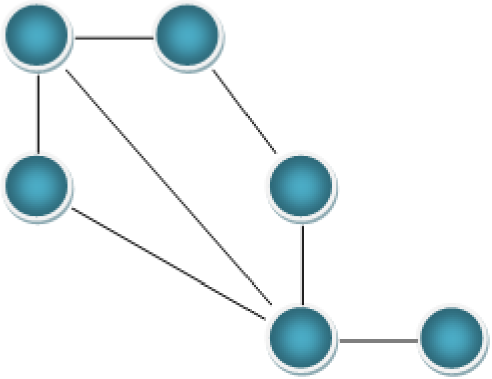
BLUETOOTH® MESH

Johan Hedberg

What is Bluetooth Mesh?

- New standard which came out in 2017
- Many-to-many, multi-hop topology
- No new Bluetooth HW required
- Broadcast & relay in a flooding/ripple fashion
- Mainly for signaling – not large data transfer
- Message publication & subscription
- Multi-level security
- Greatly extended range



Mesh in terms of Bluetooth

BR/EDR

for continuous connections

pairing (1:1)

audio streaming

- wireless speakers
- wireless headsets
- in-car infotainment

2016: 730M | 2020: 930M

Low Energy (LE)

for short burst connections

pairing (1:1)

data transfer

- sports & fitness devices
- medical & healthcare devices
- peripherals & accessories

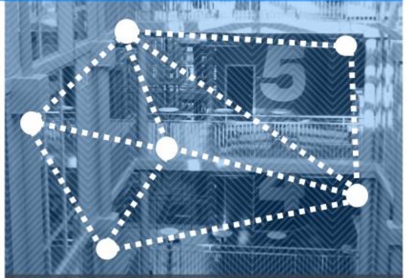
2016: 573M | 2020: 975M

broadcasting (1:m)

localized info sharing

- PoI information
- item finding
- way finding

2016: 12M | 2020: 380M



mesh networking (m:m)

