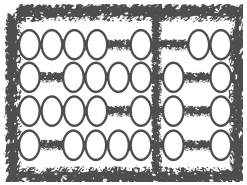
IEEE 802.15.1: Bluetooth

Prof. Juliana Freitag Borin

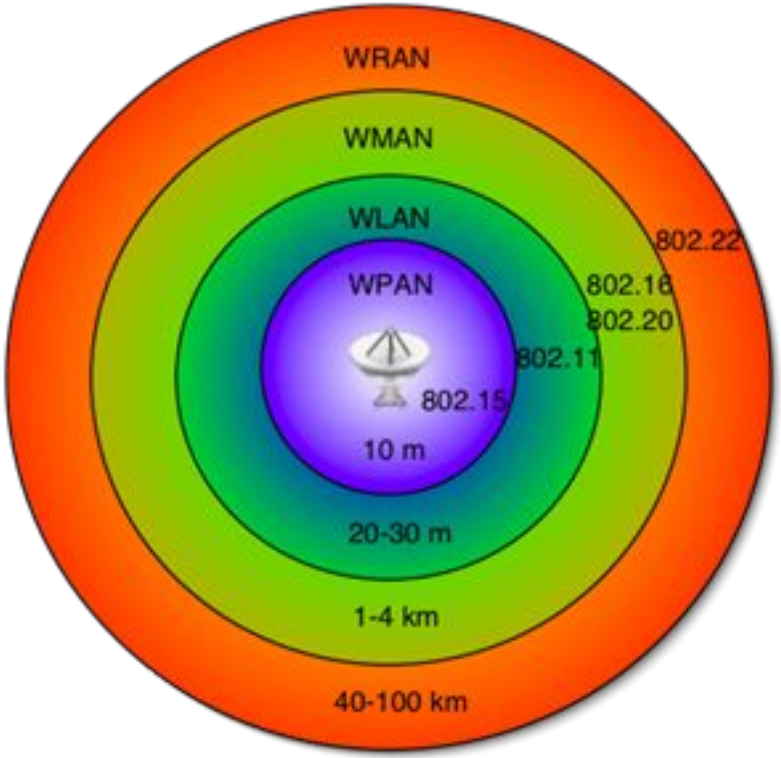


Institute of
Computing

University
of Campinas

Wireless Personal Area Networks (WPANs)

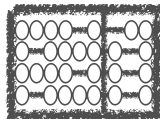
- 10 m or less



Source: https://www.cse.wustl.edu/~jain/cse574-06/ftp/wimax/index.html

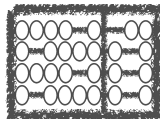
IEEE 802.15.1 and Bluetooth

- Bluetooth started with Ericsson's Bluetooth Project in 1994 for radio communication between cell phones over short distances.
- Intel, IBM, Nokia, Toshiba, and Ericsson formed Bluetooth Special Interest Group (SIG) in 1998.
- 1st version of the specification came out in 1999.
- IEEE 802.15.1 (2002) is based on Bluetooth specifications.



IEEE 802.15.1 and Bluetooth

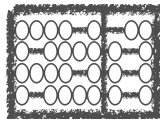
- **Bluetooth v1.1** – IEEE 802.15.1-2002
- **Bluetooth v1.2** – IEEE 802.15.1-2005 (Completed Nov. 2003): Faster Connection and Discover, adaptive FHSS – reduce radio frequency interference -, Extended Synchronous Connections – improve voice quality of audio links.
- **Bluetooth v2.0 + Extended Data Rate (EDR) (2004)**: 3Mbps.
- **Bluetooth v2.1 + EDR (2007)**: Secure Simple Pairing to speed up pairing.
- **Bluetooth v3.0 + High Speed (HS) (2009)**: 24 Mbps using Wi-Fi + Bluetooth.
- **Bluetooth v4.0 (2010)**: low energy. Smaller devices requiring longer battery life. New incompatible PHY. Bluetooth Smart.
- **Bluetooth v4.2 (Dec. 2014)**: privacy, data length extension, IP connectivity.
- **Bluetooth 5 (2016)**: 4x range, 2x speed (2Mbps), 8x the broadcasting message capacity, focus on IoT
- **Bluetooth 5.1 (2019)**: direction finding feature, cache, improved advertising channels
- **Bluetooth 5.2 (2019):** enhanced attribute protocol, LE power control, isochronous channel
- **Bluetooth 5.3 (2021):** removal of AMP, channel classification enhancement, connection subrating, enhancements to Periodic Advertising, Encryption Key Size Control
- **Bluetooth 5.4 (2023):** time-synchronized star network with bi-directional communication
- **Bluetooth 6.0 (2024):** location accuracy, energy efficiency, and audio quality



Two flavors of Bluetooth

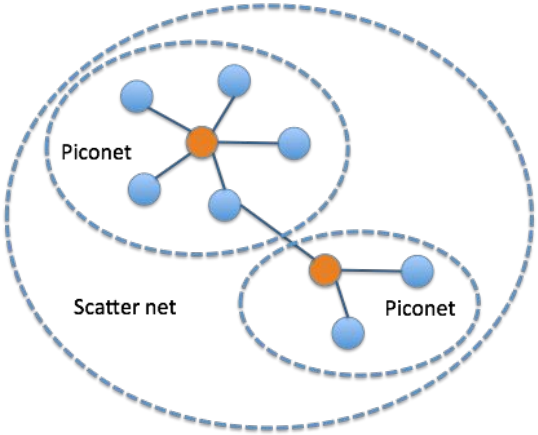
- Bluetooth BR/EDR - establishes a relatively short-range, continuous wireless connection, which makes it ideal for use cases such as streaming audio.
- Bluetooth with low energy functionality (LE) - allows for short bursts of long-range radio connection, making it ideal for Internet of Things (IoT) applications that don't require continuous connection but depend on long battery life.
- Dual-Mode - dual-mode chipsets are available to support single devices such as smartphones or tablets that need to connect to both BR/EDR devices (such as audio headsets) and LE devices (such as wearables or retail beacons).

Source: https://www.bluetooth.com/specifications/bluetooth-core-specification



Bluetooth network

- Piconet:
  - 1 master and many slaves
- Scatter net:
  - A device can participate in multiple Piconets.
  - Routing protocols not defined.
- Mesh (Bluetooth 5.0)



