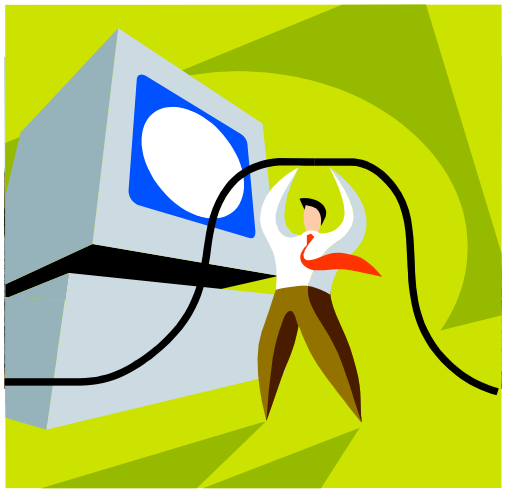
Mobile Networks

Connections and structures



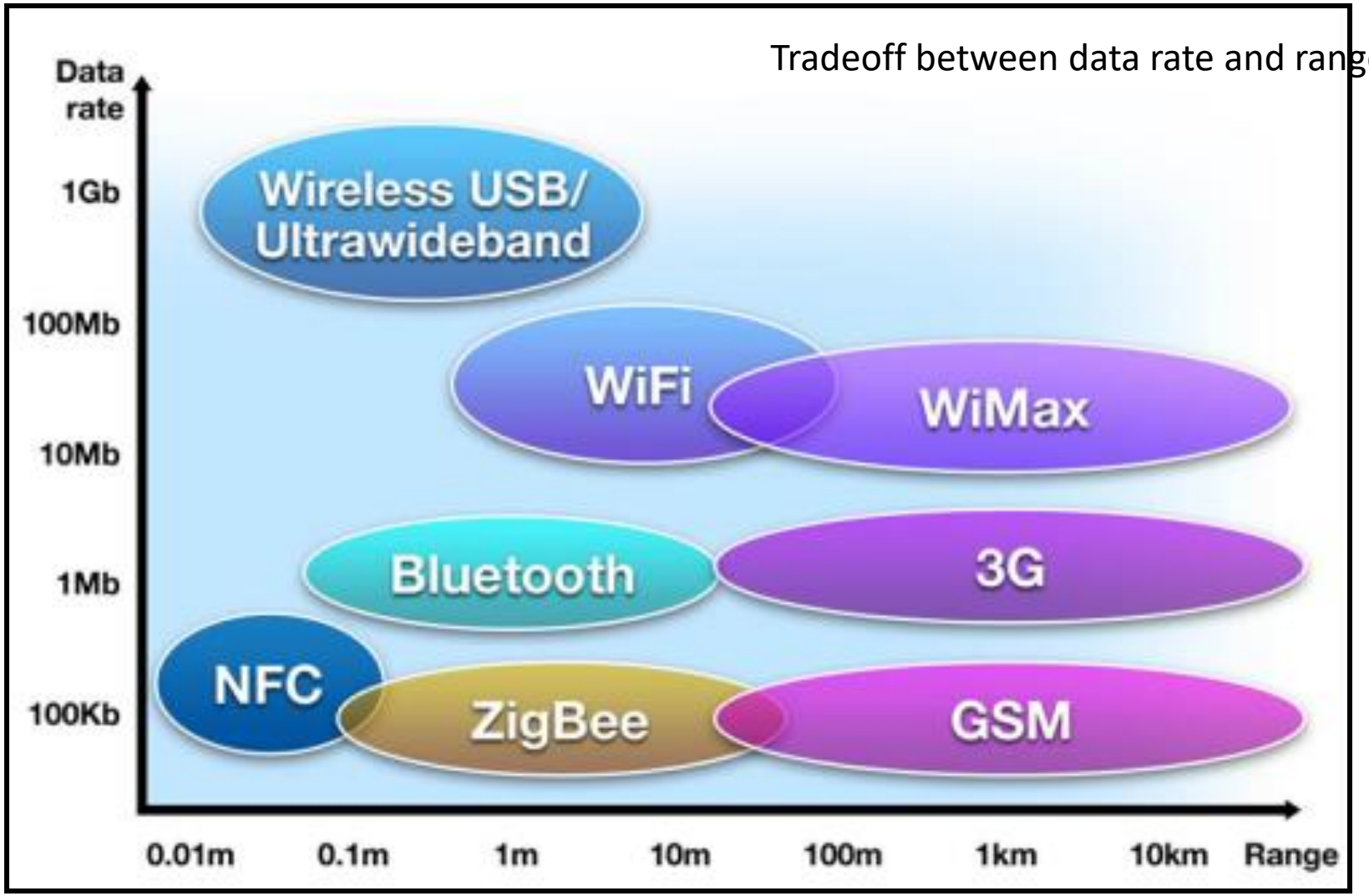
Bluetooth

WPAN

Outline

- Bluetooth networks
- Piconet operation
  - Inquiry
  - Paging
- Bluetooth stack
- Profiles and security
- BT 4.0 BLE

Comparison Between Wireless Technologies



Personal networks: when?

- Access mostly to “transported devices”
- No dominant need for Information Technologies
- No physical access to cabled networks
- No need for large communication rates
- Very low cost system required
- Consumer electronics integration is mandatory

Personal Area Networks

- Target deployment environment: communication of personal devices working together
  - Short-range
  - Low Power
  - Low Cost
  - Small numbers of devices
  - Sometimes have more “bus-like” characteristics
- PAN Standards
  - Bluetooth – Industry consortia
  - IEEE 802.15.1 – ”Bluetooth” based
  - IEEE 802.15.2 – Interoperability and coexistence
  - IEEE 802.15.3 – High data rate WPAN (UWB)
  - IEEE 802.15.4 – Low data rate WPAN (Zigbee,…)
  - IEEE 802.15.5 – Mesh Networks
  - IEEE 802.15.6 – Body Area Network

Bluetooth

- Originally for replacing “USB”, not “Ethernet”
  - Cable replacement technology
  - Later also used as Internet connection, phone, or headset
- Created by Ericsson
- PAN - Personal Area Network
  - Up to 1 Mbps connections
  - Includes synchronous, asynchronous, voice connections
  - Piconet routing
- Small, low-power, short-range, cheap, versatile radios
- Master/slave configuration and scheduling

History

1998 - Bluetooth technology is officially introduced and the BLUETOOTH SIG is formed. 1999 - Bluetooth 1.0 Specification is introduced.

2003 - The BLUETOOTH SIG overhauls the Bluetooth Core Specification with the announcement of Version 2.1.

2004 - Bluetooth Version 2.0 + EDR (Enhanced Data Rate) is introduced.

2005 - Devices using Version 2.0 + EDR begin to hit the market in late 2005.

2007 - Bluetooth Core Specification Version 2.1 + EDR is adopted by the BLUETOOTH SIG.

2009 - Bluetooth Core Specification Version 3.0 + HS (High Speed) is adopted by the BLUETOOTH SIG.

2010 - Bluetooth Core Specification Version 4.0 is adopted by the BLUETOOTH SIG.

2013 – Bluetooth 4.1

2014 – Bluetooth 4.2

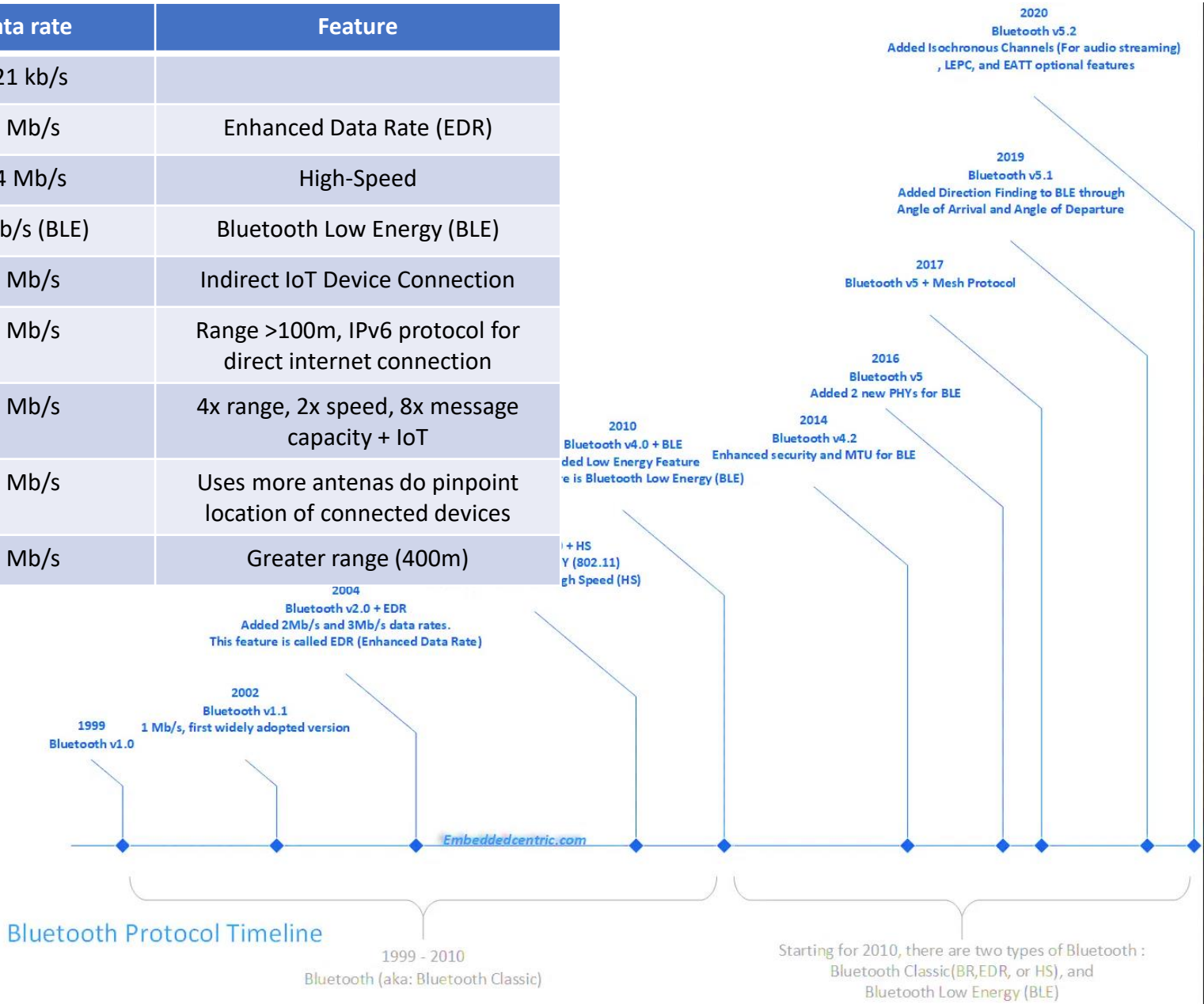
2016 – Bluetooth 5

2019 – Bluetooth 5.1, 5.2

2021 – Bluetooth 5.3

Bluetooth Versions

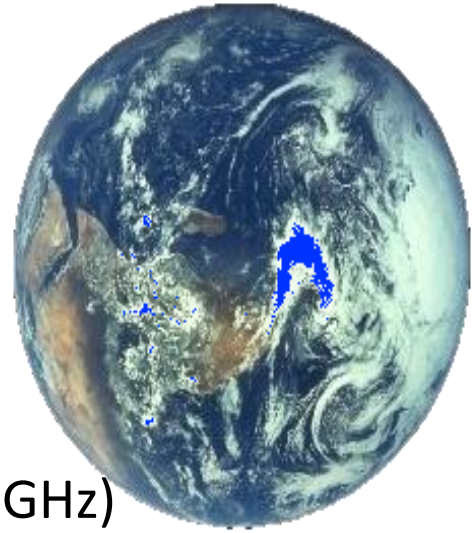
Version	Data rate	Feature
1.2	721 kb/s	
2.0 + EDR	3 Mb/s	Enhanced Data Rate (EDR)
3.0 + HS	24 Mb/s	High-Speed
4.0	1 Mb/s (BLE)	Bluetooth Low Energy (BLE)
4.1	1 Mb/s	Indirect IoT Device Connection
4.2	1 Mb/s	Range >100m, IPv6 protocol for direct internet connection
5.0	2 Mb/s	4x range, 2x speed, 8x message capacity + IoT
5.1	2 Mb/s	Uses more antenas do pinpoint location of connected devices
5.2	2 Mb/s	Greater range (400m)



Bluetooth higher speeds (in BT classic)

- Enhanced Data Rate (EDR)
  - Introduced in Bluetooth v2.0 to support faster data transfer
  - Supports a data rate up to **3 Mbps**
  - Using reduced duty cycle control (time radio is ON), EDR can provide lower power consumption
- High Speed (HS)
  - BT HS released in April 2009 (in Bluetooth version 3.0+HS)
  - Bluetooth 3.0+HS provides data transfer speeds of up to **24 Mbps**, though not over the Bluetooth link itself:
    - BT link is used for negotiation and establishment, and the high data rate traffic is carried over a collocated 802.11 link
    - HS part of the specification is not mandatory in BT 3.0
    - Only devices that display the "+HS" logo actually support Bluetooth over 802.11 high-speed data transfer