large device networks

- building automation
- sensor networks
- asset tracking

Launching mid-2017

Mesh in terms of LE roles

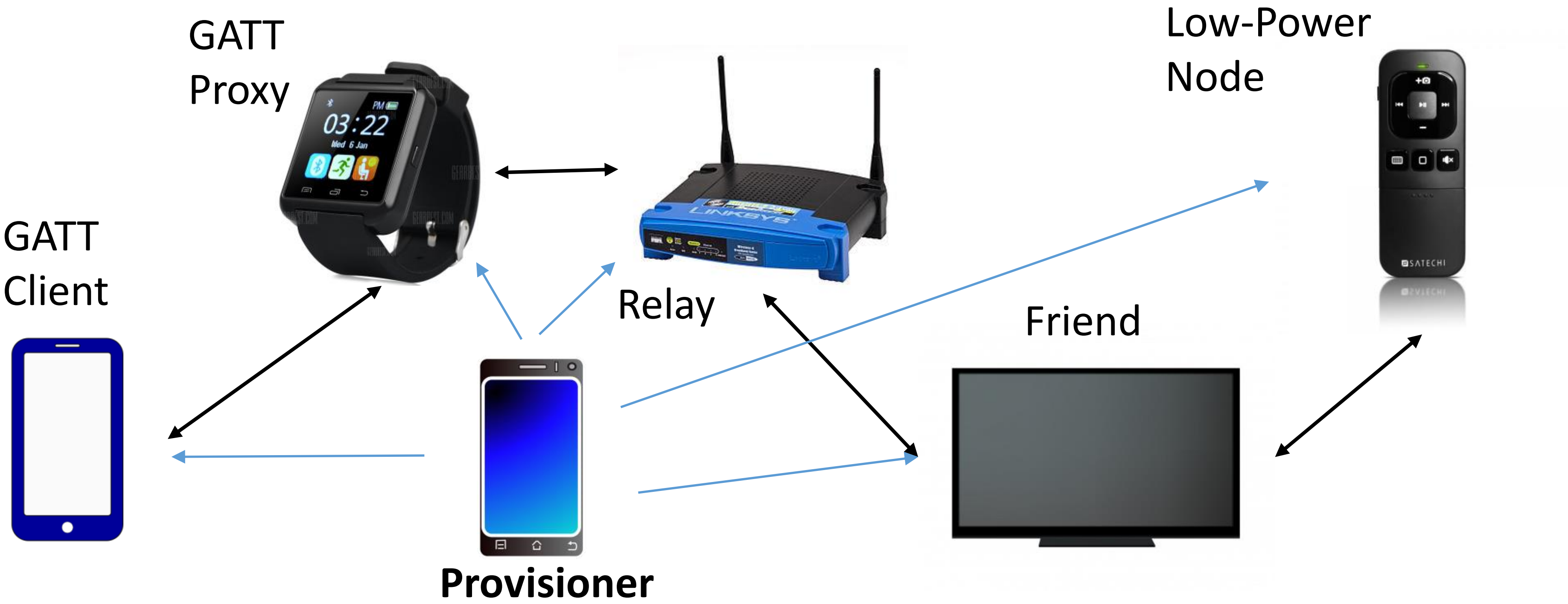
Central - Peripheral

- Connection-oriented, between two devices
- Sensor as peripheral, your phone or PC as the central

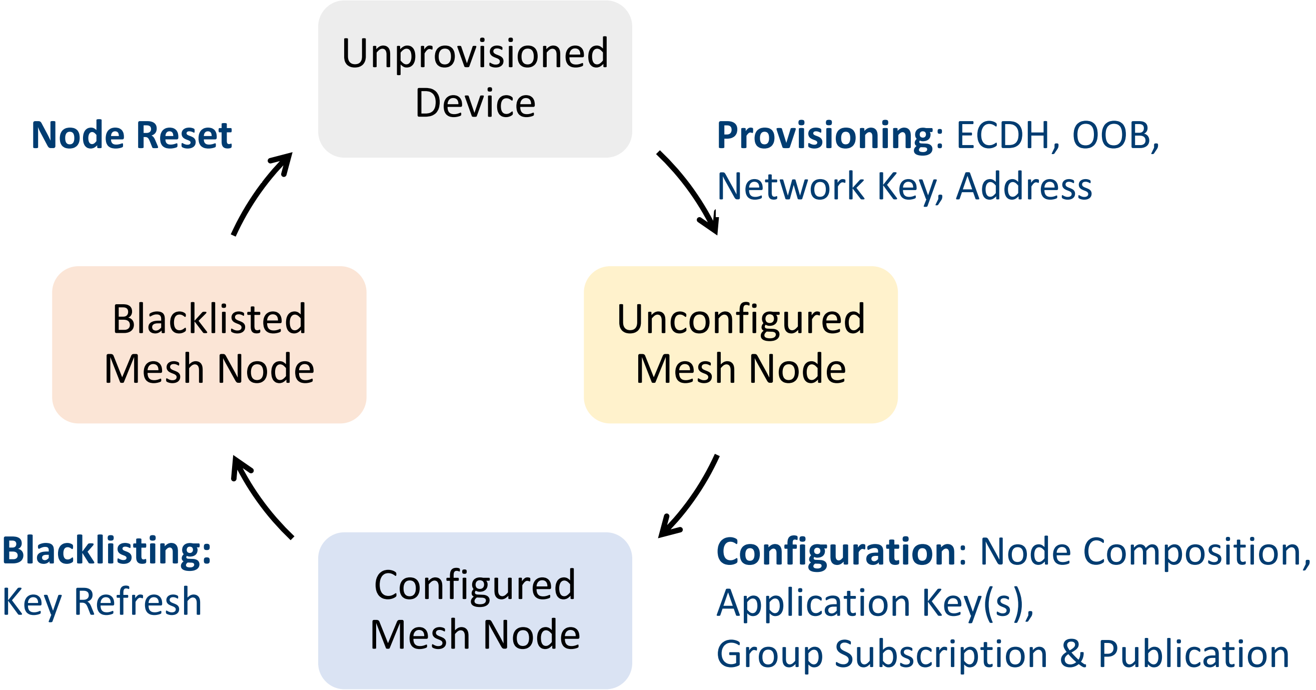
Observer - Broadcaster

- Observer scans for advertising packets
- Broadcaster sends advertising packets for everybody who is scanning
- The natural choice for Mesh

Node Types


GATT Proxy
Low-Power Node
GATT Client
Relay
Friend
Provisioner
03:22
Wed 6 Jan
LINKSYS
SATECHI

