apple</td>
<td>0x02</td>
<td>Len</td>
<td>UUID</td>
<td>Major</td>
<td>Minor</td>
<td>Tx Power</td>
</tr>
</tbody>
</table>
## Protocols on Advertising Channels

### Generic Advertising Packets (Length Tag Value)

<table>
<thead>
<tr>
<th>Len</th>
<th>Tag</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 to n</td>
<td>0 to m</td>
</tr>
</tbody>
</table>

### iBeacon

<table>
<thead>
<tr>
<th>Len</th>
<th>Vendor</th>
<th>Apple</th>
<th>0x02</th>
<th>Len</th>
<th>UUID</th>
<th>Major</th>
<th>Minor</th>
<th>Tx Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
# Protocols on Advertising Channels

## Generic Advertising Packets (Length Tag Value)

<table>
<thead>
<tr>
<th></th>
<th>1 to n</th>
<th>0 to m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Len</td>
<td>Tag</td>
<td>Value</td>
</tr>
</tbody>
</table>

## iBeacon

- Len: 1
- Vendor: Apple (0x02)
- Len: 1
- UUID: 16
- Major: 2
- Minor: 2
- Tx Power: 1

Communicating Tx Power allows “range” to be estimated.
Mesh Advertising Protocol

Generic Advertising Packets (Length Tag Value)

<table>
<thead>
<tr>
<th>1</th>
<th>1 to n</th>
<th>0 to m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Len</td>
<td>Tag</td>
<td>Value</td>
</tr>
</tbody>
</table>

Mesh Packet

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>1</th>
<th>n</th>
<th>4 or 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Len</td>
<td>“Mesh”</td>
<td>TTL</td>
<td>Mes</td>
<td>MAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>h Message</td>
<td></td>
</tr>
</tbody>
</table>
Mesh Packet

<table>
<thead>
<tr>
<th></th>
<th>n</th>
<th>4 or 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTL</td>
<td>Message</td>
<td>MAC</td>
</tr>
</tbody>
</table>

- Message is opaque at this layer
- MAC authenticates Message against “Network Key”
  - e.g. HMAC (SHA256, Message, Network Key)
- TTL used to limit extent of relays
Mesh TTLs

TTL = 0
   – never relayed

TTL = 1
   – has been relayed

TTL > 1
   – can be relayed (after decrement)
Authentication

Shared “NetworkKey” in every mesh device
  – provisioning devices is secure transfer of network key
  – let’s ignore authorization of that transfer
    • ECDH with Out of Band is the correct answer

Allows each network to have separate NetworkKey
  – only relay messages from your network(s)

64-bits is compromise between robust enough and insecure
  – can receive “random noise”
  – protect against false positives
def process_mesh_packet (message, MAC, TTL):
    for key in known_network_keys:
        known_key_MAC = HMAC_sha256 (message, key)
        if known_key_MAC == MAC:
            process_network_message (message, TTL)
        if TTL > 1 and not in_cache (message):
            transmit (message, MAC, TTL - 1)
            add_to_cache (message)
Cached Messages?

don’t retransmit messages that you’ve already retransmitted

size of cache is implementation specific
  – should be big enough for number of message in flight
  – can be bigger without external costs
  – can be smaller (at expense of more retransmissions)

removal algorithm is implementation specific
  – least recently used
  – least frequently used
Message Format

SEQ – Sequence Number
SRC – Source Device Address
DST – Destination Device Address / Group Address
OP – Operation Code
Params – Parameters for the operation
TID – Transaction Identifier
Message Format Details

Sequence number – 24 bits
  – unique for each message sent (per source device)
  – compromise between number of messages that can be sent and time before sequence numbers exhausted

SRC – 16 bits
  – allocated during provisioning of device
  – combined with SEQ to provide nonce for encryption

DST – 16 bits
  – can be a device address or a group address
  – group addresses from same name space
Limits

24-bit sequence numbers
- 16 million messages can be sent by a device before needing to re-key whole network
- assume one message every 5 seconds
- 2.6 years before re-key required

16-bit device / group addresses
- 32767 devices (one “unassigned” value reserved)
- 32767 groups (one “broadcast all” value reserved)
- my house has 18360 bricks
  - one device per brick?
Message Format Details

Opcodes – 8 or 16 bits
   – grouped into “models” – allows introspection of devices

Parameters – 1 to 10 octets
   – meaning determined by opcode

TID – Transaction Identifier – 8 bits
   – allows retransmission of messages
   – allows matching of transactions
Example Transaction – Turn it on

Switch

POWER_SET_STATE_NO_ACK

Device
Example Transactions – Dimmers

Switch

Light

LIGHT_SET_LEVEL_NO_ACK
LIGHT_SET_LEVEL_NO_ACK
LIGHT_SET_LEVEL_NO_ACK
LIGHT_SET_LEVEL_NO_ACK
LIGHT_SET_LEVEL_NO_ACK
LIGHT_SET_LEVEL_NO_ACK
LIGHT_SET_LEVEL_NO_ACK
LIGHT_SET_LEVEL_NO_ACK

LIGHT_SET_LEVEL

LIGHT_STATE
Example Transactions – What colour?
Asset Tracking

TTL = 0 means “local advertisement”
   – could be from an iBeacon or similar type of device

Assets broadcast periodically
   – other devices receive and store RSSI / last time seen

Send “Find” message into network

Devices respond with “Found” with “age” and “RSSI”
   – need to use a back-off algorithm
Routing?

no I’ve not mentioned routing
  – not because it is not interesting
  – but because routing is “easy”, “well known”

not routing is also interesting though
  – when devices send messages all devices receive them
  – all devices can remember this state
  – this allows for some interesting optimisations
    • e.g. Group State
Interesting Problems

How do you send a reliable message to many devices?
  – especially if you want to reduce congestion
  – and don’t know how many devices will respond

Asymmetric authentication of messages
  – knowledge of symmetric NetworkKey allows anybody to Rx and Tx messages
  – asymmetric signatures are “HUGE”

How do you provision 10,000 devices a day?
  – and it must be secure
Conclusions

Mesh can be made simple or complex

Biggest problem is not “routing”
  – provisioning
  – device configuration
  – adhoc network management

Security is always important
  – symmetric key and sufficiently sized MAC
  – think of the attacks – (“replay”, “trash can”).join (“attack”)
thank you