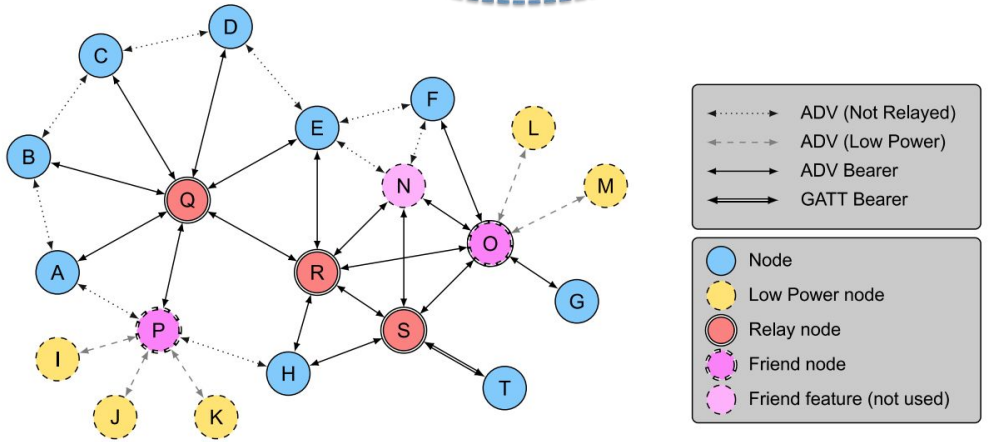


Figure 2.8: Example topology of a mesh network

Source: Mesh protocol Bluetooth Specification

BLE System Architecture

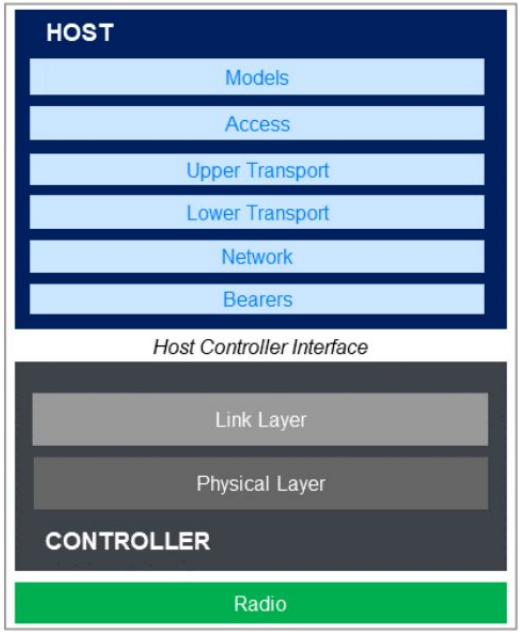
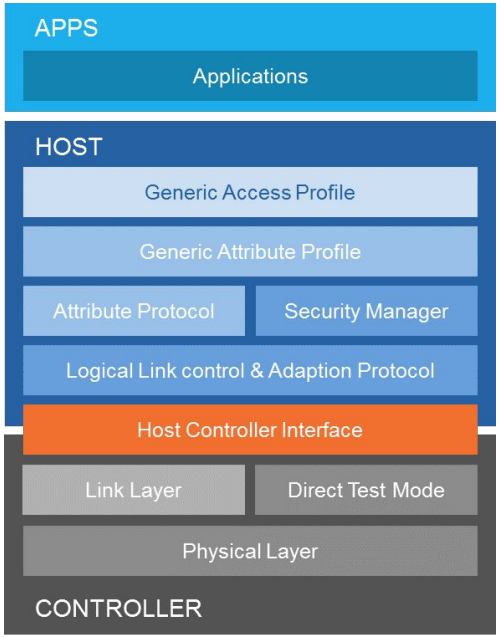
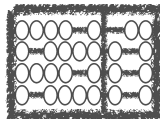


Figure 4 - The Bluetooth mesh stack

Source: The Bluetooth® Low Energy Primer

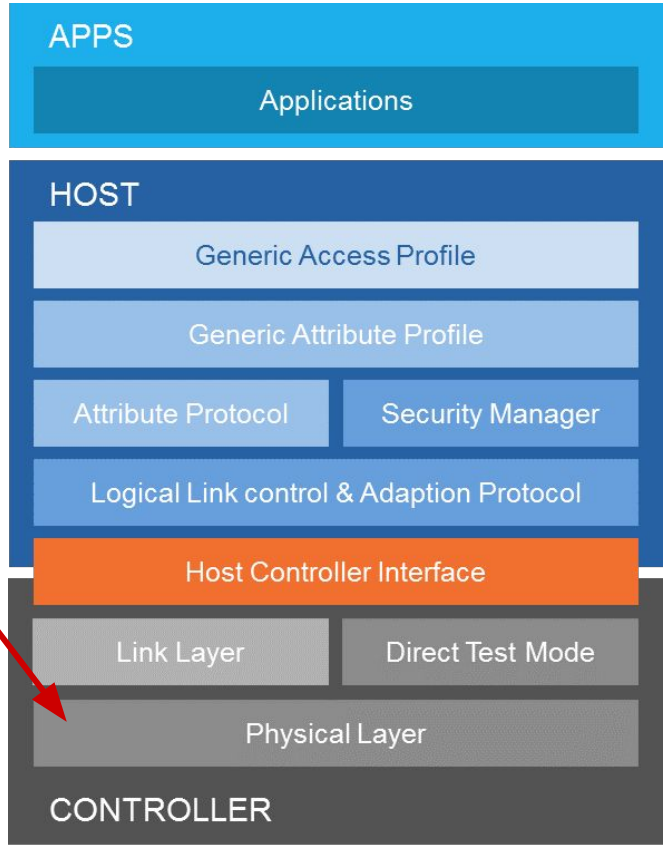


Source: https://www.bluetooth.com/specifications/bluetooth-core-specification



BLE System Architecture

Controls transmission/receiving of the 2.4Ghz radio with Bluetooth communication channels. BR/EDR provides channels (79) with narrower bandwidth, while LE uses fewer channels (40) but broader bandwidth.

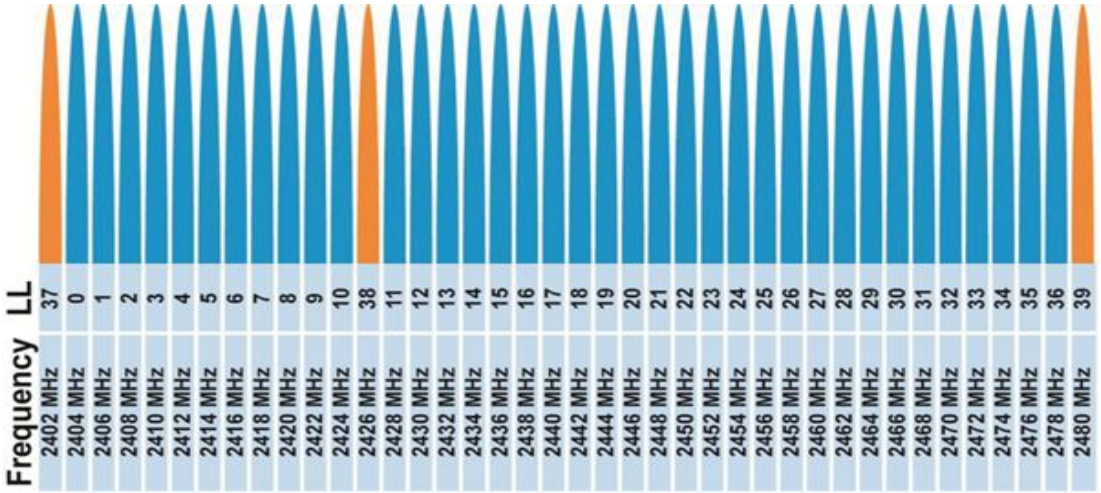


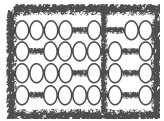
Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

Physical Layer: BLE 4.x

- 2.4 GHz ISM
- 1 Mbps (*theoretical upper limit, 5-10KB/s in a real scenario*)
- 40 Channels on 2 MHz spacing
- Frequency hopping
- Range: can reach 30 m or more *line-of-sight*, but typically closer to 2 to 5 m.

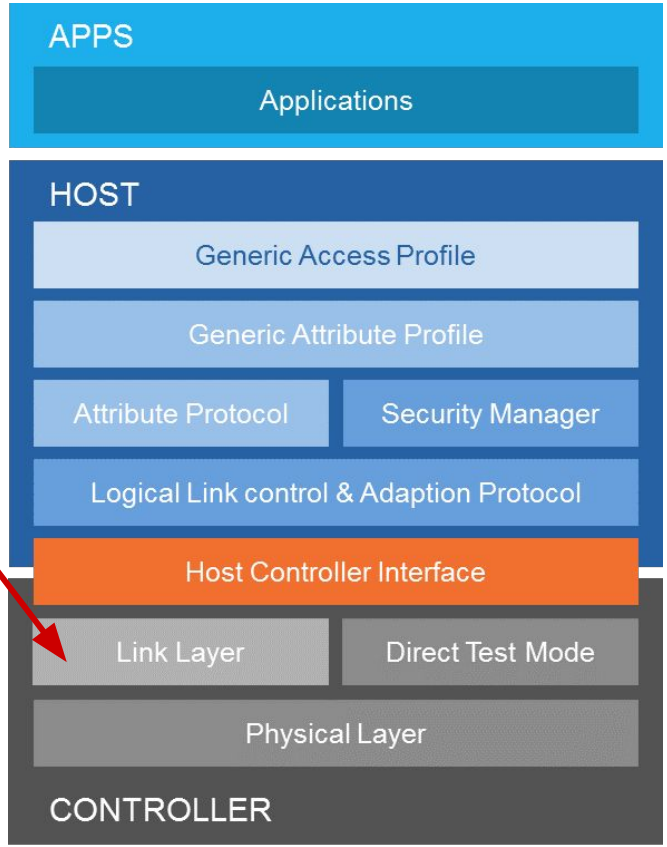
3 advertising channels and 37 data channels



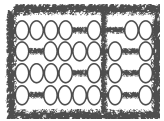


BLE System Architecture

Defines packet structure/channels, discovery/connection procedure and sends/receives data.