Bluetooth features



- Radio network, on the 2.4 GHz, world-wide!
- Airplane friendly!
- FH (Frequency Hopping) spread spectrum:
  79 (23 - .jp .es .fr) channels (de 2.402GHz - 2.480GHz)
  (We'll see how this works in the next slide!)
- Defines a master that syncrhonizes everyone to his hop-pattern.
- Defines two types of networks:
  - **piconets**
  - **scaternets**
- Maximum 8 devices per piconet (1 master + 7 slaves)
- Transmission rate: 720 Kb/s (max), assymetrical variable

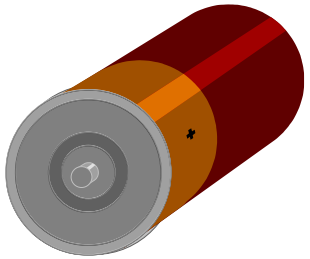
Frequency Hopping Spread Spectrum (FHSS)

- Signal broadcast over seemingly random series of frequencies
- Receiver hops between frequencies in sync with transmitter
  - Each frequency has the bandwidth of the original signal
  - Dwell time is the time spent using one frequency
- Spreading code determines the hopping sequence
  - Must be shared by sender and receiver (e.g. standardized)
- Eavesdroppers hear unintelligible blips
- Jamming on one frequency affects only a few bits
  - Typically large number of frequencies used
    - Improved resistance to jamming

Bluetooth classic vs. cable

Topology	Max. 7 simultaneous lines	1 line = 1 cable
Flexibility	Crosses walls, bodies, etc.	Line-of-sight, physical path
Transmission rate	1 MSPS, 720 Kbps	115Kbps - 400Mbps
Power	0.1 watts active power	0.05 watts or more
Dimensions	25 mm x 13 mm x 2 mm, several grams	Typical 1-2 metros. Weight varies with size
Cost	ci. 5 €/access	~ €4-€100/meter
Range	~ 10 meters	Typical 1-2 metros. Size = range.
Geographic coverage	~similar everywhere.	Cables and connections vary along the world.
Security	Link layer, SS radio. Very safe.	Ideal.

Low power



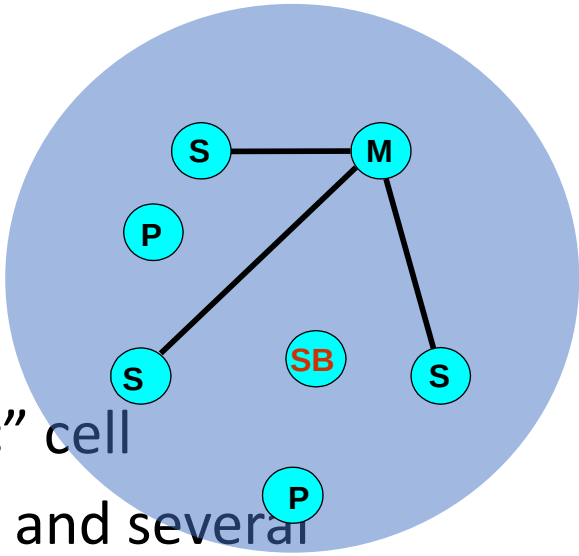
- Global architecture for low power
  - Hold and Park mode: 60 µA current
    - Connected device, but not operating
    - Device operates after a 2 ms wait process.
    - In Hold: keeps its AMA (**Active Member Address**); in Park has to free AMA, and later has to claim it back
- Transmission power ~1mW
  - 100mW classes also exist
- Standby Current < 0.3 mA

  $\Rightarrow$ 3 months
- Voice mode: 8-30 mA

  $\Rightarrow$ 75 hours
- Data mode (medium): 5 mA (0.3-30mA, 20 kbit/s, 25%)

  $\Rightarrow$ 120 hours

Piconets



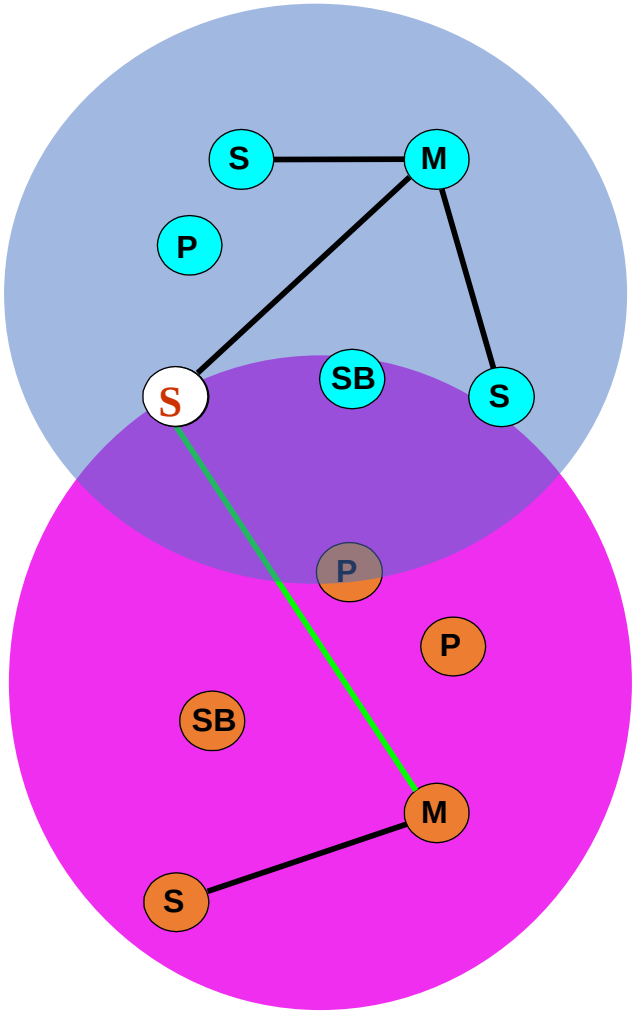
- Bluetooth devices connected in an “ad-hoc” cell
- There is a **master** with up to 7 active slaves and several hundreds parked.
  - Slaves only communicate with master
  - Slaves must wait for permission from master
- Master defines radio parameters (”clock” and “deviceID”)
  - Channel, hopping sequence, timing, …
- Each piconet has an unique FH pattern (e and a single ID)
- Each piconet has a maximum bandwidth (1MSPS)
- A slave in one piconet can also be part of another piconet
  - Either as a master or as a slave
  - If master, it can create **scatternets**

P=Parked
SB=Standby

M=Master
S=Slave

Scatternet

- Connection of several piconets
- Through a common device (bridge) (M/S)
- One device can be M/S at the same time
  - Or at least Slave in two piconets
  - Bridge node “stay” in a piconet for some time, then switch to another piconet by changing hop sequence.
- Global system BW unlimited, but piconet BW always <1Mbps
- Impact on piconets is minimal for < 10 piconets.
- Potentially any device can share piconets
  - Reality: limitations on commercial stacks



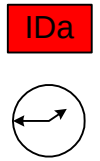
M=Master P=Parked
S=Slave SB=Standby

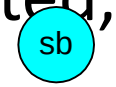
Outline

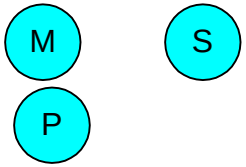
- Bluetooth networks
- Piconet operation
  - Inquiry
  - Paging
- Bluetooth stack
- Profiles and security
- 802.15.x

Piconet operation

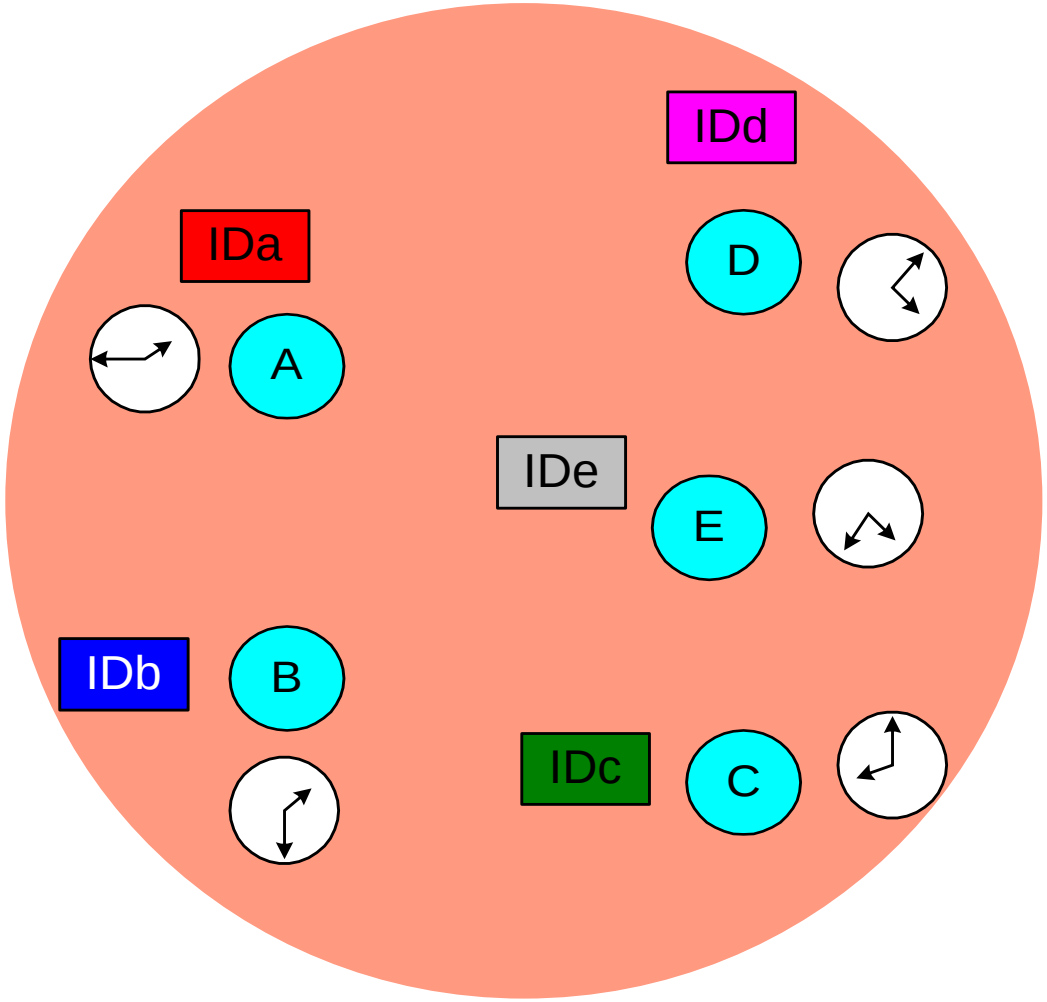
- FH-SS: all devices must share the same hopping pattern:
  - M*aster* provides clock and deviceID such that:
    - deviceID (48-bits) defines hopping pattern
    - Clock defines phase inside the pattern.
- If a device is inside a piconet, and is not connected, it must be in *standby*
- There are two types of piconet addresses (7+200...)
  - *Active Member Address* (AMA, 3-bits)
  - *Parked Member Address* (PMA, 8-bits)



