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

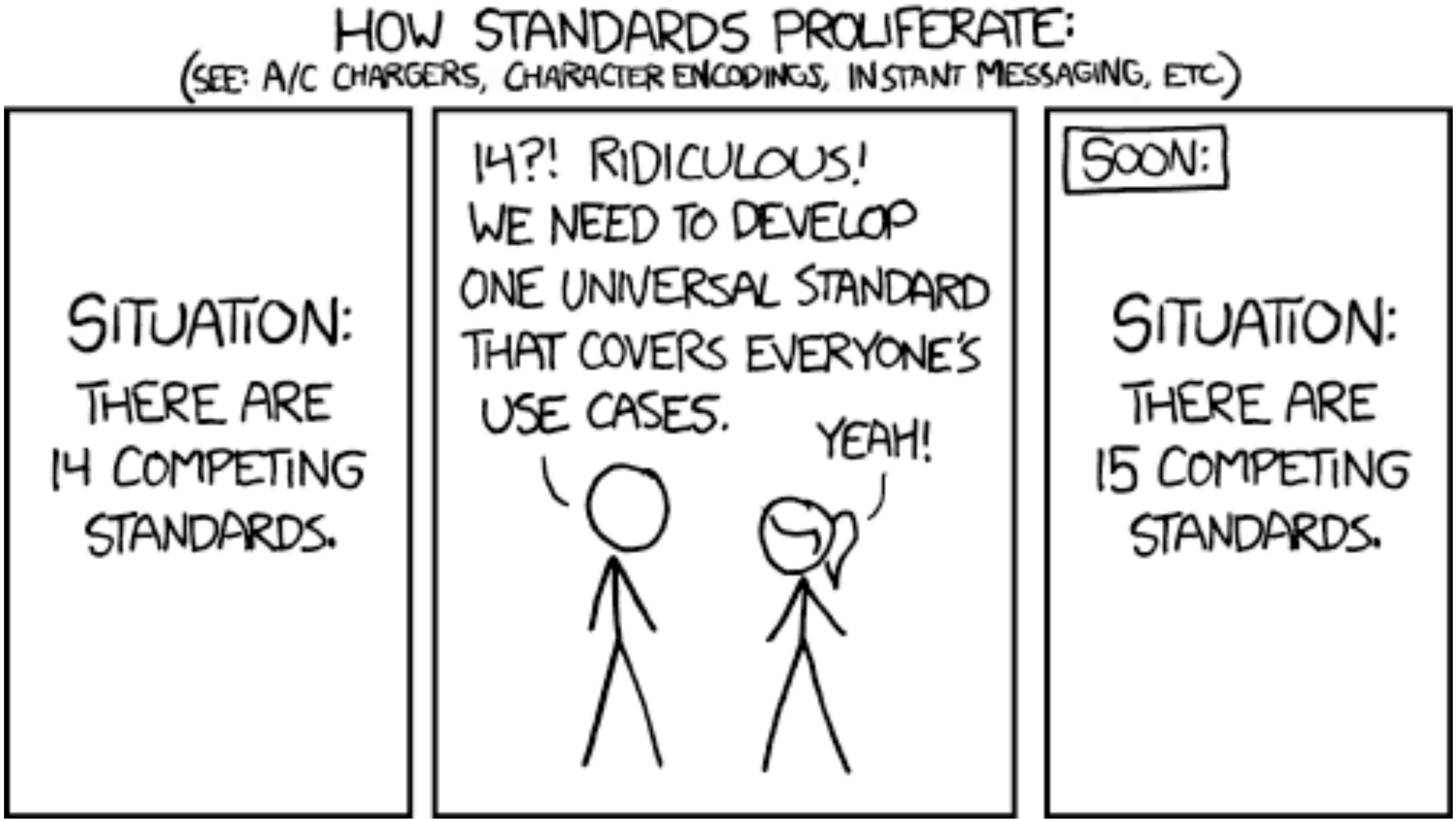
Connection Based

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



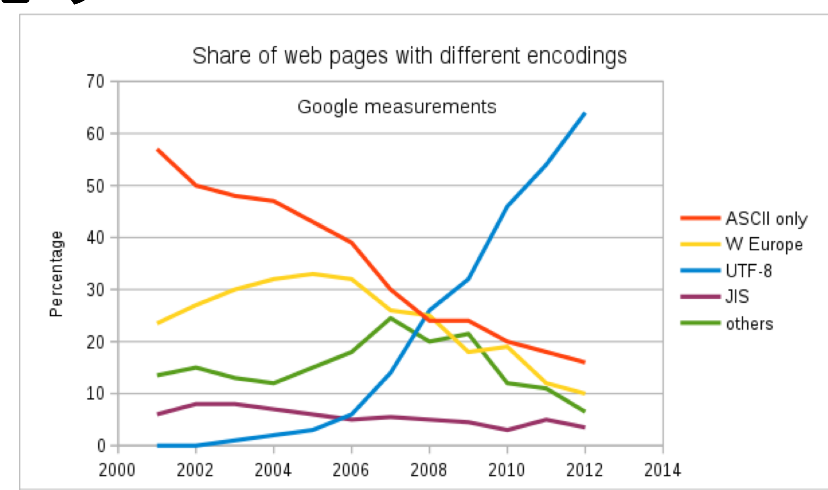
https://xkcd.com/927/

There is only one known exception to this rule...

Unicode / UTF-8

I think it is only because of Emoji 

私はそれが絵文字のためだけだと思う



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"

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 0 dB
- receive at -95 dB
- *n* walls
- each wall causes 12 dB signal loss

implies up to maximum 95 dB "pathloss"

assume pathloss model of $40 + 25 \times \log(d)$

85 dB implies ~158 metre range

