BLE 101 – Bluetooth Low Energy

Agenda

- Bluetooth low energy defined
- Architectural Overview
- Stack Architecture
  - Physical Layer
  - Link Layer
  - HCI Layer
  - L2CAP Layer
  - Security Manager Protocol
  - Attribute Protocol
  - Generic Attribute Profile
  - Generic Access Profile
  - Applications
- Comparison of LE to BR/EDR

Agenda

- Bluetooth low energy defined
- Architectural Overview
- Stack Architecture
  - Physical Layer
  - Link Layer
  - HCI Layer
  - L2CAP Layer
  - Security Manager Protocol
  - Attribute Protocol
  - Generic Attribute Profile
  - Generic Access Profile
  - Applications
- Comparison of LE to BR/EDR

What is Bluetooth low energy?

- Evolution of current Bluetooth standard
  - Open and license free standard
  - Easily integrated within existing Bluetooth technology
- Focus on ultra-low power consumption
  - Ideal for devices with very low battery capacity
- Faster connections
  - “I got it”
  - “I got it” “I want more” “Here it is”

New Technology?

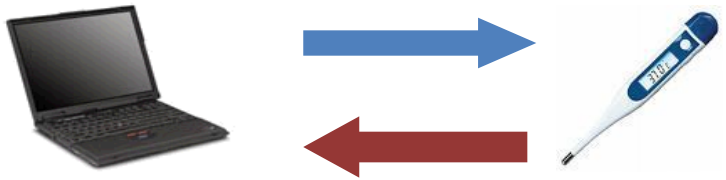
- Yes
  - efficient discovery / connection procedures
  - very short packets
  - asymmetric design for peripherals
  - client server architecture

- No
  - reuse existing BR radio architecture
  - reuse existing HCI logical and physical transports
  - reuse existing L2CAP packets

Why Bluetooth low energy?

- Everything optimized for power consumption

- Button Cell will be the main power supply for peripherals
  - < 15ma peak current
  - < 1µa average current



- Everything has STATE
  - devices expose their state
  - these are servers
- Clients can use the state exposed on servers
  - read it – get current temperature
  - write it – increase set point temperature for room
- Servers can tell clients when state updates
  - notify it – temperature up to set point

- Client Server Architecture
  - proven architecture for web-infrastructure

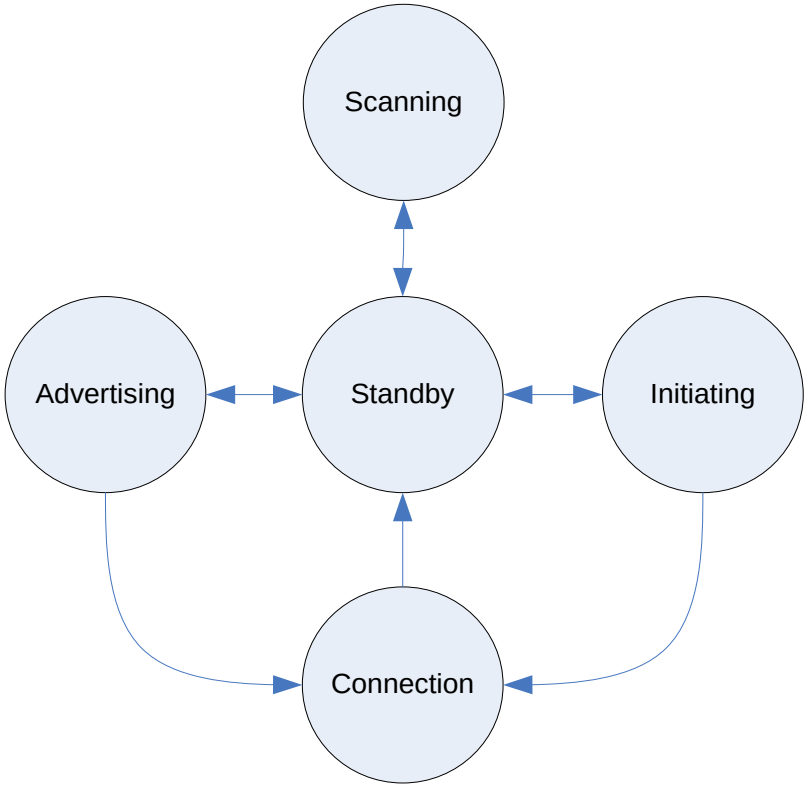
- Gateways allow interconnect of internet & low energy
  - weighing scales send reports to doctor
  - home security web site shows all windows closed
  - assisted living for your parents allows low cost monitoring
  - sports data immediately uploaded via cellular phone

Agenda

- Bluetooth low energy defined
- Architectural Overview
- Stack Architecture
  - Physical Layer
  - Link Layer
  - HCI Layer
  - L2CAP Layer
  - Security Manager Protocol
  - Attribute Protocol
  - Generic Attribute Profile
  - Generic Access Profile
  - Applications
- Comparison of LE to BR/EDR

Operating States and Roles

State		State Description
Standby		Does not transmit or receive packets
Advertising		Broadcasts advertisements in advertising channels
Scanning		Looks for advertisers
Initiating		Initiates connection to advertiser
Connection	Master Role	Communicates with device in the Slave role, defines timings of transmissions
	Slave Role	Communicates with single device in Master Role



- Master/Slave only
  - No scatternet
  - No role switches

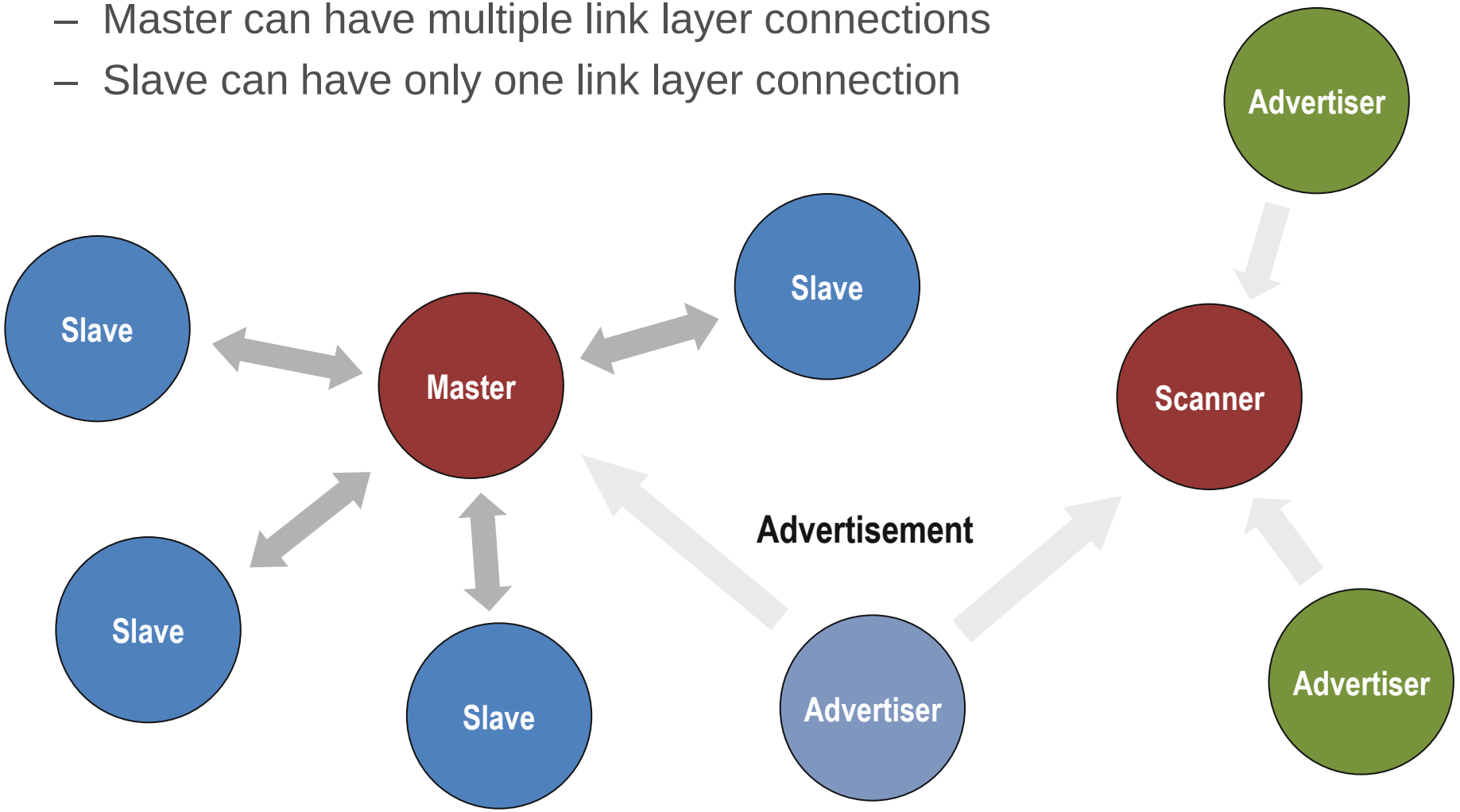
- Bluetooth low energy devices may have more than one instance of the Link Layer state machine at any one time
  - However a Bluetooth low energy device cannot be master and slave at the same time

Multiple State Machine States and Roles		Advertising	Scanning	Initiating	Connection	
					Master	Slave
Advertising		No	Yes	Yes*	Yes *	Yes *
Scanning		Yes	No	Yes	Yes	Yes
Initiator		Yes *	Yes	No	Yes	No
Connection	Master	Yes *	Yes	Yes	Yes	No
	Slave	Yes *	Yes	No	No	No

* Only advertising packets that will not result in Link Layer entering a Slave Role

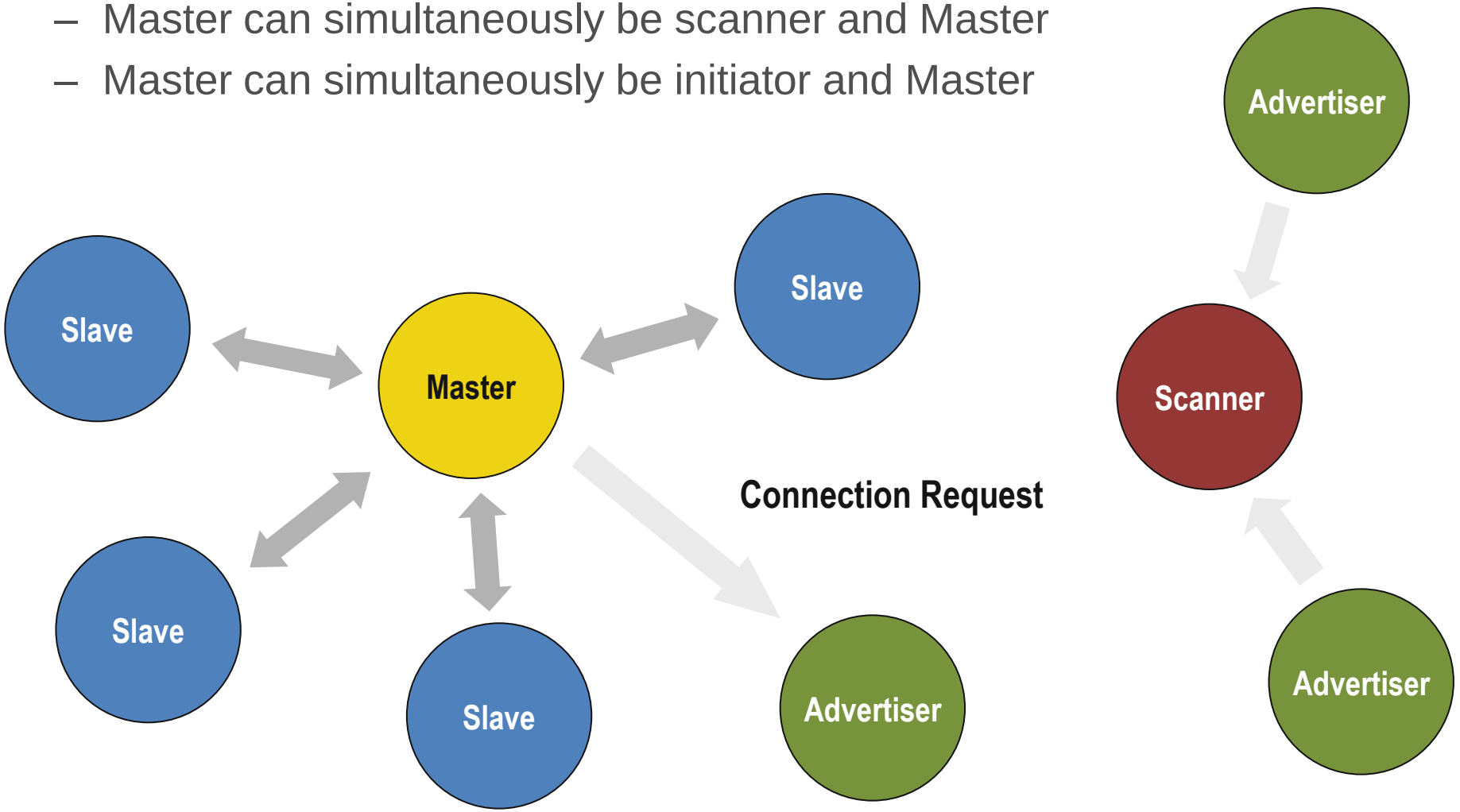
Topology Example

- Star topology
  - Master can have multiple link layer connections
  - Slave can have only one link layer connection



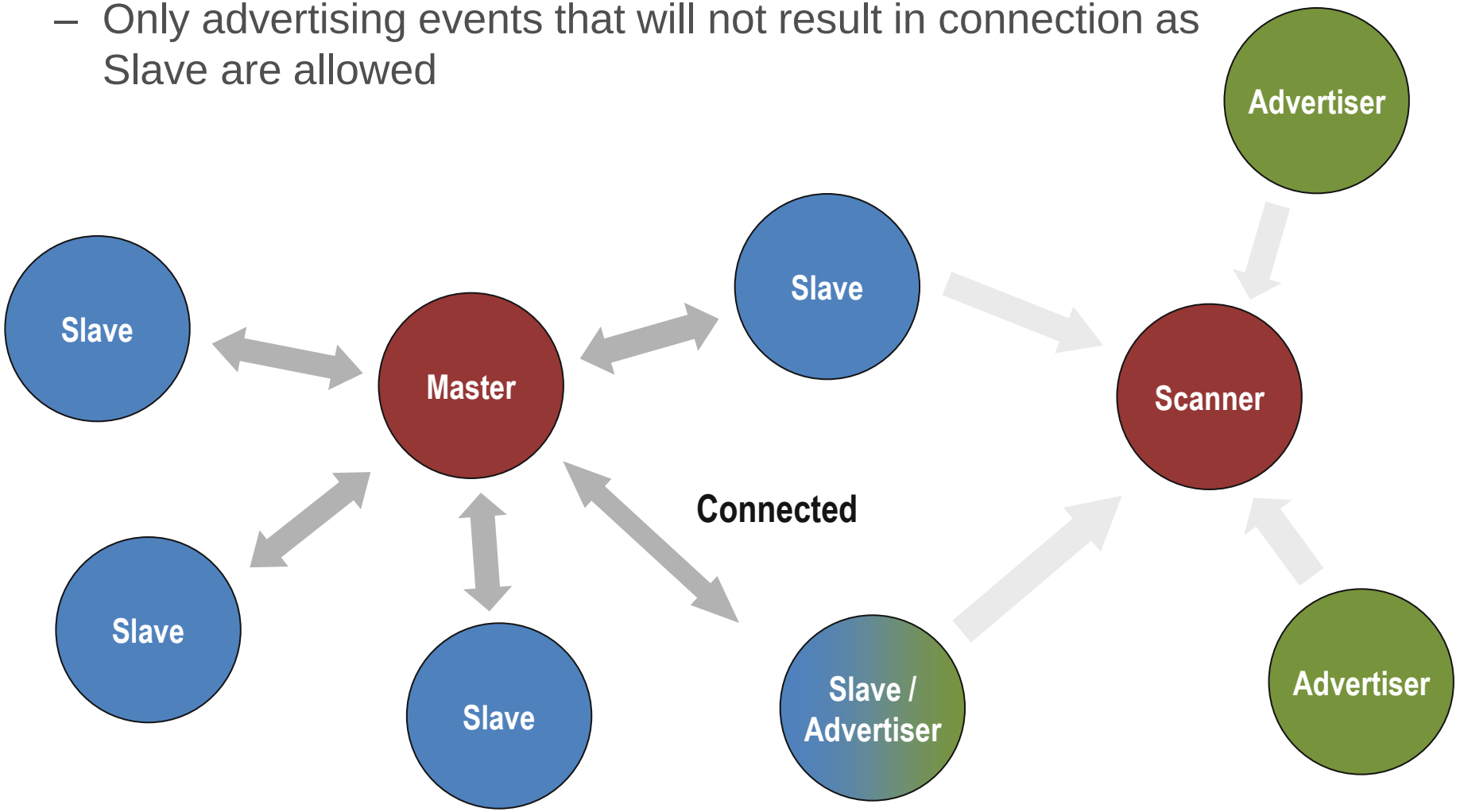
Topology Example

- Initiation of connection requests
  - Master can simultaneously be scanner and Master
  - Master can simultaneously be initiator and Master



Topology Example

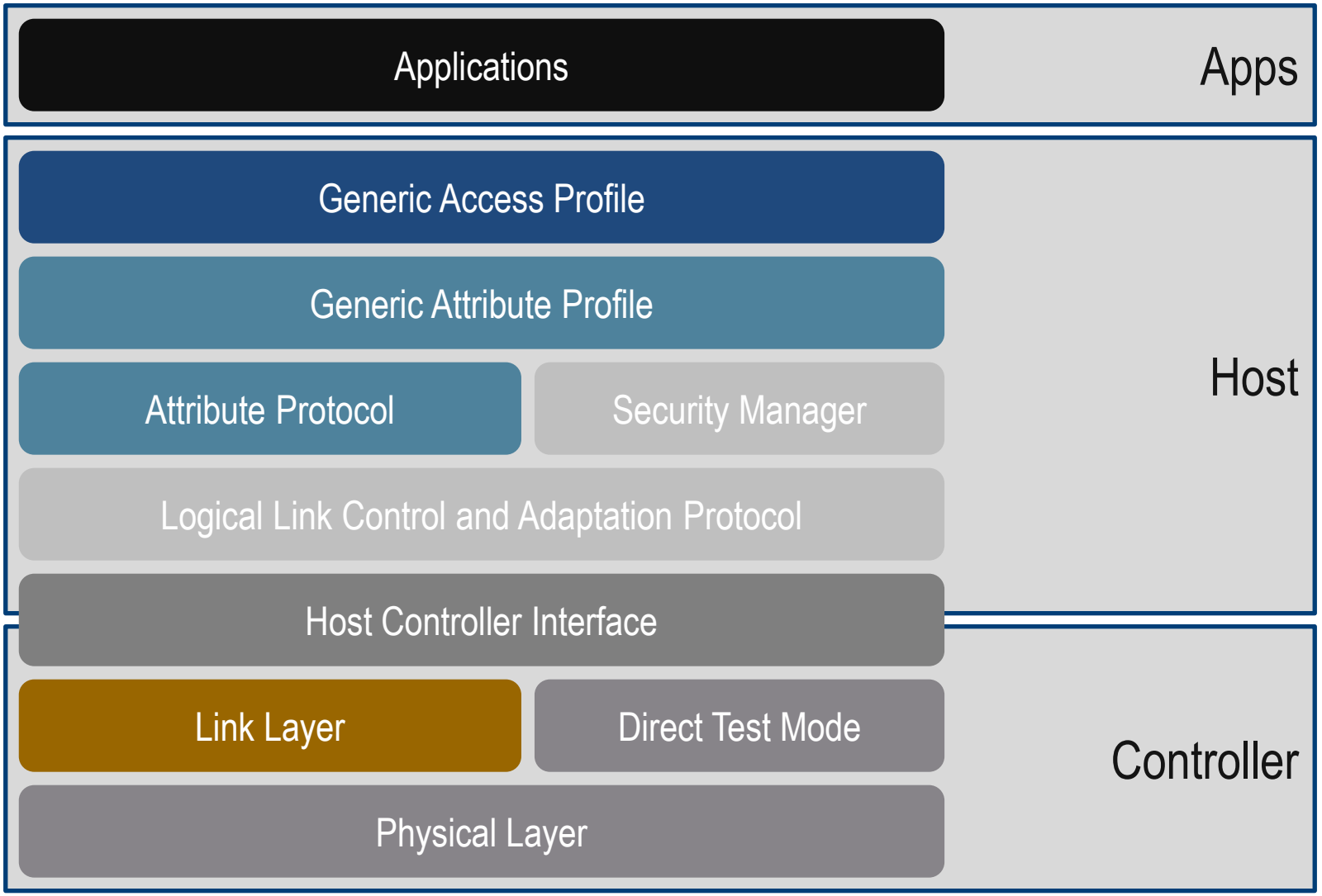
- Master and Slave can both act as Advertiser
  - Only advertising events that will not result in connection as Slave are allowed