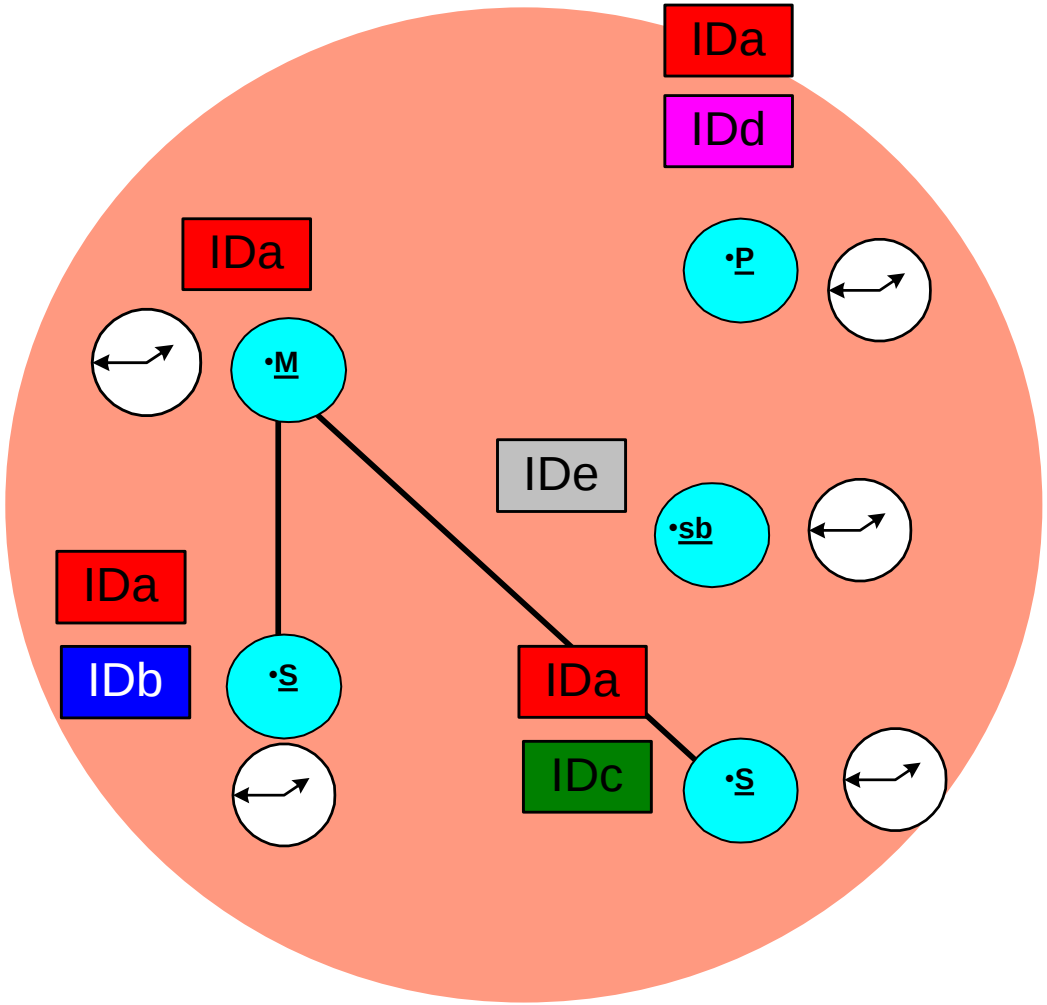




Piconet before setup



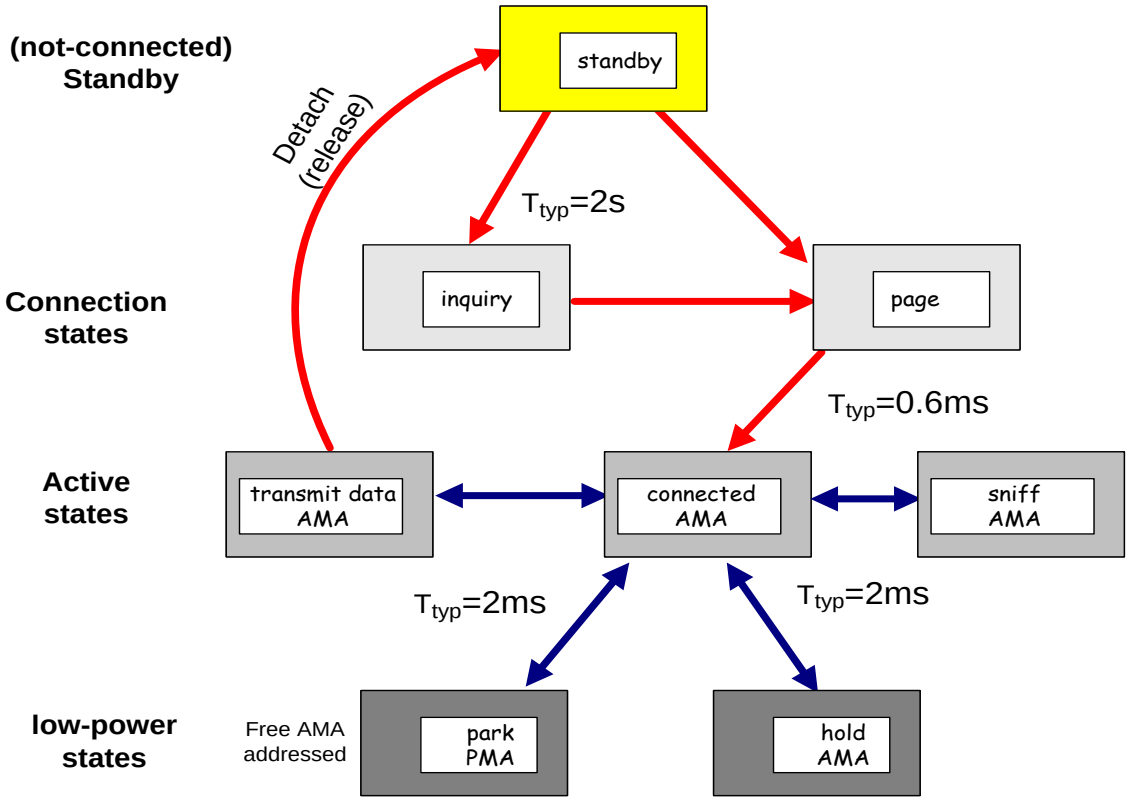
Piconet in operation



Piconet built!

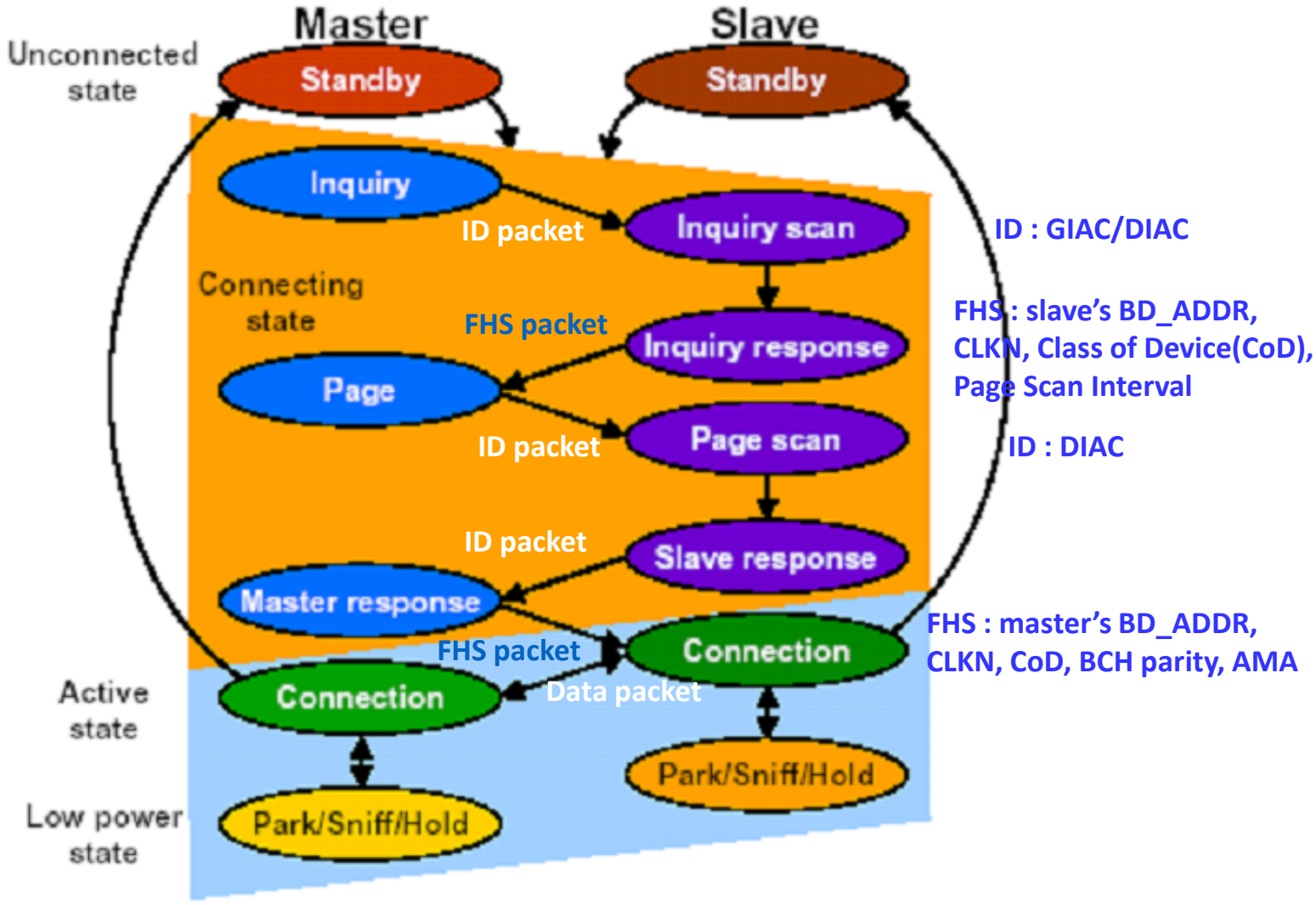
Device states

- Standby
  - Waiting to join a piconet
- Inquire
  - Ask about radios to connect to (discover nodes)
- Page
  - Connect to a specific radio
- Connected
  - Actively on a piconet (master or slave)
- Park/Sniff/Hold
  - Low Power connected states



Connection Procedure

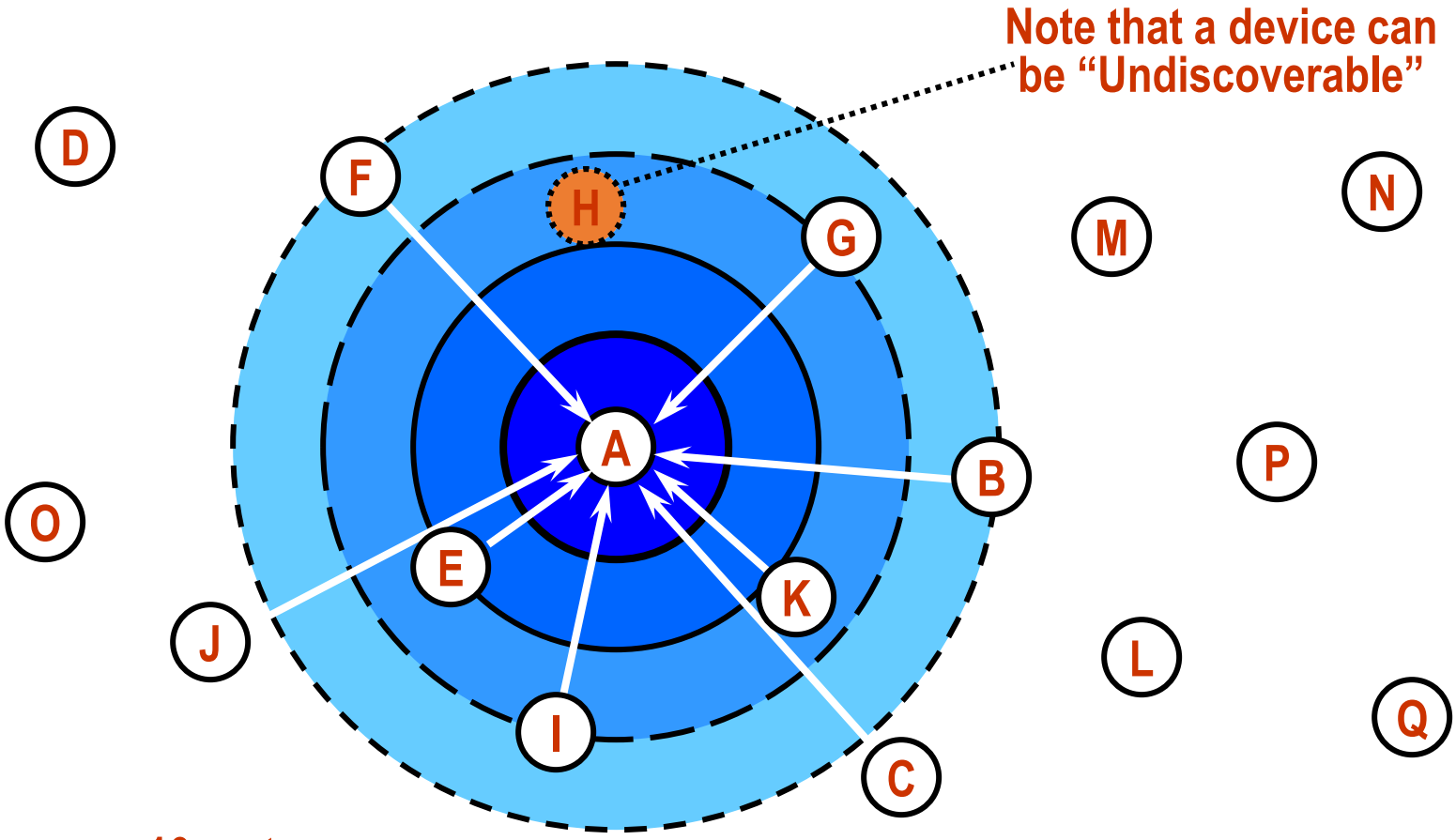
General Inquiry Access Code (GIAC)
Dedicated Inquiry Access Code (DIAC)



Low-Power Operation in BT classic

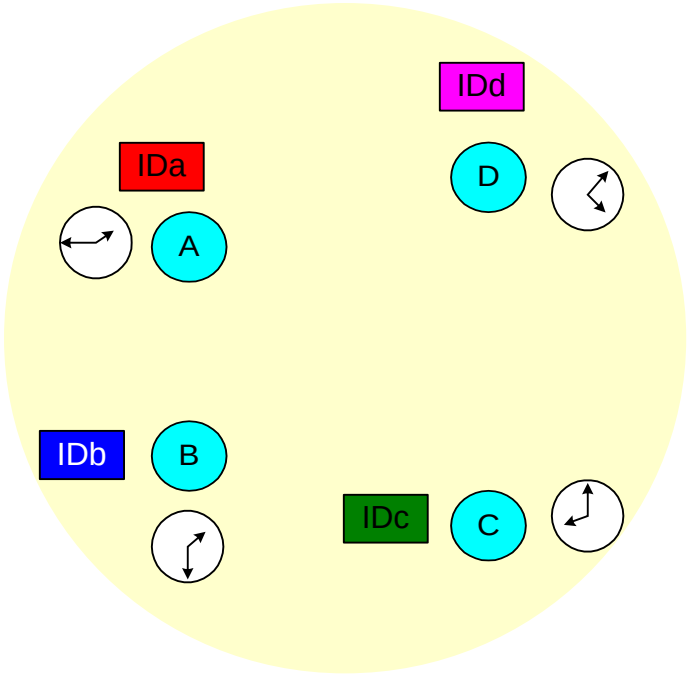
- 3 modes:
  - Hold: node sleeps for specified interval.
    - Master can put slaves in hold while searching for new members, attending another piconet, etc.
    - No ACL packets (Asynchronous Connection-Less) → general data packets
      - (there is also Synchronous Connection Oriented → Audio)
  - Sniff: slave low-duty cycle mode.
    - Slave wakes up periodically to talk to master.
    - Fixed “sniff” intervals.
  - Park:
    - Very low power state.
    - Used to admit more than 7 slaves in piconet.
      - Slave gives up its active member address.
      - Receives “parked” member address.
    - Wakes up periodically listening for broadcasts which can be used to “unpark” node.

