UNIVERSIDAD
COMPLUTENSE
MADRID

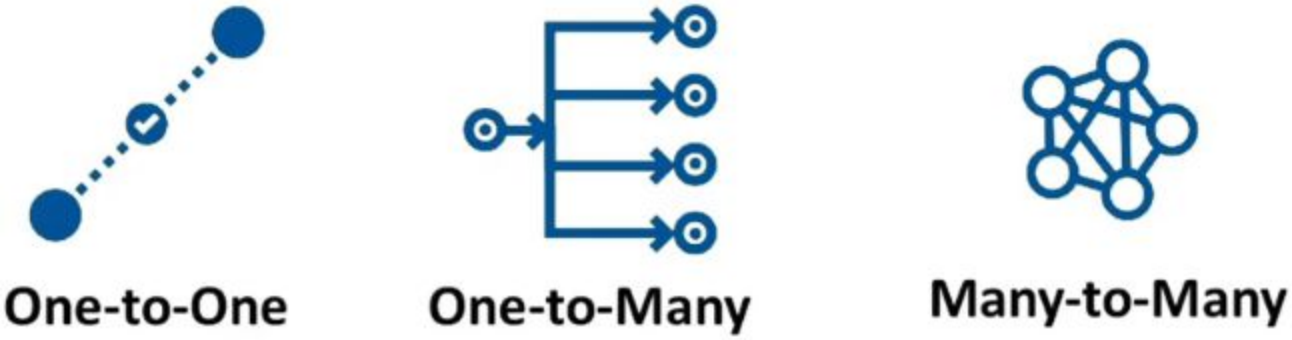
Bluetooth Mesh

Networks and Protocols 1

Facultad de Informática

Bluetooth Mesh

- Released in 2017
- Independent of the Bluetooth standard
  - Compatible, uses its radio and advertisement packages but it defines its own stack
  - Supported by the Bluetooth SIG
- Purpose:
  - Extend the range of the BLE networks
  - Increase the range of BLE for industrial applications
    - Home Automation

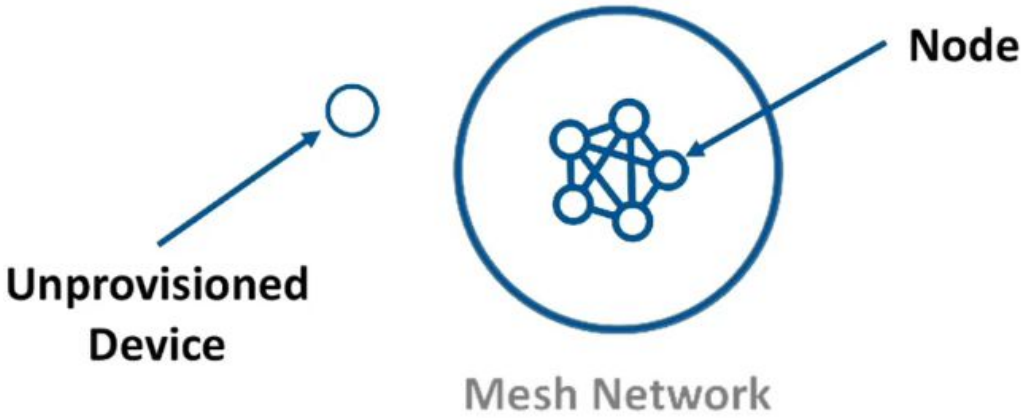


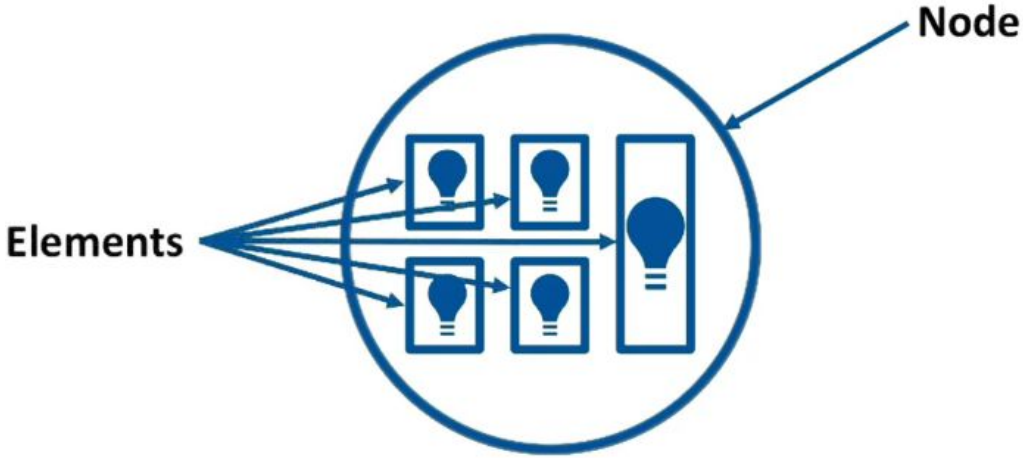
Uses BLE technology

- Uses the BLE controller to send packages
  - Only uses the Advertising and Scanning States
  - It does not use secure BLE connections
- Supports all BLE versions
  - It does not support some of the novelties introduced in Bluetooth 5, like extended advertisements

Basic Terminology

- Node: device that is part of a Mesh network
- Unprovisioned device: still not part of the Mesh
- A node can be composed of several elements that can be controlled independently