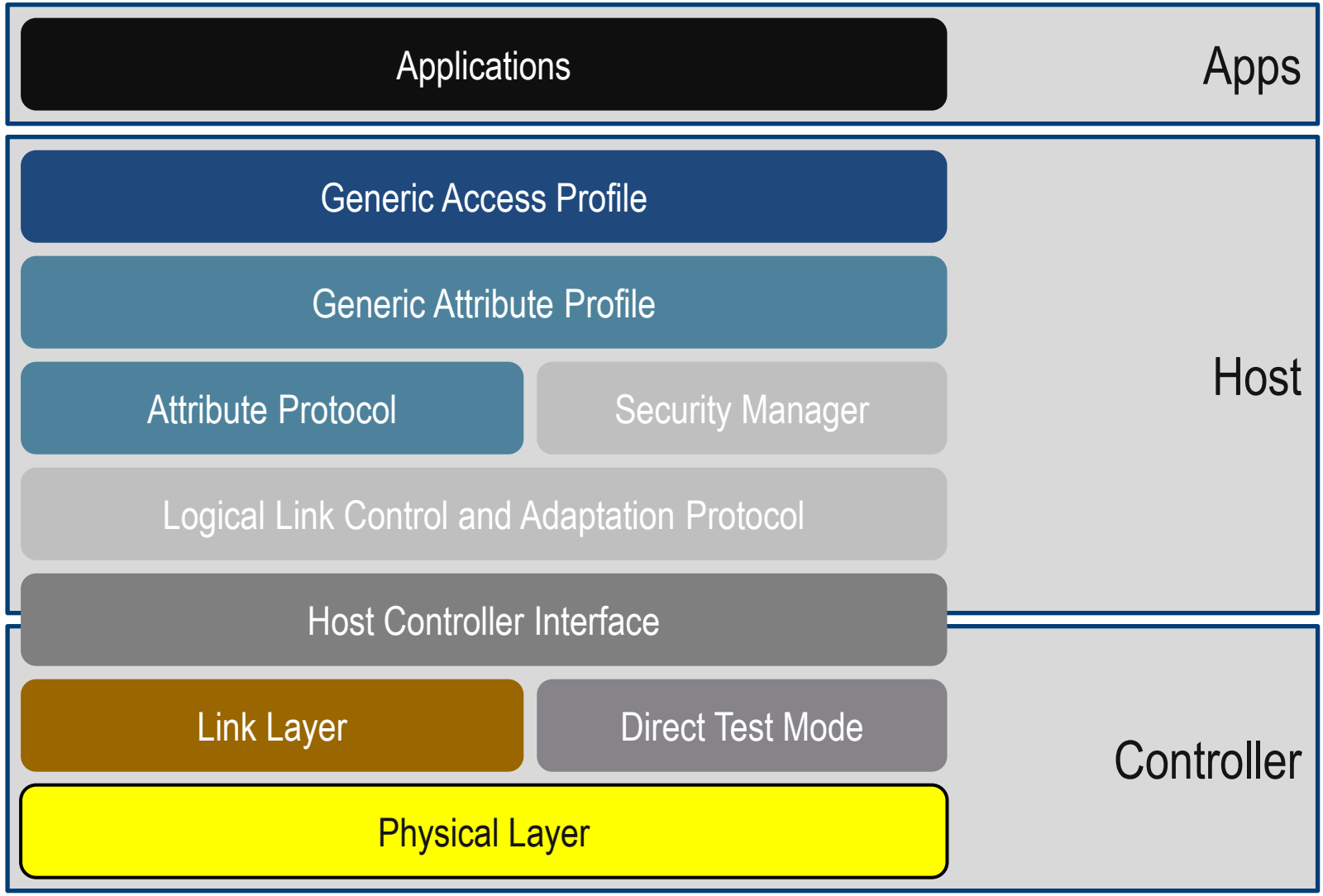
Agenda

- Bluetooth low energy defined
- Architectural Overview
- Stack Architecture
  - Physical Layer
  - Link Layer
  - HCI Layer
  - L2CAP Layer
  - Security Manager Protocol
  - Attribute Protocol
  - Generic Attribute Profile
  - Generic Access Profile
  - Applications
- Comparison of LE to BR/EDR

Stack Architecture

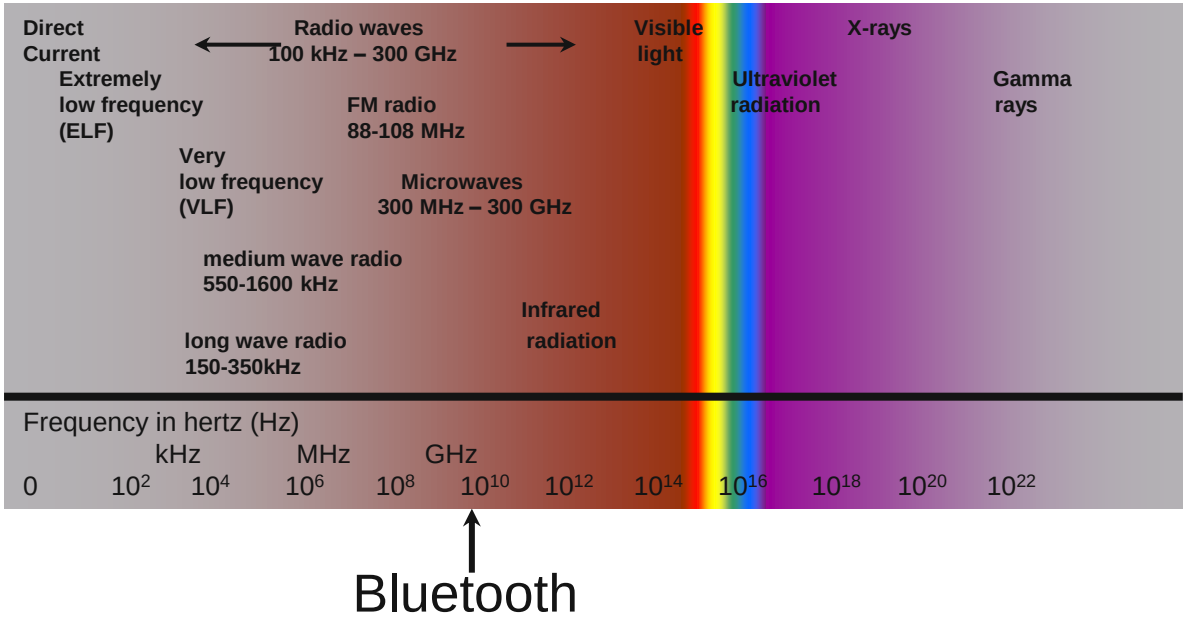


Physical Layer



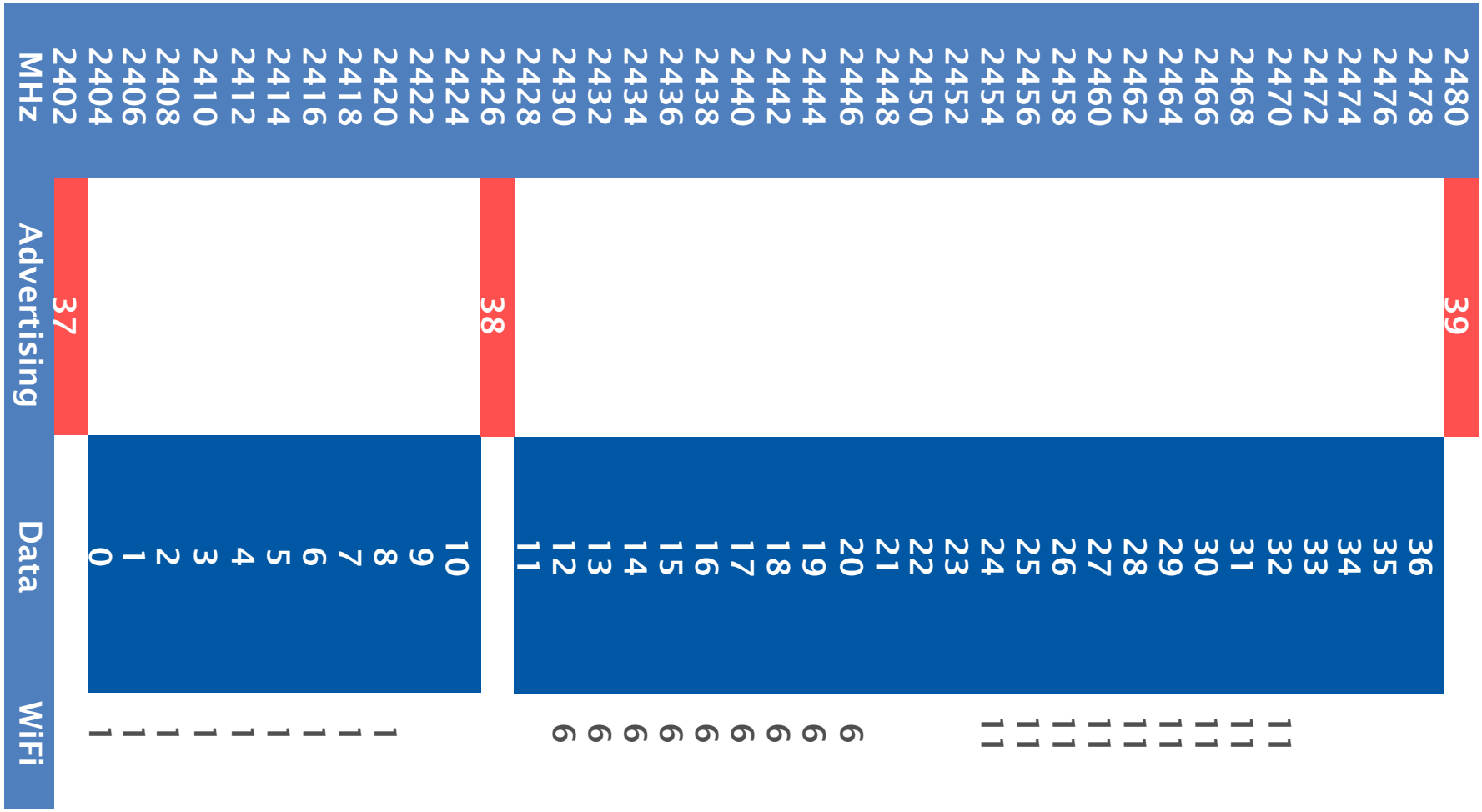
Spectrum Usage

- The 2.4GHz ISM band is a free for all for anyone who wants to use it.



- The 2.4GHz ISM Band is also used by:
  - Microwave Ovens
  - Digital Cordless Phones
  - 802.11b/g

Bluetooth low energy Frequency Plan



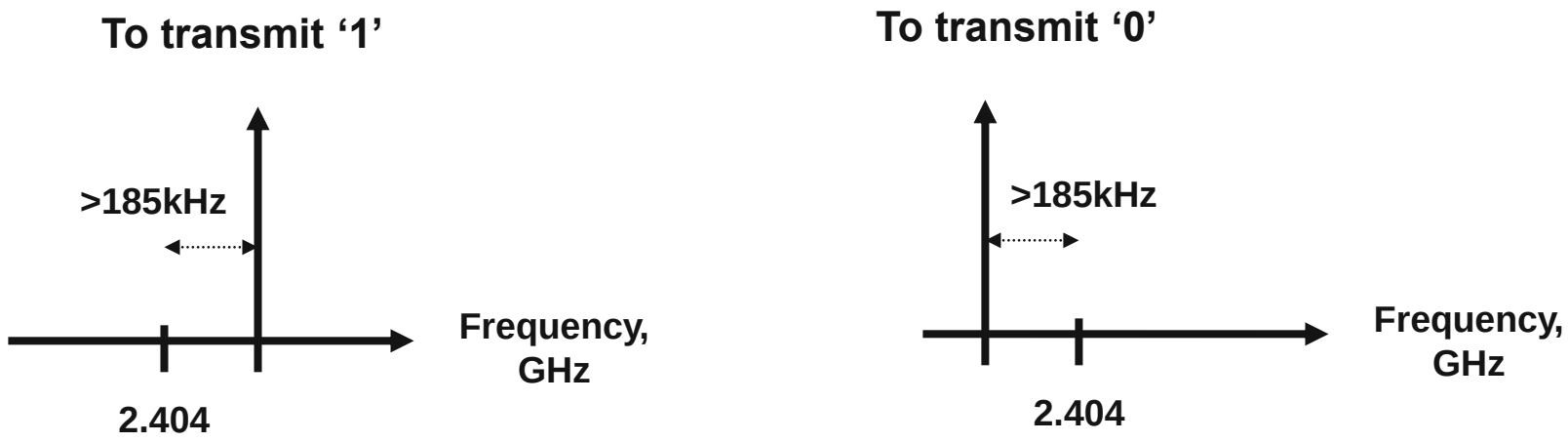
Lower guard band of 2MHz, upper guard band of 3.5MHz

Modulation Scheme

- Data is transmitted using Gaussian Frequency Shift Keying, GFSK
- FSK uses two different frequencies to transmit a binary ‘1’ or ‘0’
- For Bluetooth low energy the two frequencies are:
  - fc + Δ for ‘1’
  - fc - Δ for ‘0’ where fc = frequency of current hop and Δ = >185kHz

Modulation Example

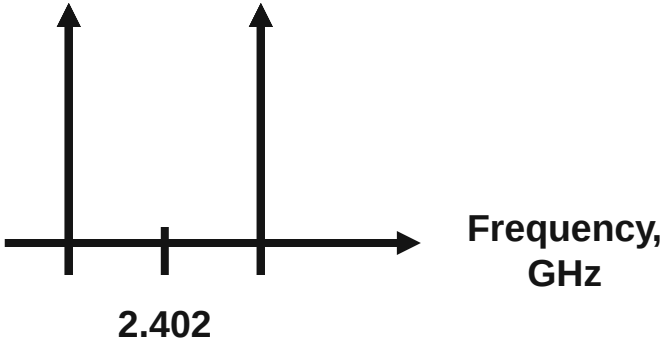
- For channel 0 (Frequency 2.402GHz)

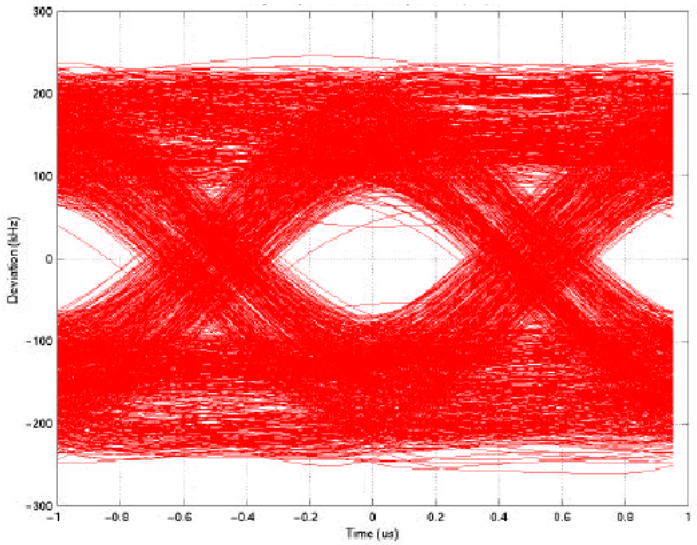


- During one time slot the data can change value every 1µs, so the transmit frequency oscillates back and forth around the center channel frequency.

Modulation Example cont.

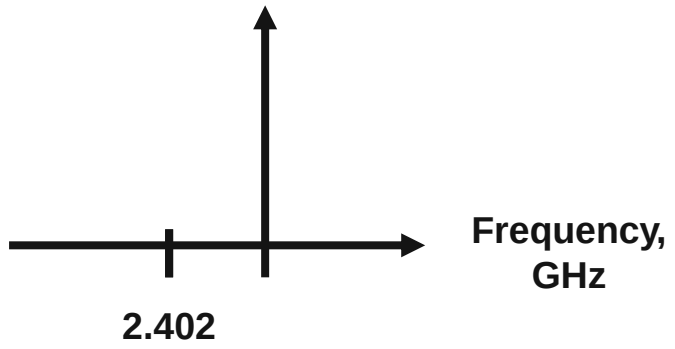
- If the frequency change is allowed to occur ‘instantaneously’ this can lead to inter-symbol interference (ISI) at the receiver which makes it difficult to interpret what state the bit is trying to represent. This will result in high bit errors in the transmitted data.

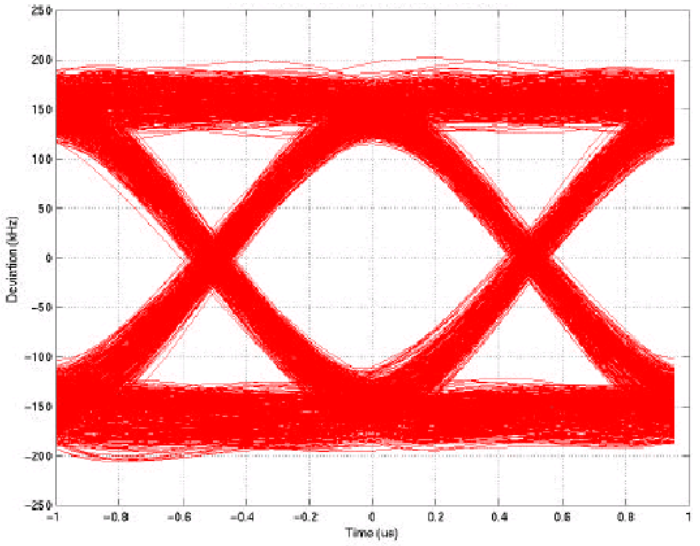




Modulation Example cont.

- To reduce the spectral spreading that causes ISI, Bluetooth low energy uses a 0.5BT Gaussian Filter to slow the transitions between the two frequencies.



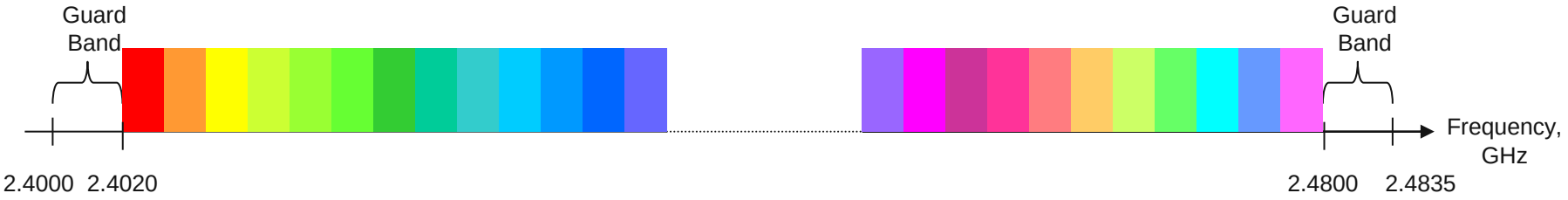


Overcoming Interference

- Due to the unrestricted nature of the ISM band, Bluetooth low energy must overcome interference from other systems and minimize its interference on other systems.
- Bluetooth low energy does this by using a Frequency Hopping Spread Spectrum (FHSS) technique.
- This spreads the RF power across the spectrum which reduces interference and the spectral power density.