Walls significantly reduce distance

Number of Walls	Available Pathloss after Walls (dB)	Resultant Distance (m)
0	95	158
1	83	52
2	59	5.4
3	23	-0.29

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)

1	**1 to n**	**0 to m**
Len	Tag	Value

iBeacon

1	**1**	**2**	**1**	**1**	**16**	**2**	**2**	**1**
Len	Vendor	Apple	0x02	Len	UUID	Major	Minor	Tx Power

Protocols on Advertising Channels

Generic Advertising Packets (Length Tag Value)

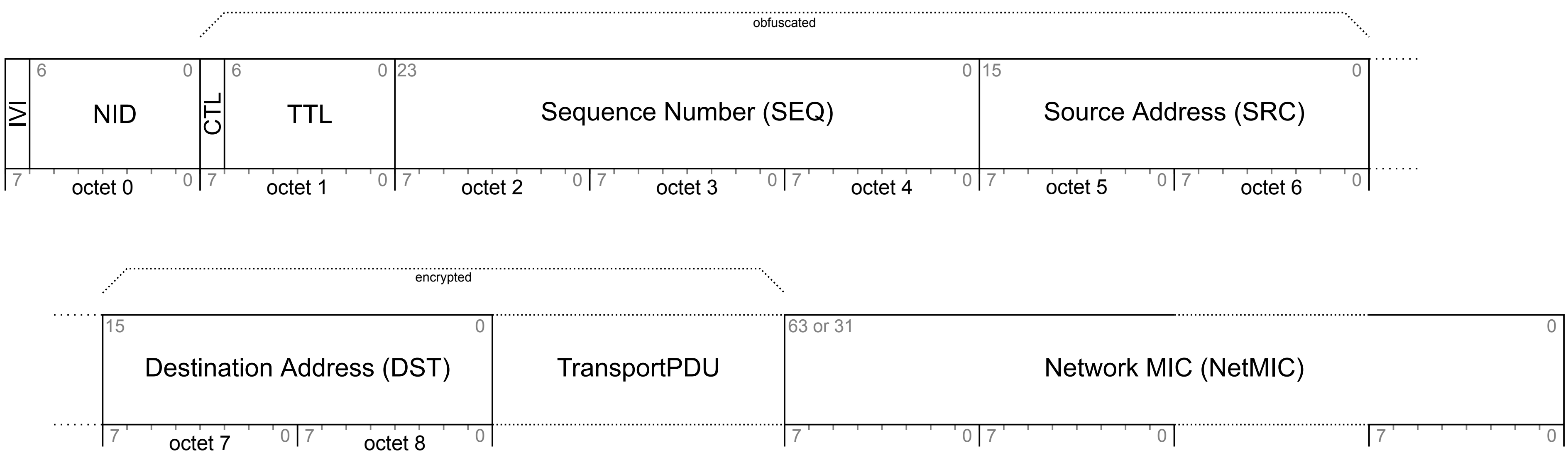
1	**1 to n**	**0 to m**
Len	Tag	Value

Mesh Message

1	**1**	**18 to 29**
Len	“Mesh”	Network PDU

Mesh Message

Network PDU



IVI / NID

Initialization Vector (least significant bit)