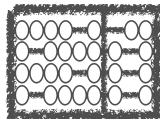


Source: https://www.bluetooth.com/specifications/bluetooth-core-specification



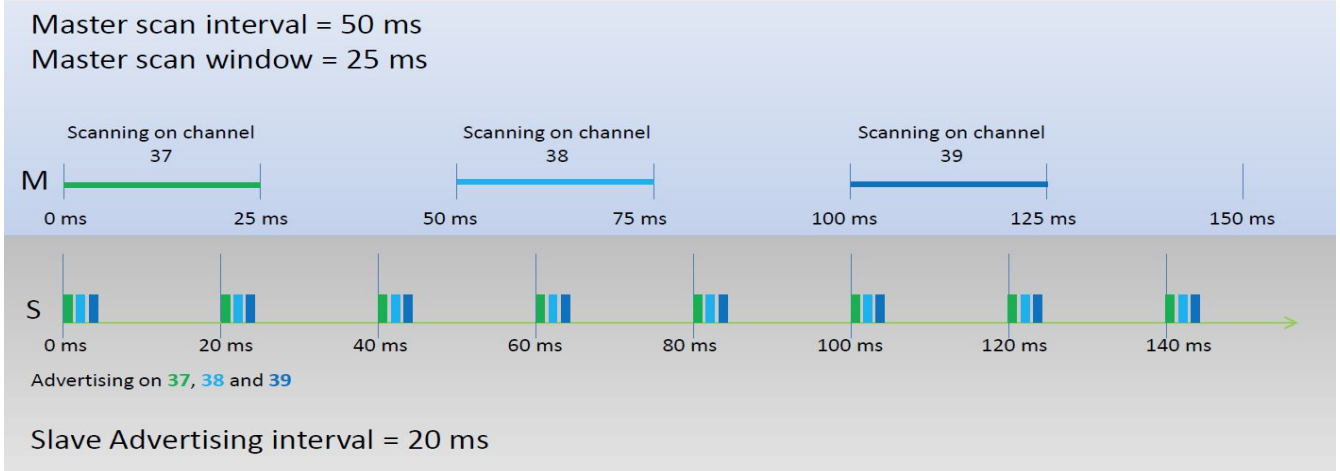
Link Layer: Advertising and Scanning

- One packet format and two types of packets: advertising and data packets
- Advertising packets serve two purposes:
  - To broadcast data for applications that do not need the overhead of a full connection establishment
  - To discover slaves and to connect to them
- Advertising packets are sent at a fixed rate defined by the advertising interval (20 ms to 10.24.s)



Link Layer: Advertising and Scanning (piconet)

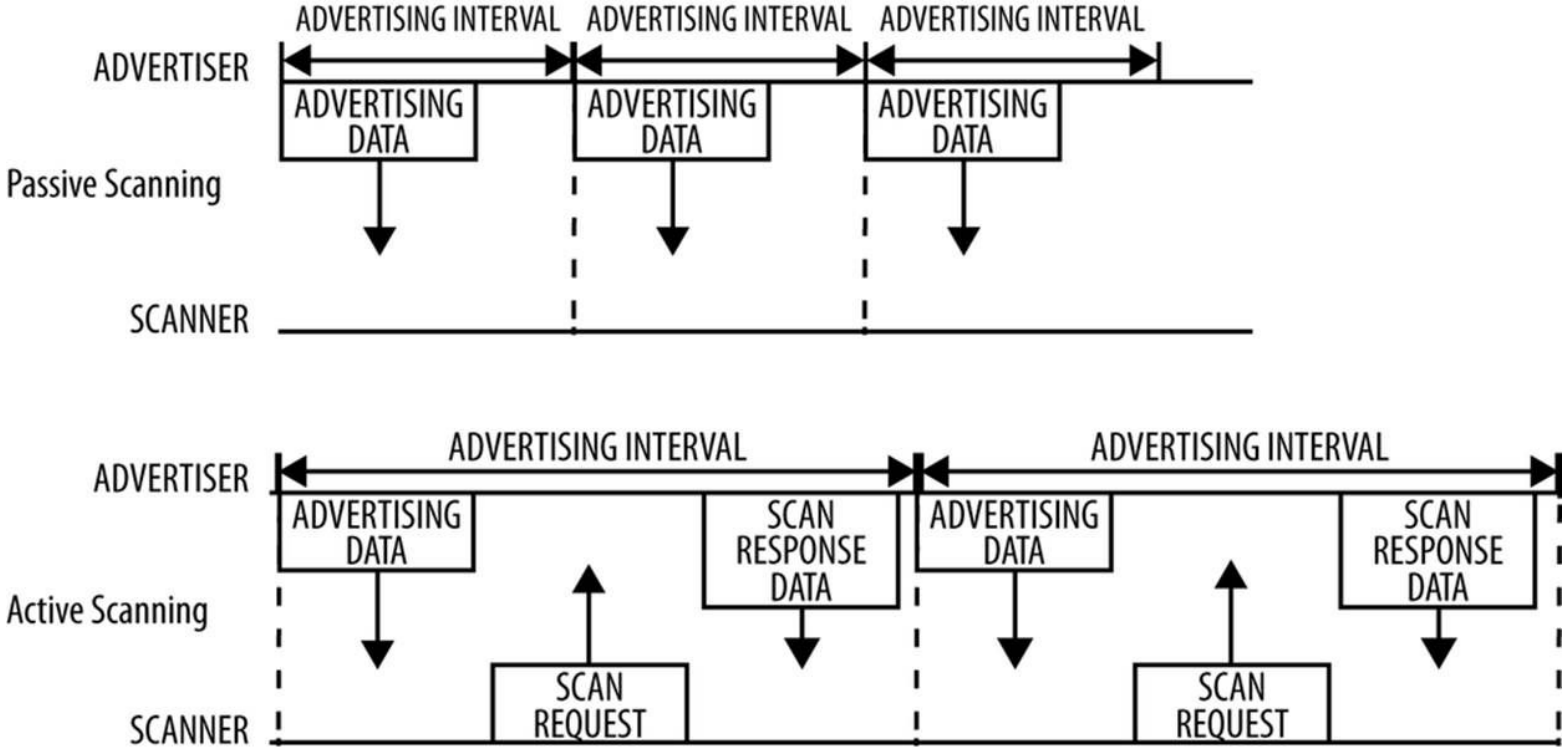
BLE Advertising



From this we can see that the following advertising packets will be picked up by the scanning device: t=0 ch=37, t=20 ch=37, t=60 ch=38, t=100 ch=39 and t=120 ch=39.

Source: https://devzone.nordicsemi.com

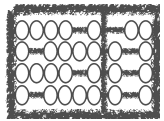
Link Layer: Advertising and Scanning (piconet)



Link Layer: Advertising and Scanning

Classification of advertising packets:

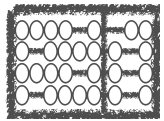
- Connectability:
  - Connectable: a scanner can initiate a connection upon reception of such packet
  - Non-connectable: a scanner cannot initiate a connection (broadcast only)
- Scannability:
  - Scannable: a scanner can issue a scan request upon reception of such packet
  - Non-scannable: a scanner cannot issue a scan request upon reception of such packet
- Directability:
  - Directed: contains only the advertiser's and the target scanner's Bluetooth addresses in its payload. This packet is, therefore, connectable.
  - Undirected: not targeted to any particular scanner.



Link Layer: Connections (piconet)

Connection establishment:

1. Initiator starts scanning to look for advertisers accepting connections requests.
2. Upon detection of a suitable advertising slave, the initiator sends a connection request packet.
3. If advertiser responds, the connection is established by the initiator. The initiator becomes the master and the advertiser becomes the slave.



Link Layer: Connections (piconet)