Frequency Hopping Spread Spectrum - FHSS

- Bluetooth low energy splits the spectrum up into 37 1MHz wide channels data channels
- FHSS occurs while in a connection
- The frequency hops follow a hop-length that is pseudo-random per connection
  - Communicated in the “Connection Request”
  - Provides instant adaptive frequency hopping capability
  - Can be updated using a channel update message

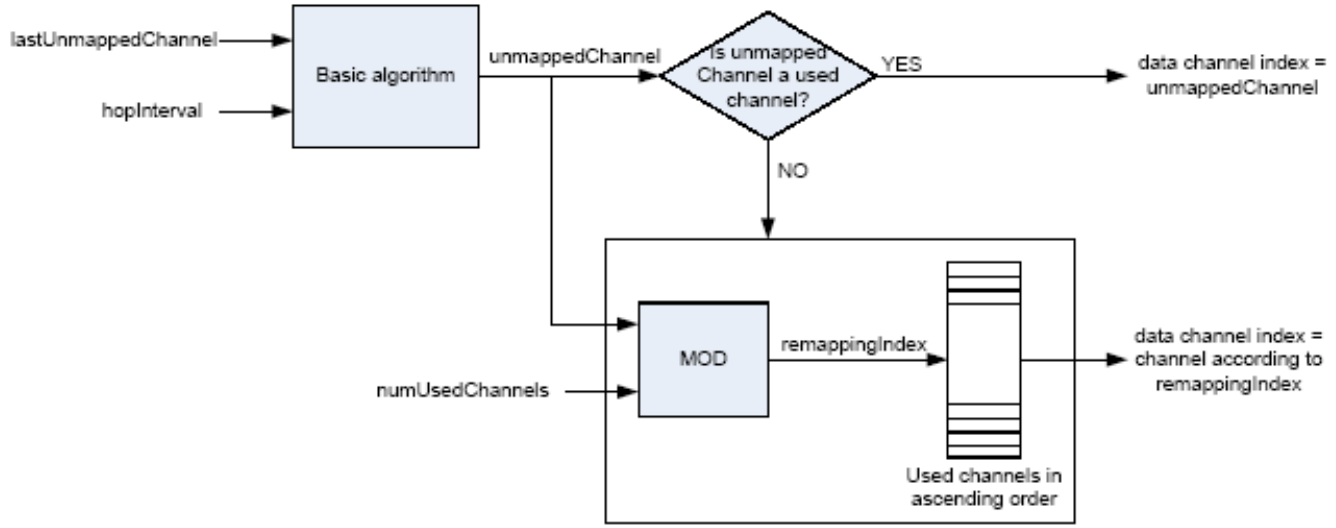


Bluetooth low energy Frequency Hopping

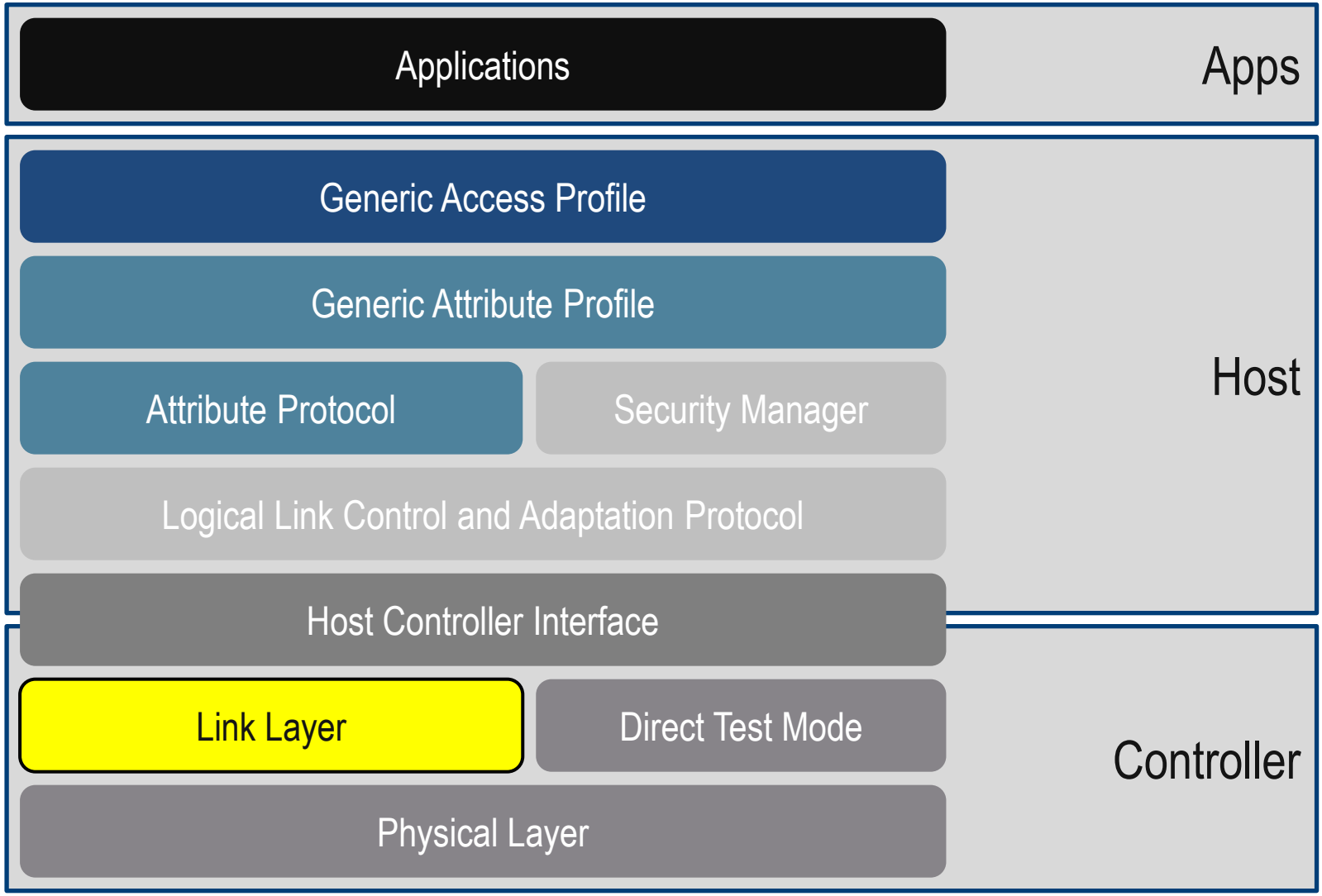
- Definitions:
  - Used channel – used for the connection
  - Unused channel – not used for the connection

- Pseudo-random hop length communicated in “Connection Request” by master
  - fn+1 = (fn + hop) mod 37

- Unused channels are remapped



Link Layer



Bluetooth low energy addresses

- Device addresses
  - Public
    - 48-bit address obtained from IEEE Registration authority
    - BD_ADDR in dual-mode devices

  - Private
    - Optional for Bluetooth low energy devices
    - Changes frequently – enables privacy

- Access addresses
  - 32-bit pseudo random access address
  - Changes with each link layer connection

Device Filtering

- Devices maintain a “white list”
  - Storage of device addresses for device filtering

- Filter policy can be set to:
  - Advertiser
    - Process scan/connection requests from devices in white list
    - Process all scan/connection requests (default advertiser filter policy)
    - Process connection requests from all devices but only scan requests in white list
    - Process scan requests from all devices but only connection requests in white list
  - Scanner
    - Process advertising packets from devices in white list
    - Process all advertising packets (default scanner filter policy)
  - Initiator
    - Process connectable advertising events from devices in white list
    - Process connectable events only from single device specified by host

White Lists

- No need to send every advertising packet to Host
  - only send information from devices in white list

- Allows “connect to white list” semantics
  - a master can automatically connect to a set of devices
  - will connect when sees adverts from these devices
  - allows very fast connections from many devices

Physical Channels

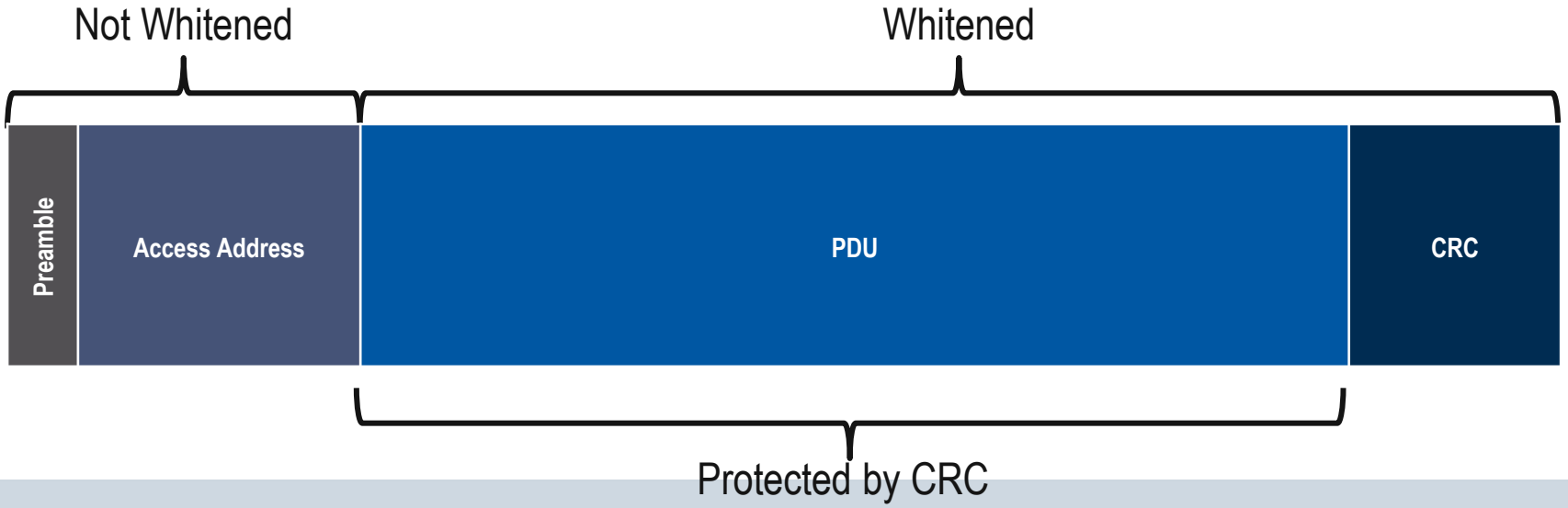
- ISM band split into 40 channels of two types
  - Advertising Channels
  - Data channels

- Advertising Channels
  - Frequencies
    - 2402 (37), 2426 (38), 2480 (39)
  - Usage
    - Discovering devices
    - Initiating a connection
    - Broadcasting data

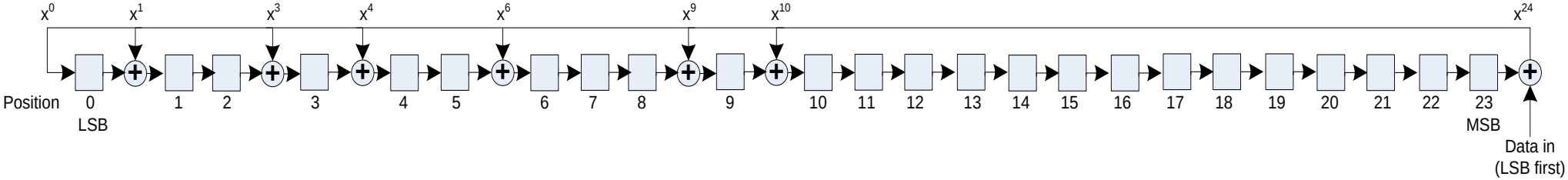
- Data Channels
  - Frequencies
    - 2404-2424 (0-10), 2428-2478 (11-36)
  - Usage
    - Communicating between connected devices

One Packet Format

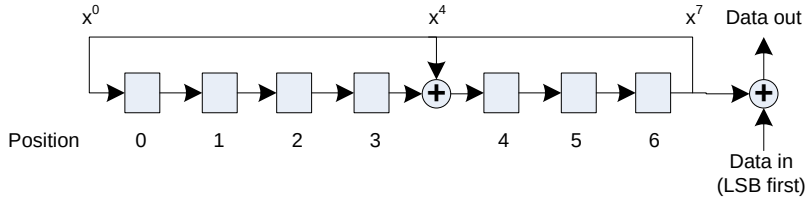
- Used for Advertising and Data Channel Packets
- Preamble (0x55, 0xAA)
  - Frequency synchronization, symbol timing estimation, AGC training
- Access Address
  - Advertising packets – always 0x8e89bed6
  - Data packets – different for each link layer connection
- Packet Data Unit
  - Defined based upon packet types



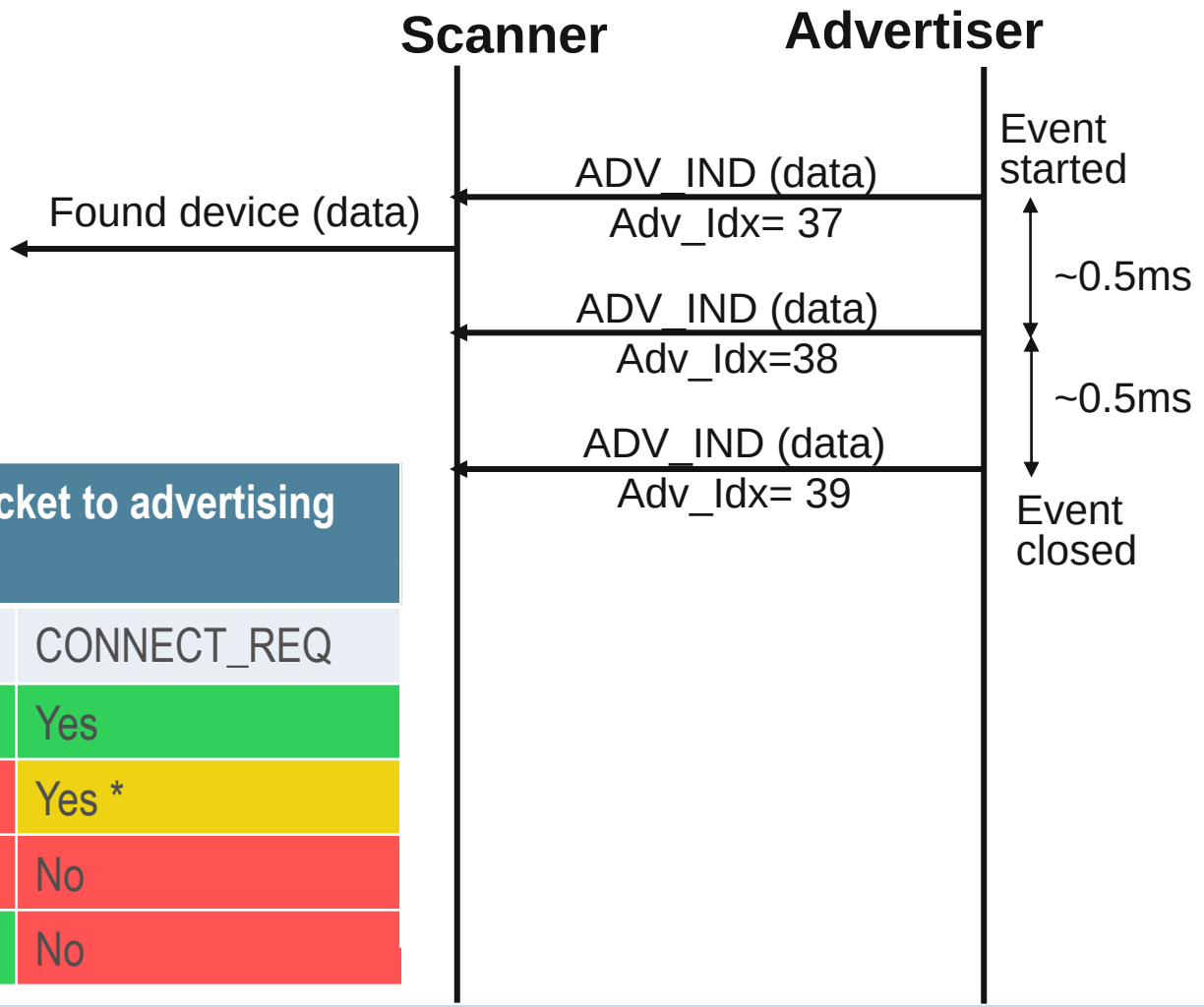
- CRC
  - Calculated over entire PDU
  - Detects all 1, 2, 3, 5, and odd bit errors in PDU
  - Shift register is preset
    - Data channel - with value from CONNECT_REQ
    - Advertising channel – 0x555555



- Data Whitening
  - ‘Random’ data can contain long strings of ‘1’s or ‘0’s.
  - Long durations at one level can cause an unwanted DC bias to be present in received data which can upset some data slicers.
  - To counter this the PDU and CRC is ‘whitened’
  - Uses same LFSR as Bluetooth BR



Advertising

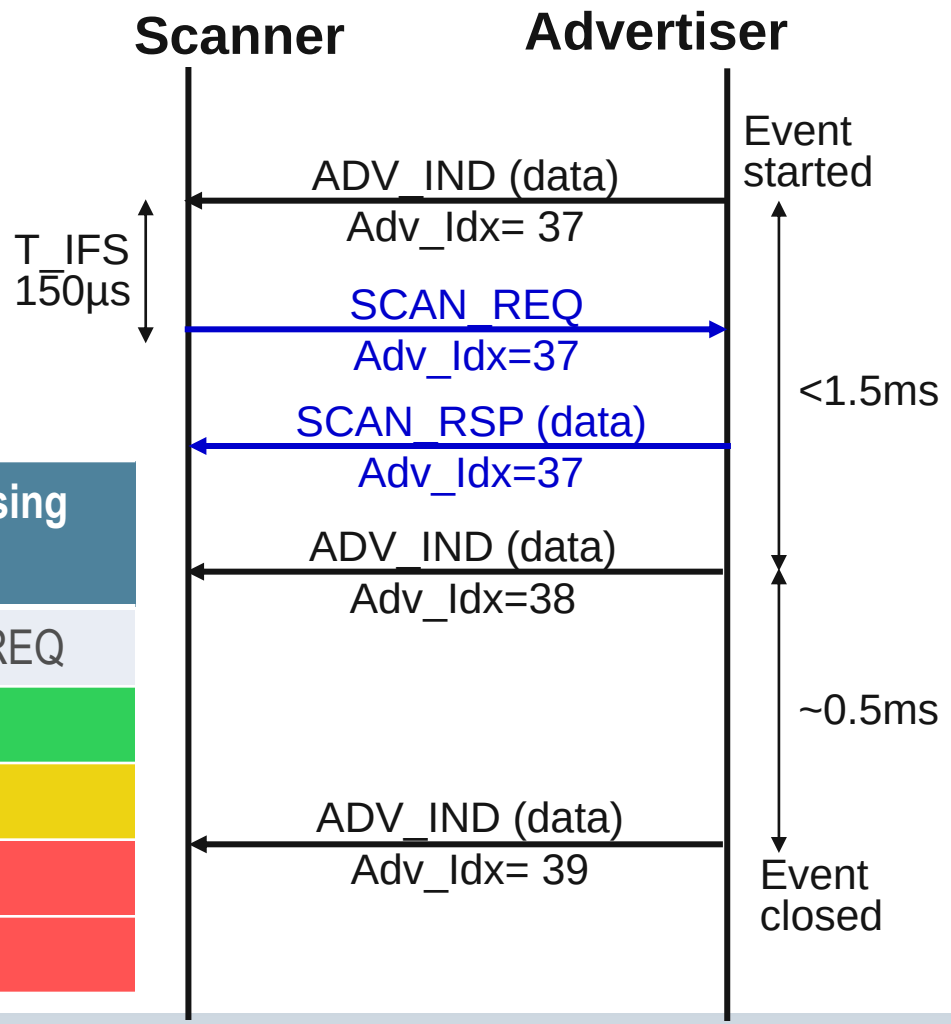


Advertising Event	Response packet to advertising event	
	SCAN_REQ	CONNECT_REQ
ADV_IND	Yes	Yes
ADV_DIRECT_IND	No	Yes *
ADV_NONCONN_IND	No	No
ADV_DISCOVER_IND	Yes	No