Network Identifier

- 7-bit hash of Network Key
- not-human generated

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
  - 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

Relaying

Receive Packet Pseudo-Code

```
def process_mesh_packet (TTL, message, MIC):
    for key in known_network_keys:
        known_key_MIC = authenticate (message, key)
        if known_key_MIC == MIC:
            process_network_message (message, TTL)
            if TTL > 1 and !in_cache (message):
                transmit (TTL – 1, message)
                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 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

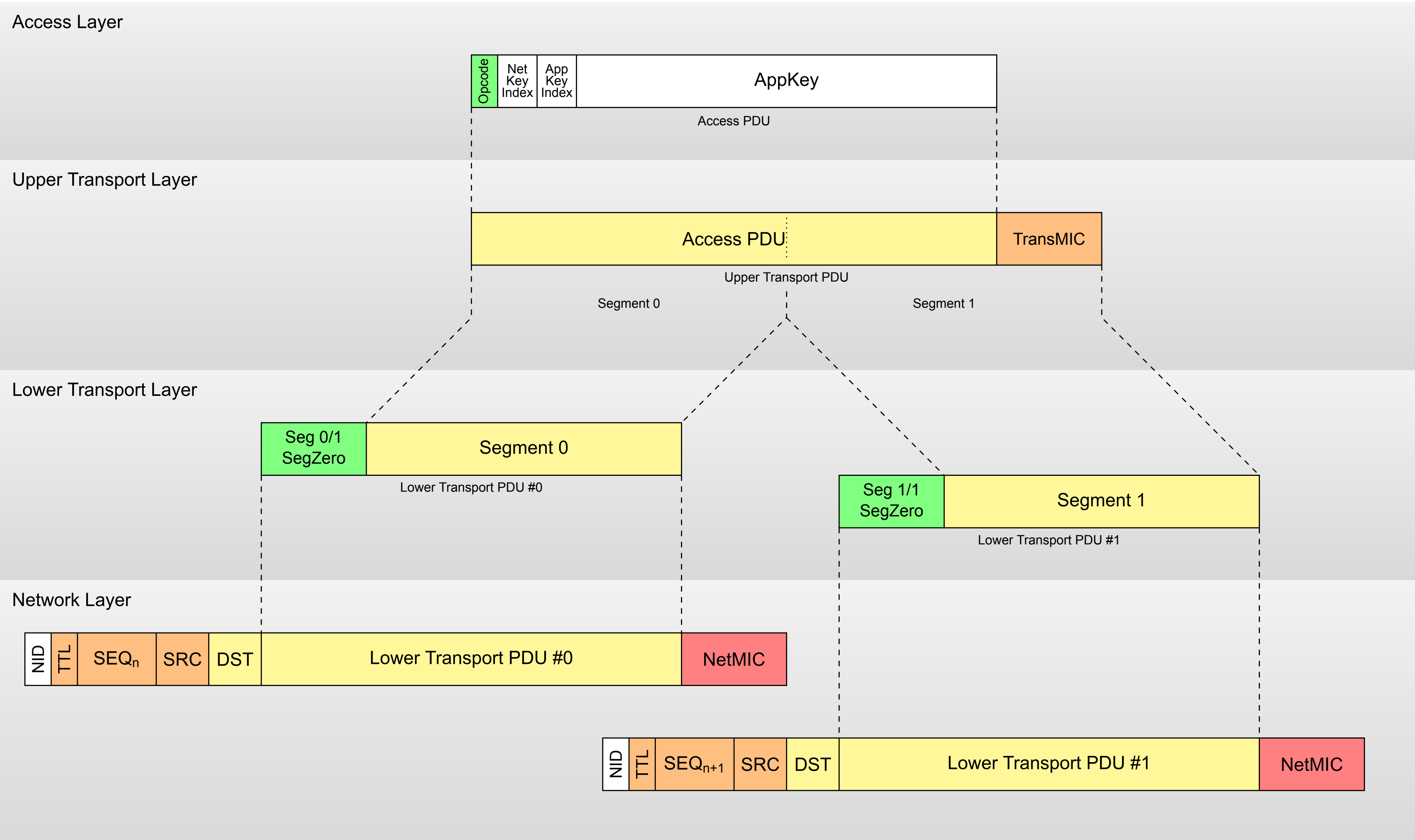