Device Discovery Illustrated



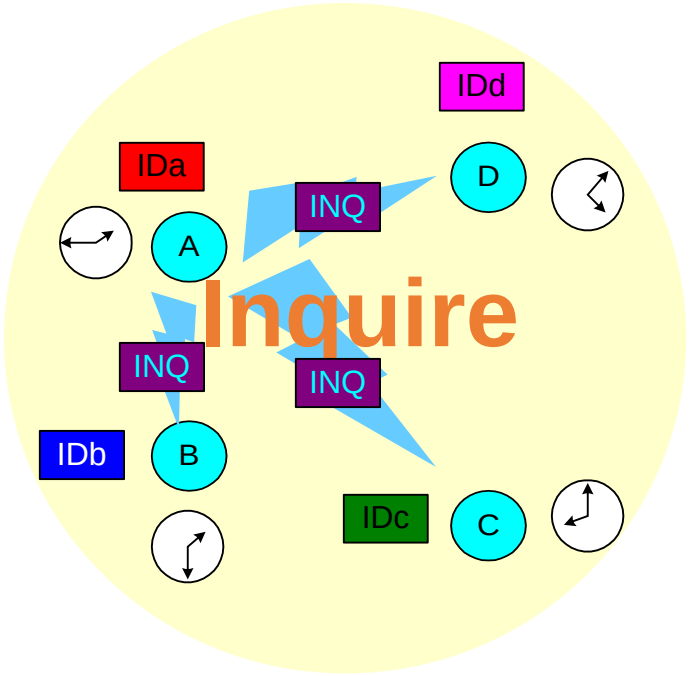
After inquiry procedure, A knows about others within range

Scanning units



Device A wants to search for stations

Scanning units

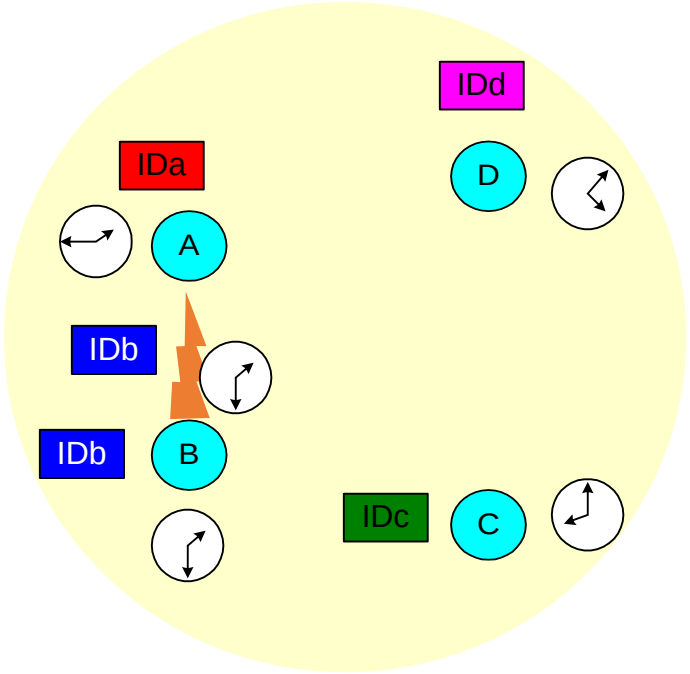


Device A wants to search for stations

A does an inquire (page with ID 000)

Devices B,C,D are doing an inquire scan

Scanning units



Device A wants to search for stations

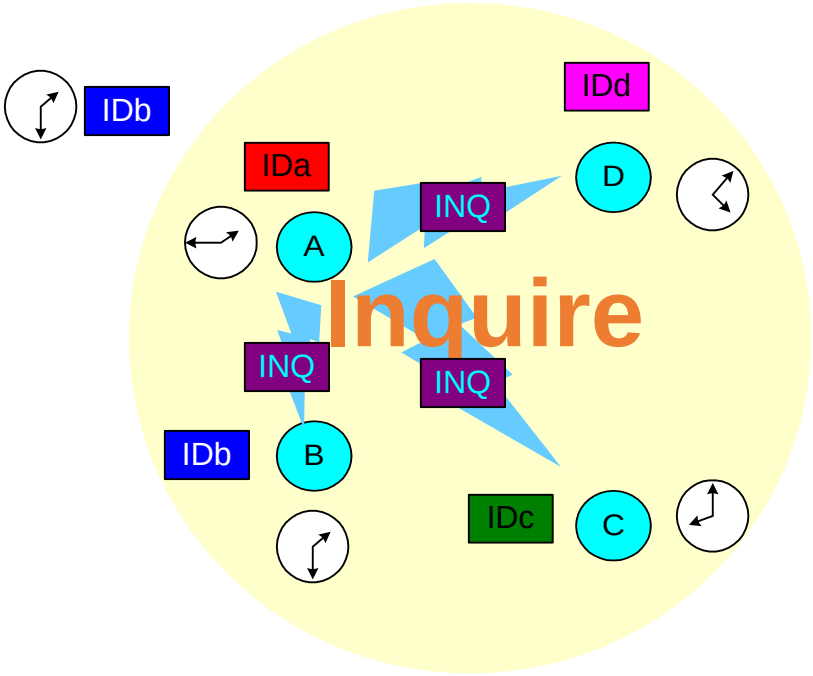
A does an inquire (page with ID 000)

Devices B,C,D are doing na inquire scan

B answers with FHS packet

Contains *DeviceID* and *Clock*

Scanning units

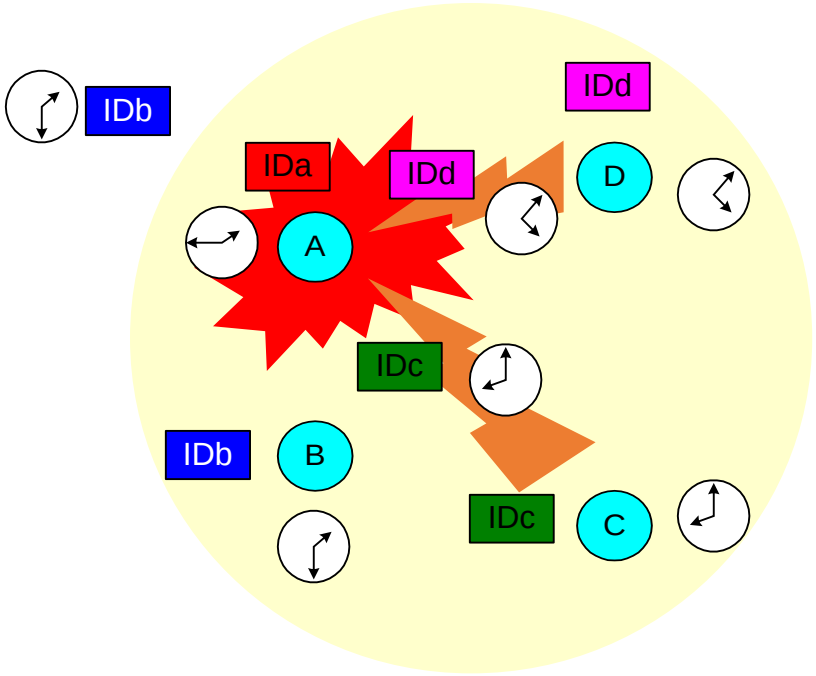


Device A wants to search for stations

- A does an inquire (page with ID 000)
  - Devices B,C,D are doing an inquire scan
- B answers with FHS packet
  - Contains *DeviceID* and *Clock*

A does an inquire again

Scanning units



A wants to search for stations

A does an inquire again

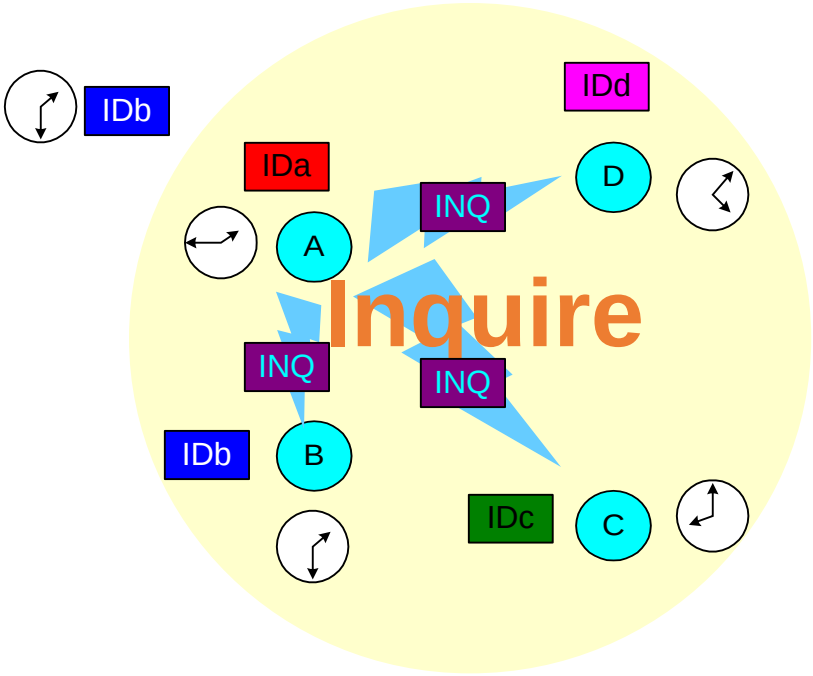
C e D answer at the same time with FHS packet