* If initiator address matches

Active Scanning

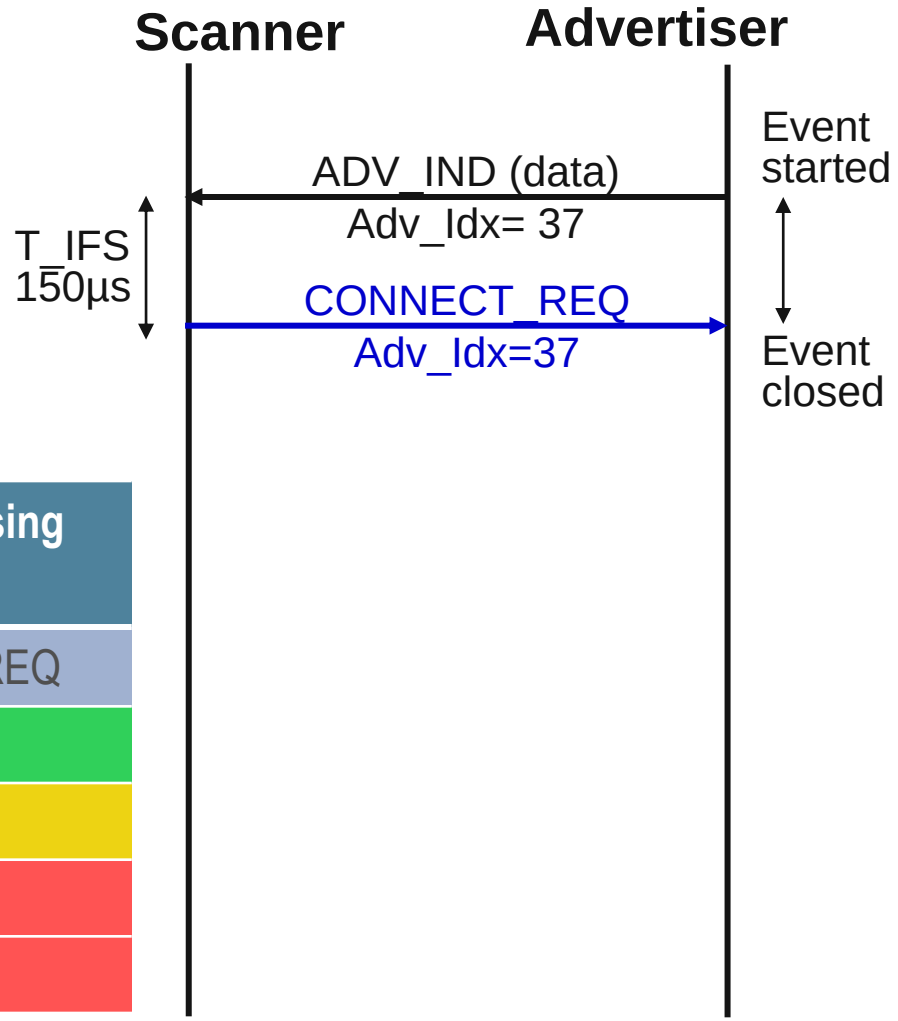
Advertising Event	Response packet to advertising event	
	SCAN_REQ	CONNECT_REQ
ADV_IND	Yes	Yes
ADV_DIRECT_IND	No	Yes *
ADV_NONCONN_IND	No	No
ADV_DISCOVER_IND	Yes	No



* If initiator address matches

- CONNECT_REQ includes the following data:
  - Transmit window size
  - Transmit window offset
  - Connection interval
  - Slave latency
  - Connection Timeout
  - Hop sequence
  - Channel Map
  - CRC initialization value

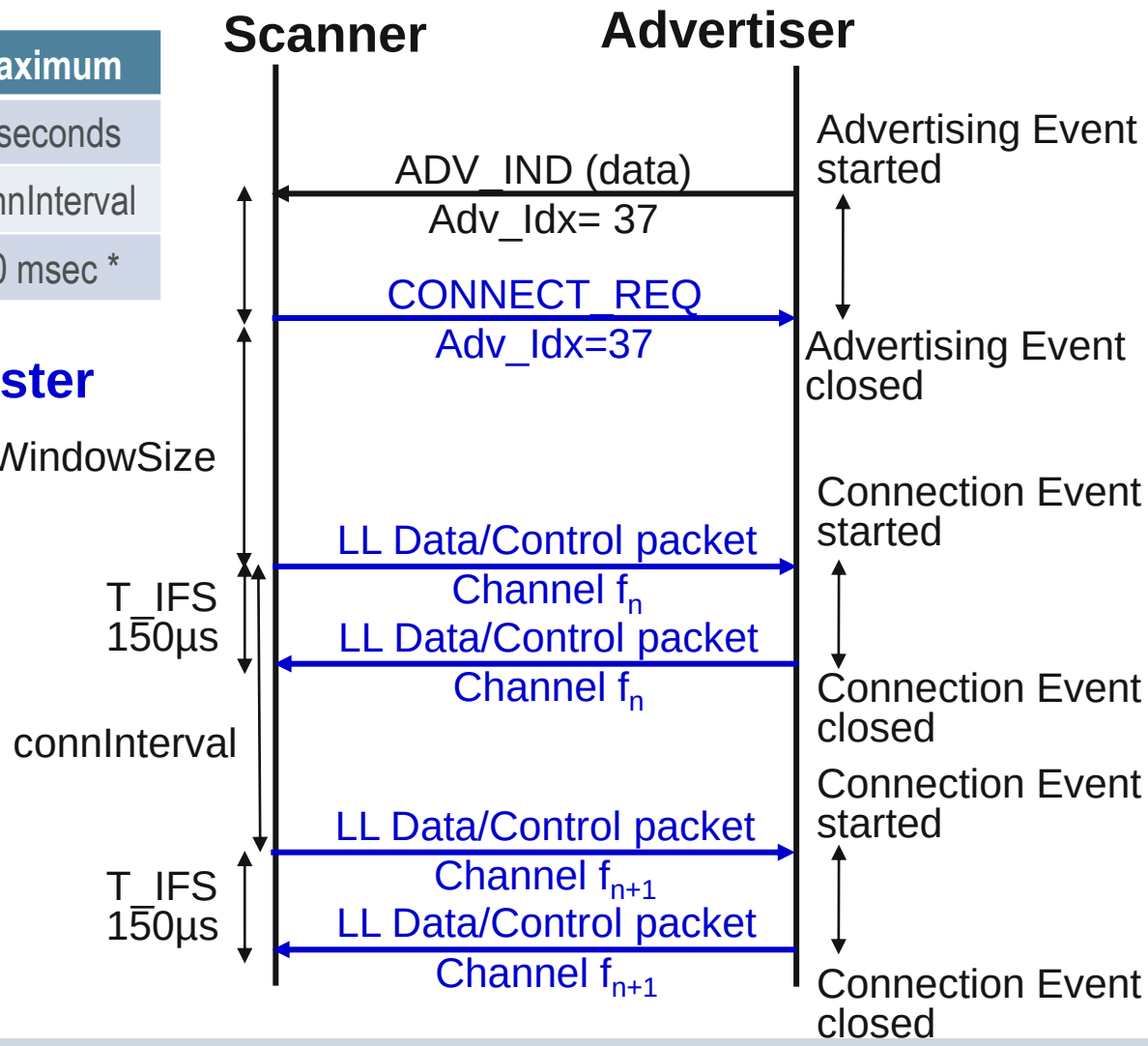
Advertising Event	Response packet to advertising event	
	SCAN_REQ	CONNECT_REQ
ADV_IND	Yes	Yes
ADV_DIRECT_IND	No	Yes *
ADV_NONCONN_IND	No	No
ADV_DISCOVER_IND	Yes	No



Parameter	Minimum	Maximum
connInterval	7.5 msec	4 seconds
WindowOffset	0	connInterval
WindowSize	1.25 msec	10 msec *

Master

1.25ms < t < WindowOffset+WindowSize



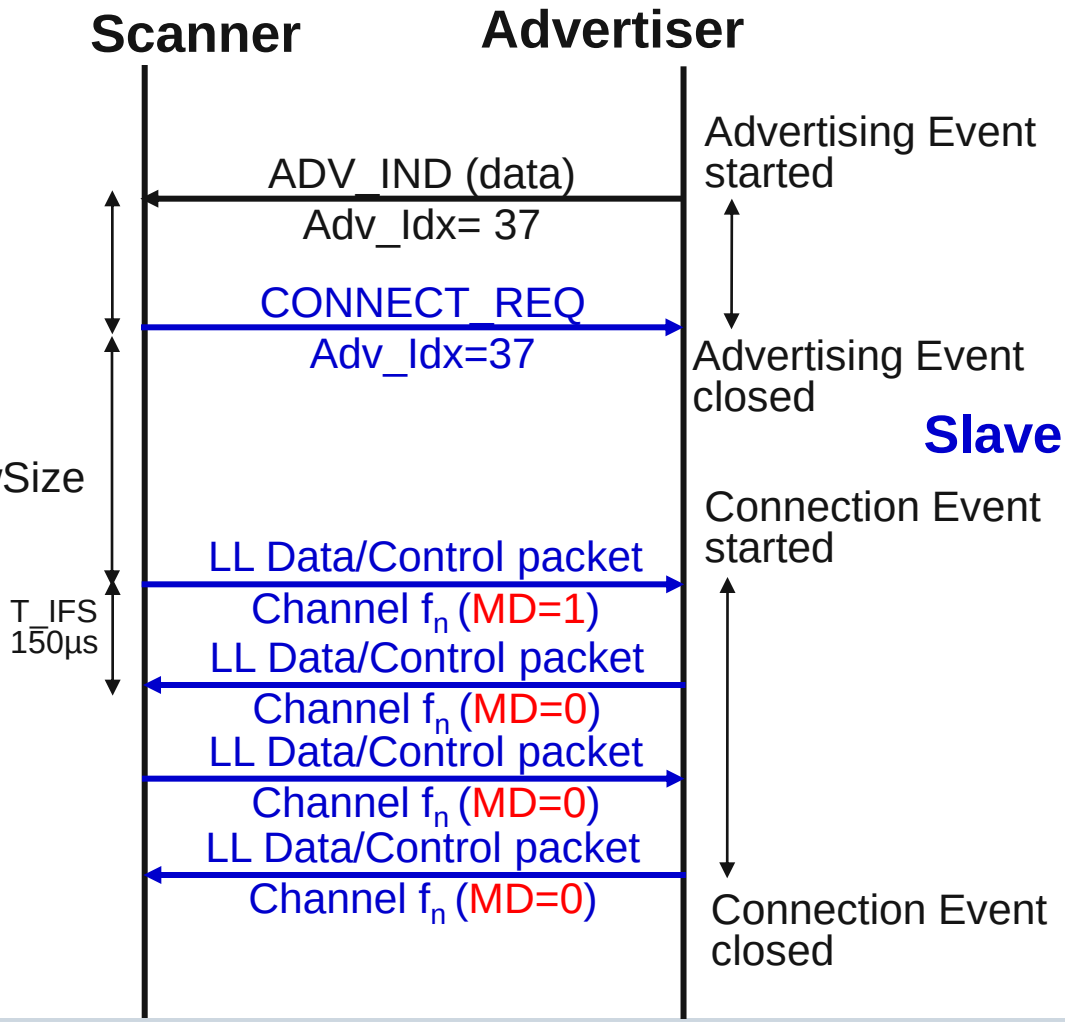
Connection using More Data (MD)

Parameter	Minimum	Maximum
connInterval	7.5 msec	4 seconds
WindowOffset	0	connInterval
WindowSize	1.25 msec	10 msec *

Master

1.25ms < t < WindowOffset+WindowSize

		Master	
		MD=0	MD=1
Slave	MD=0	Master closes connection	Master transmits Slave listens
	MD=1	Master transmits Slave listens	Master transmits Slave listens

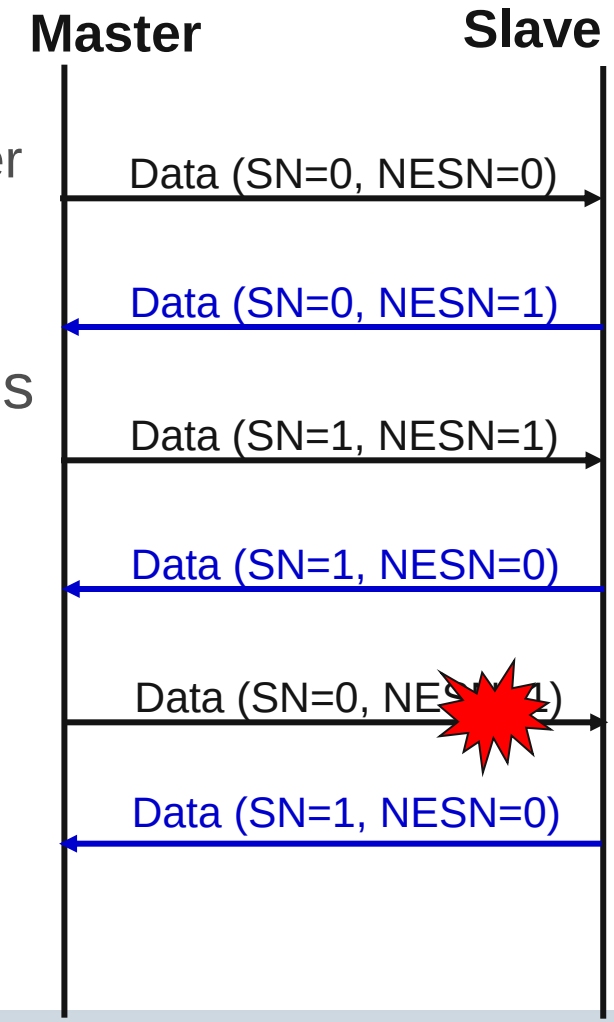


Connection Termination

- Master initiated termination - transmit TERMINATE_IND packets to slave until:
  - Acknowledgement from Slave
  - Slave latency + 6 connection events
- Slave initiated termination – transmit TERMINATE_IND packets to master until
  - Acknowledgement from Master
  - 6 connection events
- Connection supervision timeout

Acknowledgement and Flow Control

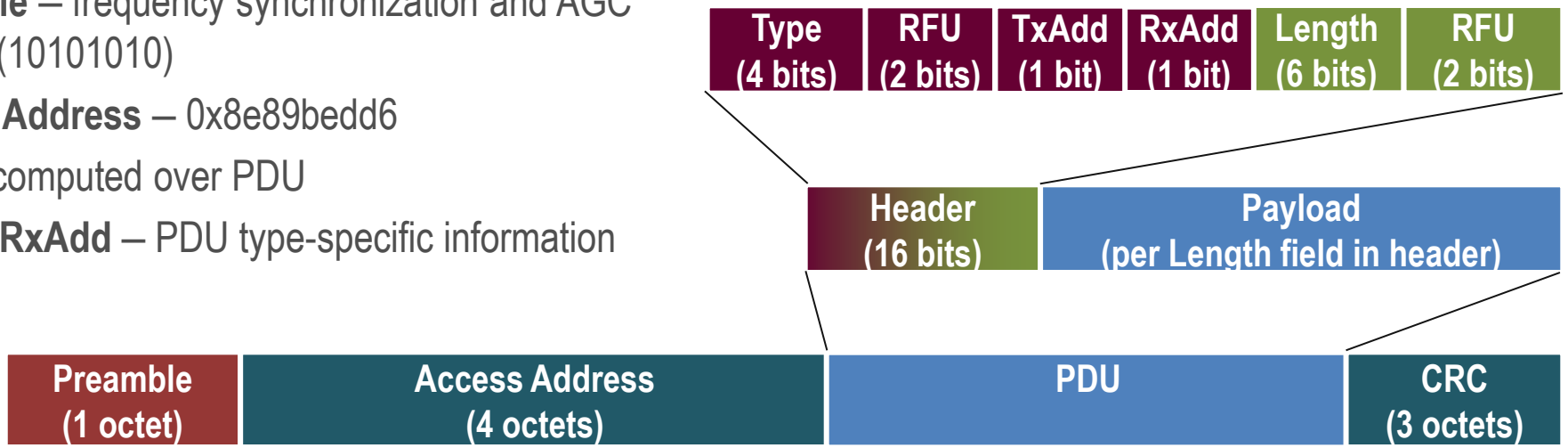
- Acknowledgements embedded in header of every Data channel PDU
  - Single bit Sequence Number (SN)
  - Single bit Next Expected Sequence Number (NESN)
- Packet is retransmitted until the NESN is different from the SN value in the sent packet
  - Enables lazy acknowledgement for significant power savings



Air Interface Packets – Advertising Packets

Type	Packet	Usage
0000	ADV_IND	Connectable undirected advertising event
0001	ADV_DIRECT_IND	Connectable directed advertising event
0010	ADV_NONCONN_IND	Non-connectable undirected advertising event
0011	SCAN_REQ	Scan request for further information from advertiser
0100	SCAN_RSP	Response to scan request from scanner
0101	CONNECT_REQ	Connect request by Initiator
0110	ADV_DISCOVER_IND	Discoverable undirected advertising event

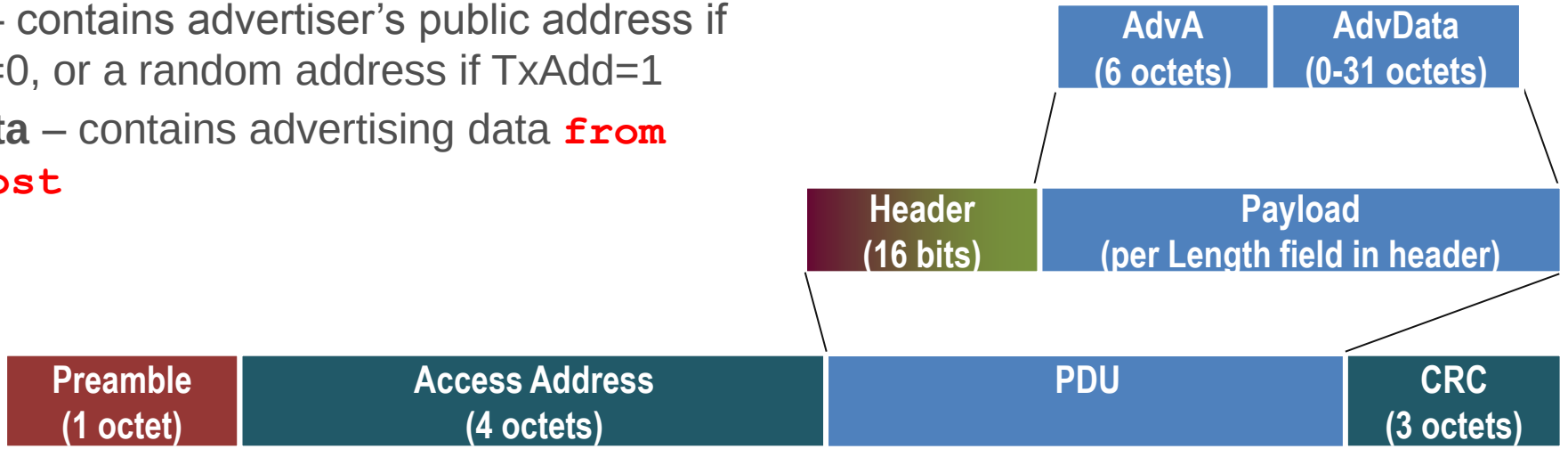
- **Preamble** – frequency synchronization and AGC training (10101010)
- **Access Address** – 0x8e89bedd6
- **CRC** – computed over PDU
- **TxAdd, RxAdd** – PDU type-specific information



Air Interface Packets – Advertising PDUs (Undirected)

Type	Packet	Usage
0000	ADV_IND	Connectable undirected advertising event
0010	ADV_NONCONN_IND	Non-connectable undirected advertising event
0110	ADV_DISCOVER_IND	Discoverable undirected advertising event

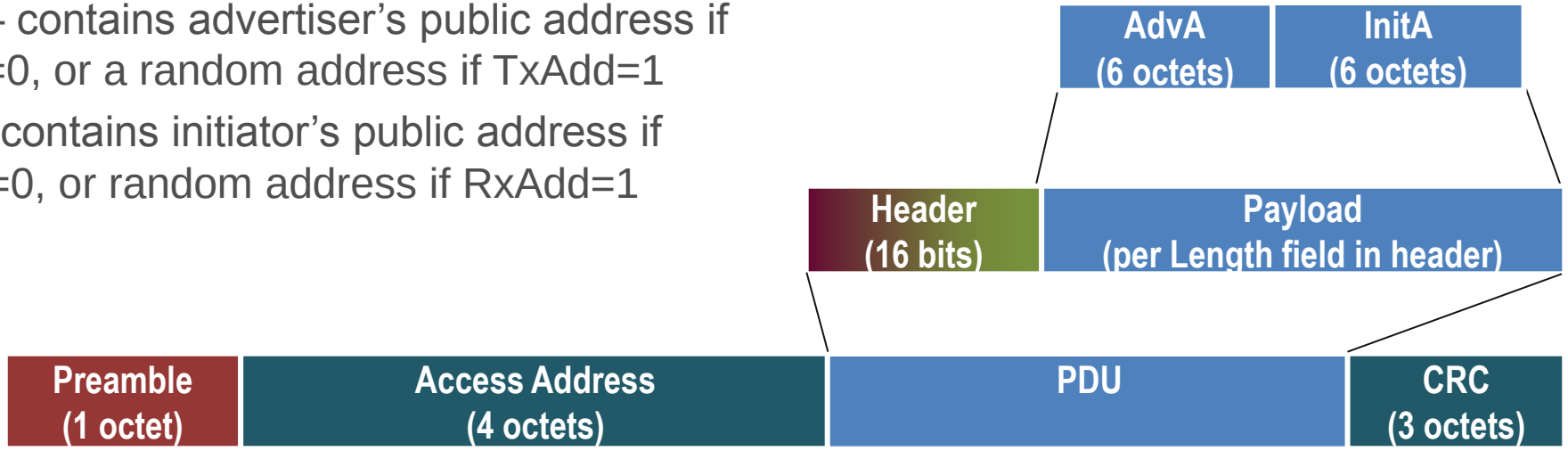
- **AdvA** – contains advertiser's public address if TxAdd=0, or a random address if TxAdd=1
- **AdvData** – contains advertising data **from the host**