Basic Terminology

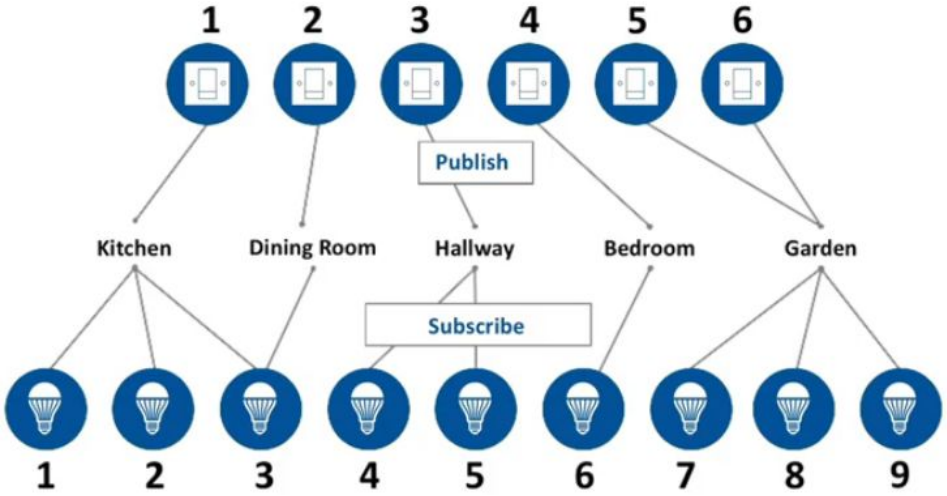
- States:
  - The functionality of the elements is defined as a set of states the element can be in and the messages that can be sent to act on the state of the element
  - E.g.: a lamp can be defined as an element with two states: on/off
- Properties:
  - Add additional context to a state
  - E.g.: external or internal temperature
- State transitions:
  - Define the state changes

Messages

- Send between the nodes that form the network
- Control the nodes, transmit information, report state
- Two categories:
  - Acknowledged: require the receiver to send an ACK
  - Unacknowledged: the receiver does not send an ACK
- Three types:
  - get: request state of a node
  - set: modify state of a node
  - status: response to a get with state information or send automatically (timer)

Addresses

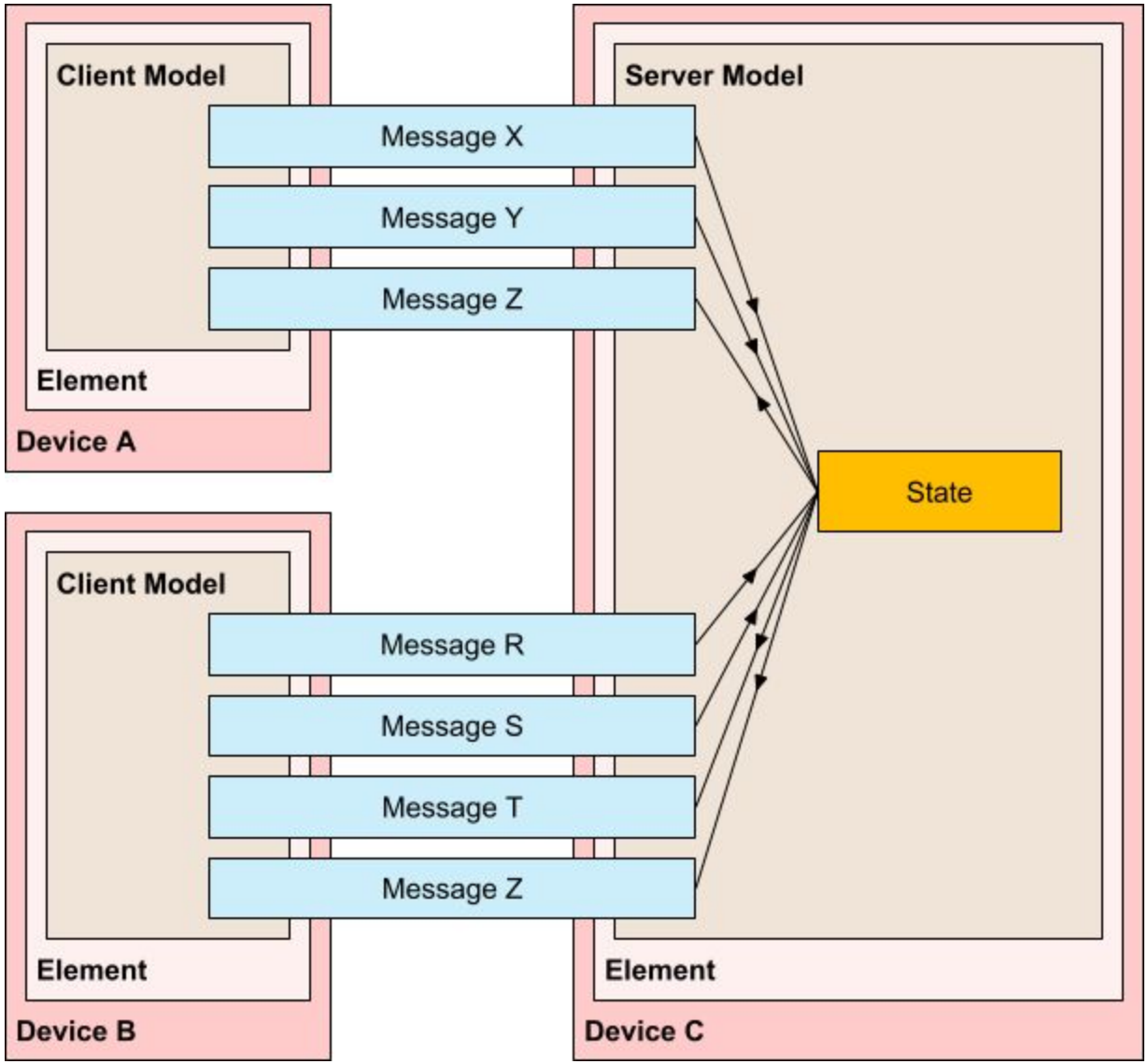
- Identify source and destination of a message
- Three types:
  - Unicast: identify a single node, assigned during provisioning
  - Group: identify a set of nodes, generally used to group nodes that are physically close to each other (e.g.: all the lights in a room)
    - SIG Fixed: registered, well known
    - Dynamic: created by the user
  - Virtual: assigned to one or more elements of different nodes.