Packets are corrupted

A does not answer

C and D will wait an random number of slots

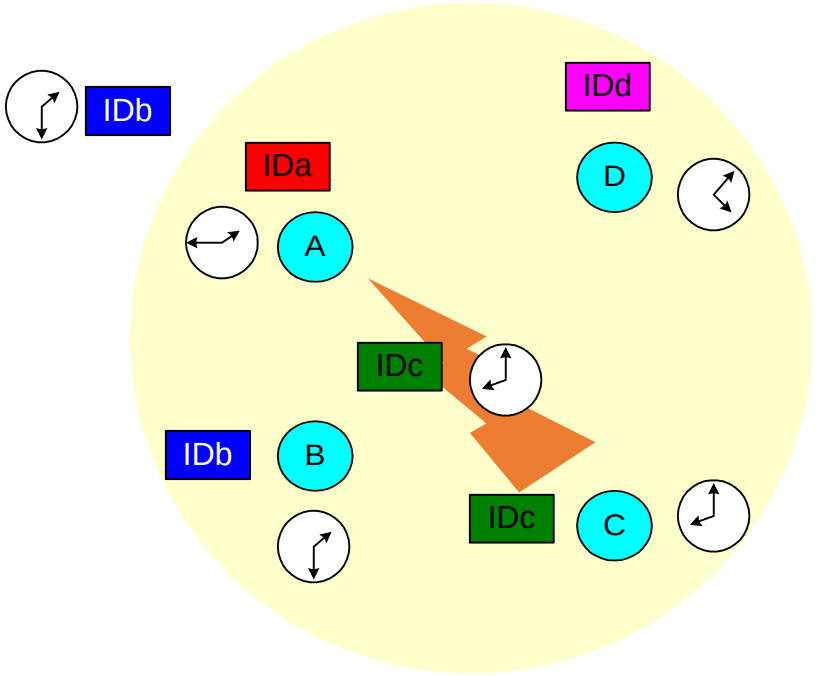
Scanning units



A wants to search for stations

A does an inquire again

Scanning units

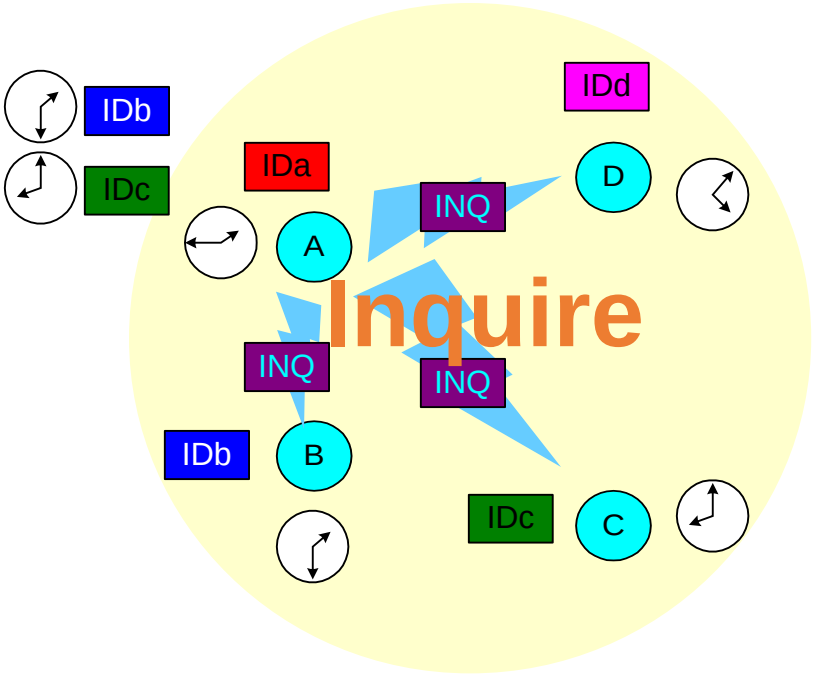


A wants to search for stations

A does an inquire again

C answers with FHS packet

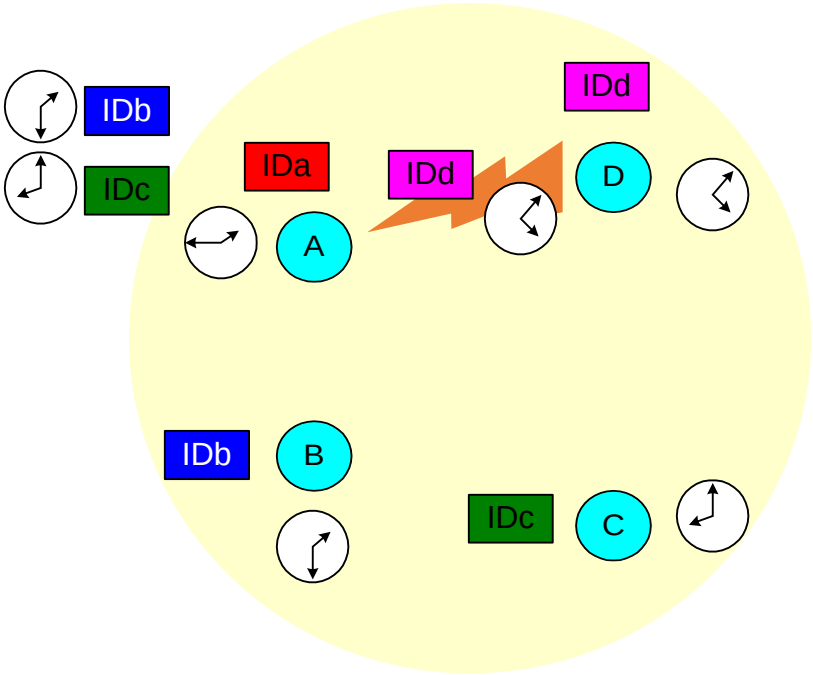
Scanning units



A wants to search for stations

A does an inquire again

Scanning units

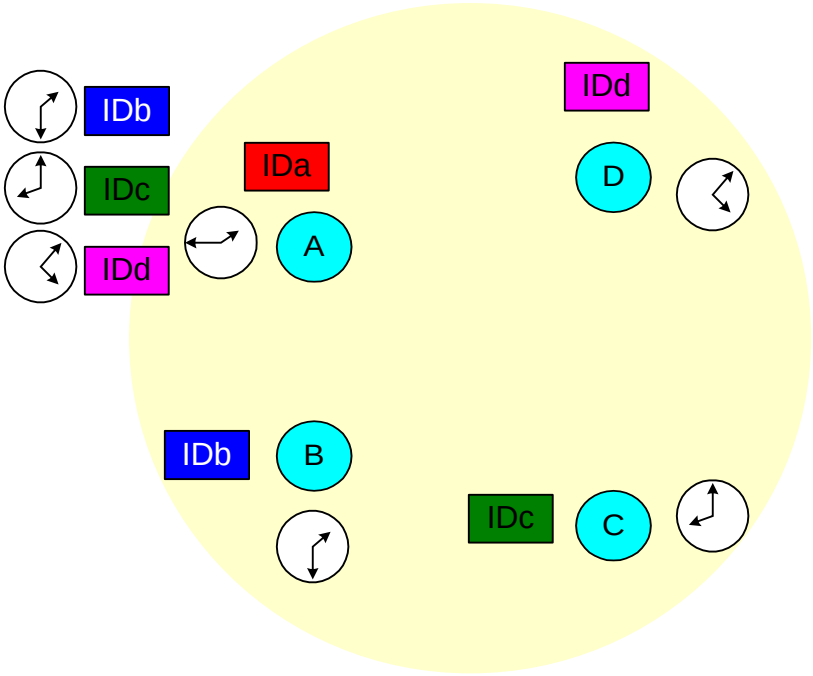


A wants to search for stations

A does an inquire again

D answers with FHS packet

Scanning units



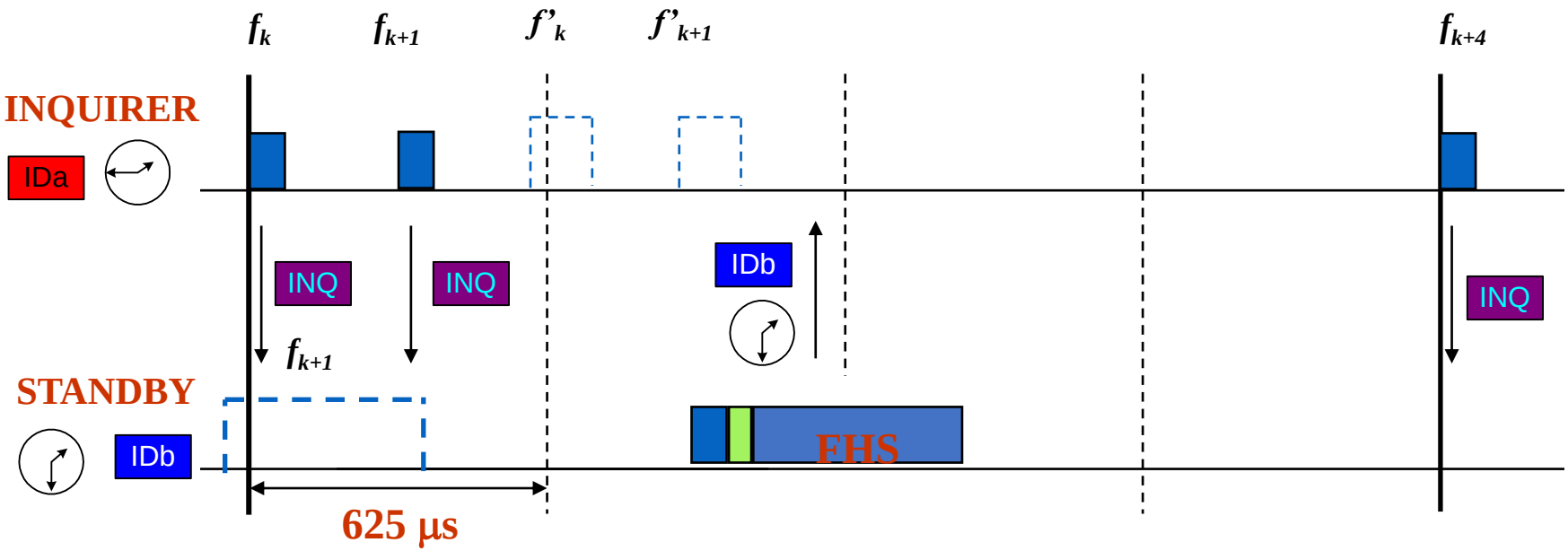
A has all the information it needs about the units in the cell.

Inquiry scanning: summary

- Inquiry scanning has a common address
  - and a common frequency pattern (from 32 frequencies)
- All devices can page this address (and become masters)
- All machines hearing an inquiry will answer the inquiry request
- There is a detector (*correlator hit*) in the slaves, that detects inquiries, before answering with a FHS providing:

  Device ID and Clock
- A machine in low power waits a random time before answering again to a scan
- If there is a collision on answering to a scan, they also wait a random period before answering again.

Timing: Inquiry



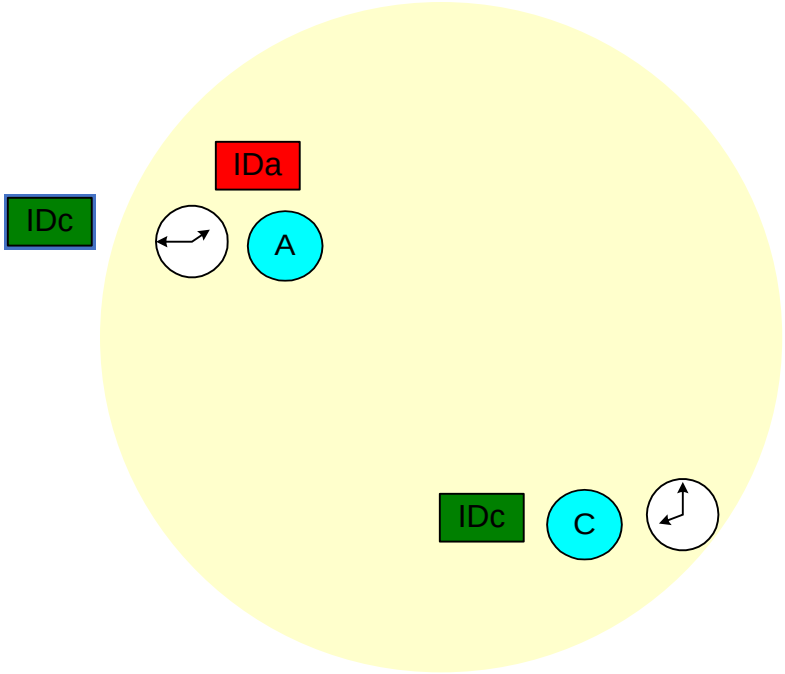
Inquiry requires two packets before the slave answers.

Paging: Will you connect to me?

- Very similar to inquire
- Still have not synchronized clocks or frequencies
- Establishes actual Piconet connection with a device that it knows about
- Connection process involves a 6 steps of communication between the the master and the slave

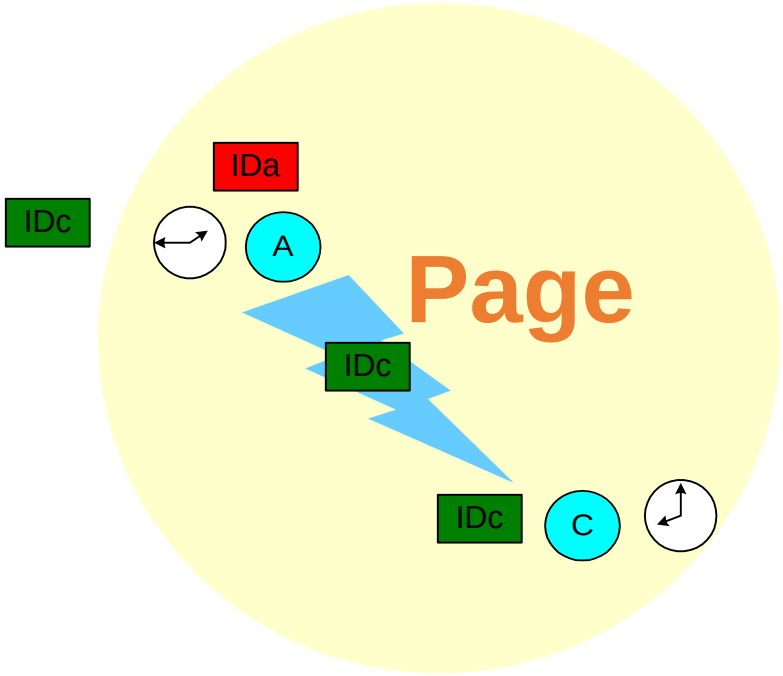
Step	Message	Direction	Hopping Pattern	Pattern Source and Clock
1	Slave ID	Master to Slave	Page	Slave
2	Slave ID	Slave to Master	Page Response	Slave
3	FHS	Master to Slave	Page	Slave
4	Slave ID	Slave to Master	Page Response	Slave
5	1st Master Packet	Master to Slave	Channel	Master
6	1st Slave Packet	Slave to Master	Channel	Master