Air Interface Packets – Advertising PDUs (Directed)

Type	Packet	Usage
0001	ADV_DIRECT_IND	Connectable directed advertising event

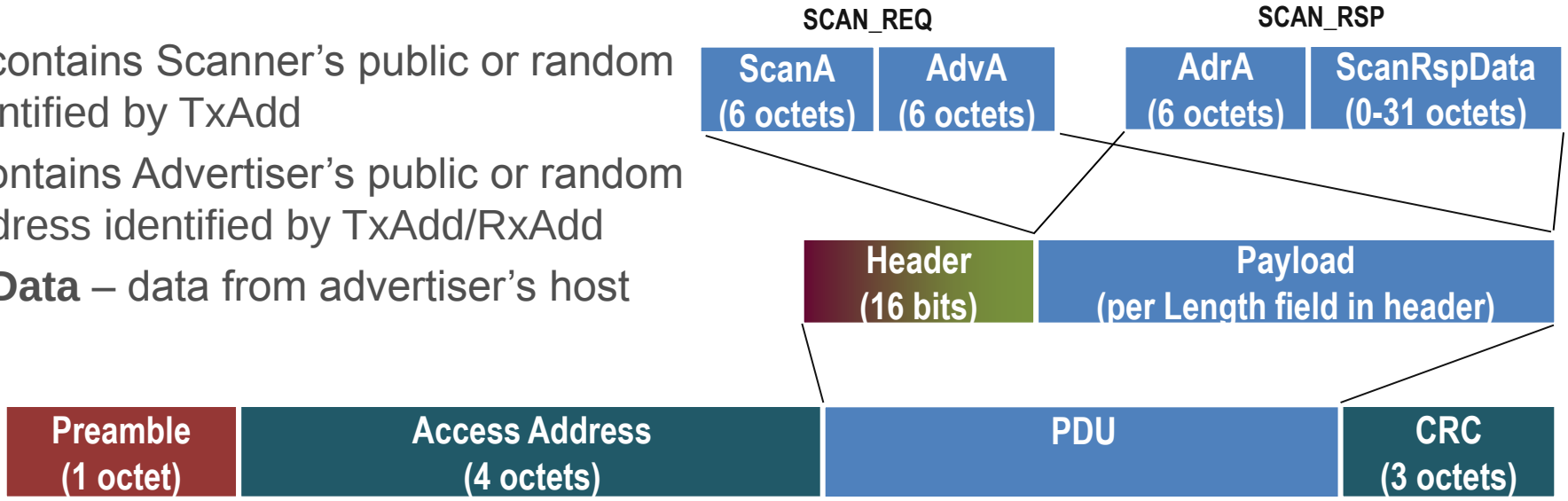
- **AdvA** – contains advertiser's public address if TxAdd=0, or a random address if TxAdd=1
- **InitA** – contains initiator's public address if RxAdd=0, or random address if RxAdd=1



Air Interface Packets – Scanning PDUs

Type	Packet	Usage
0011	SCAN_REQ	Scan request for further information from advertiser
0100	SCAN_RSP	Response to scan request from scanner

- **ScanA** – contains Scanner's public or random device identified by TxAdd
- **AdvA** – contains Advertiser's public or random device address identified by TxAdd/RxAdd
- **ScanRspData** – data from advertiser's host

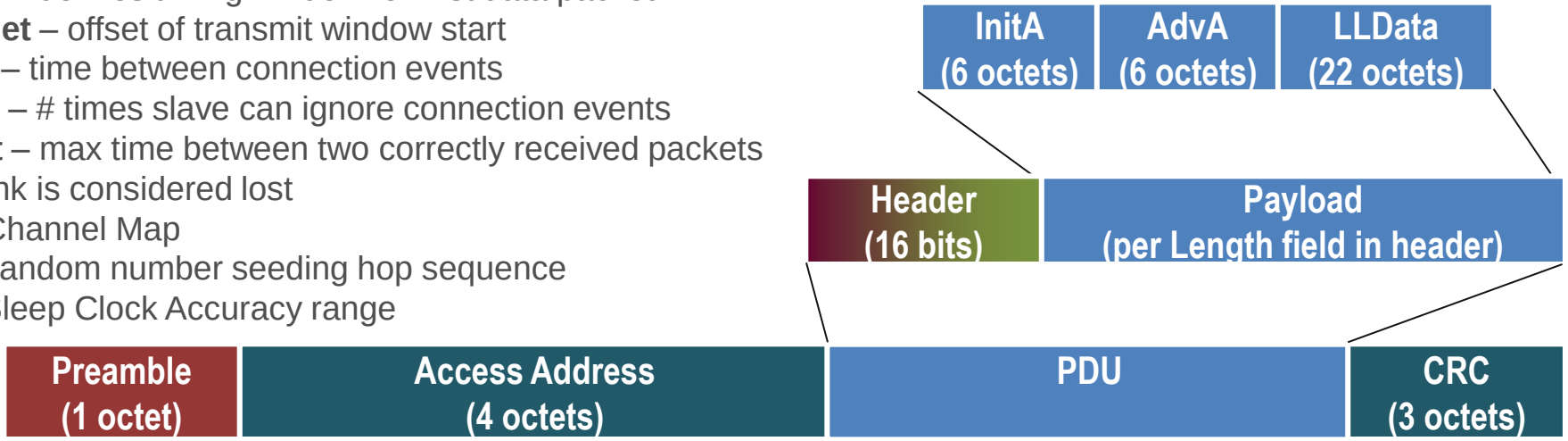


Air Interface Packets – Initiating PDUs

Type	Packet	Usage
0101	CONNECT_REQ	Connect request by Initiator

AA (4 octets)	CRCInit (3 octets)	WinSize (1 octets)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	ChM (5 octets)	Hop (5 bits)	SCA (3 bits)

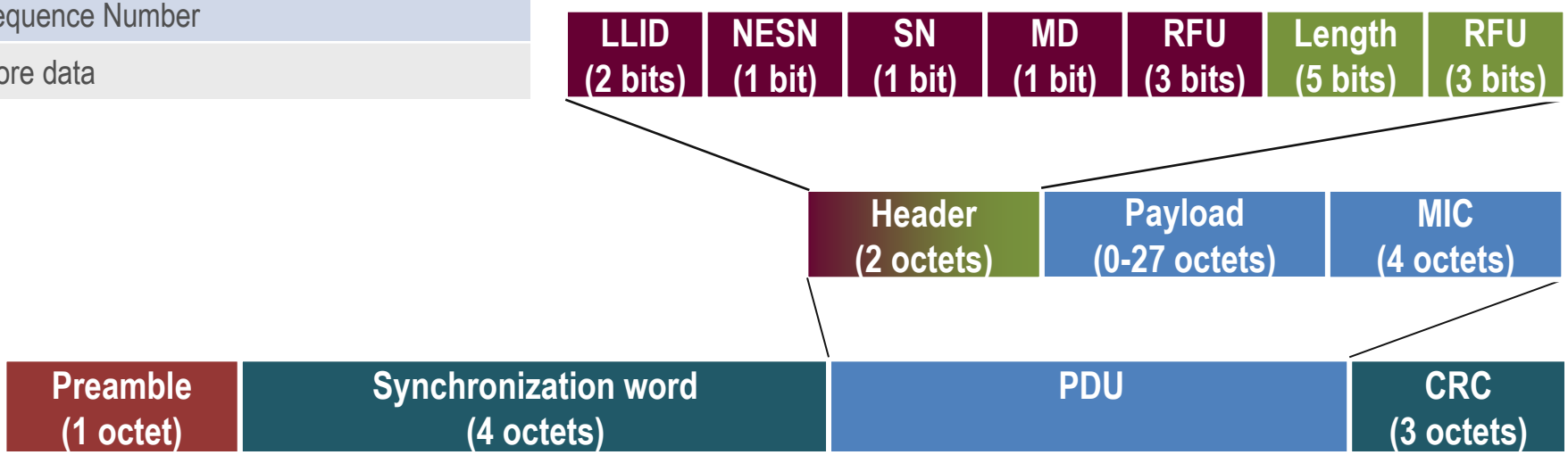
- **InitA** –initiator's public/random address based on TxAdd
- **AdvA** –advertiser's public/random address based on RxAdd
- **AA** – contains Link Layer's connection address
- **CRCInit** –initialization value for CRC calculation
- **WinSize** – defines timing window for first data packet
- **WinOffset** – offset of transmit window start
- **Interval** – time between connection events
- **Latency** – # times slave can ignore connection events
- **Timeout** – max time between two correctly received packets before link is considered lost
- **ChM** – Channel Map
- **Hop** – Random number seeding hop sequence
- **SCA** – Sleep Clock Accuracy range



Air Interface Packets – LL Data Channel

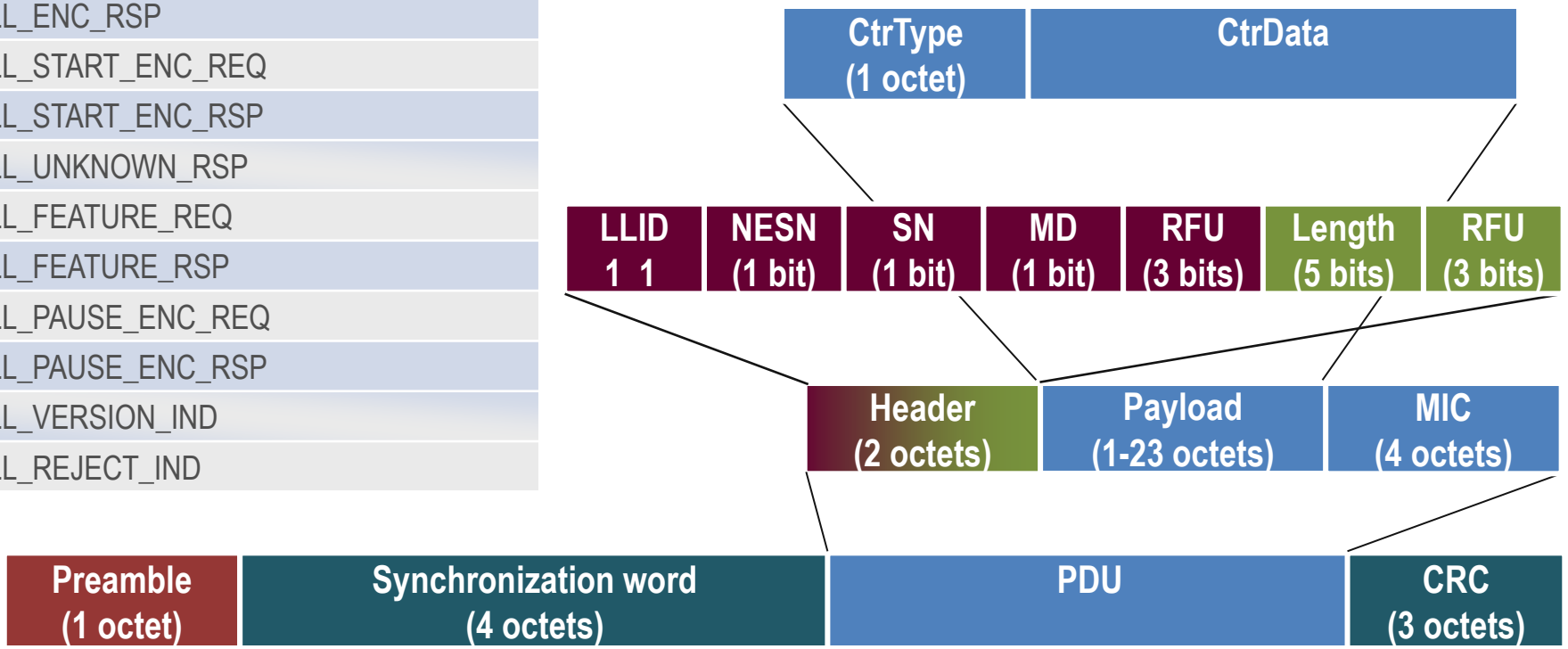
Field	Purpose and Encoding
LLID	**0x01** = Continuation/empty L2CAP packet **0x02** = Start of an L2CAP packet **0x03** = LL Control packet
NESN	Next Expected Sequence Number
SN	Sequence Number
MD	More data

- **Preamble** – frequency synchronization and AGC training (`01010101`) or (`10101010`)
- **Synchronization word** – 32 bit link layer connection access address
- **CRC** – computed over PDU
- **MIC** – Message Integrity Code, for use with encrypted links



Air Interface Packets – LL Control Packets

Opcode	Control packet name
0x00	LL_CONNECTION_UPDATE_REQ
0x01	LL_CHANNEL_MAP_REQ
0x02	LL_TERMINATE_IND
0x03	LL_ENC_REQ
0x04	LL_ENC_RSP
0x05	LL_START_ENC_REQ
0x06	LL_START_ENC_RSP
0x07	LL_UNKNOWN_RSP
0x08	LL_FEATURE_REQ
0x09	LL_FEATURE_RSP
0x0a	LL_PAUSE_ENC_REQ
0x0b	LL_PAUSE_ENC_RSP
0x0c	LL_VERSION_IND
0x0d	LL_REJECT_IND



Encryption

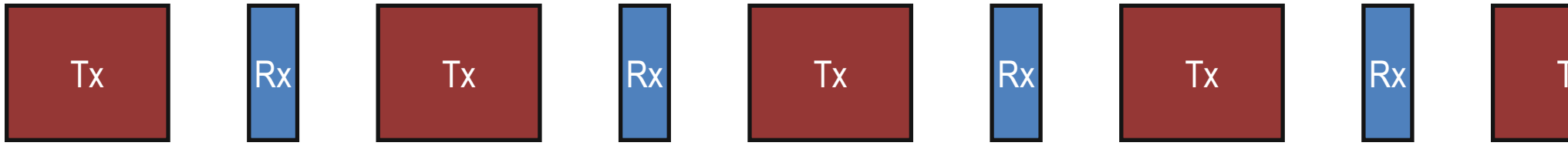
- Uses CCM algorithm
  - Counter with CBC-MAC (Cipher Block Chaining-Message Authentication Code)
  - Defined in IETF RFC 3610
- 32-bit Message Integrity Code on every encrypted packet
- Based on “Encryption Root” key in slave
  - Generates a Long Term Key (LTK)
  - LTK shared with master to allow faster reconnections

Packet Timings

- Peer device transmits 150 µs after last packet
- Minimum size packet = 80 µs

  (Preamble + Access Address + Header + CRC)
- Maximum size packet = 328 µs

  (Preamble + Access Address + Header + Payload + MIC + CRC)



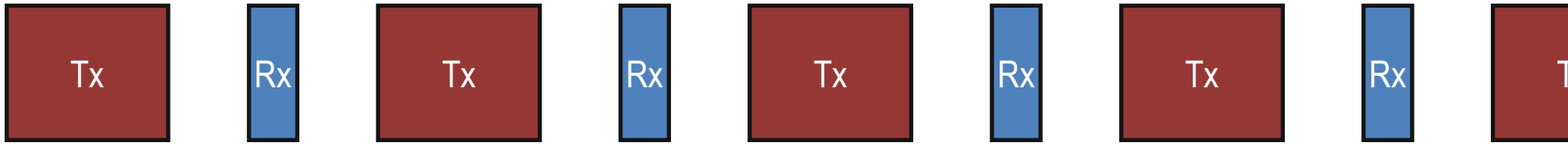
Maximum Data Rate

- Asymmetric Tx/Rx Packet Sequence

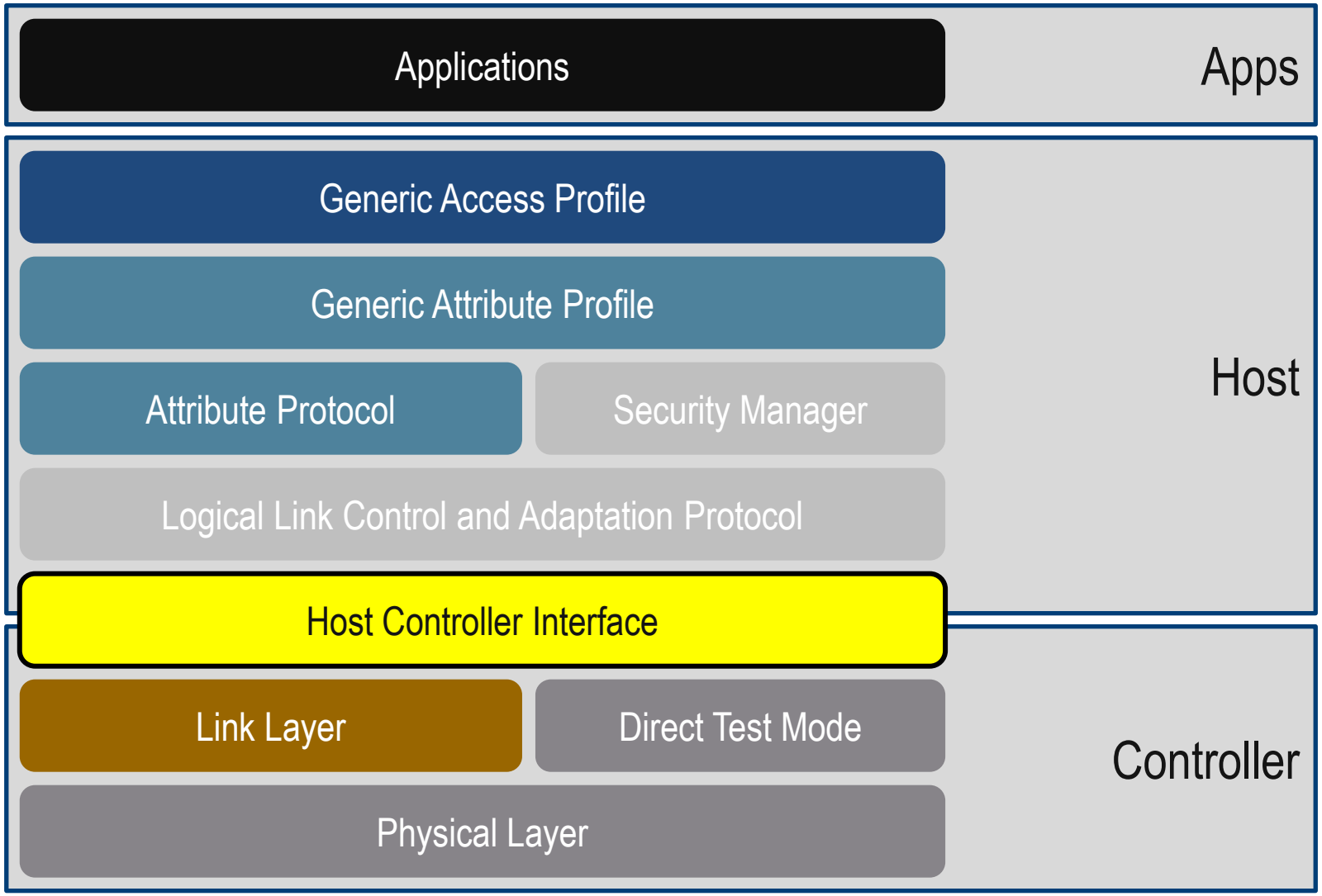
  328 + 150 + 80 + 150 = 708 µs

  Transmitting 27 octets of application data

  ~305 kbps



Host Controller Interface



Host Controller Interface (HCI)

- Defines physical connection between a host (e.g. PC) and a host controller (e.g. Bluetooth module).
- Extension of HCI specified in the Bluetooth Core Specification
- The specification defines several interfaces:
  - UART
  - USB
  - SD
  - 3-wire UART
- It also defines messages that are passed across the HCI interface.