24-bit sequence numbers

- 16 million messages can be sent by a device before needing to re-key whole network
- assume one message every 250ms
- 48 days before new IV Index is required
  - IV Index is 32-bit value
  - 571,000,000 years before re-key required

16-bit device / group addresses

- 32767 devices
- 32768 groups

Transport Layer



Application Format Details

Opcodes – 8 or 16 bits

- grouped into “models” – allows introspection of devices

Parameters – 1 to 11 octets

- meaning determined by opcode

TID – Transaction Identifier – 8 bits

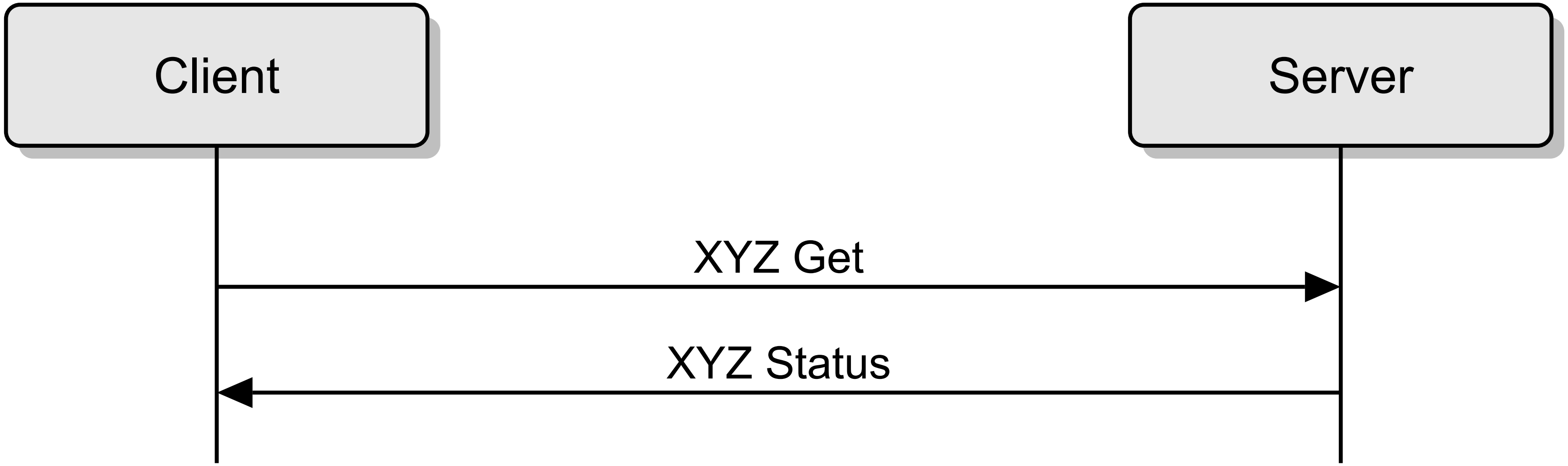
- allows retransmission of messages
- allows matching of transactions