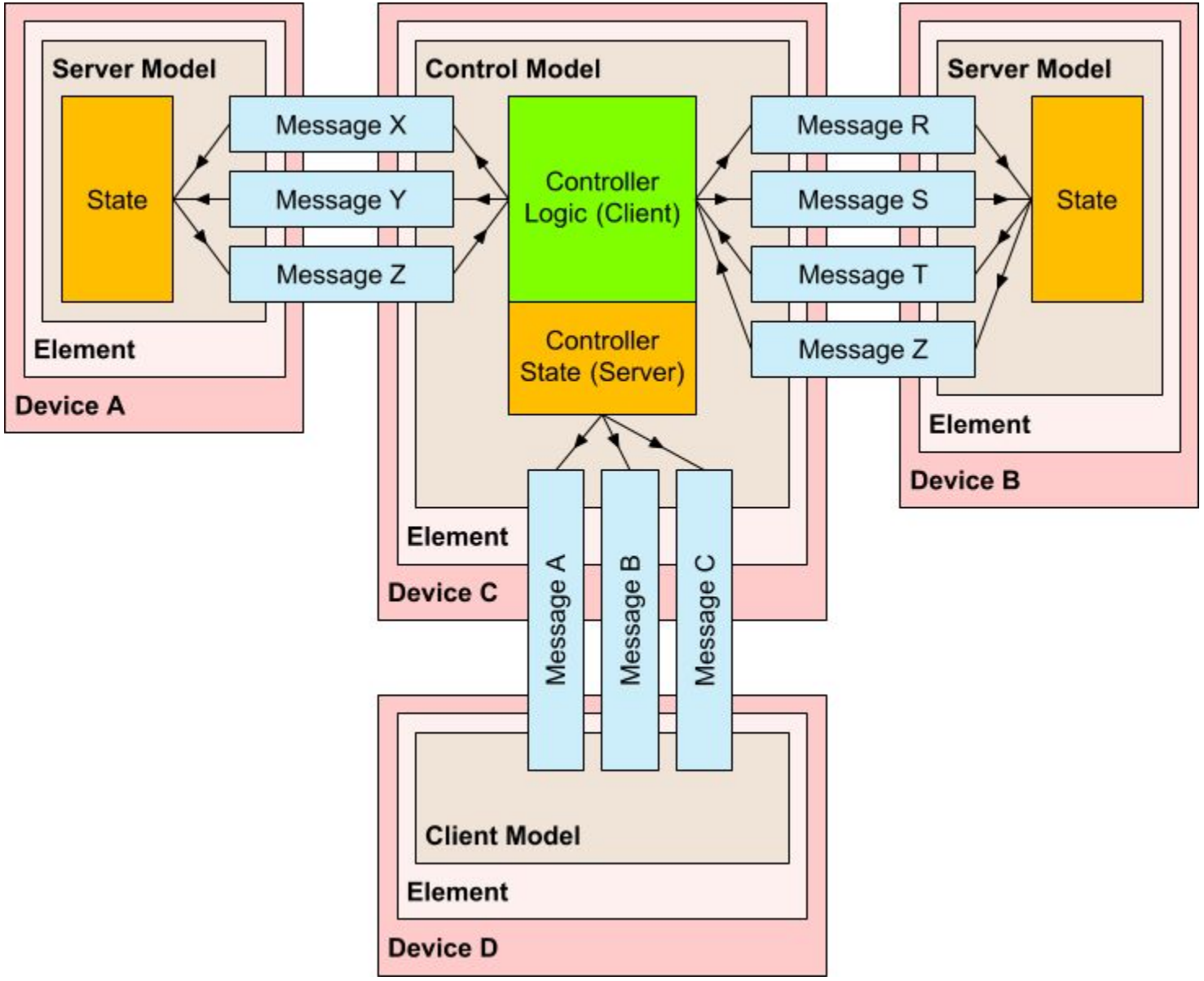
Model

- An element is composed of one or more Models
  - Define the functionality of the element (like the application)
- Can be extended but not modified (backward compatibility)
  - An unextended model is known as a root model
- Three categories:
  - **Server model**: a collection of states, transitions and constraints between them, and the messages that can be received and send in each of the states
  - **Client model**: does not define states, it defines the messages that can be sent to interact with the corresponding server model
  - **Control model**: has a server model and a client model, using the client to communicate with other servers and the server to interact with his clients.
    - Can be used to automate certain changes and control the nodes in the network

Server & Client models



Control model



Scenes

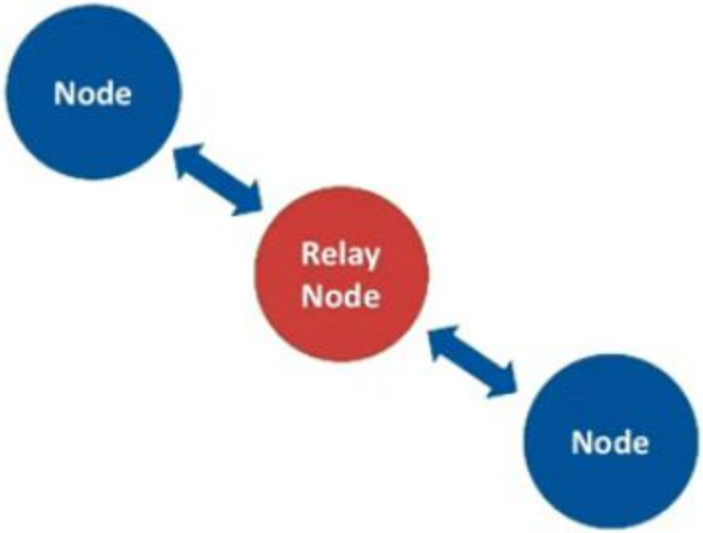
- A collection of stored states
  - E.g.: lights 1 and 3 on, lights 2 and 4 off
- Have a 16 bits id number, unique in the network
- Used to make several changes with only one command
  - Can be triggered on demand or programmed for a certain moment