HCI Commands and Events

- Device Setup
  - Reset
- Controller Commands & Events
  - Flow Control
  - Obtain local Information
  - Configuration
- Device Discovery
  - Advertising Reports
  - Scan Setup
- Connection Setup and State
  - Setup and tear-down of connections
- Remote Information
  - Obtain remote device information
- Physical Links
  - Set host channel classification
- Host Flow Control
  - Set host control
  - Configure white lists
- Link Information
  - RSSI
  - Channel maps
- Testing
  - Places device into special test mode
- Generic events
  - Can occur at any time

HCI Commands and Events

Generic Events
Command Complete
Command Status
Hardware Error

Device Setup
Reset

Controller Flow Control
Read Buffer Size
Number of Completed Packets
LE Read Buffer Size

Controller Information
Read Local Version Information
Read Local Supported Commands
Read Local Supported Features
Read BDADDR
LE Read Local Supported Features
LE Read Supported States

Controller Configuration
LE Set Advertise Enable
LE Set Advertising Data
LE Set Advertising Parameters
LE Set Random Address
LE Set Scan Response Data
LE Set Random Address

Black – existing commands
Black italicized – existing events
Red – new commands
Red italicized – new events

Device Discovery
LE Advertising Report
LE Set Scan Enable
LE Set Scan Parameters

Connection Setup
Disconnect Command
Disconnection Complete
LE Connection Complete
LE Create Connection Cancel
LE Create Connection

Remote Information
Read Remote Version Information
Read Remote Version Information Complete
LE Read Remote Used Features
LE Read Remote Used Features Complete

Connection State
LE Connection Update
LE Connection Update Complete

Physical Links
LE Set Host Channel Classification

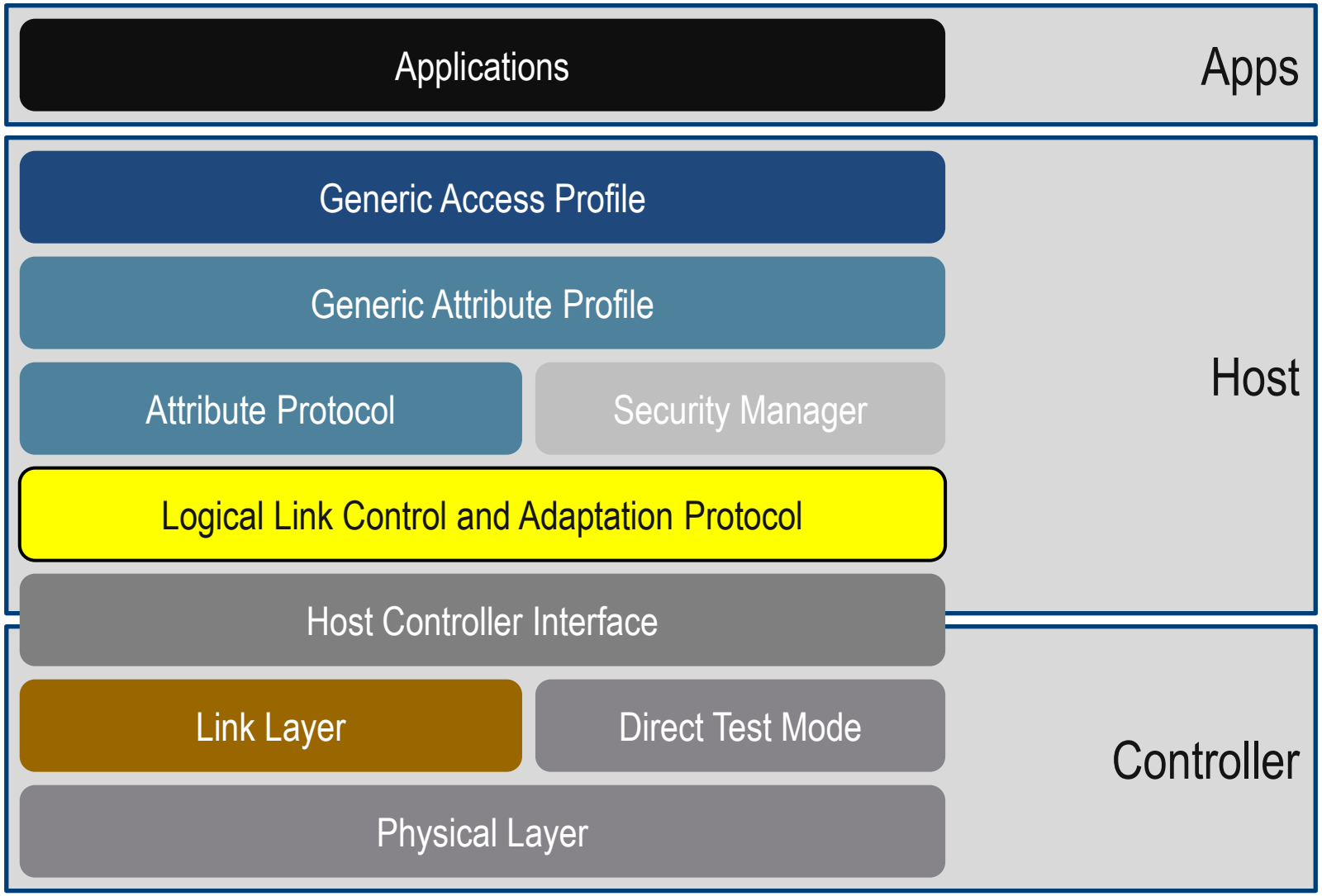
Test
LE Receiver Test
LE Transmitter Test
LE Test End

Host Flow Control
Host Buffer Size
Set Event Mask
Set Controller To Host Flow Control
Host Number of Completed Packets
Data Buffer Overflow
LE Add Device to White List
LE Clear White List
LE Read White List Size
LE Remove Device from White List
LE Set Event Mask

Link Information
Read Transmit Power Level
Read RSSI
LE Read Advertising Channel TX Power
LE Read Channel Map

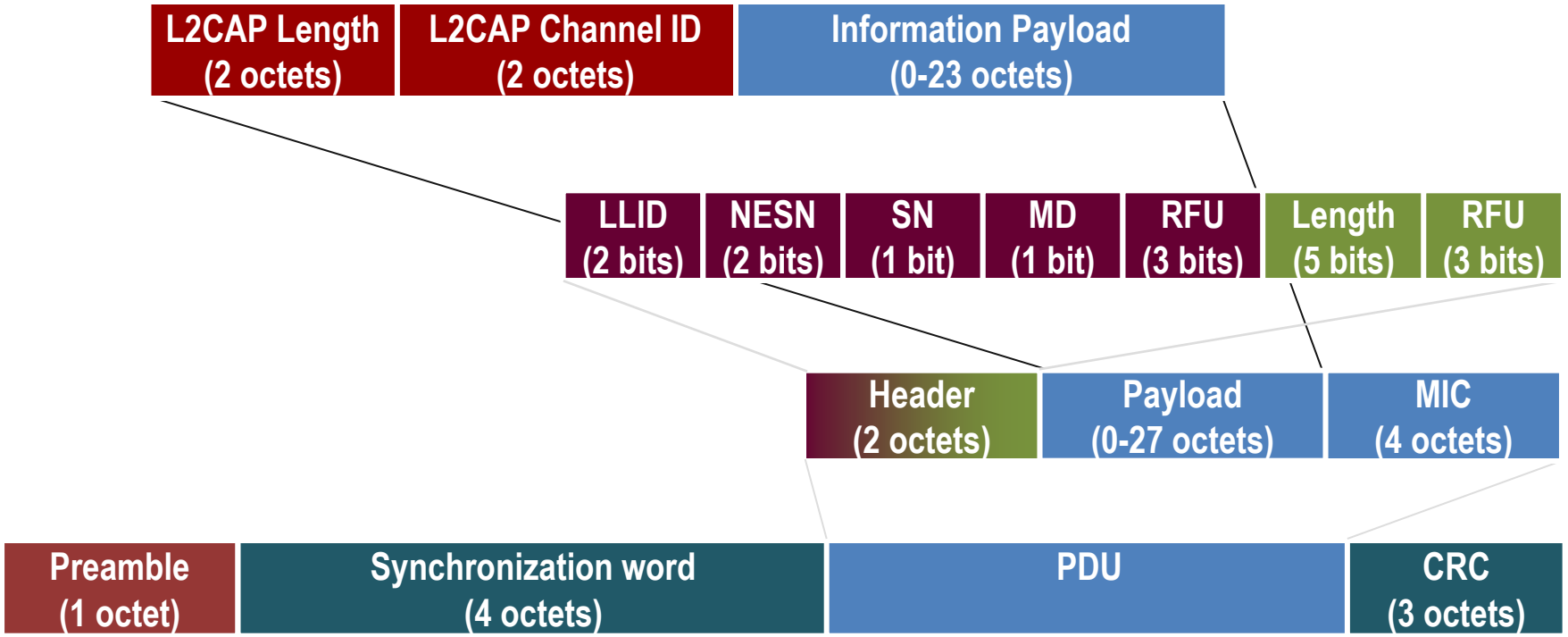
Authentication and Encryption
Encryption Change
Encryption Key Refresh Complete
LE Encrypt
LE Long Term Key Requested
LE Long Term Key Requested Reply
LE Long Term Key Requested Negative Reply
LE Rand
LE Start Encryption

Logical Link Control and Adaptation Protocol



Logical Link Control and Adaptation Protocol (L2CAP)

- Provides fixed channel data services to upper layer protocols
- Provides protocol multiplexing capability through concept of channels
  - Channel Identifier is local name representing a logical channel
  - Channels are bi-directional



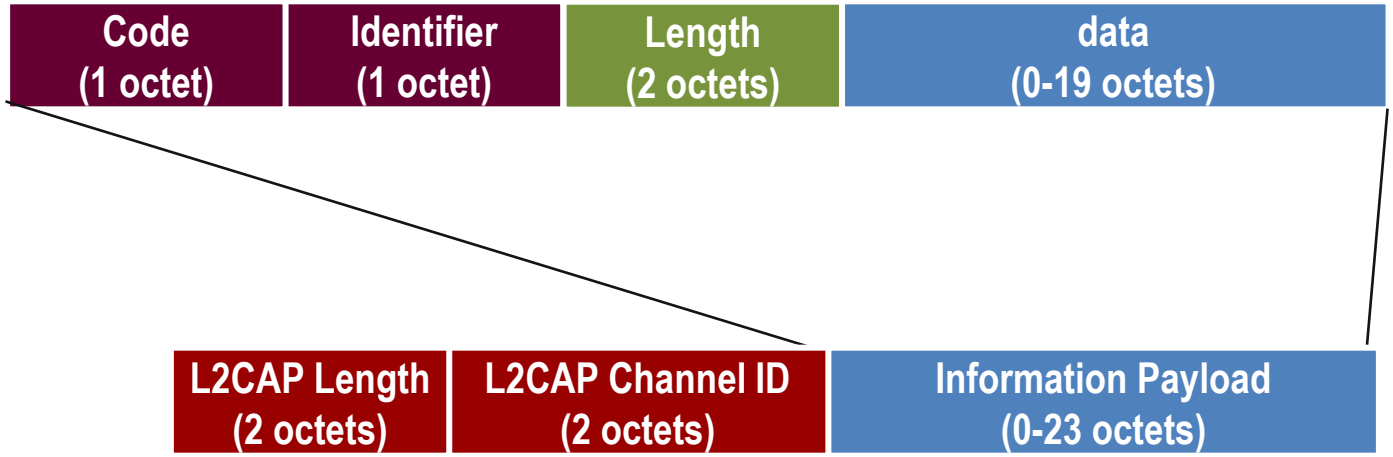
L2CAP Channel Types

- L2CAP in Bluetooth low energy operates in Basic Mode
  - Offers only fixed channel types
  - Connection oriented channels are not used in BTle

Channel Type	Local CID (sending)	Remote CID (receiving)
Attribute Protocol	0x0004 (fixed)	0x0004 (fixed)
Signaling	0x0005 (fixed)	0x0005 (fixed)
Security Manager Protocol	0x0006 (fixed)	0x0006 (fixed)

- Response timeouts used to terminate channel when remote endpoint is unresponsive

Code	Description	Usage
0x00	Reserved	Reserved
0x01	Command Reject	Sent in response when unknown command code or inappropriate response
0x12	Connection Parameter Update Request	Allows slave device to request new connection parameter targets
0x13	Connection Parameter Update Response	Master response to slave Connection Parameter Update Request

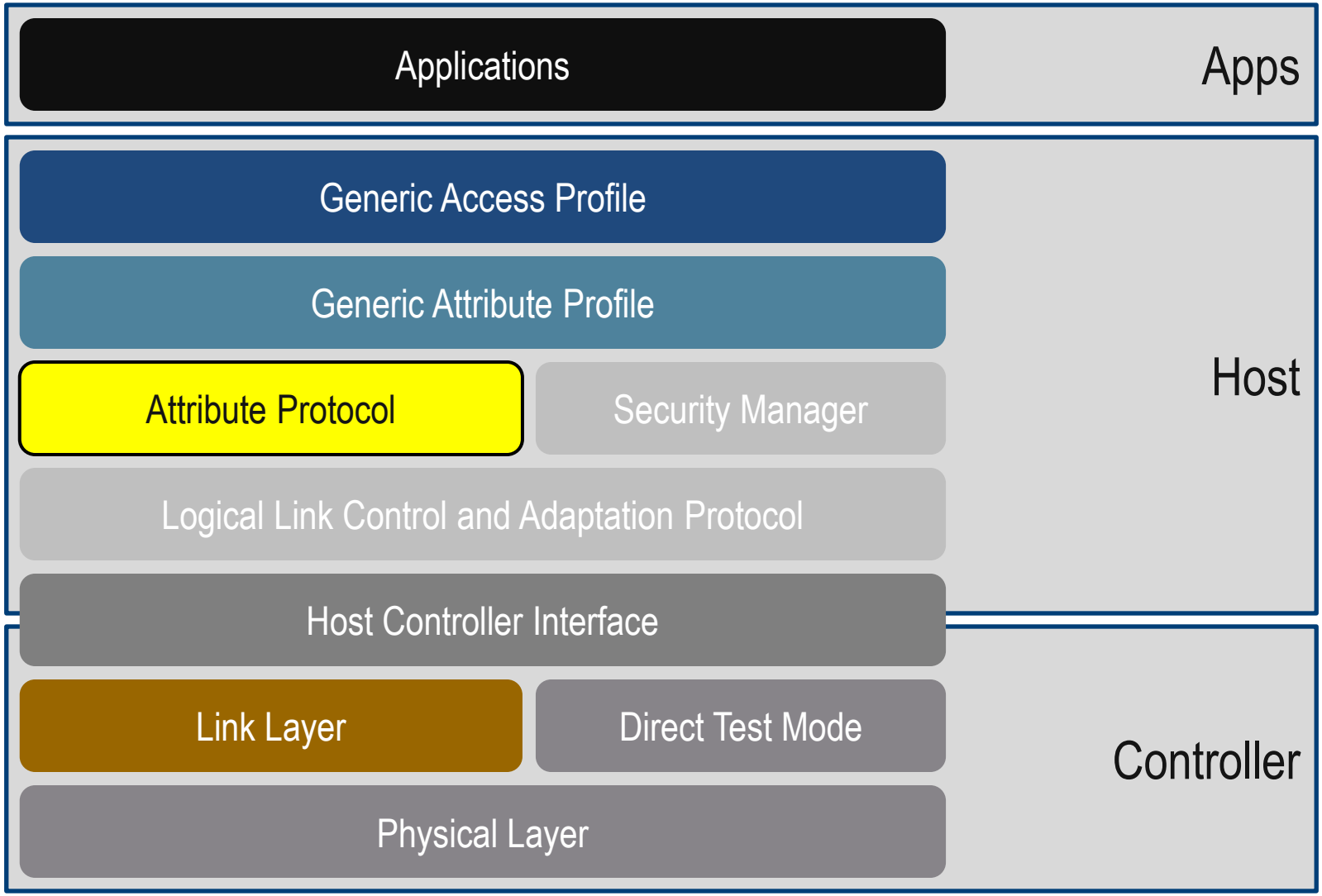


L2CAP Signaling – Connection Parameter Update

- Slave can request update to connection parameters
  - minimum and maximum connection interval (connInterval)
  - slave latency, i.e. number of times it can ignore connection events from master
  - Connection timeout

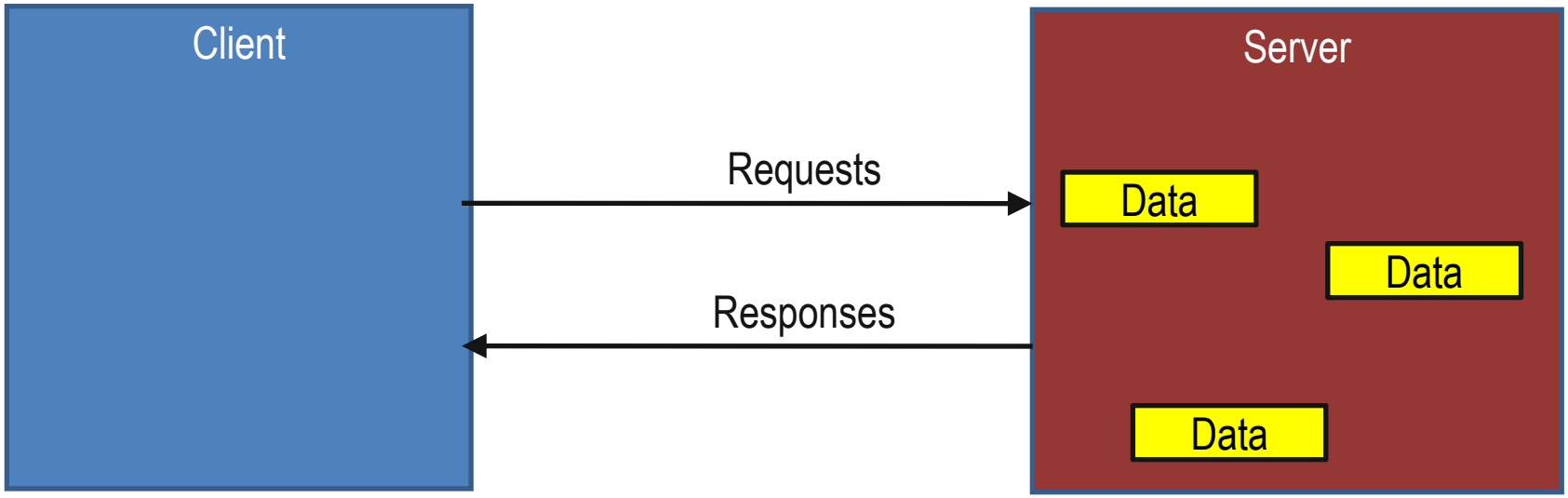
- Master can choose to accept or reject the parameters

Attribute Protocol



- Client Server Architecture
  - servers have data
  - clients request data to/from servers

- Servers expose data using Attributes

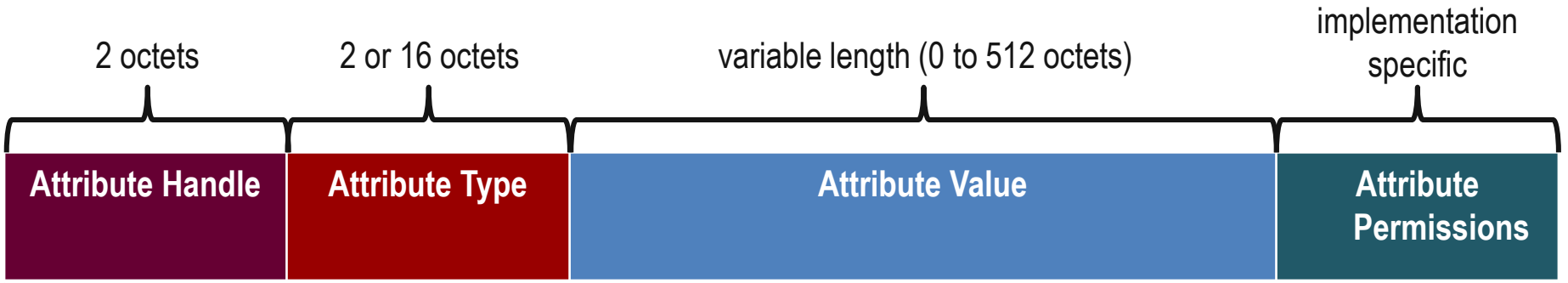


The Attributes of Attributes

- Attributes have ***values***
  - Array of up to 512 octets, fixed or variable length
- Attributes have ***handles***
  - Used to address an individual attribute by a client
- Attributes have a ***type***
  - <<UUID>>, determines what the value means
  - Defined by GAP, GATT, “Characteristic Specifications”
- Attributes have ***permissions***
  - Read, Write
  - May require authentication or authorization to read or write

Handle	Type	Value	Meaning
0x0009	«Device Name»	0x54656d706572617475726520536e736f72	“Temperature Sensor”
0x0022	«Battery State»	0x04	Discharging
0x0098	«Temperature»	0x0802	20.5 ºC