Publish / Subscribe

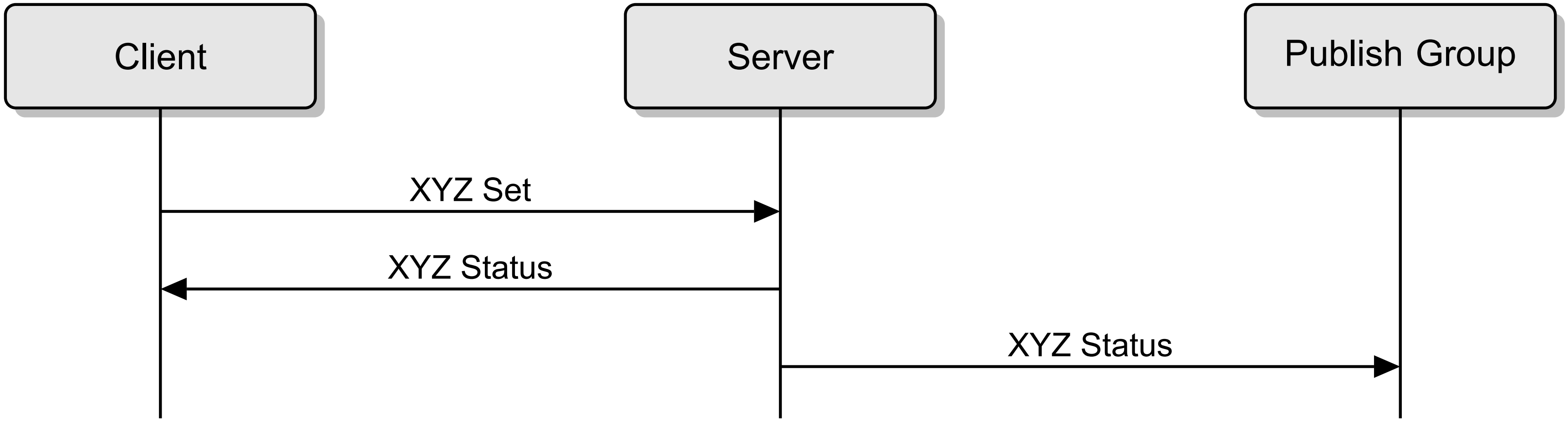
Models (services within a mesh node)

- can publish information
- can publish information periodically
- can be asked for current state
- can be asked to set state

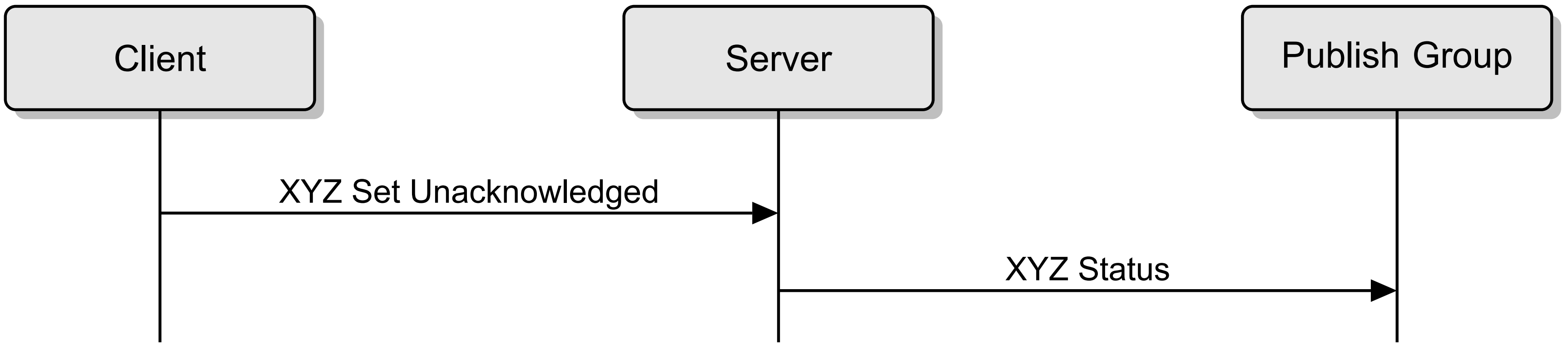
Getting State



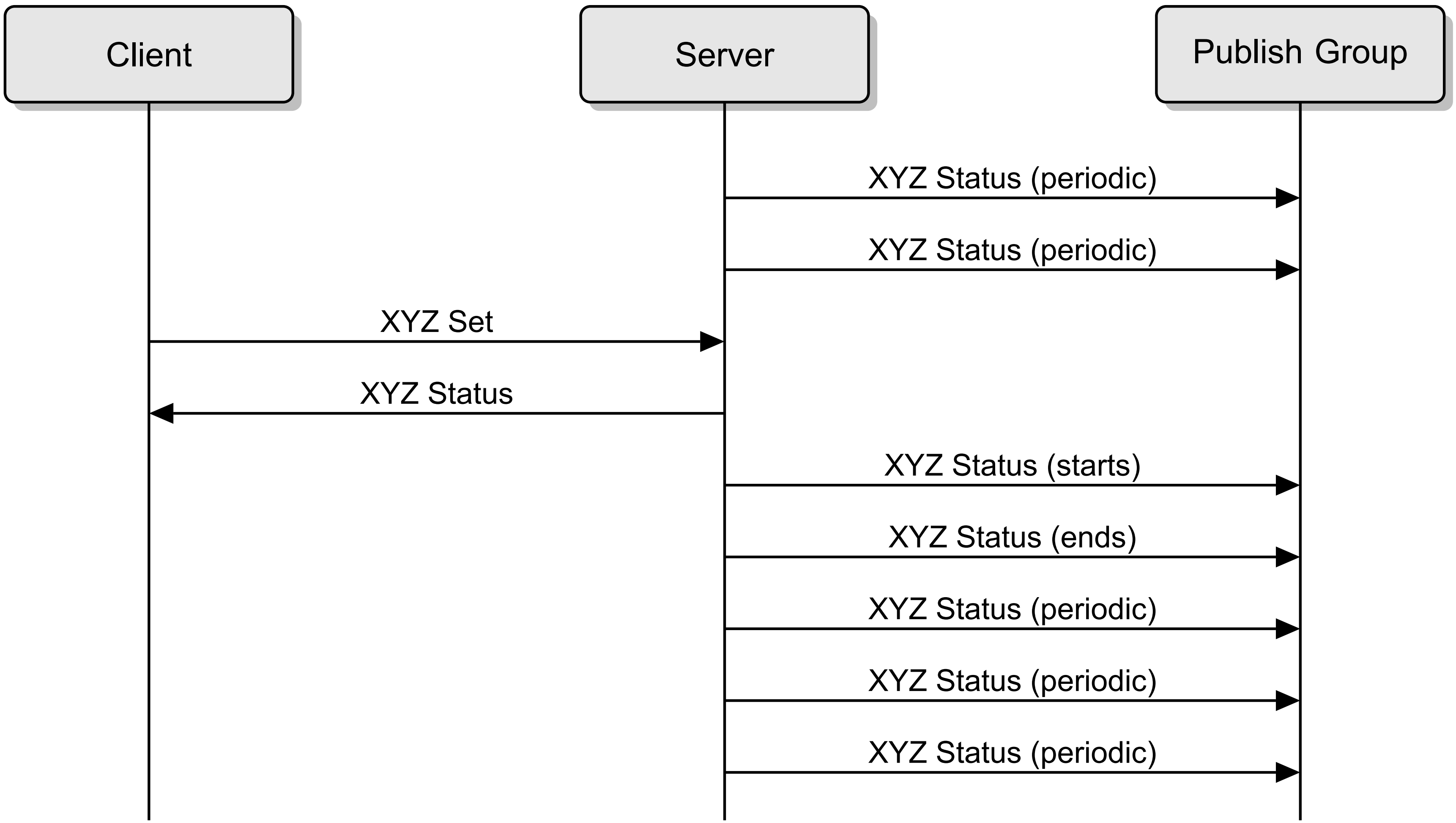
Setting State (with publishing)



Setting State (not acknowledged)



Setting State (with periodic publishing)



Deciding names of things difficult

Originally we called "Unreliable Set"

- Marketing thought this sounded terrible
- “How do you sell an unreliable feature?”

Changed to "Unacknowledged Set"

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 / when