Type of nodes

- All nodes can send and receive mesh messages
- Nodes can have additional features that give them additional capabilities
  - Relay
  - Proxy
  - Friend
  - Low power
- A node can have 0 or more features
- Features can be enabled and/or disabled dynamically at any moment

Relay node

- Supports the Relay feature
- Can transmit messages from other nodes, allowing them to traverse the whole network
- Time To Live (TTL)
  - Determines if a packet is going to be relayed or not
  - A TTL of 1 or 0 implies that the packet is not relayed
    - 0: the message has not been relayed
    - 1: may have been relayed but will not be relayed again
  - Max value 127



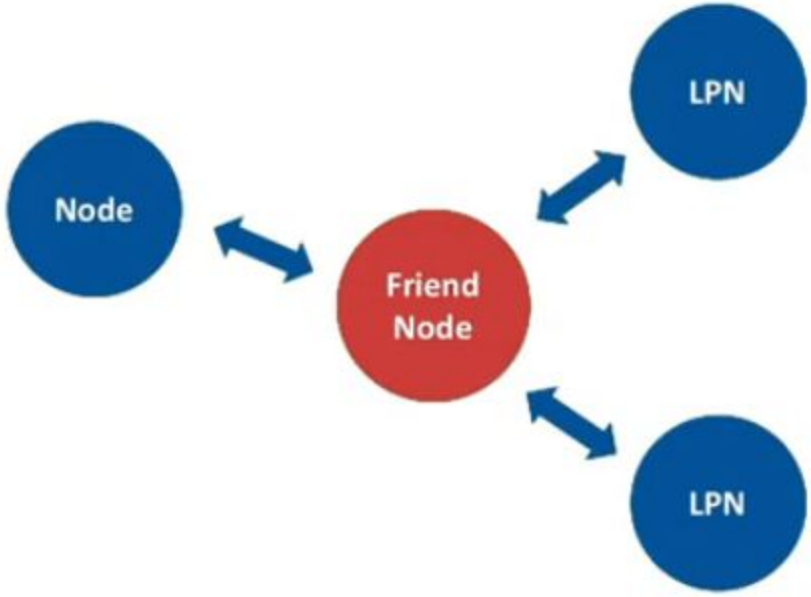
Low Power and Friend nodes

Low Power Node (LPN)

- energy constrained
- asleep/low power mode most of the time, with the radio powered off
- E.g.: sensor nodes

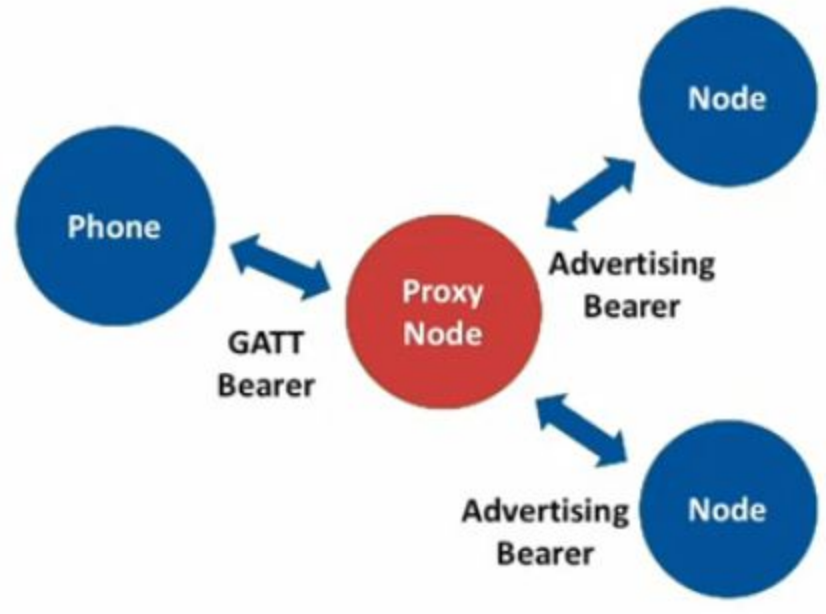
Friend Node (FN)

- Not energy constrained
- Radio is always on
- Listen to messages, store them and relay them to the LPN when they are ready
  - The LPN requests the stored messages to its FN