Logical Attribute Presentation

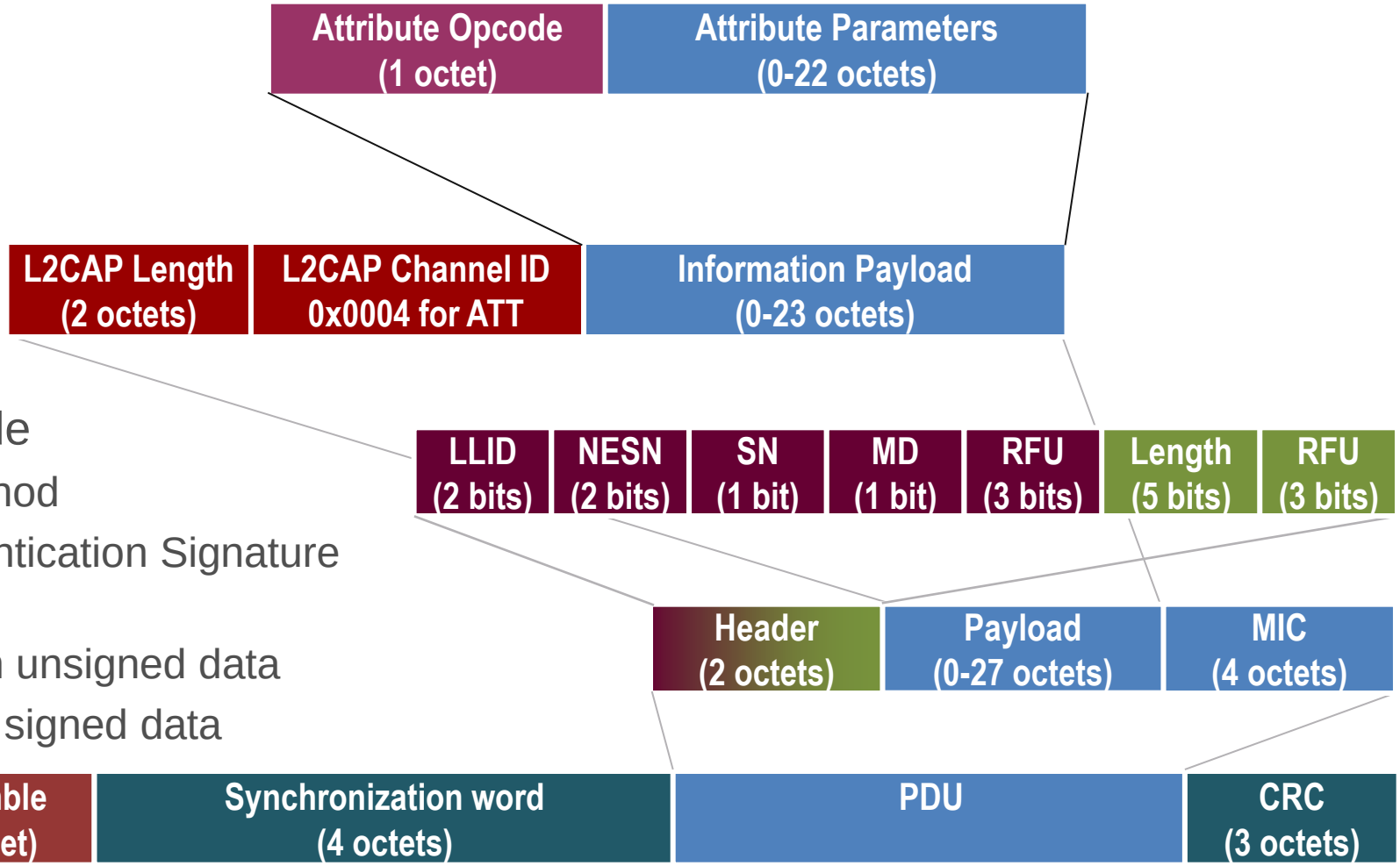


Protocol Methods

Protocol PDU Type	Sent by	Description
Request	Client	Client requests something from server – always causes a response
Response	Server	Server sends response to a request from a client
Command	Client	Client commands something to server – no response
Notification	Server	Server notifies client of new value – no confirmation
Indication	Server	Server indicates to client new value – always causes a confirmation
Confirmation	Client	Confirmation to an indication

- Client can only send one request at a time – request completes after response received
- Server can send only one indication at a time – indication completes after confirmation
- Commands and Notifications can be sent at any time

Attribute PDU format

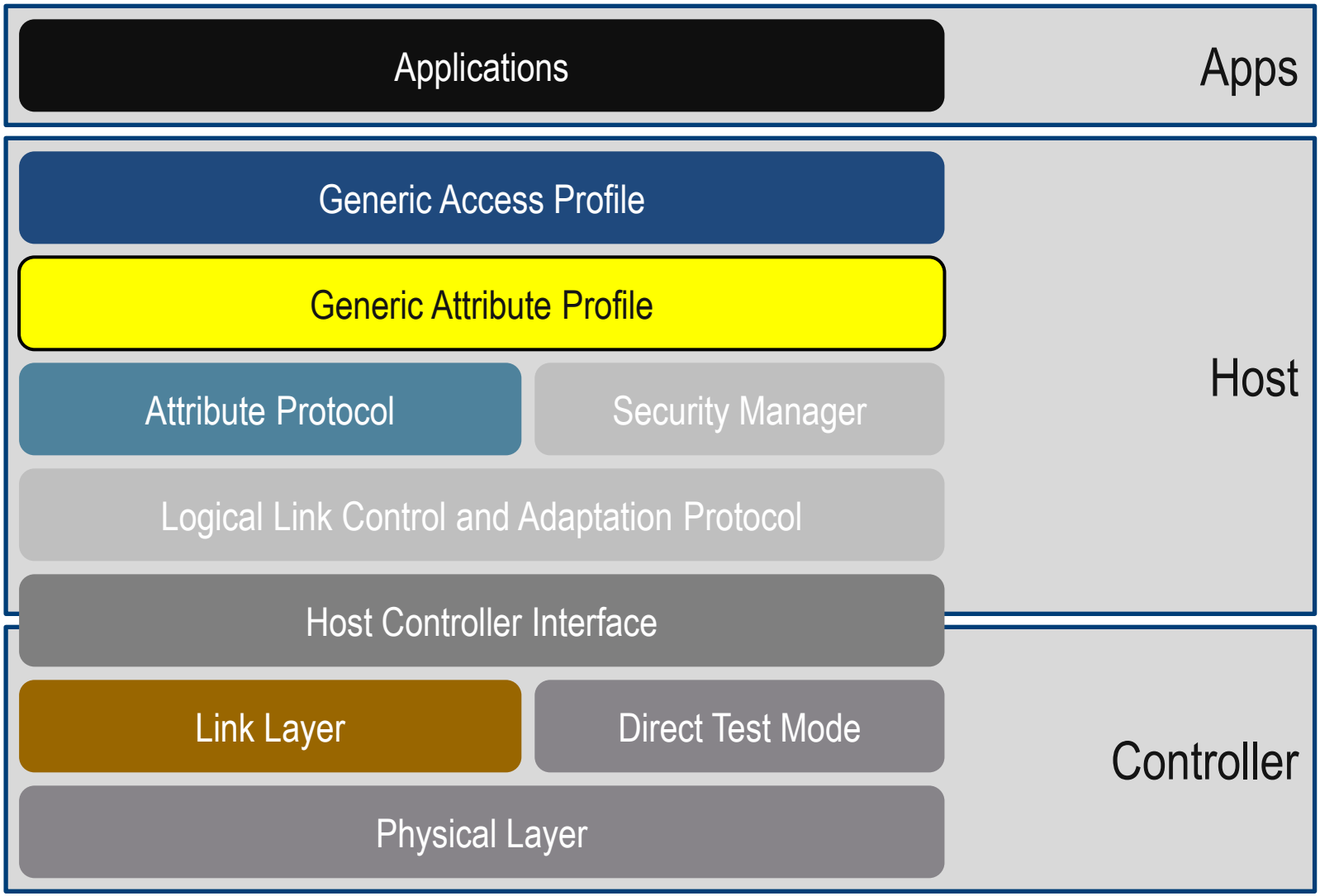


- Attribute Opcode
  - bit 6-0 : Method
  - bit 7 : Authentication Signature Flag
    - If 0, then unsigned data
    - If 1 then signed data

Attribute Commands

Name	Description
Error Response	Something was wrong with a request
Exchange MTU Request / Response	Exchange new ATT_MTU
Find Information Request / Response	Find information about attributes
Find By Type Value Request / Response	Find specific attributes
Read By Group Type Request / Response	Find specific group attributes and ranges
Read By Type Request / Response	Read attribute values of a given type
Read Read / Response	Read an attribute value
Read Blob Request / Response	Read part of a long attribute value
Read Multiple Request / Response	Read multiple attribute values
Write Command	Write this – no response
Write Request / Response	Write an attribute value
Prepare Write Request / Response	Prepare to write a value (long)
Execute Write Request / Response	Execute these prepared values
Handle Value Notification	Notify attribute value – no confirmation
Handle Value Indication / Confirmation	This attribute now has this value

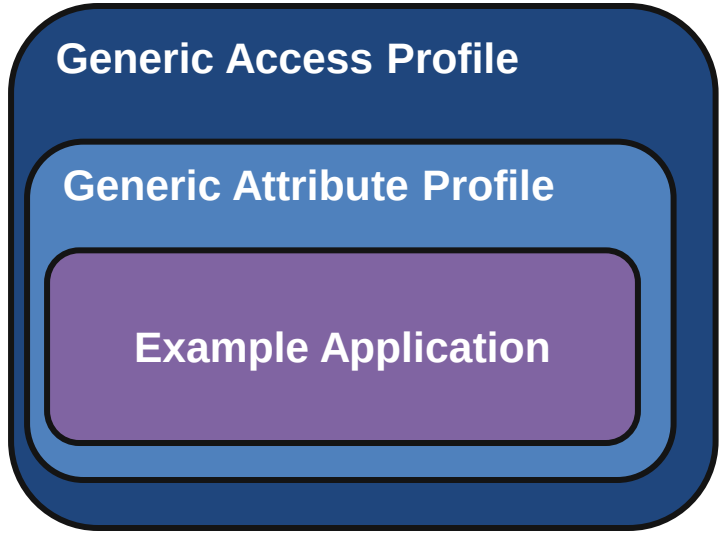
Generic Attribute Profile

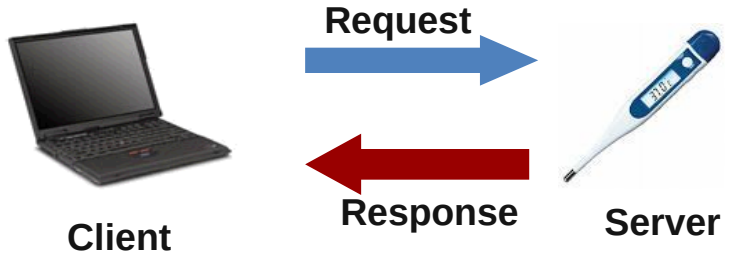


Generic Attribute Profile (GATT)

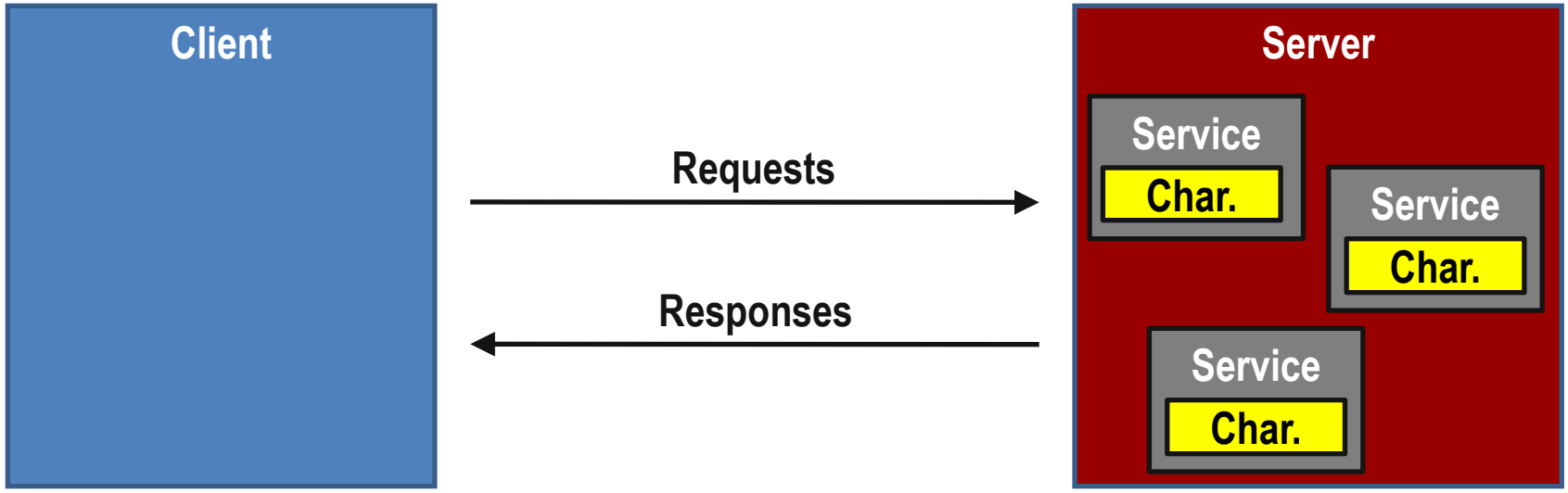
- Defines framework for using Generic Attribute Protocol

- Configurations and Roles
  - Client
    - Initiates commands and requests toward server
    - Receives responses, indications, and notifications from server
  - Server
    - Accepts commands and requests from client
    - Sends responses, indications, and notifications to client





- Same client server architecture as Attribute Protocol
  - except that data is encapsulated in “Services”
  - data is exposed in “Characteristic”



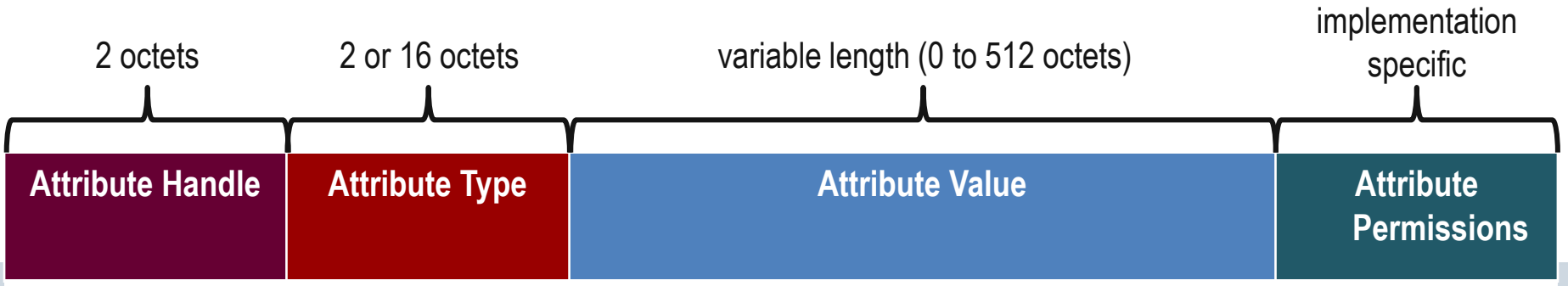
GATT Definitions

- Service – set of related characteristics and how these are used
  - Primary Services – exposes primary usable functionality of device
    - Can be included by another service
  - Secondary Services – intended to be referenced by primary services

- Characteristics – related attributes that describe state of device
  - Features available (readable, indicatable, etc.)
  - Handle
  - Representation (units, exponent, data type) – i.e. data dictionary

GATT uses Attribute Protocol

- Attribute Protocol defines a server with a set of attributes
  - addressable with a handle
  - typed using a UUID
  - Includes data in an attribute value
  - Includes permissions



GATT Attribute Grouping

Handle	Type	Value	Permissions
0x0001	«Primary Service»	«GAP»	R
0x0002	«Characteristic»	{r, 0x0003, «Device Name»}	R
0x0003	«Device Name»	“Temperature Sensor”	R
0x0004	«Characteristic»	{r, 0x0006, «Appearance»}	R
0x0006	«Appearance»	«Thermometer»	R
0x000F	«Primary Service»	«GATT»	R
0x0010	«Characteristic»	{r, 0x0012, «Attribute Opcodes Supported»}	R
0x0012	«Attribute Opcodes Supported»	0x00003FDF	R
0x0020	«Primary Service»	«Temperature»	R
0x0021	«Characteristic»	{r, 0x0022, «Temperature Celsius»}	R
0x0022	«Temperature Celsius»	0x0802	R*

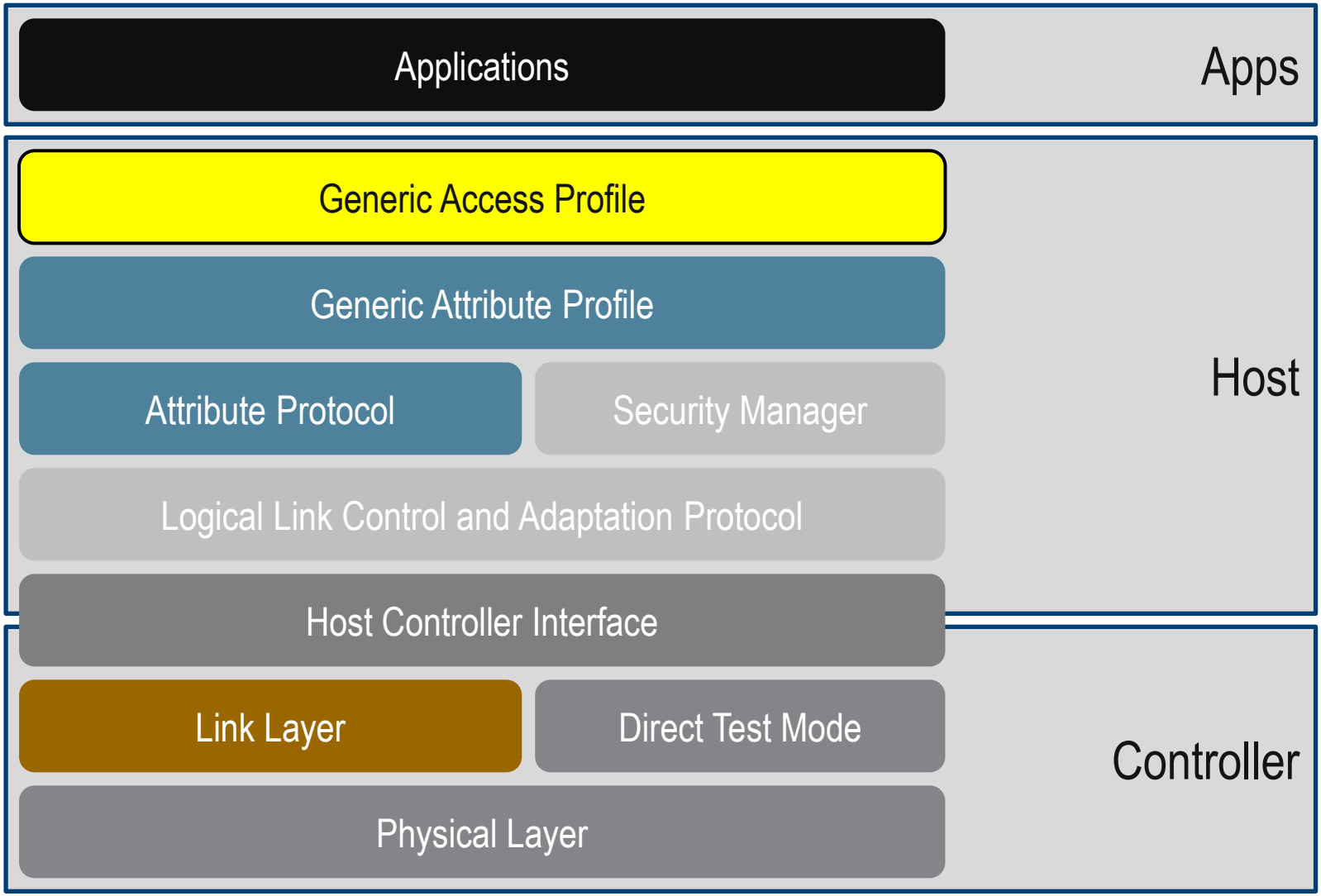
<<Characteristics>> Properties

- Broadcast
- Read
- Write without response
- Write
- Notify
- Indicate
- Authenticated Signed Writes
- Extended Properties

GATT Features and Procedures

Procedure	Sub-Procedures
Server Configuration	Exchange MTU
Primary Service Discovery	Discovery All Primary Service Discover Primary Service by Service UUID
Relationship Discovery	Find Included Services
Characteristic Discovery	Discover All Characteristics of a Service Discover Characteristics by UUID
Characteristic Descriptor Discovery	Discover All Characteristic Descriptors
Characteristic Value Read	Read Characteristic Value Read Using Characteristic UUID Read Long Characteristic Values Read Multiple Characteristic Values
Characteristic Value Write	Write Without Response Write Without Response With Authentication Write Characteristic Value Write Long Characteristic Values Reliable Writes
Characteristic Value Notifications	Notifications
Characteristic Value Indications	Indications
Characteristic Descriptors	Read Characteristic Descriptors Read Long Characteristic Descriptors Write Characteristic Descriptors Write Long Characteristic Descriptors