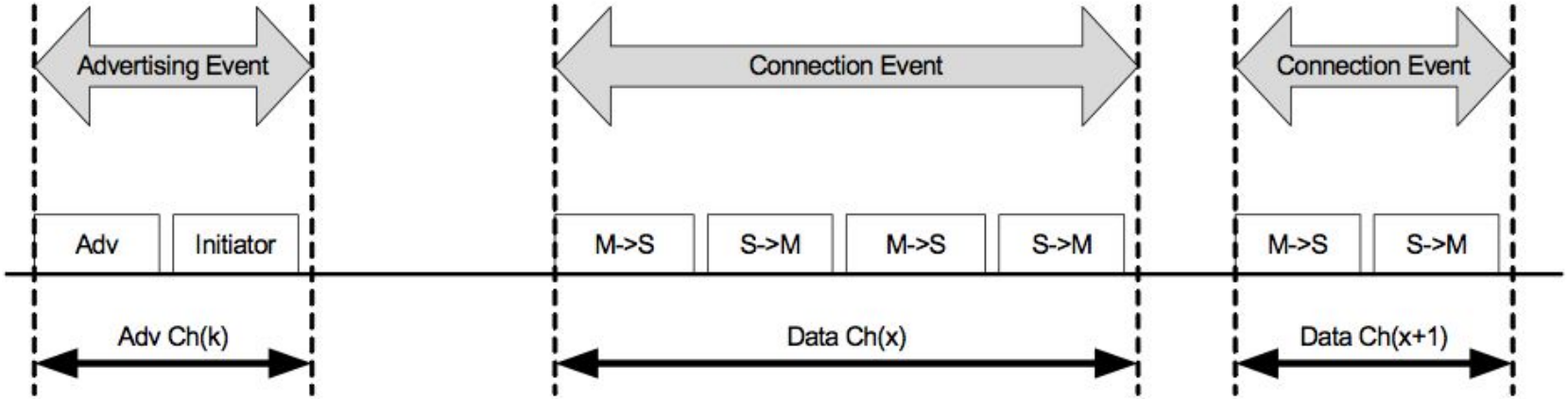
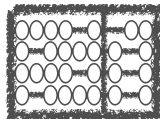
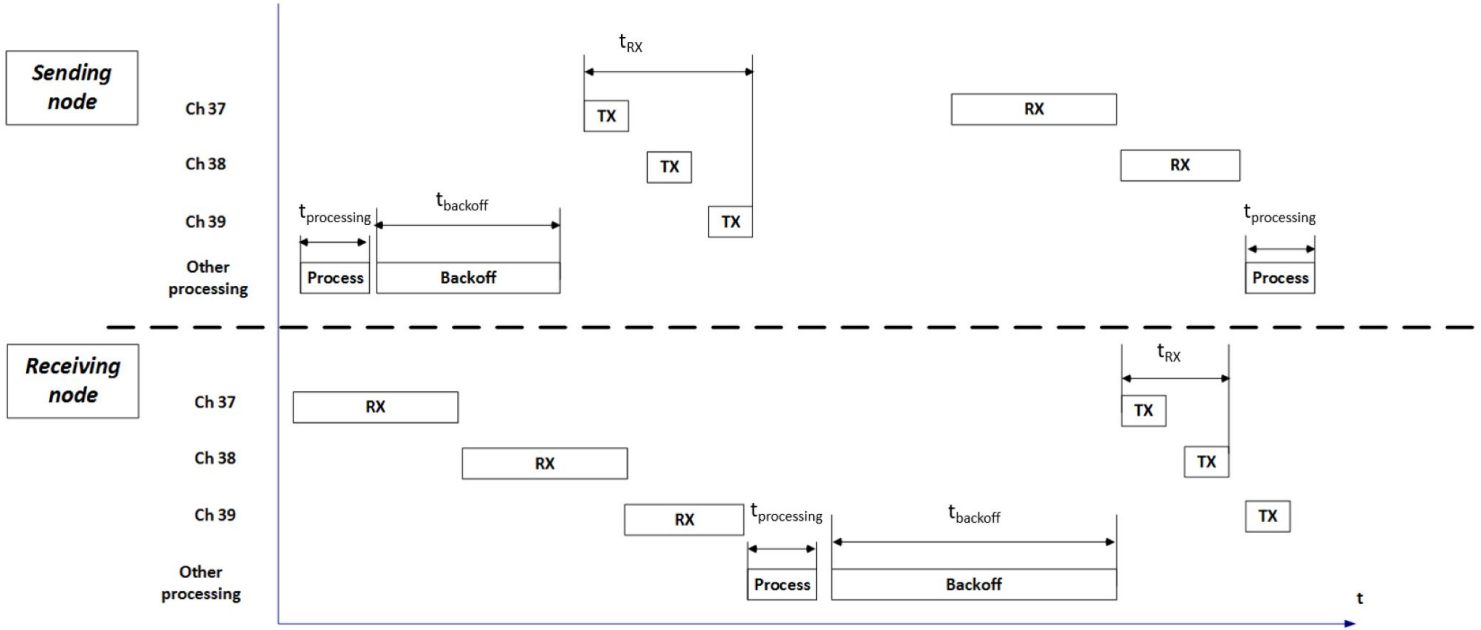


Figure 1.4: Connection Events

Source: Bluetooth Specification Version 4.2.



Link Layer: Mesh communication

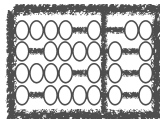


Source: Baert, M.; Rossey, J.; Shahid, A.; Hoebeke, J. *The Bluetooth Mesh Standard: An Overview and Experimental Evaluation*. Sensors 2018, 18, 2409. https://doi.org/10.3390/s18082409

Link Layer: Connections

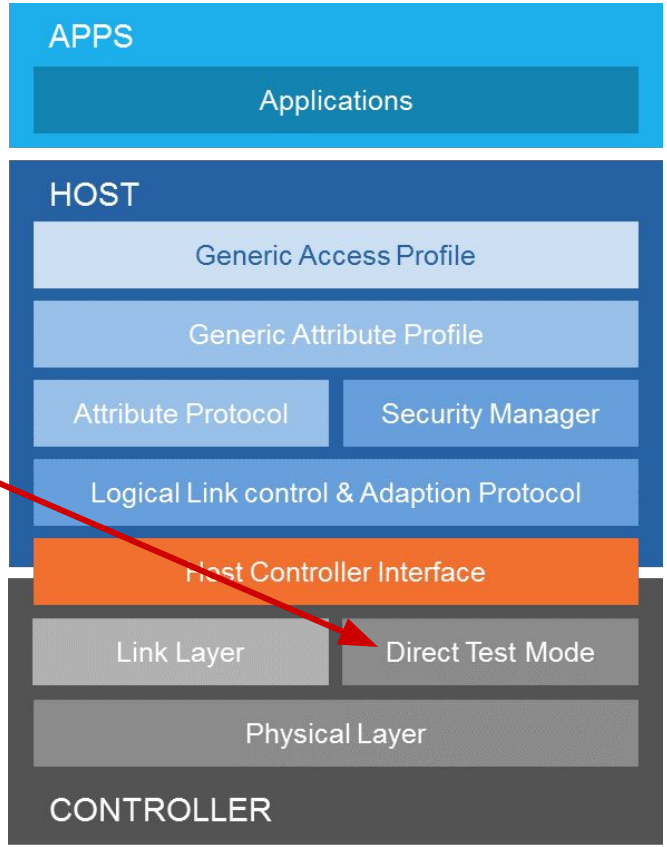
Connection parameters communicated by the master:

- **Connection interval:** the time between the beginning of two consecutive connection events (7.5 ms to 4s).
- **Slave latency:** number of connection events that a slave can skip without resulting in disconnection.
- **Connection supervision timeout:** maximum time between two received valid data packets before a connection is considered lost.



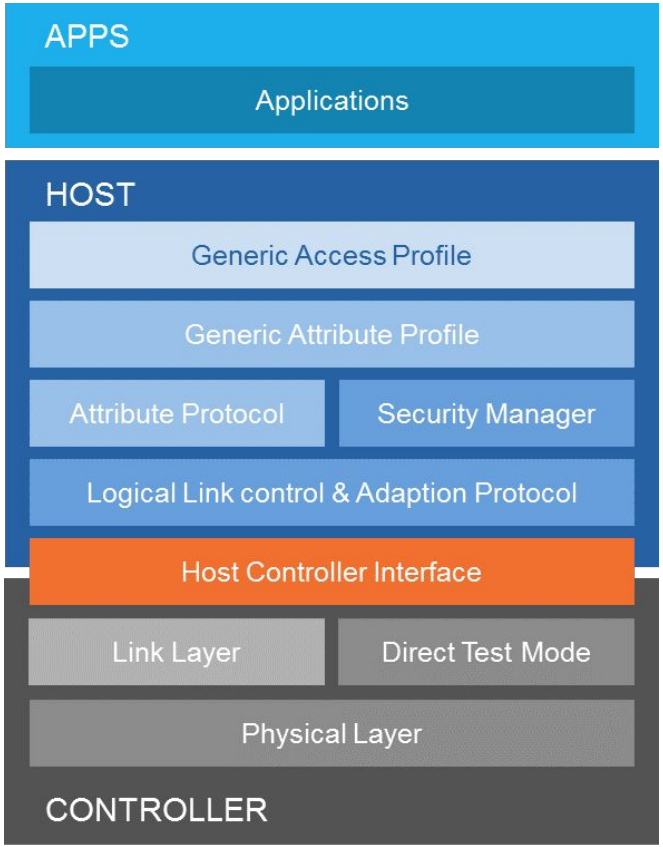
BLE System Architecture

Allows testers to instruct the PHY layer to transmit or receive a given sequence of packets, submitting commands to it either via the HCI or via a 2-wire UART interface.



Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

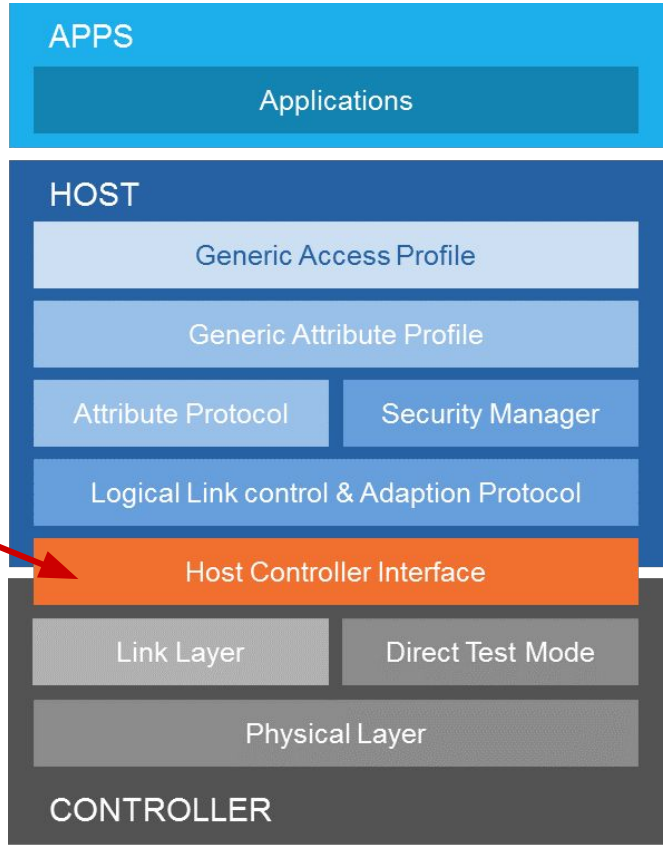
BLE System Architecture



Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

BLE System Architecture

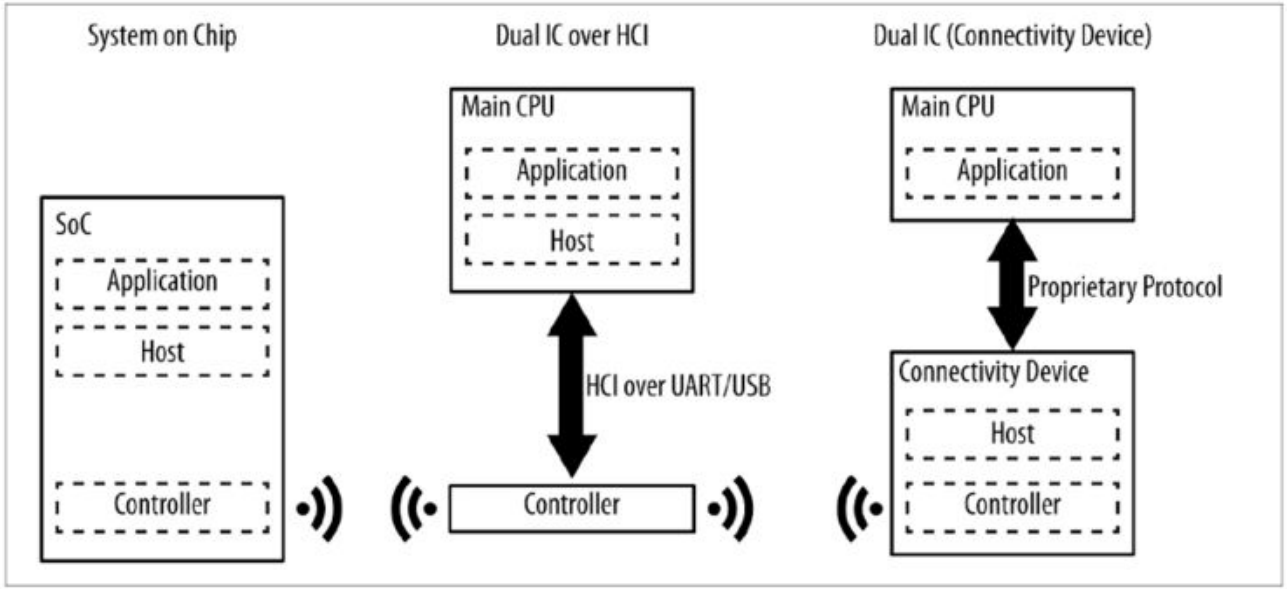
Optional standard interface between the Bluetooth controller subsystem (bottom three layers) and the Bluetooth host.



Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

Host Controller Interface (HCI)

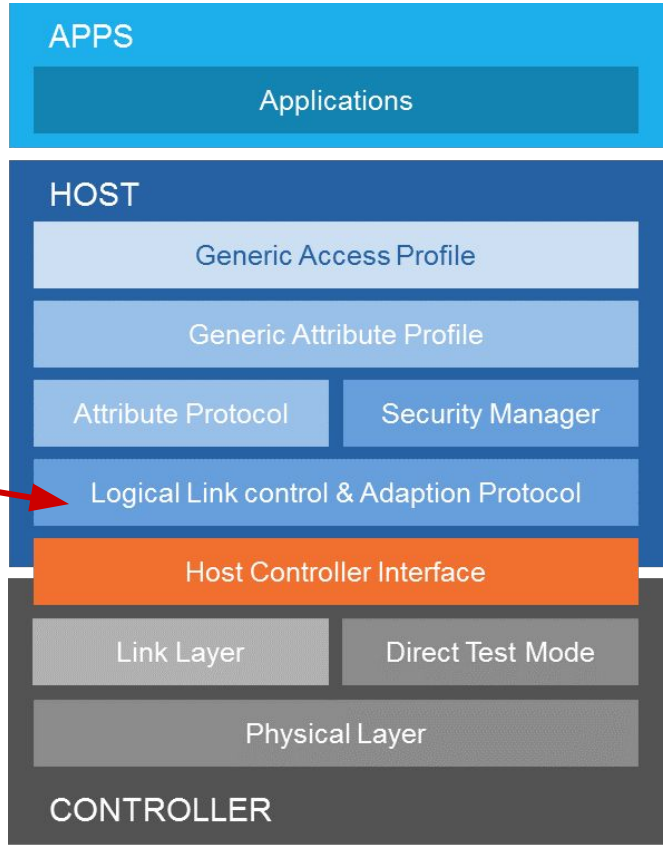
Standard protocol that allows for the communication between a host and a controller to take place across a serial interface.



Source: K. Townsend *et all*, Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking. O'Reilly Media, 2014.

BLE System Architecture

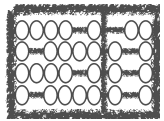
A packet-based protocol that transmits packets to the HCI or directly to the Link Manager in a hostless system. Supports **higher-level protocol multiplexing**, packet segmentation and reassembly, and the conveying of quality of service information to higher layers.