Send “Find” message into network

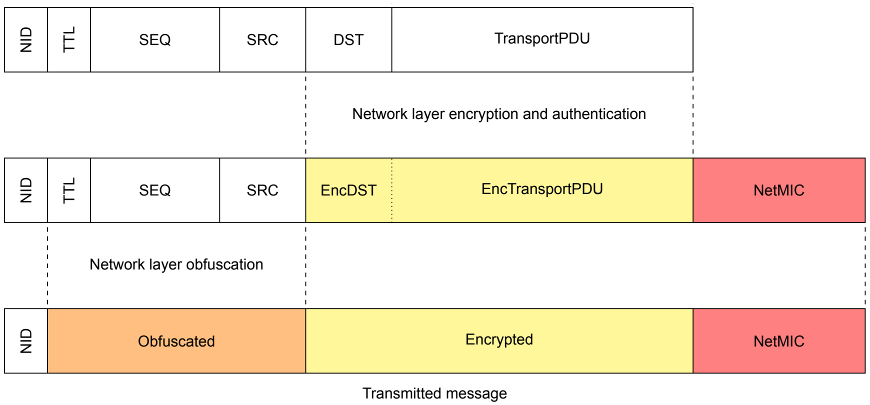
Devices respond with “Found” with “age” and “RSSI”

- need to use a back-off algorithm

Security Considerations

Two layers of security (Network / Application)

- Network
- Application



Encryption / Authentication

Obfuscation

"Trash can" attack

- discard device in the trash can (rubbish bin)
- must re-key whole network
    - application keys and network keys

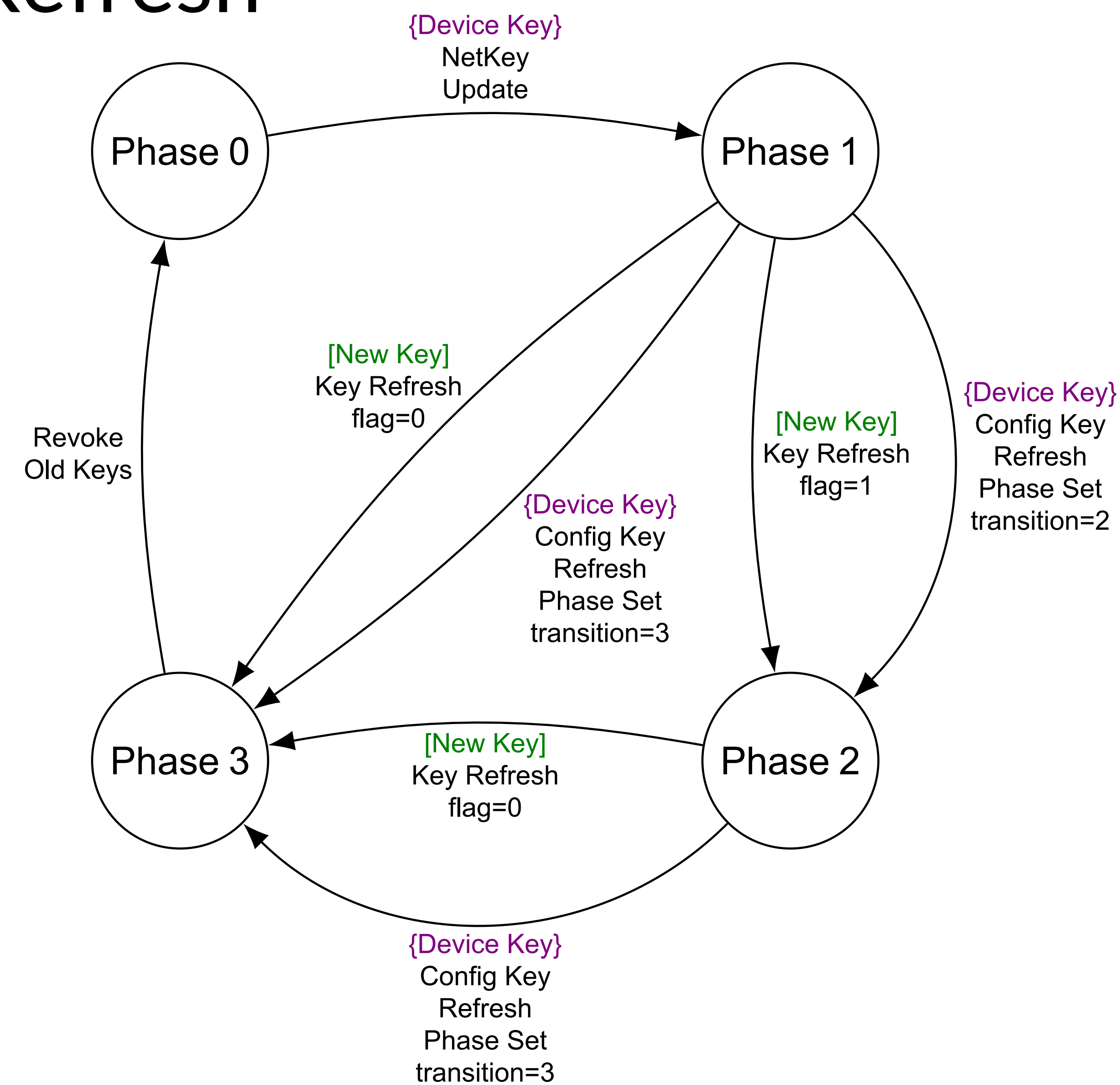
Device Key / Application Key / Network Key

Device Key
- allows "provisioner" to communicate only one device

Application Key
- allows an application to communicate with others