Generic Access Profile

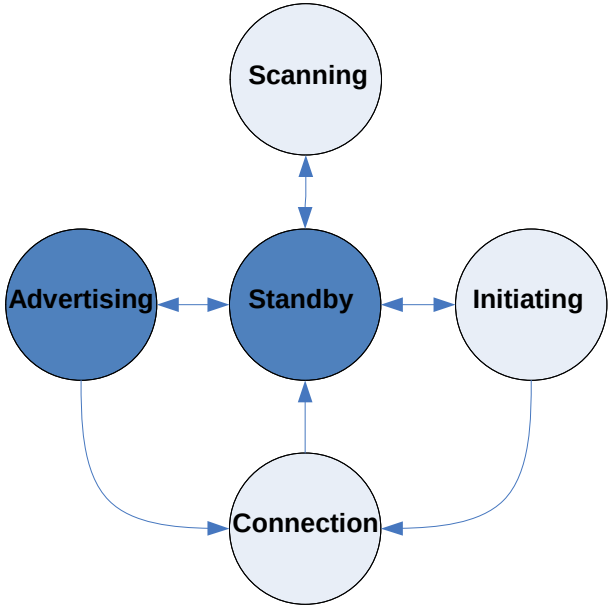


Generic Access Profile - GAP

- Defines profile roles
  - Broadcaster
  - Observer
  - Peripheral
  - Central
- Defines procedures for:
  - Discovering identities, names, and basic capabilities
  - Creating bonds
  - Exchange of security information
  - Establishing connections
  - Resolvable Private addresses
- Defines Advertising and Scan Response Data formats
- All profiles are built upon GAP

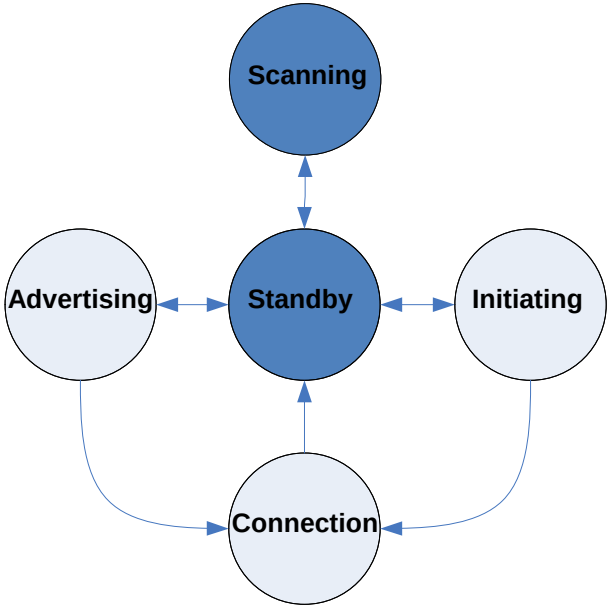
Broadcaster

Sends advertising events
Can include characteristics and service data
Doesn't need receiver
Can be discoverable if it does have receiver



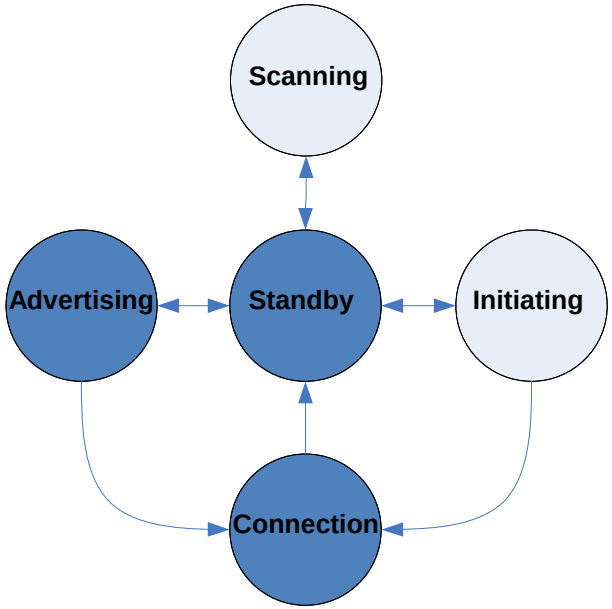
Observer

Receives advertising events
Listens for characteristics and service data
Doesn't need transmitter
Can discover devices if it does have transmitter



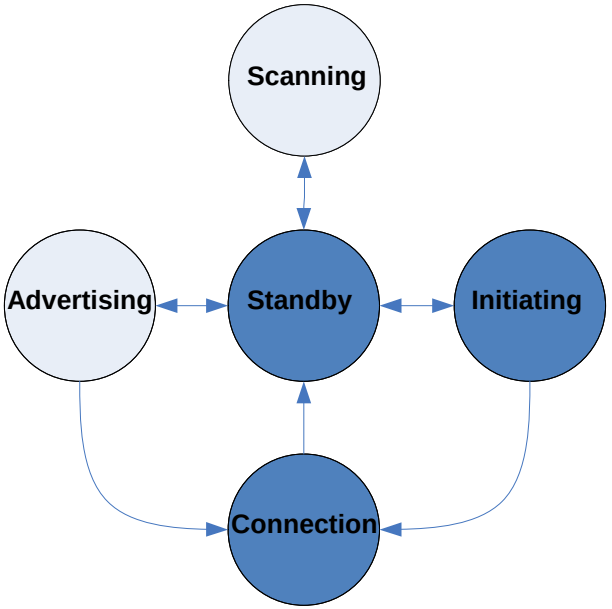
Peripheral

Has transmitter and receiver
Always slave
Connectable advertising



Central

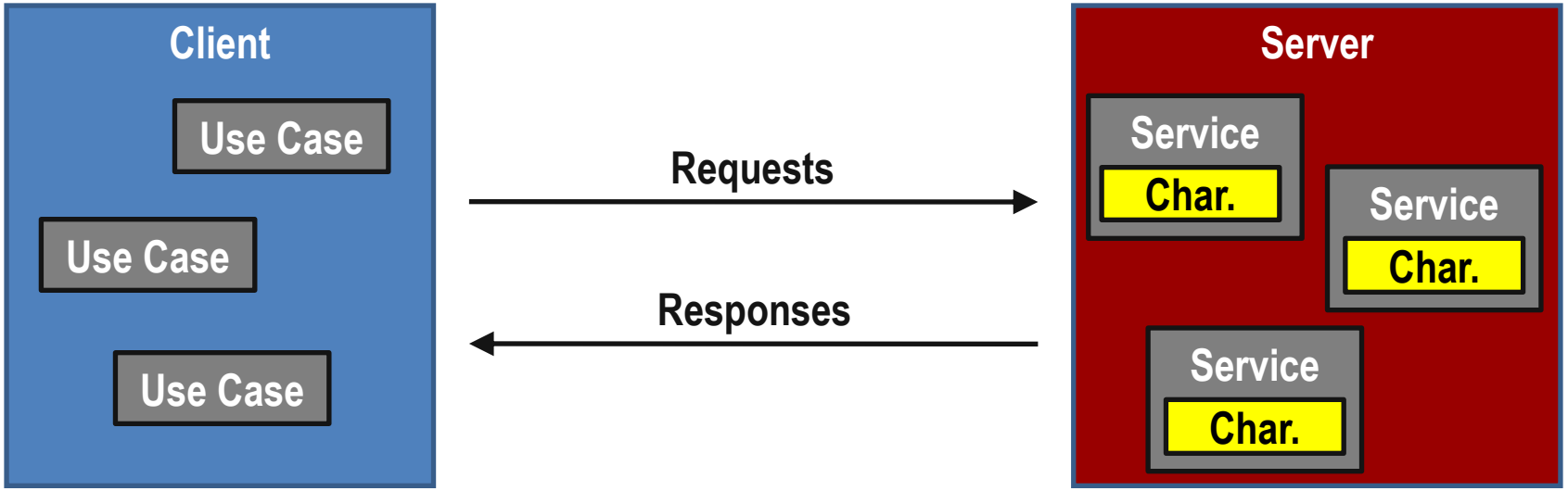
Has transmitter and receiver
Always master
Never advertises



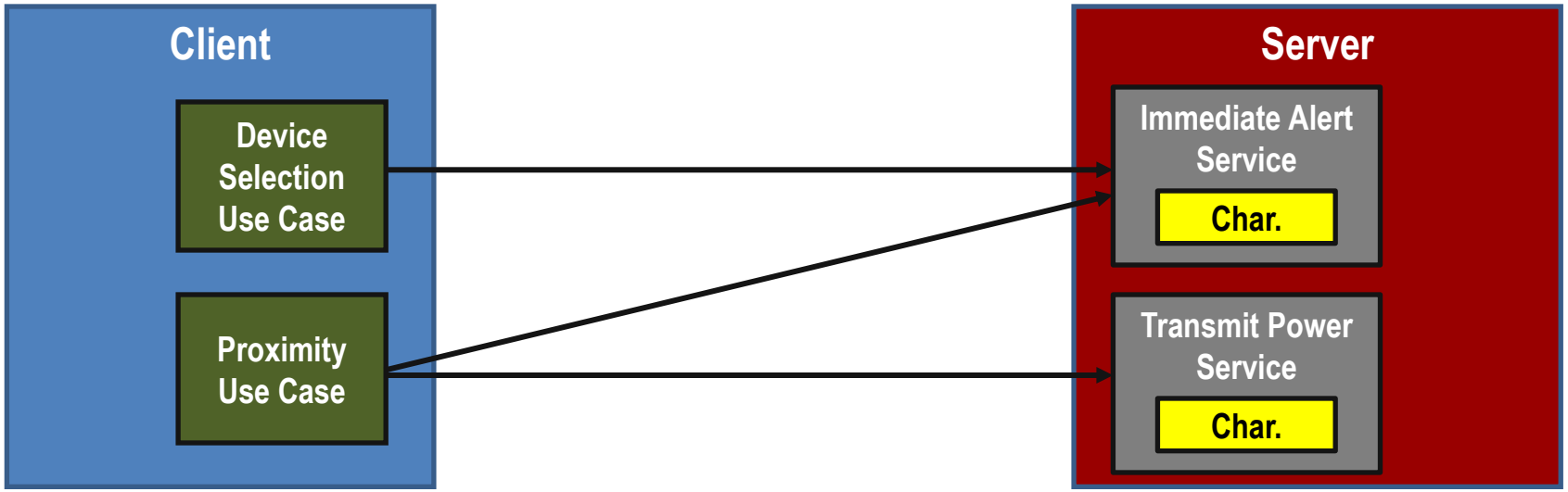
Advertising Data

Tag Value	Description
0x01	Flags (LE limited/general Discoverable mode, BR/EDR support)
0x02	Non-complete list of 16-bit Service UUIDs
0x03	Complete list of 16-bit Service UUIDs
0x06	Non-complete list of 128-bit Service UUIDs
0x07	Complete list of 128-bit Service UUIDs
0x08	Non-complete shortened local name
0x09	Complete local name
0x0A	Tx Power Level (-127 dBm to +127 dBm)
0x12	Slave Connection Interval Range (min, max)
0x14	Service Solicitation for 16 bit Service UUIDs
0x15	Service Solicitation for 128 bit Service UUIDs
0x16	Service Data (16 bit Service UUID, service data)
0xFF	Manufacturer Specific Data (Company Identifier Code, data)

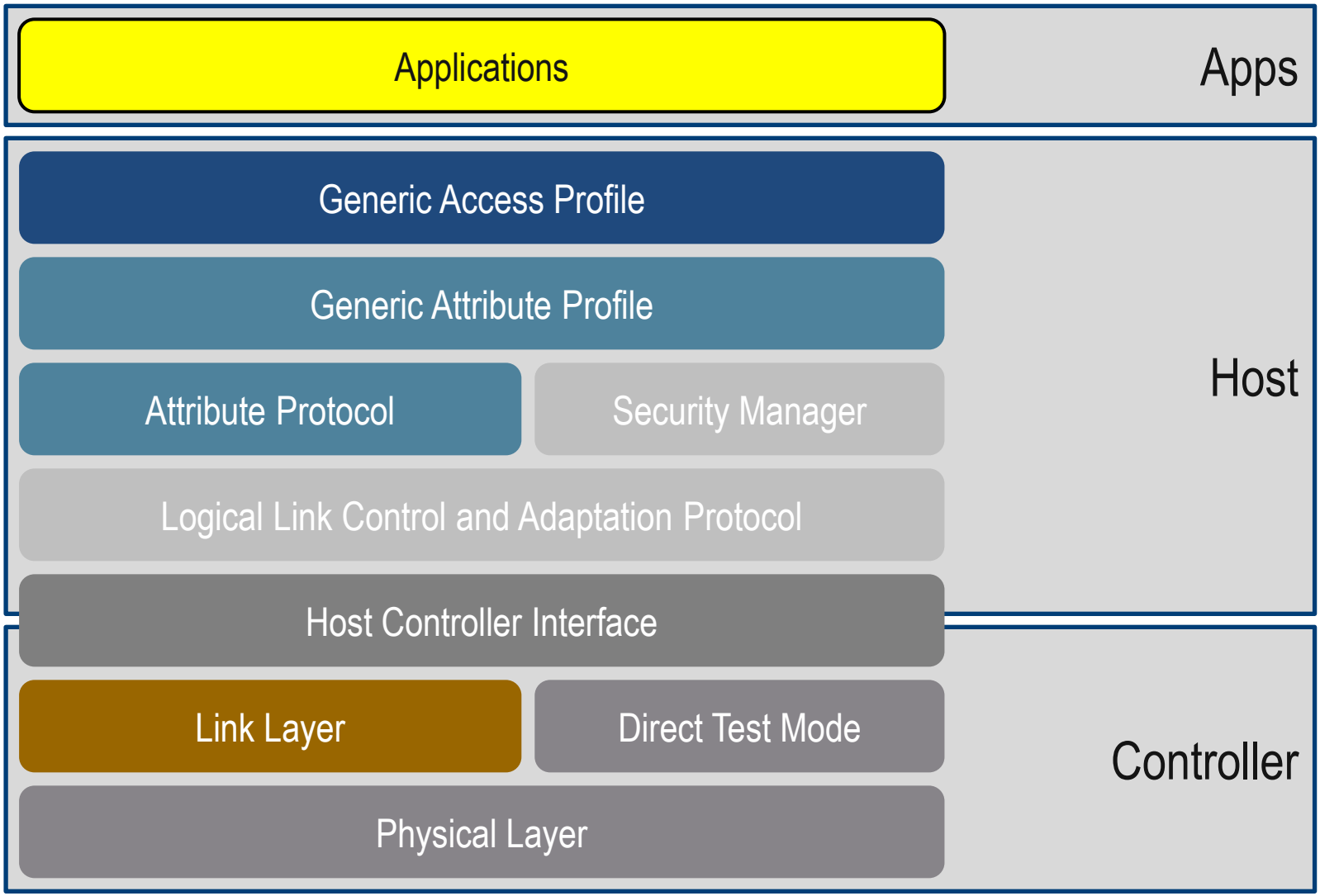
- Client Server Architecture
  - Services – exposes behavior that have characteristics
  - Use Cases– define how to use services on a peer



- There is not a one-to-one link between services and use cases
- Clients implement use cases, Servers implement services
- Use cases can use multiple services



Applications



- An Application uses a set of Use Cases
  - Use Cases use a set of Services on a peer device
    - Services expose Characteristics
    - Services define behavior exposed by Characteristics

- Bluetooth SIG creates Use Case, Requirements, and Design Documents (UCRDD)
  - Specifies User Scenarios
  - Defines Profiles
    - Includes roles for the Client and Server
    - Services required
  - Defines Services
    - Includes Characteristics and associated data formats

Example: Proximity UCRDD

- User Scenarios
  - Leaving a phone behind
  - Leaving keys behind
  - Child straying too far
  - Hospital patient from bed
  - Automatic PC Locking & Unlocking
  - Automatic PC Locking & Authenticated Unlocking
- Roles
  - Proximity Monitor
  - Proximity Reporter
- Proximity Profile
  - Specifies services used
  - Specifies GAP requirements for discoverability/connectability
- Services
  - Link Loss Service
  - Immediate Alert Service
  - Tx Power Service

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

Agenda

- Bluetooth low energy defined
- Architectural Overview
- Stack Architecture
  - Physical Layer
  - Link Layer
  - HCI Layer
  - L2CAP Layer
  - Security Manager Protocol
  - Attribute Protocol
  - Generic Attribute Profile
  - Generic Access Profile
  - Applications
- Comparison of LE to BR/EDR

Comparison with Classic Bluetooth

Feature	BR/EDR	LE	Notes
RF Channels	79	40	2 MHz spacing in LE
Modulation	GFSK	GFSK	Simple and effective
Modulation Index	0.25 to 0.35	0.45 to 0.55	Wider signal – more robust
Max Tx Power	+20 dBm (class 1) +4 dBm (class 2)	+10 dBm	No “class” structure +10 dBm regulatory limit
Rx Sensitivity (typical)	-85 dBm	-85 dBm	Pathloss = 90 dB for BR Pathloss = 95 dB for LE
Range (typical)	30 meters	50 meters	Modulation Index, increased power for class 2

Comparison with Classic Bluetooth

Feature	BR/EDR	LE	Notes
Packet Format	6 (BR / EDR)	2 (LE)	ID, FHS, DM, DH, 2-DH, 3-DH - Advertising / Data
Ack Packet Len	126 µs	80 µs	63% shorter
8 octet Packet	214 µs	144 µs	67% shorter
Max Packet Size	2875 µs = 1021 octets	328 µs = 27 octets	LE very short
Max Data Rate	2178.1 kb/s	305 kb/s	EDR much faster
Time to transfer 1Mbyte	DH1 = 18.2 s, DH5 = 8.8 s, 3-DH5 = 2.9 s	13.9 s (LE)	LE less efficient for large packets
CRC Strength	16	24	LE stronger
Encryption	Safer+	AES-128	LE stronger

Comparison with Classic Bluetooth

Feature	BR/EDR	LE	Notes
Authentication	once	every packet	more secure
Acknowledge	immediate	sliding window	lower power
Topology	flexible	pure star	LE simpler
Discoverable	Inquiry Scanning 11.25 ms / 1.25 s	Advertising 1.25 ms / 1.25 s	10x lower power
Connectable	Page Scanning 11.25 ms / 1.25 s	Advertising 1.25 ms / 1.25 s	10x lower power
Discoverable + Connectable	Inquiry + Page Scan 22.5 ms / 1.25 s	Advertising 1.25 ms / 1.25 s	20x lower power
LMP PDUs	75	14	5x simpler
Feature Bits	59	1	59x simpler

Comparison with Classic Bluetooth

Feature	BR/EDR	LE	Notes
Connection time	20 ms (R0 Page Scan)	2.5 ms	8x quicker
LMP negotiation time	min 5 ms ~ 50 ms	no negotiation required	instant
L2CAP connection setup time	min 5 ms ~ 50 ms	uses fixed channel no negotiation required	instant
Time to send application data	30 ms ~ 120 ms	3 ms	10x quicker
Time to AFH	1.25 ms	instant	no penalty for coexistence
Time to Sniff Subrating	2.5 ms	instant	no penalty for low power

Comparison with Classic Bluetooth

Feature	BR/EDR	LE	Notes
Protocols Supported by Host	14	3	Only ATT required for all plain text applications
Min protocols for application	3 (SDP, L2CAP, App Protocol)	2 (ATT, L2CAP)	LE uses ATT for service discovery and apps
1 MB using BR	8.81 s	13.93 s	BR 60% faster for data
1 MB using EDR	2.93 s	13.93 s	EDR ~5x faster
1 MB using HS	< 1s	13.93 s	LE very slow
L2CAP overhead	4 to 12 octets	4 octets	LE basic headers only
L2CAP configuration options	7	0	Nothing configured in LE
L2CAP commands	17	1	LE very simple

Agenda

- Bluetooth low energy defined
- Architectural Overview
- Stack Architecture
  - Physical Layer
  - Link Layer
  - HCI Layer
  - L2CAP Layer
  - Security Manager Protocol
  - Attribute Protocol
  - Generic Attribute Profile
  - Generic Access Profile
  - Applications
- Comparison of LE to BR/EDR

Additional Information and Training

- Bluetooth low energy specification can be found on the Bluetooth website, www.bluetooth.org/Technical/Specifications/adopted.htm

- Online training is also available on the Bluetooth website, www.bluetooth.org/Events/Training/LowEnergyTraining.htm