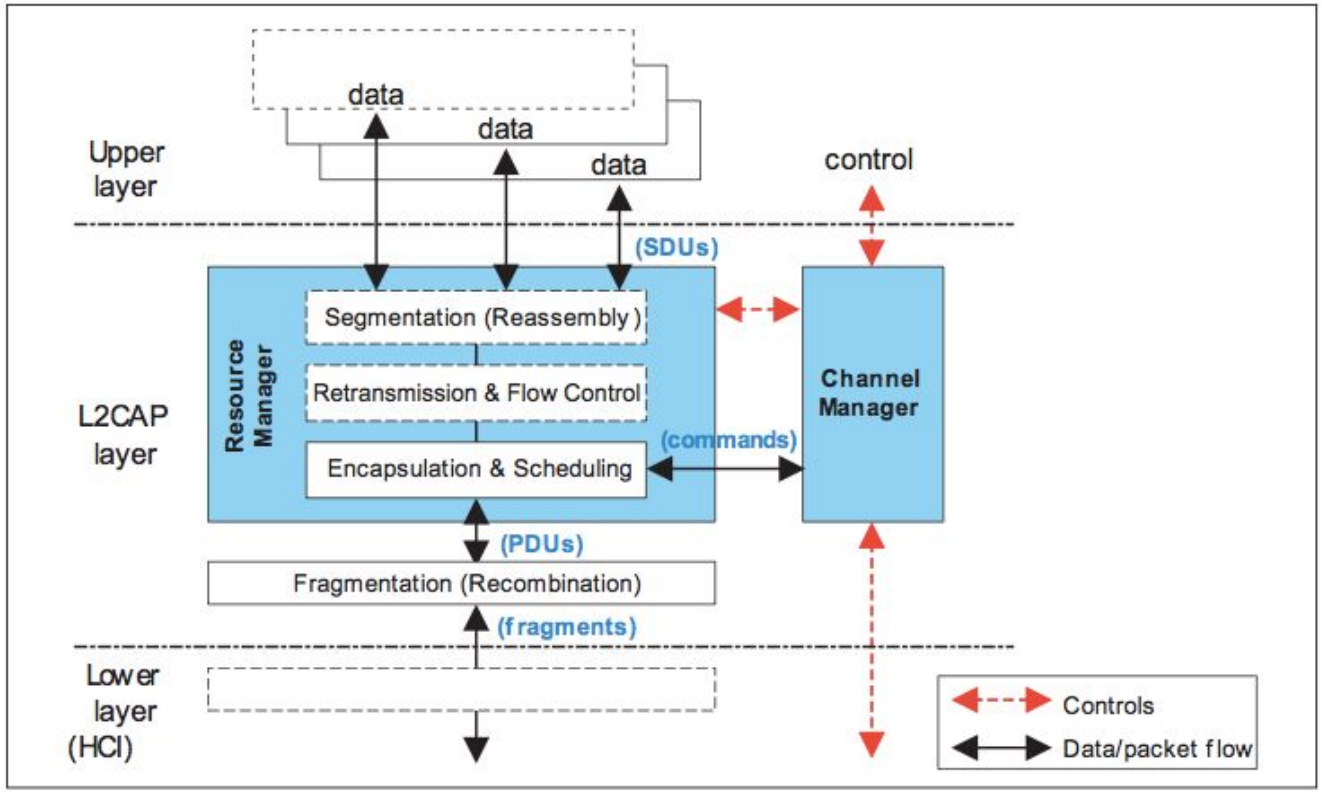
Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

Logical Link Control and Adaptation Protocol (L2CAP)

- Protocol multiplexer: encapsulates messages from the upper layer protocols into the standard BLE packet format and vice versa.
- Fragmentation and recombination:
  - on the transmit side it breaks large messages from upper layers into chunks that fit into the 27-byte maximum payload size of BLE;
  - on the reception side it takes multiple packets that have been fragmented and recombines them into one large packet to send to the upper layers.
- Flow control: window based flow control for each channel
- Quality of Service: connection establishment process allows the exchange of information regarding expected QoS.



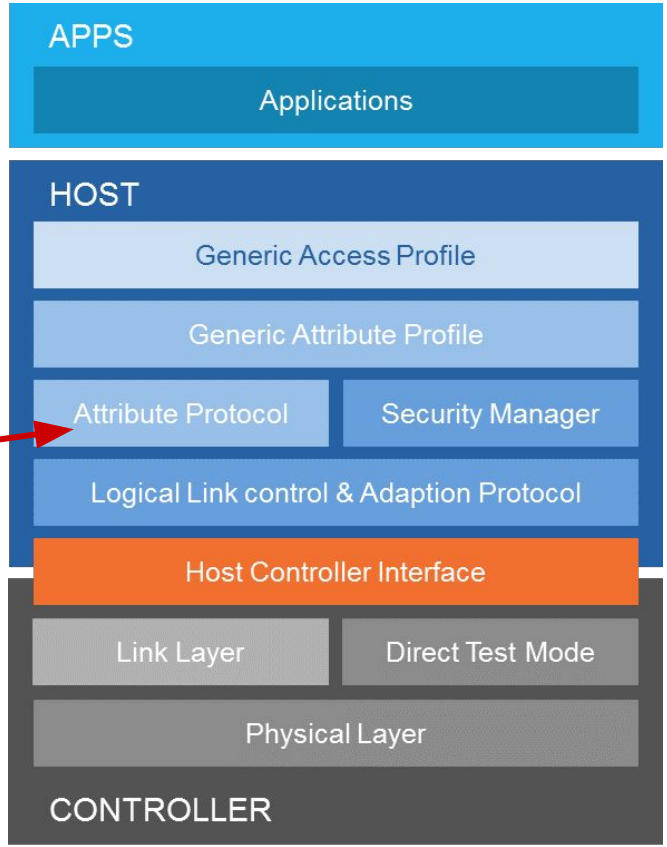
Logical Link Control and Adaptation Protocol (L2CAP)



Source: Bluetooth Specification Version 4.2.

BLE System Architecture

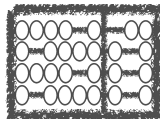
Defines the **client/server protocol** for data exchange once a connection is established. Attributes are grouped together into meaningful services using the Generic Attribute Profile (GATT). ATT is used in LE implementations and occasionally in BR/EDR implementations.



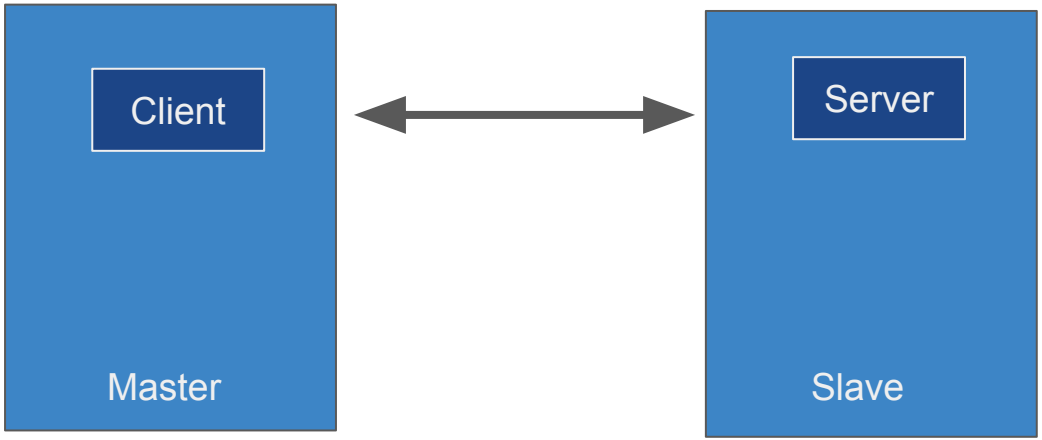
Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

Attribute Protocol (ATT)

- Implements the peer-to-peer protocol between an attribute server and an attribute client.
- The ATT client communicates with an ATT server on a remote device over a dedicated fixed L2CAP channel.
- Attribute - discrete value that has three properties:
  - type: specifies what the attribute represents (Bluetooth SIG defined a set of types)
  - handle: uniquely identifies an attribute on a server
  - permissions: controls whether the attribute can be read or written or whether the attribute value should be sent over an encrypted link.



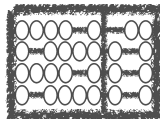
Attribute Protocol (ATT): Client-Server



Example: BLE-enabled temperature sensor. This sensor (*server*) periodically measures the ambient temperature and makes it available for other devices (*clients*) to read.

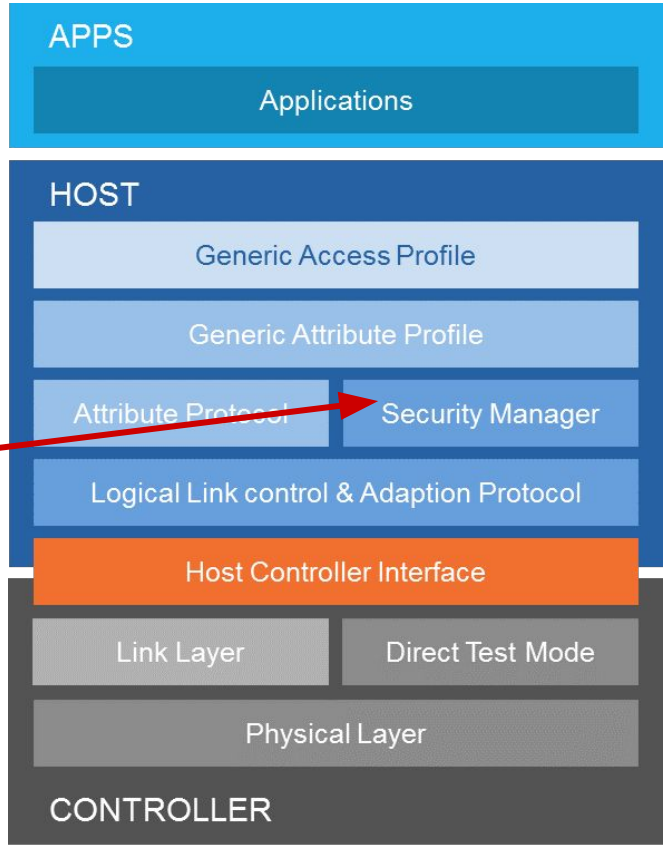
- Attributes: the temperature sensor exposes its temperature reading through an attribute:

Attribute Value: A 32-bit floating-point value representing the temperature in Celsius.
Attribute Type: Temperature Measurement
Attribute Handle: 0x1234 (just an example handle)
Attribute Permissions: read



BLE System Architecture

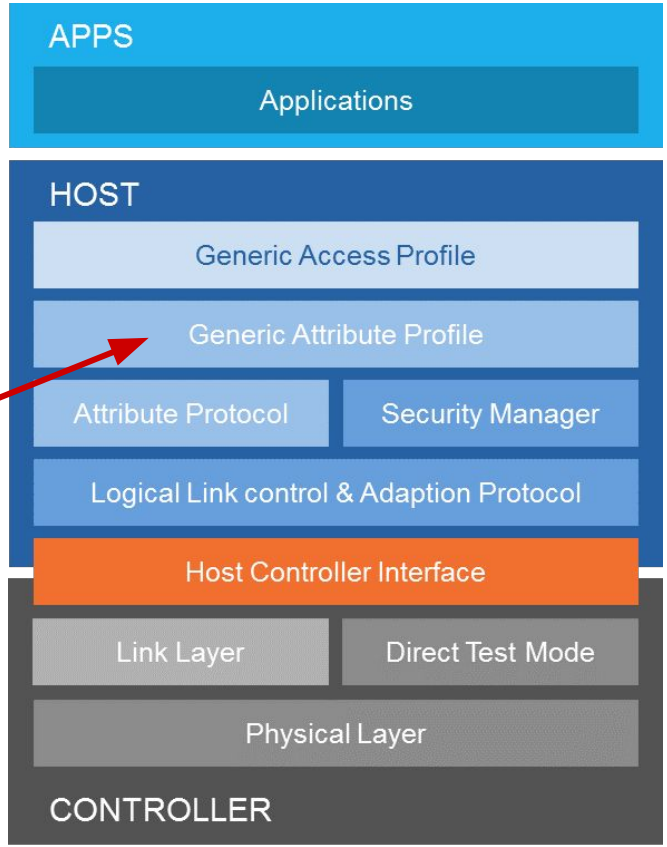
Defines the protocol and behavior that manages pairing integrity, authentication and encryption between Bluetooth devices, and provides a toolbox of security functions that other components use to support almost any level of security needed by diverse applications.



Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

BLE System Architecture

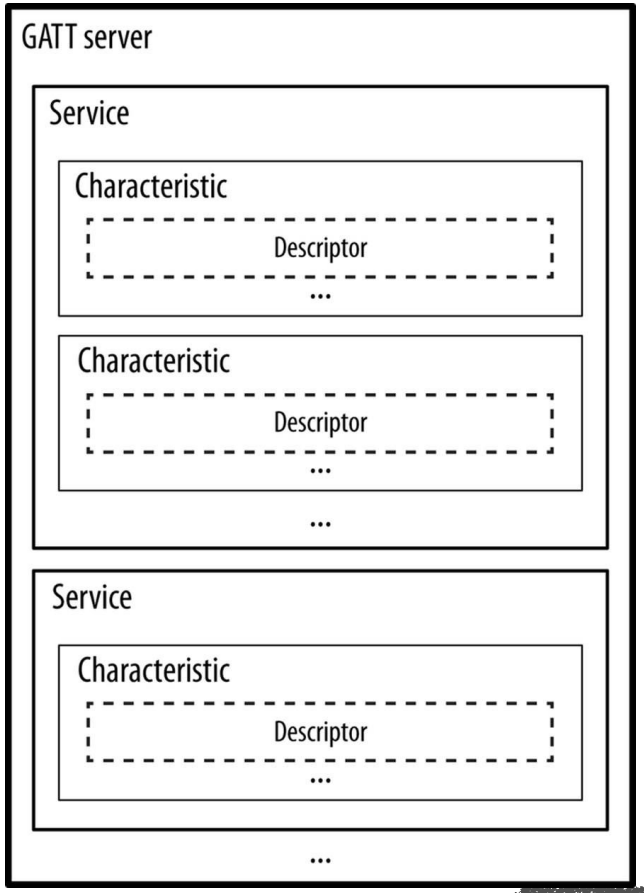
Using the Attribute Protocol, GATT groups services that encapsulate the behavior of part of a device and describes a use case, roles and general behaviors based on the GATT functionality. Its service framework defines procedures and formats of **services** and their **characteristics**, including discovering, reading, writing, notifying and indicating characteristics, as well as configuring the broadcast of characteristics. GATT is used only in Bluetooth LE implementations.

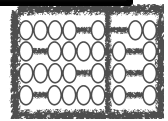


Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

GATT

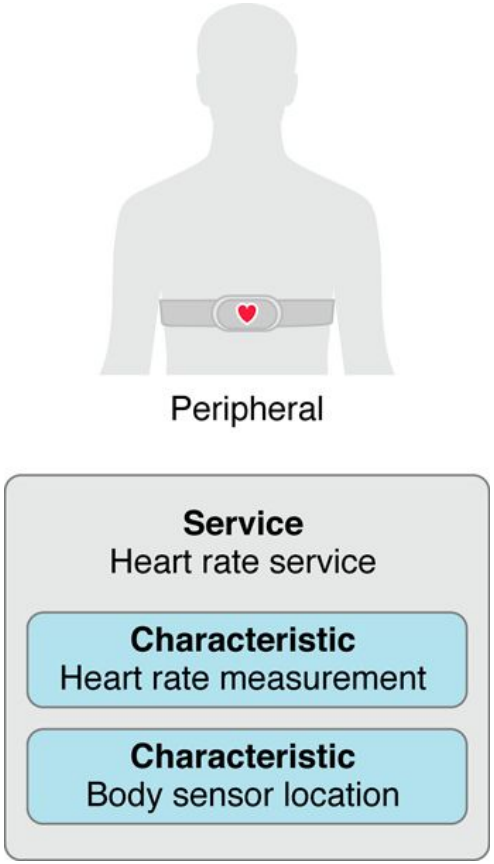
- Attributes in a GATT server are grouped into services.
- Each service can contain zero or more characteristics.
- Characteristics can include zero or more descriptors.





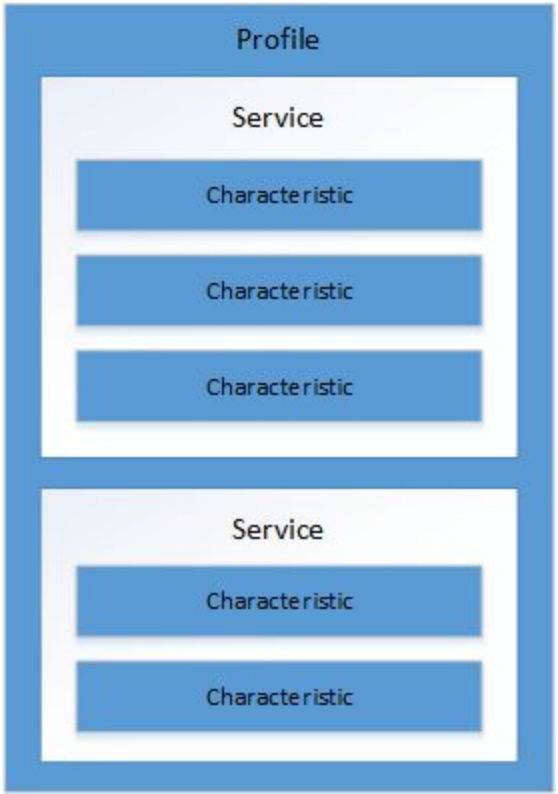
GATT

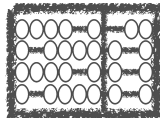
For example, a heart rate service may contain one characteristic that describes the intended body location of the device's heart rate sensor and another characteristic that transmits heart rate measurement data.