Node Lifecycle



Unprovisioned Device

Provisioning: ECDH, OOB, Network Key, Address

Unconfigured Mesh Node

Configuration: Node Composition, Application Key(s), Group Subscription & Publication

Configured Mesh Node

Blacklisting: Key Refresh

Blacklisted Mesh Node

Node Reset

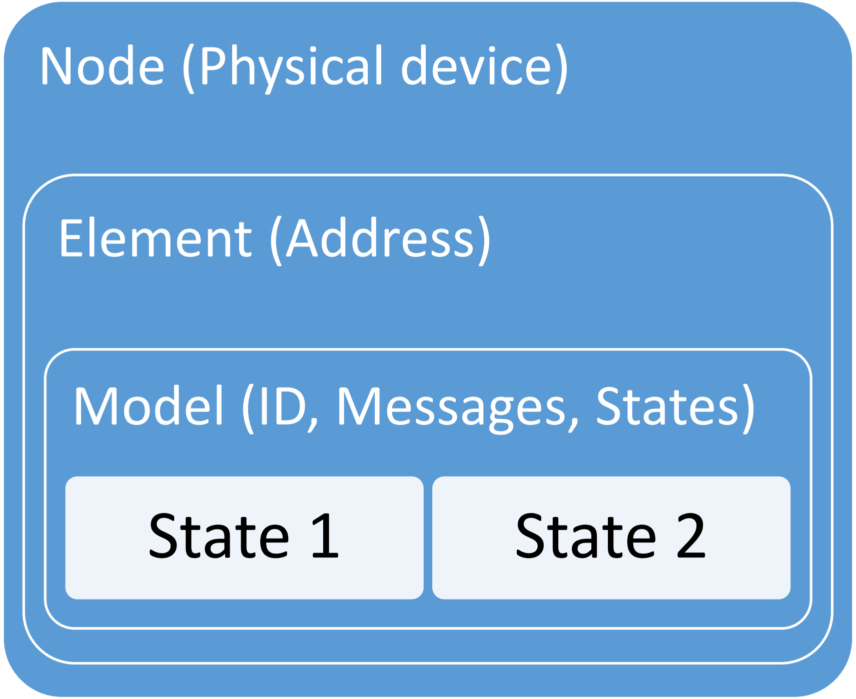
Node Composition: Elements & Models

Elements

- Unique Network Address
- Implements one or more Models

Models

- OpCode addressing
- States & Messages
- Client & Server


Node (Physical device)

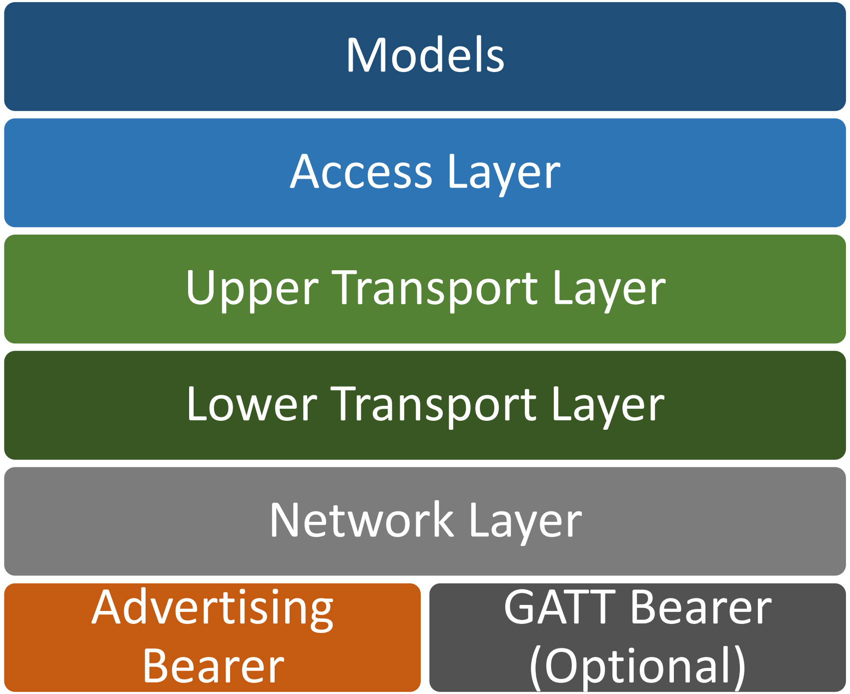
Element (Address)