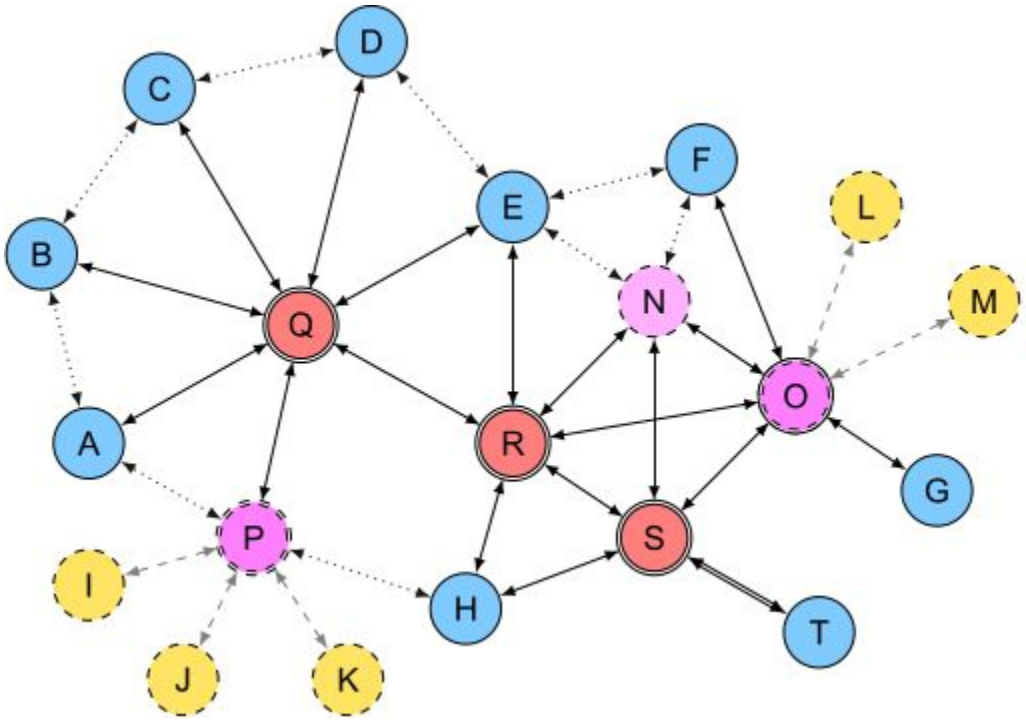


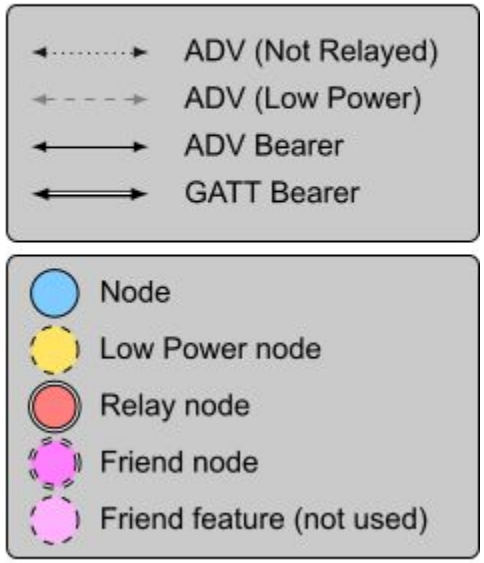
Proxy Nodes

- They have the two stacks: BLE and Mesh
- Use GATT to expose a proxy service to a BLE client
  - Allow a BLE device that does not support BLE Mesh to interact with the network
  - Read/Write of attributes are translated to mesh operations
- E.g.: smartphone used to configure a mesh network or to change the states of the nodes