Source: https://developer.apple.com

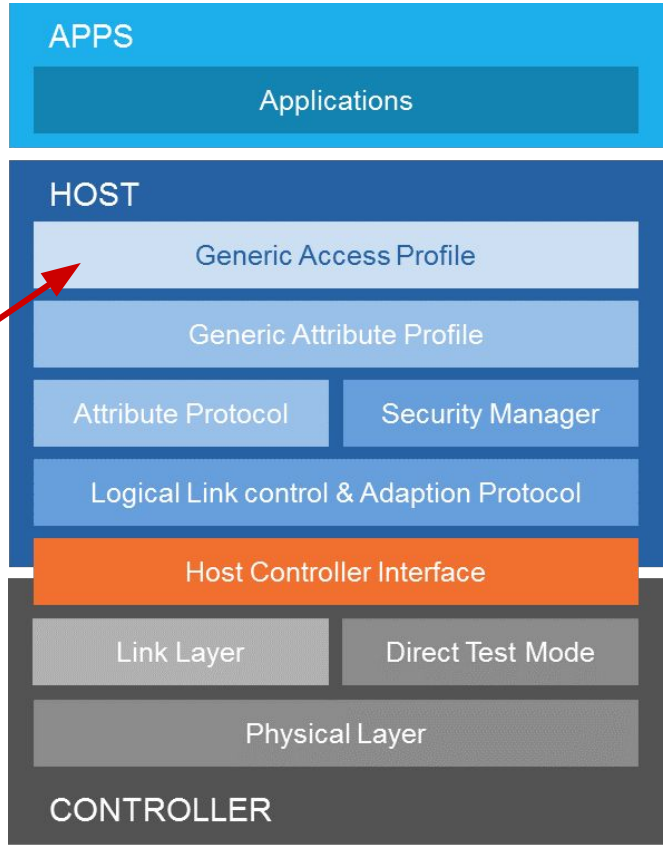
GATT: Profiles





BLE System Architecture

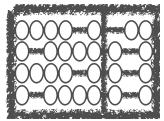
Works in conjunction with GATT in Bluetooth LE implementations to define the **procedures and roles related to the discovery of Bluetooth devices and sharing information**, and link management aspects of connecting to Bluetooth devices.



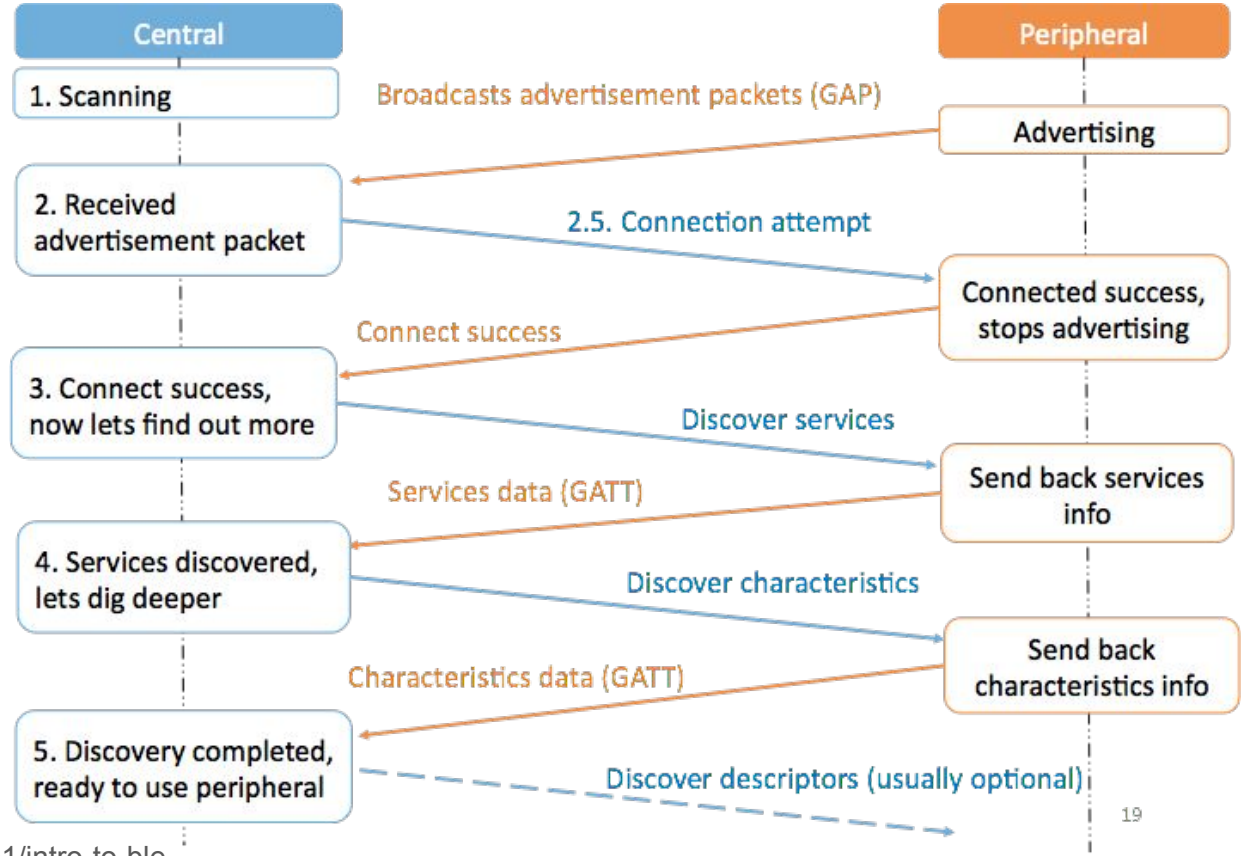
Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

Generic Access Profile (GAP)

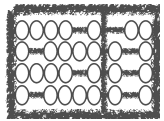
- Allows Bluetooth Low Energy devices to interoperate with each other.
- GAP specifies four roles that a device can adopt to join a BLE network:
    - **Broadcaster:** periodically sends out advertising packets with data.
    - **Observer:** listens for data embedded in advertising packets from broadcasting peers.
    - **Central:** corresponds to the Link Layer master. A device capable of establishing multiple connections to peers.
    - **Peripheral:** corresponds to the Link Layer slave. This role uses advertising packets to allow centrals to find it and, subsequently, to establish a connection with it.



GAP



Source: https://github.com/yeokm1/intro-to-ble



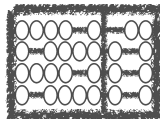
Broadcaster use case

BLE beacons and the Physical Web:

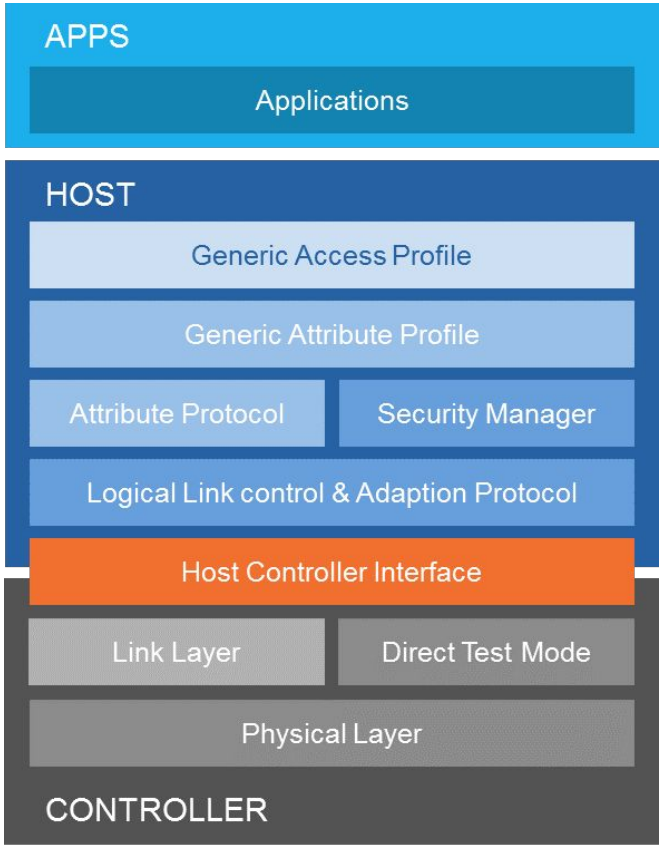
https://www.youtube.com/watch?v=1yaLPRgtlR0

Eddystone format:

https://developers.google.com/beacons/eddystone



BLE System Architecture



Source: https://www.bluetooth.com/specifications/bluetooth-core-specification

Bibliography

- IEEE 802.15-15-15-0107-01-007a, January 2015
- Bluetooth Core Specification Version 4.2.
- Bluetooth Core Specification Version 5.0.
- https://www.bluetooth.com/specifications/bluetooth-core-specification.
- K. Townsend et all, Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking. O'Reilly Media, 2014.