Model (ID, Messages, States)

State 1

State 2

Mesh Protocol Layers



Models	states / messages / behavior
Access Layer	opcodes, multiplexing models
Upper Transport Layer	heartbeat/friendship, application encryption & authentication
Lower Transport Layer	segmentation & reassembly
Network Layer	message format, network encryption & authentication
Advertising Bearer GATT Bearer (Optional)	message transport

Anatomy of a Mesh Network PDU

IVI	NID	CTL	TTL	SEQ	SRC	DST	Transport PDU	NetMIC

Field Name	Bits	Notes
IVI	1	Least significant bit of IV Index
NID	7	Value derived from the NetKey used to identify the Encryption Key and Privacy Key used to secure this PDU
CTL	1	Network Control
TTL	7	Time To Live
SEQ	24	Sequence Number
SRC	16	Source Address
DST	16	Destination Address
TransportPDU	8 to 128	Transport Protocol Data Unit
NetMIC	32 or 64	Message Integrity Check for Network

Mesh Network Addresses

- 16-bit Network address with several categories/ranges

Unassigned	0000000000000000	No address assigned (typically used when not publishing or subscribing)
Unicast	0xxxxxxxxxxxxxxx	Every element has a unique unicast address
Virtual	10xxxxxxxxxxxxxx	Special group addresses authenticated using a 128-bit virtual label UUID
Group	11xxxxxxxxxxxxxx	Fixed (all nodes, all friends, etc) or dedicated (application specific)

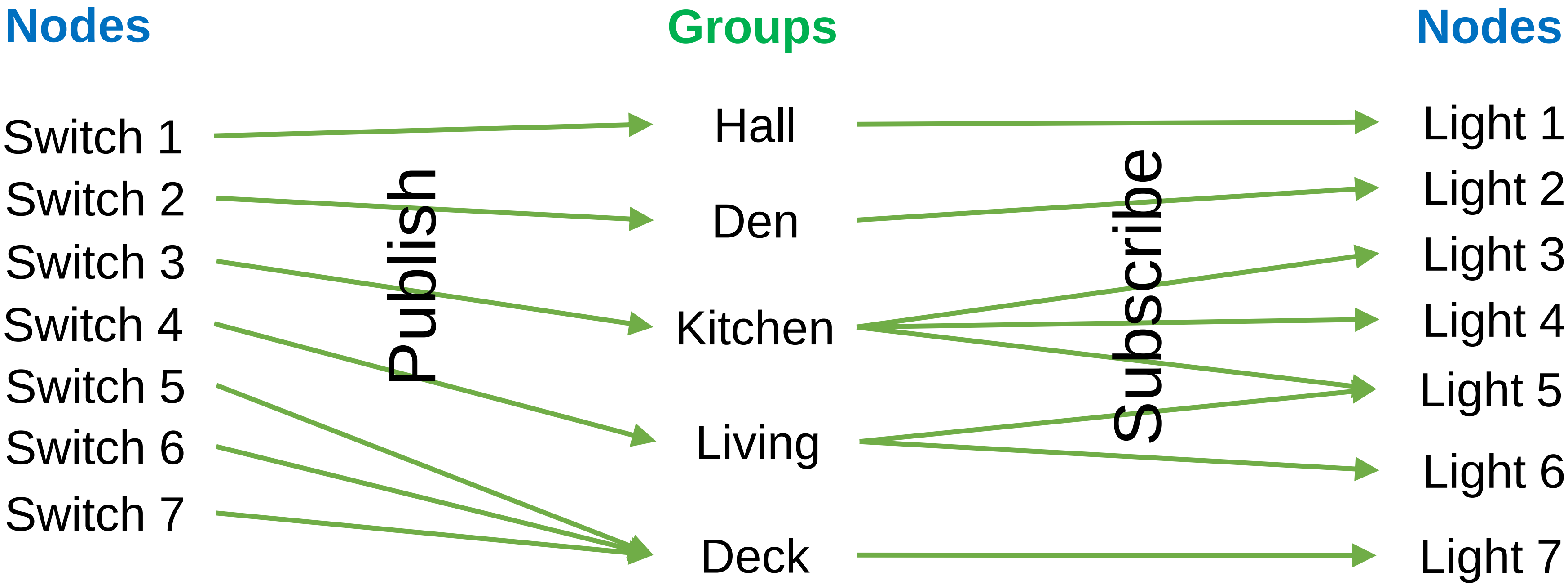
Relaying

- Time-to-Live (TTL, 7-bit, i.e. max 127)
- Decrypt with Network Key
- Decrement TTL