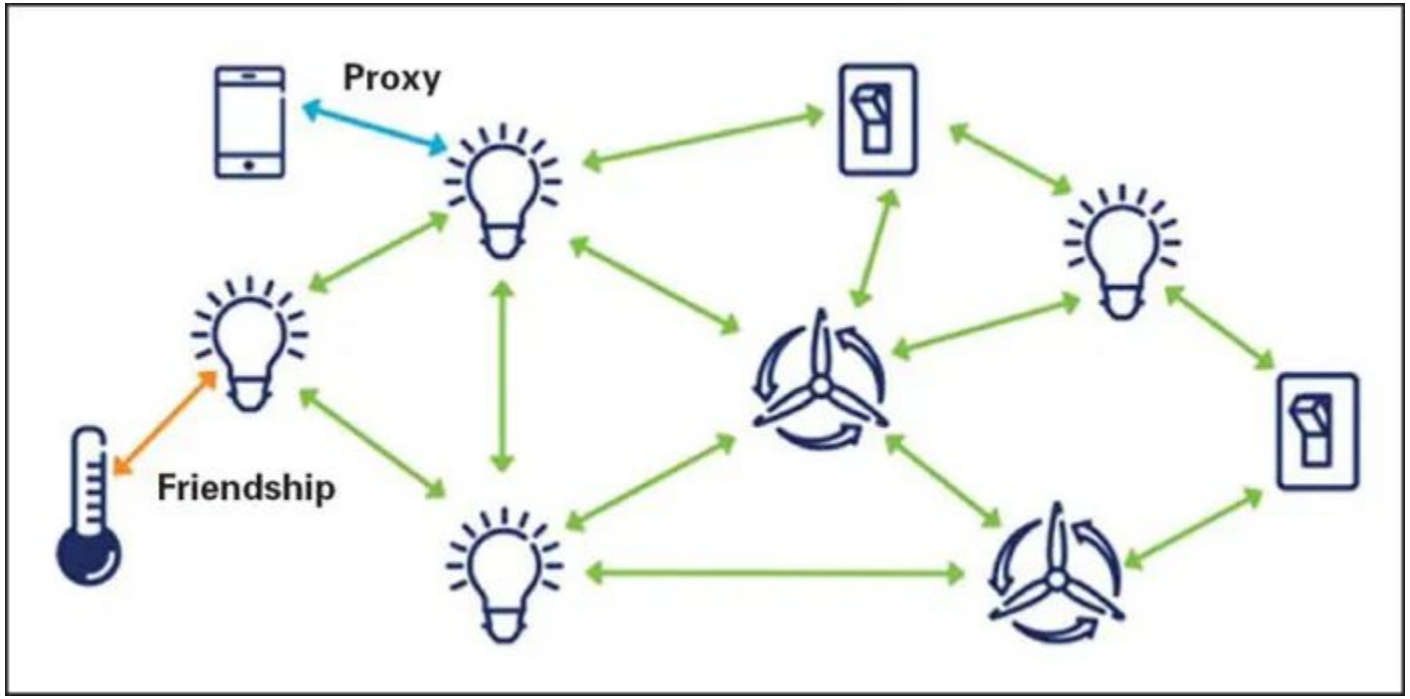


E.g.: type of nodes

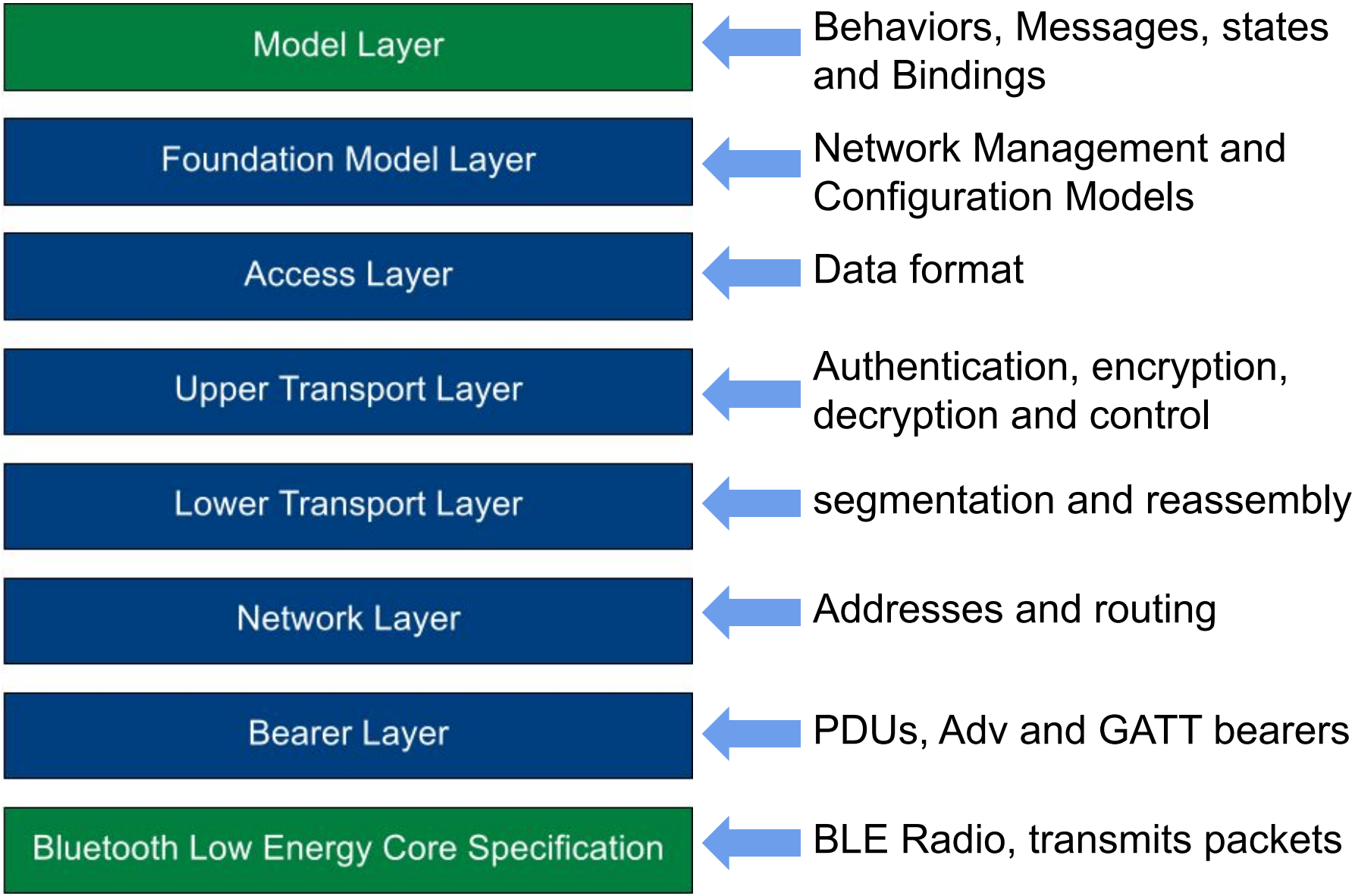




E.g.: home automation



Architecture



Routing: Managed Flooding

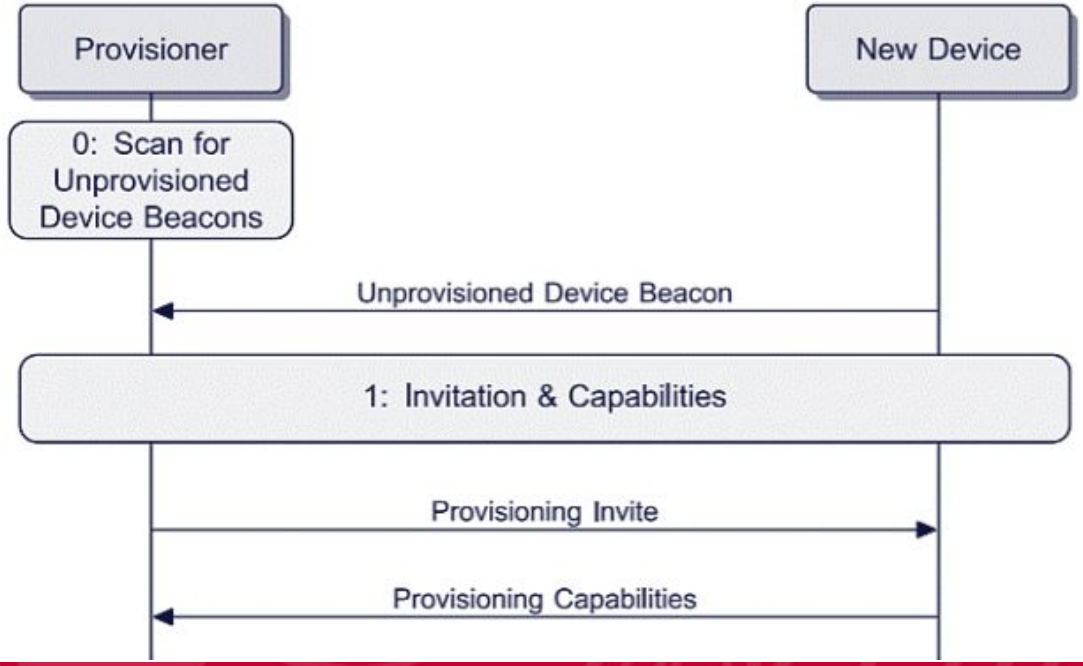
- The messages are broadcasted, not sent individually to a node
- Flooding
  - Any message sent from one node reaches all the nodes in its radio reach
  - Relay nodes retransmit the message to all the nodes in its radio reach
    - A message can reach a node from multiple paths (duplicates)
- Managed Flooding
  - Heartbeats: transmitted periodically by a node to indicate that it is active
  - TTL: messages discarded when TTL reaches 1
  - Messages are cached: detected duplicates are not relayed
  - Friends and LPN