- door bell / door lock / lighting

Network Key
- allows a network to defined
- home network / guest network

Key Refresh


{Device Key}
NetKey
Update
Phase 0
Phase 1
Revoke
Old Keys
[New Key]
Key Refresh
flag=0
{Device Key}
Config Key
Refresh
Phase Set
transition=3
[New Key]
Key Refresh
flag=1
{Device Key}
Config Key
Refresh
Phase Set
transition=2
Phase 3
Phase 2
[New Key]
Key Refresh
flag=0
{Device Key}
Config Key
Refresh
Phase Set
transition=3

Low Power Considerations

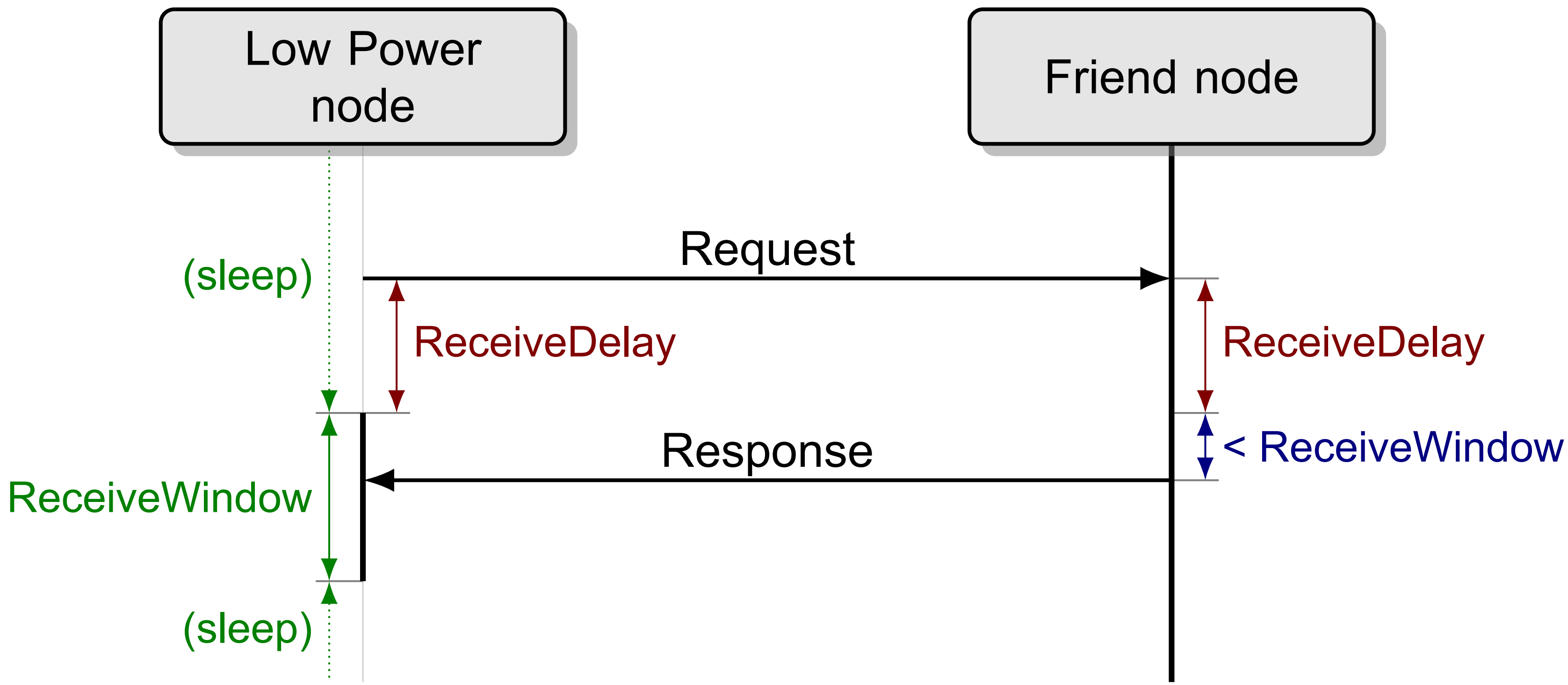
Many devices will be powered by batteries

Cannot listen all the time

Friendship / Low Power Nodes

- LPNs find a local friend
- Friend promises to queue up all* incoming messages
- LPN polls Friend when it wants to check for messages
- Friend responds at a known time with new messages
    - and also new security information

Friendship requests / responses



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”
  - e.g. ECDSA signatures (using P-256 curve) are 64 octets

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
    - Group State

Conclusions

Mesh can be made simple or complex

Biggest problem is not “routing”

- provisioning
- device configuration
- adhoc network management

Security is always important

- symmetric keys and sufficiently sized MAC
- think of the attacks – (“replay”, “trash can”).join (“attack”)

thank you