If TTL > 0:

- Re-encrypt with Network Key
- Send out to Network
- Application layer payload remains encrypted & untouched
  - Relay Node may not even have the Application Key

Publish & Subscribe



Nodes

Switch 1
Switch 2
Switch 3
Switch 4
Switch 5
Switch 6
Switch 7

Publish

Groups

Hall
Den
Kitchen
Living
Deck

Subscribe

Nodes

Light 1
Light 2
Light 3
Light 4
Light 5
Light 6
Light 7

Security Features

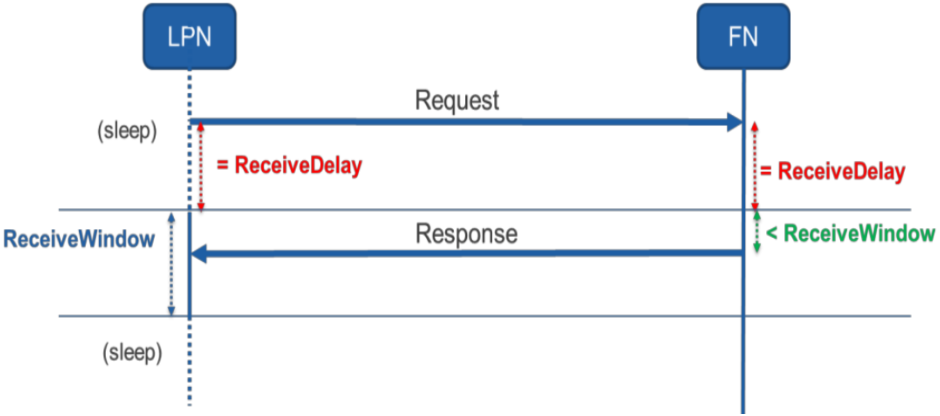
- Authentication during provisioning
- Two level encryption
  - Network
  - Application
- Replay protection
  - IV Index (32-bits)
  - Sequence number (24 bits)
  - IV Index Update procedure
- Key Refresh
  - Node Blacklisting

Segmentation & Reassembly

- A message can be either unsegmented or segmented
- Payload
  - Unsegmented: 15 bytes
  - Segmented: 12 bytes per segment, max 32 segments = 384 bytes
  - Contains 4 or 8 byte MIC at the end, reducing usable payload size
- Unsegmented messages are inherently unreliable
- Segments of a segmented message are acknowledged by the receiver
  - One-segment “segmented” message can be used for reliable sending

Friendship

- 100% duty-cycle scanning needed for reliability, but consumes a lot of power
- Mix of battery & mains powered nodes
- Solution: pair up stable power supply nodes (Friends) with Low Power Nodes (LPNs)
- Friends queue up messages for the LPN
- LPN queries the Friend periodically if there are any messages for it



IMPLEMENTATION STATUS & PLANS

Support in Zephyr* OS

- Available starting with Zephyr 1.9
- All mandatory features implemented
- Tested against multiple other implementations
- Ported to MyNewt
  - Multiple valuable fixes ported back to Zephyr
- Demos possible with many popular supported Zephyr boards
  - Come to the Zephyr booth to see it in action!
- Minimum RAM footprint (entire OS with Mesh) is ~12kB
  - Fits even the most constrained 16k boards, like BBC micro:bit

*Other names and brands may be claimed as the property of others.

Support in Linux*

- meshctl tool released with BlueZ 5.47
  - GATT Client
  - PB-GATT Provisioner
- Ongoing work both in user space (BlueZ) and kernel
  - Advertising & Scanning managed in the kernel
    - Controlled through mgmt API extensions
  - Essentially everything else in a user space meshd

*Other names and brands may be claimed as the property of others.

Future development

- Mesh Vendor HCI Extensions
  - Supported both by Linux & Zephyr
- More features
  - Friend support for Zephyr
- More standard models
- More demos with various boards

QUESTIONS?

Disclaimer

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at www.intel.com.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Intel Corporation is under license.

*Other names and brands may be claimed as the property of others.

intel®