Provisioning procedure

- Used to add nodes to the network
- Normally done with a smartphone or tablet
  - known as the provisioner
- 5 steps:
  - Beaconing
  - Invitation
  - Public Key Exchange
  - Authentication
  - Provisioning Data Distribution

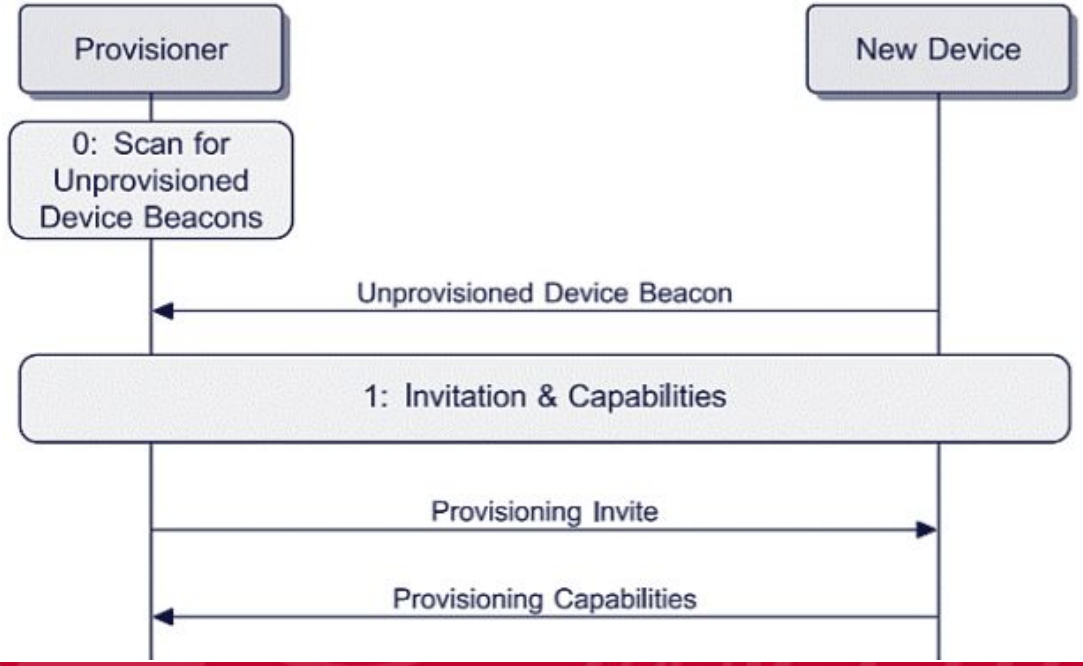
Step 1: beaconing

- The device sends beacons in BLE advertisements
  - New advertisement type
- Initiated by the user with a sequence of key presses in the device to be provisioned