Master Paging Slave



- Paging:
  - Assumes that the master has the *Device ID* and *Clock*

Master Paging Slave

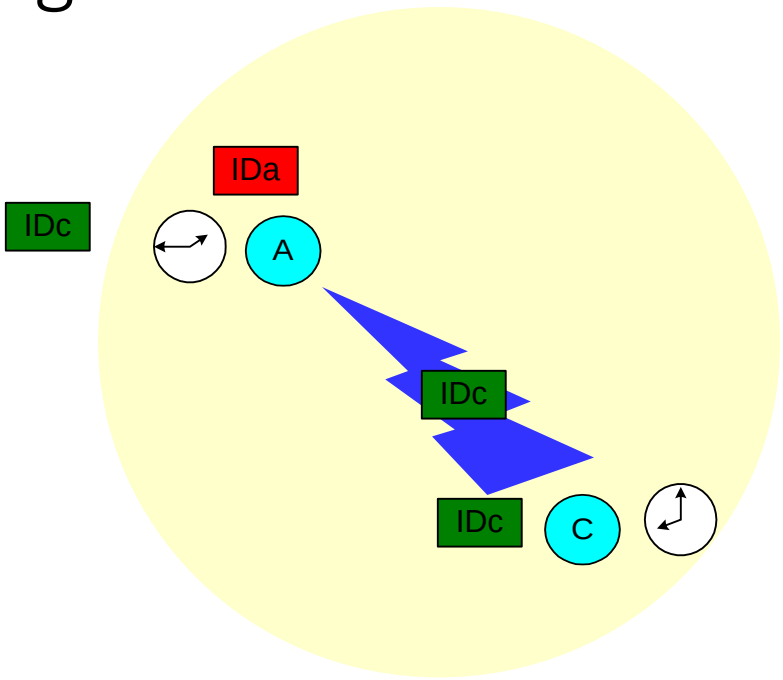


- Paging:

  Assumes that the master has the *Device ID* and *Clock*

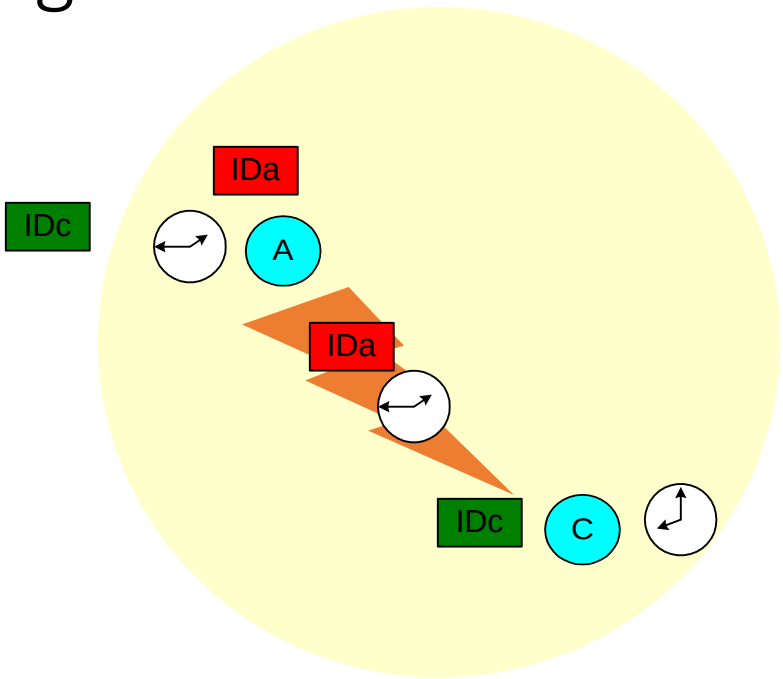
  – *A* pages C with the *deviceID* of C

Master Paging Slave



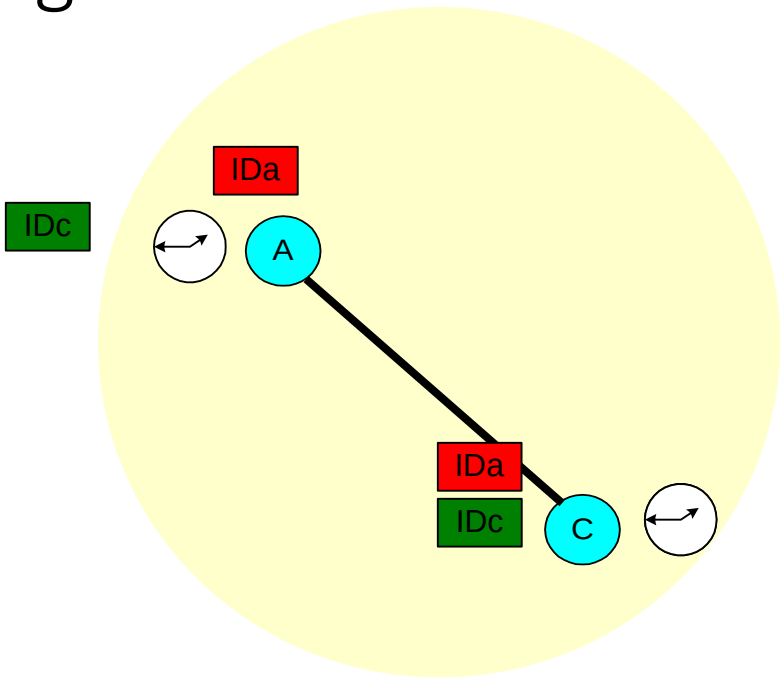
- Paging: master has the *Device ID* and *Clock*
  - *A* pages C with the *deviceID* of C
  - C answers A with his *deviceID*

Master Paging Slave



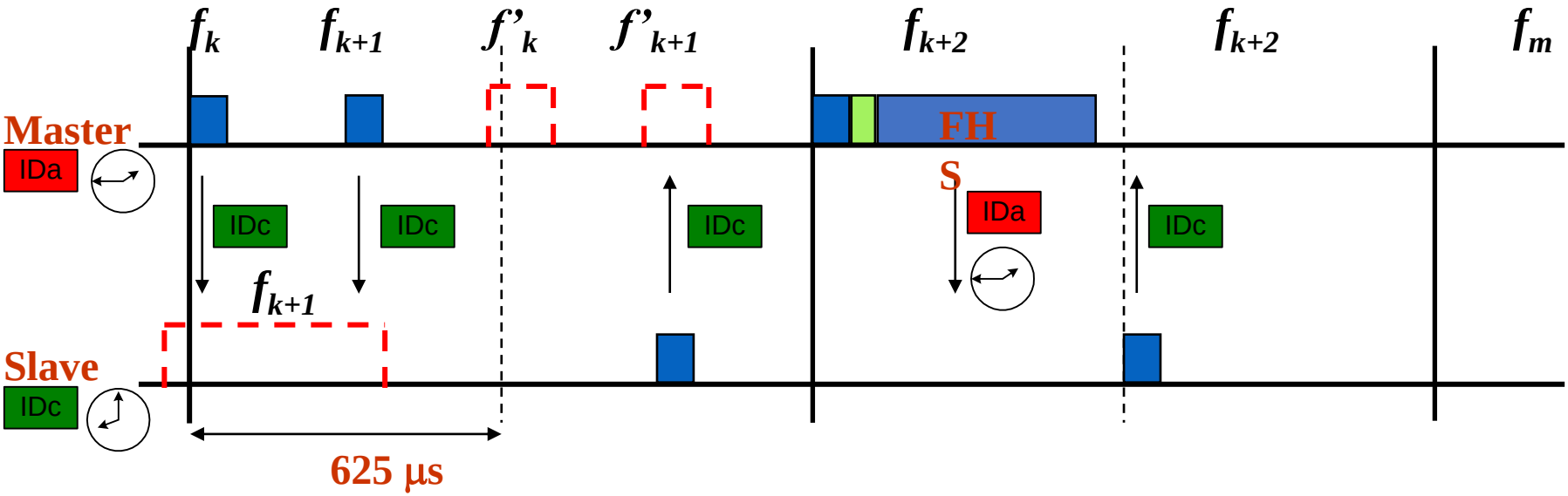
- Paging: master has the *Device ID* and *Clock*
  - *A* pages C with the *deviceID* of C
  - C answers A with his *deviceID*
  - A send C his *deviceID* and *Clock* (FHS packet)

Master Paging Slave



- Paging: master has the *Device ID* and *Clock*
  - *A* pages C with the *deviceID* of C
  - C answers A with his *deviceID*
  - A send C his *deviceID* and *Clock* (FHS packet)
  - A becomes master of C

Time: Master Paging Slave

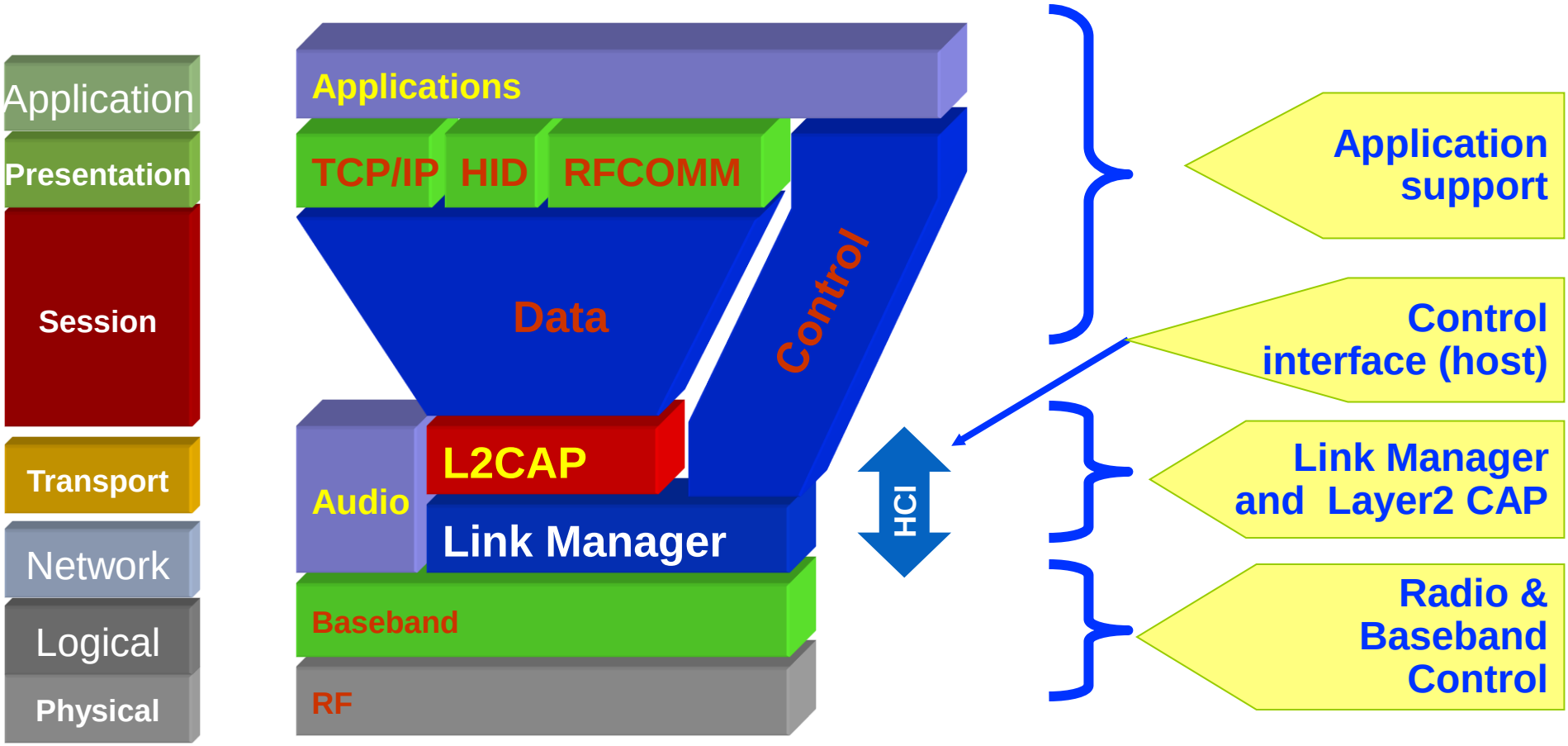


- Master pages slave (packet has slave's ID) at the paging frequency of the slave (1 of 32)
  - Master send a train of 16 fqs in the slave hop set.
    - Slave ID sent twice in the slave frequency
    - Master waits for two answers in the slave frequency
  - If it does not work, master will send
- Slave listens for 11 ms (page scan)
  - If it identifies packets, slave wakes up and sends packets in that frequency.
  - Master answers with FHS (*Device ID* e *Clock*)
  - Slave joins piconet.