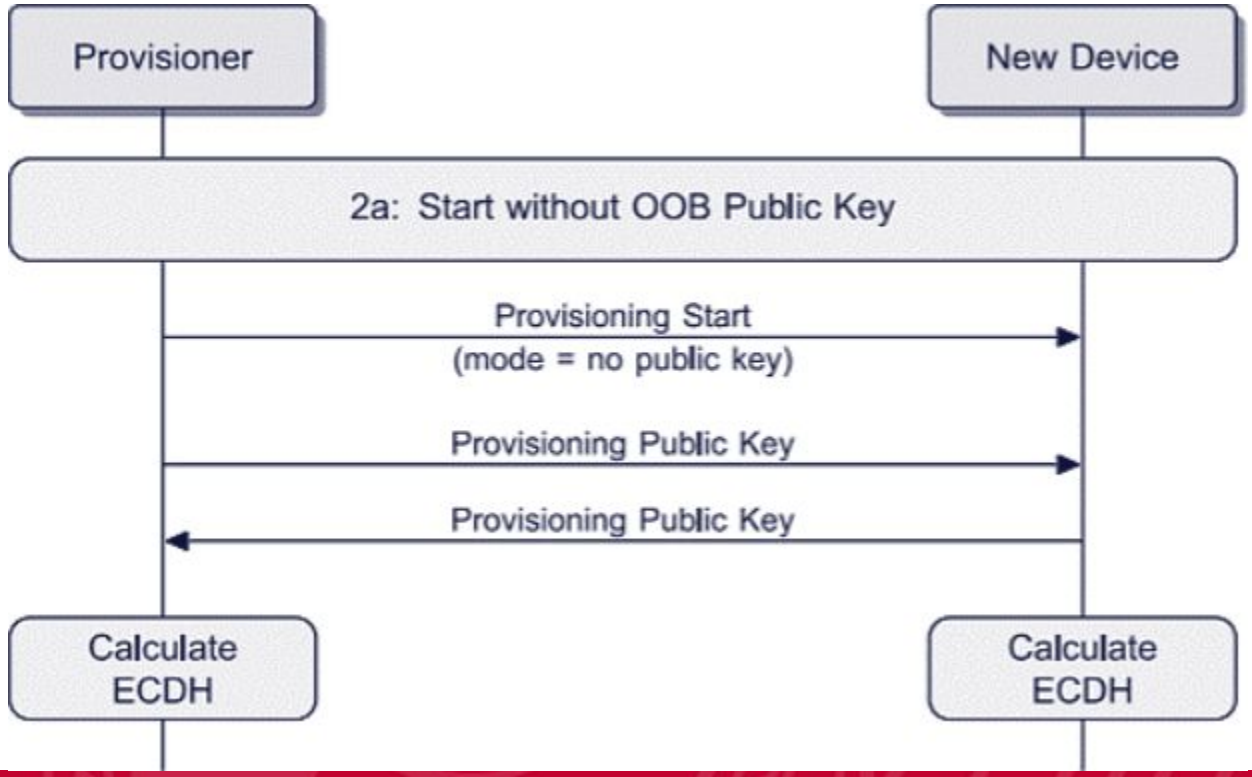
Step 2: Invitation

- When the provisioner detects the device it sends an invitation message (Provisioning invite PDU)
- The device responds with the information about itselfs (Provisioning Capabilities PDU)
  - Number of elements in the device, set of security algorithms supported, OOB and IO capabilities



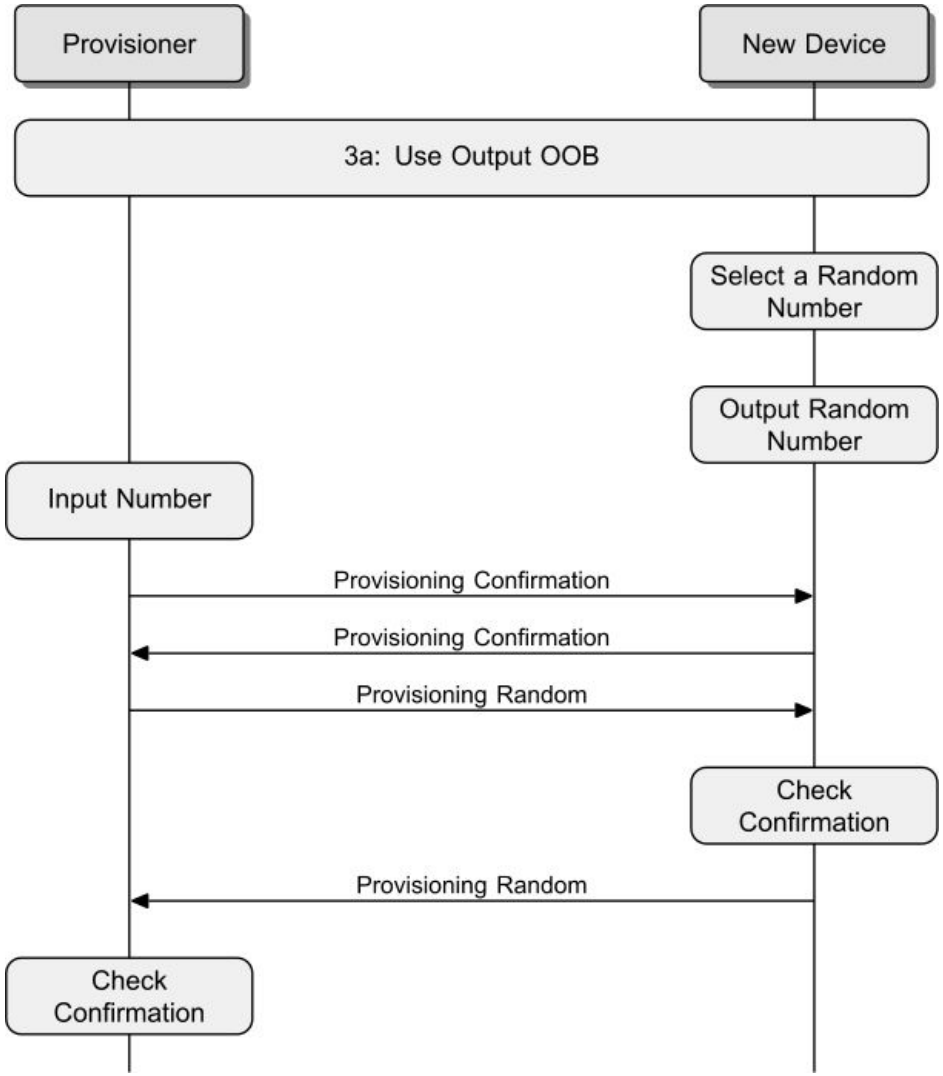
Step 3: public keys interchange

- Bluetooth Mesh uses symmetric and asymmetric keys
  - Uses an Elliptic Curve Diffie-Hellman algorithm
- The public keys interchange can be performed OOB



Step 4: Authentication

- User interaction
  - Depends on the IO capabilities
  - E.g.: Output OOB
- Confirmation
- Session key generation
  - ECDH
- With communications encrypted
  - Network key
  - Device key
  - IV index
  - Unicast address



Security in Bluetooth Mesh

- Is mandatory
- All messages are encrypted
- Three independent levels, with different keys
  - Network security, uses two keys derived from the network key
    - Network encryption key
    - Privacy key, for address obfuscation
  - Application security: uses application keys
  - Device security: uses the device key, for the node configuration
- The keys can be modified during the network life
- Protection against trash-can attacks
  - Trashed nodes are added to a black list
  - Keys are refreshed

Other security aspects

- Privacy: obfuscated addresses
  - Privacy key derived from the network key
  - Hinder the device tracking based on their addresses
- Defense against replay attacks
  - Use of sequence numbers (SEQ) in all messages
  - Use of the IV index, permits to extend the range of the sequence numbers so that they do not overflow in thousands of years

References

- Mesh profile specification
  - https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457092
- Espressive SDK API for BLE Mesh
  - https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/esp-ble-mesh/ble-mesh-index.html