Outline

- Bluetooth networks
- Piconet operation
  - Inquiry
  - Paging
- Bluetooth stack
- Profiles and security
- 802.15.x

stack Bluetooth



Bluetooth includes:

- A HW description
- an environment for applications

L2CAP – Logical Link Control and Adaptation Protocol

LMP – Link Manager Protocol

HID – Human Interface Device

RFCOMM – serial cable emulation (ETSI)

Bluetooth Protocol

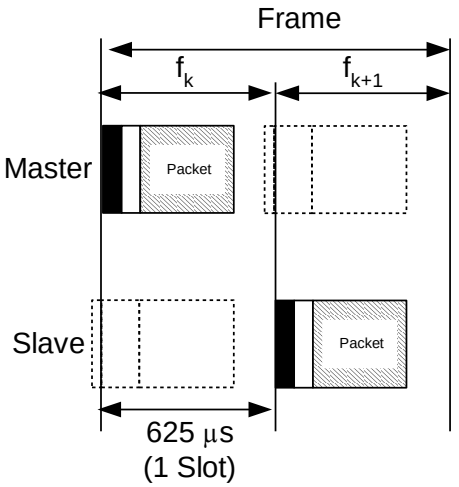
- Radio layer
  - Defines requirements for a Bluetooth radio transceiver
  - Handles conformity to 2.4GHz band
  - Establishes specifications for using Spread-Spectrum Frequency Hopping
  - Classifies device into one of three power classes:
    - long range; (Class 1 - 100mW, 100m)
    - normal/standard range; (Class 2 - 2.5mW, 10m)
    - short range; (Class 3 - 1 mW, 1m)

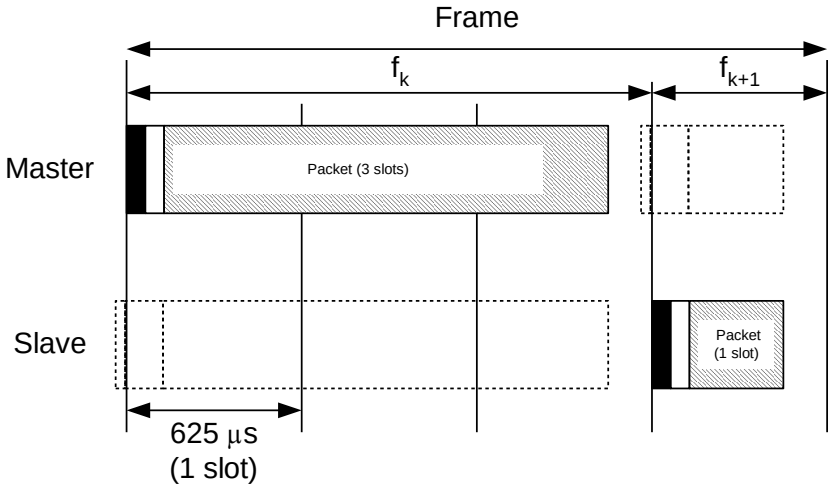
RF: international 2.4Ghz band

- 2.4GHz issues
  - Channel BW limited to 1MHz
  - Spread spectrum must be used
  - Multiple independent networks can interfere
  - Microwave ovens also use these frequencies.
  - ICs at 2.4 GHz need huge current levels.
- Bluetooth remedies
  - 1 Mb/s baud taxes exploit BW at maximum.
  - Voice coding (CVSDM) allow high speed operation.
  - Fast “frequency hoping” and small packets to avoid interference.
  - Interface with the channel minimizes power consumption.
  - Interface specifications are relaxed enough to allow its integration in low power chipsets.

Radio Layer

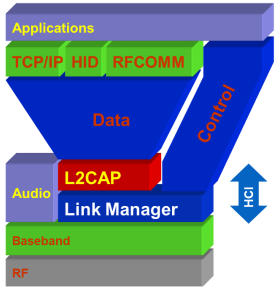
- Radio: FH SS
  - 79/23 channels of 1 Mb/s
  - Hoping: per slot
    - Packets have 1, 3, or 5 slots of 625 μs.
    - Hoping (nominal) 1600 times per second
  - Frame includes two packets
    - Transmission followed by reception
  - Radio designed to low cost and universal usage
    - (noise, synchronous action technology 2.4GHz, etc…)





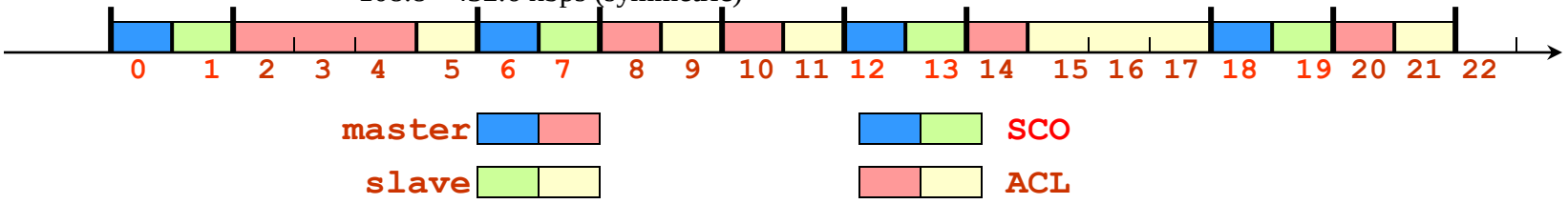
Baseband in Bluetooth

- Manages physical channels and logical lines
  - Controls device addressing, channel control, power-saving operations, and flow control and synchronization among devices
  - Implements TDD aspects: master and slave switch in communications
- Works closely with Link controller:
  - Manages link (a)synchronism
  - Controls paging and inquiries
  - Controls power save modes

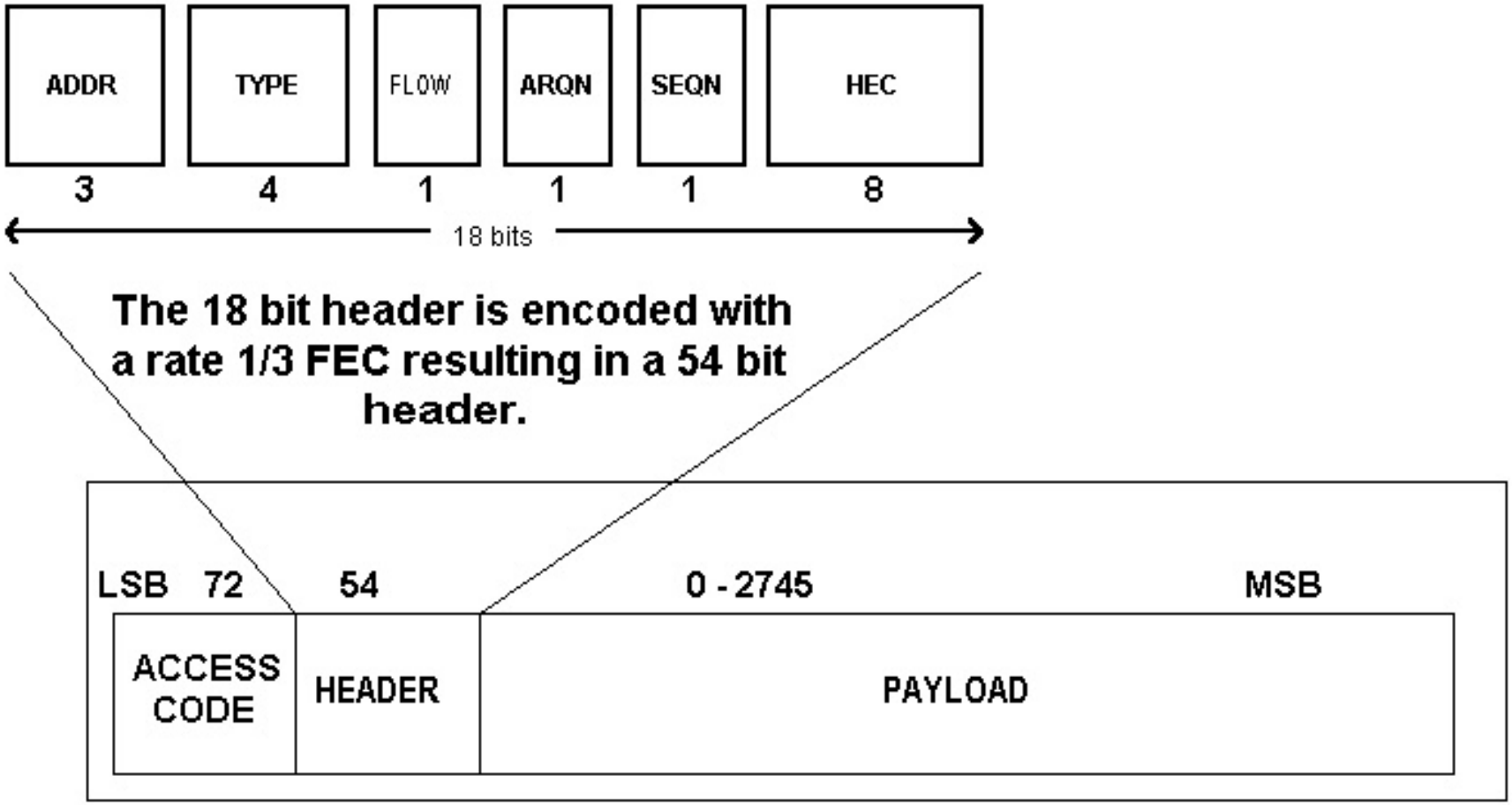


Baseband link types

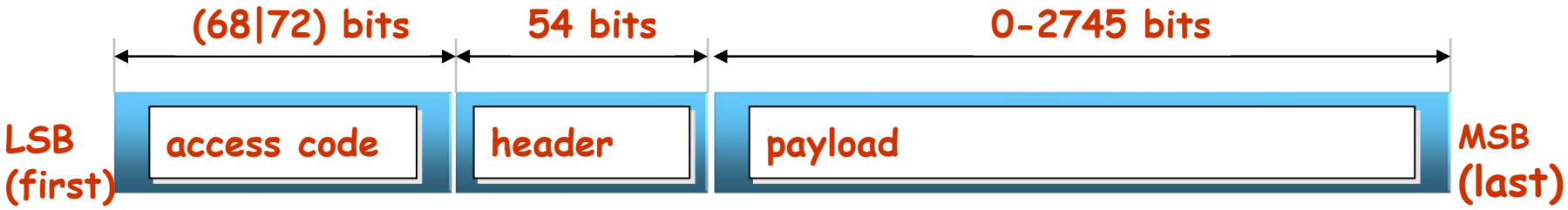
- **Polling-based (TDD) frame transmissions**
  - **1 slot: 0.625msec (max 1600 slots/sec)**
  - **master/slave slots (even-/odd-numbered slots)**
  - **polling: master always "polls" slaves**
- **Synchronous connection-oriented (SCO) link**
  - **"circuit-switched"**
    - periodic single-slot frame assignment
  - **symmetric 64Kbps full-duplex**
- **Asynchronous connection-less (ACL) link**
  - **Frame switching**
  - **asymmetric bandwidth**
    - variable frame size (1-5 slots)
      - max. 721 kbps (57.6 kbps return channel)
      - 108.8 - 432.6 kbps (symmetric)



Baseband Packet

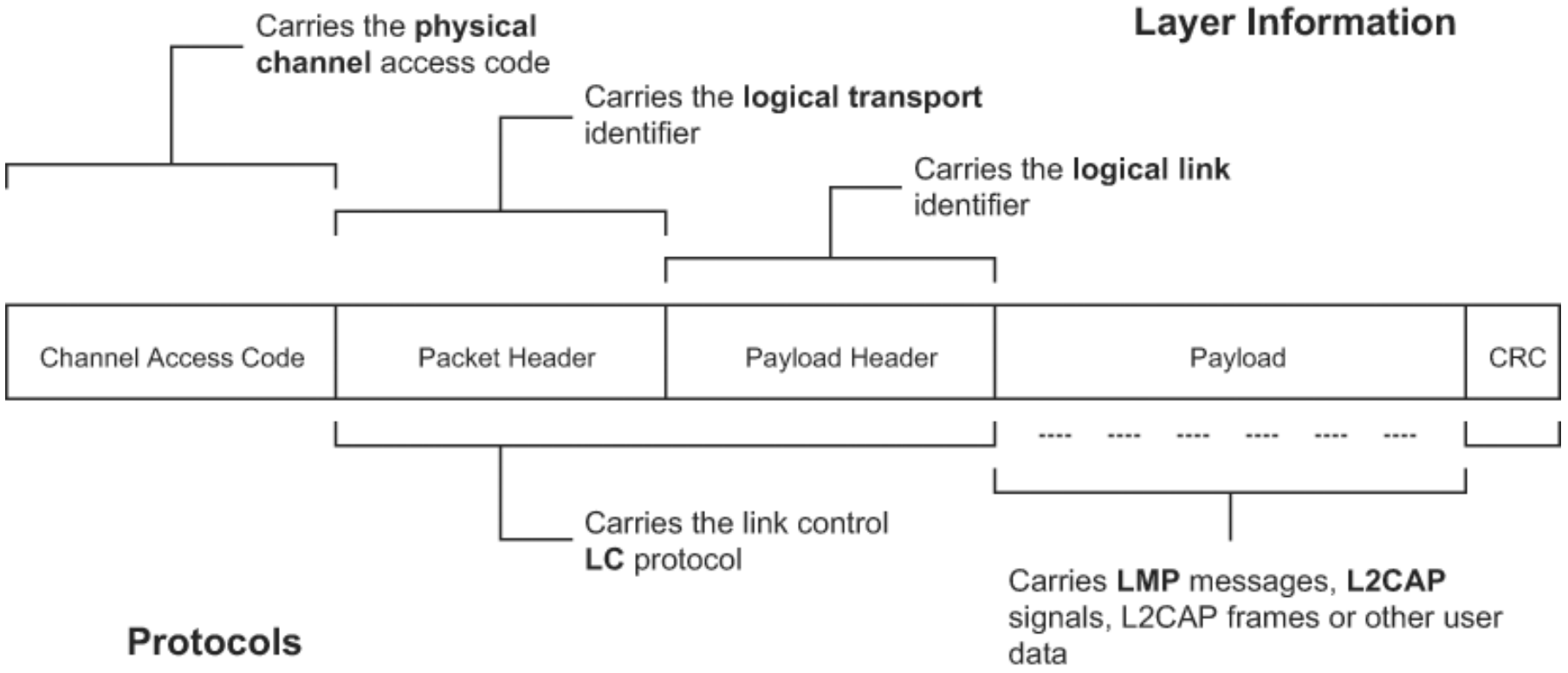


Baseband Frame



- **Access Code**: time synchronization, offset, paging, inquiry.
  - Channel Access Code (CAC), piconet identification, synchronization, DC offset.
  - Device Access Code (DAC), paging and replies.
  - Inquiry Access Code (IAC), inquiries (GIAC, general; DIAC, dedicated)
- **Header**: packet acknowledgement and numbering, flow control, slave address, error checking
- **Payload**: voice, data or both (DV packets).
  - When data, the payload has additional internal header

Bluetooth Frame : Role of fields



ACCESS CODE - based on identity and system clock of Master

Provides means for synchronization; Unique for channel;

Used by all frames on the channel

Packets (common)

TYPE	NAME	#	DESCRIPTION
Common	ID	1	Carries device access code (DAC) or inquiry access code (IAC).
	NULL	1	NULL packet has no payload. Used to get link information and flow control. Not acknowledged.
	POLL	1	No payload. Acknowledged. Used by master to poll the slaves to know whether they are up or not.
	FHS	1	A special control packet for revealing Bluetooth device address and the clock of the sender. Used in page master response, inquiry response and frequency hop synchronization. 2/3 FEC encoded.
	DM1	1	To support control messages in any link type. can also carry regular user data. Occupies one slot.

Packets: Synchronous Connection-oriented

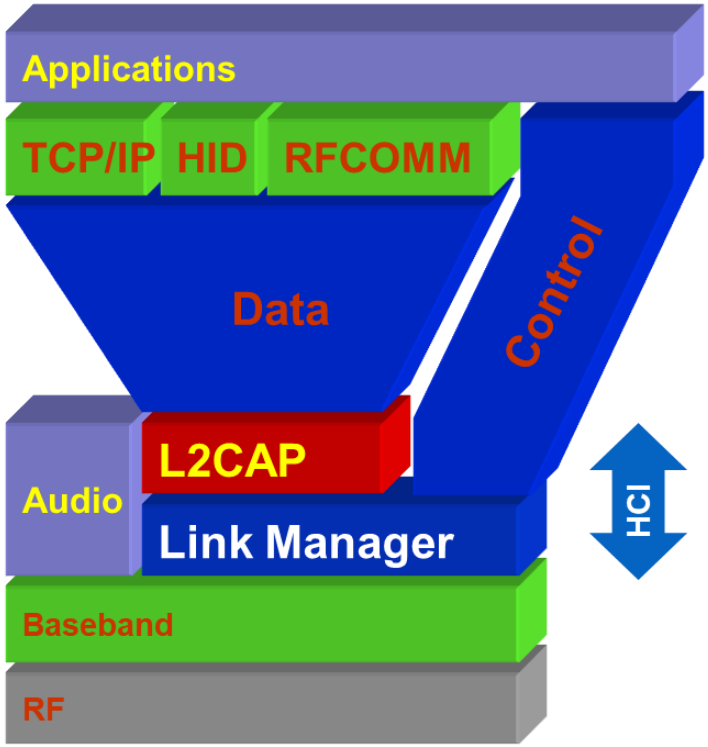
SCO	HV1	1	Carries 10 information bytes. Typically used for voice transmission. 1/3 FEC encoded.
	HV2	1	Carries 20 information bytes. Typically used for voice transmission. 2/3 FEC encoded.
	HV3	1	Carries 30 information bytes. Typically used for voice transmission. Not FEC encoded.
	DV	1	Combined data-voice packet. Voice field not protected by FEC. Data field 2/3 FEC encoded. Voice field is never retransmitted but data field can be.

Packets : Assynchronous Connection-Less

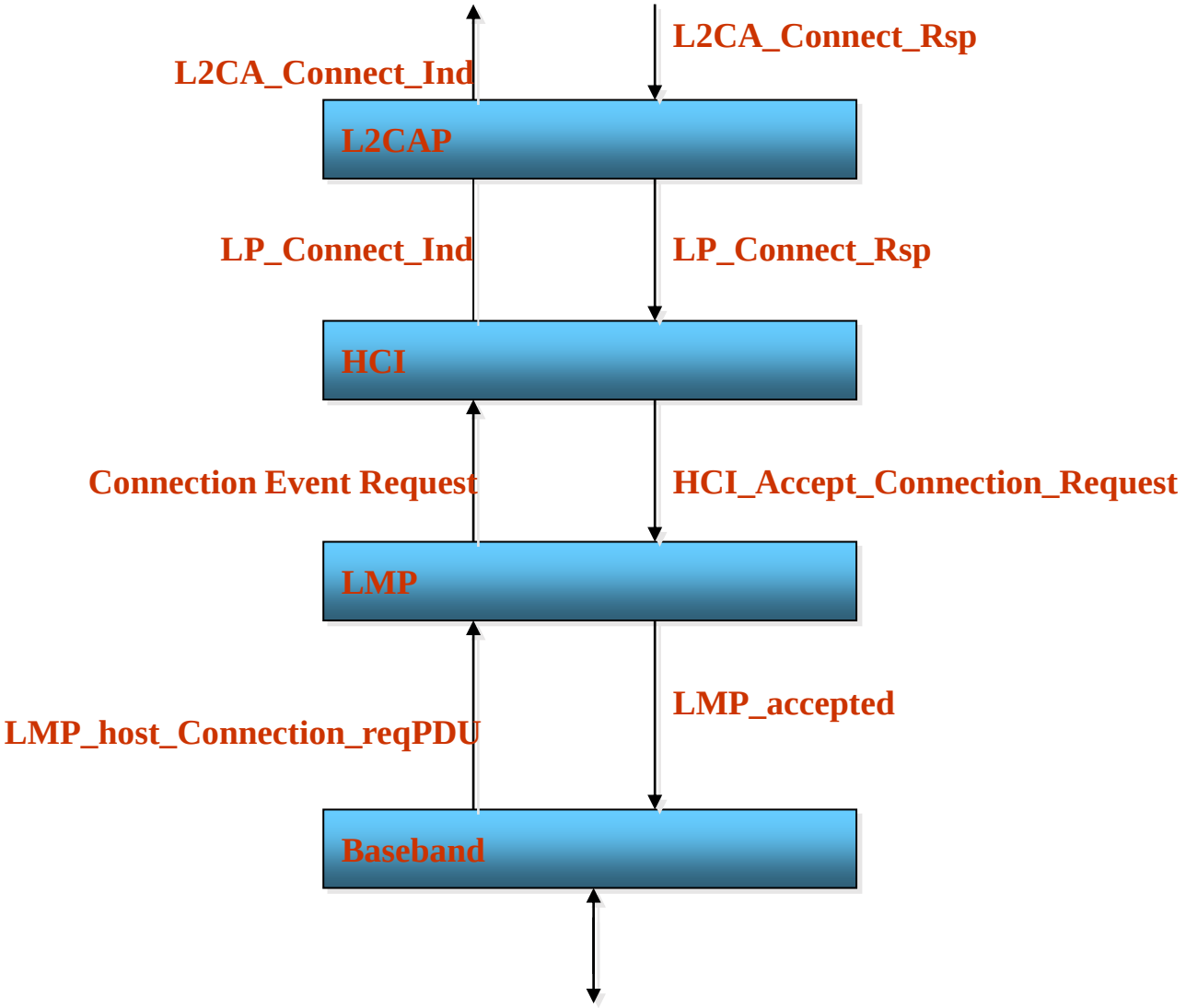
ACL	DM1	1	Carries 18 information bytes. 2/3 FEC encoded.
	DH1	1	Carries 28 information bytes. Not FEC encoded.
	DM3	3	Carries 123 information bytes. 2/3 FEC encoded.
	DH3	3	Carries 185 information bytes. Not FEC encoded.
	DM5	5	Carries 226 information bytes. 2/3 FEC encoded.
	DH5	5	Carries 341 information bytes. Not FEC encoded.
	AUX1	1	Carries 30 information bytes. Resembles DH1 but no CRC code.

Adaptation protocols

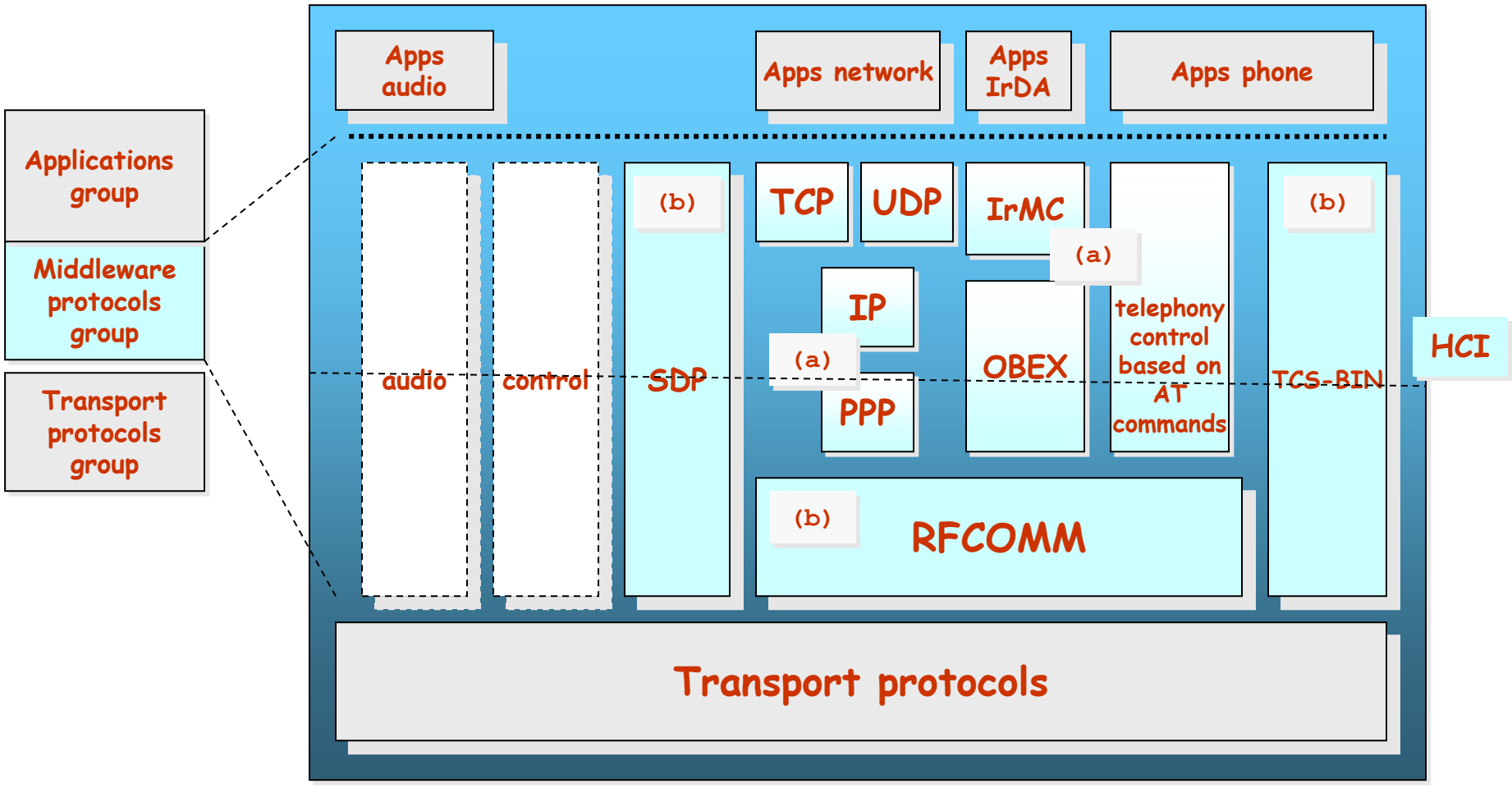
- Link Manager
  - carries out link setup, above baseband, with authentication, link configuration and other protocols
    - Support protocol multiplexing
      - BT may support other protocols besides IP
    - Segmenting and reassembly
- Link Layer Control & Adaptation (L2CAP)
  - Link control protocol, provides connection-oriented and connectionless data services to upper layer protocols
    - Handles ACL and SCO connections
    - Handle QoS specifications per connection (logical channel)
    - Manages concepts as “group of connections”
- Host Controller Interface (HCI)
  - Allows command line access to the baseband layer and LM for control and status information
    - Current interfaces: USB; UART; RS-232
  - Made up of three parts:
    - HCI firmware, HCI driver, Host Controller Transport Layer



Interlayer communication



Protocols (middleware)



a: common protocol
b: Bluetooth dedicated protocol

SDP: Service Discovery Protocol
OBEX: Facilitates binary transfers between BT devices
TCP-BIN: Telephony-control protocol binary (call control)

Middleware

- Service Discovery Protocol (SDP)
  - Provides a way for applications to detect which services are available and their characteristics
  - Protocol question ⮜⮞ answer
    - (search and browsing of services)
  - Defines a format for service registry
    - Information provided by the service *attributes,* a name (ID) + value
    - IDs can be universal (UUID)
- Protocol reusage
  - BT aims to reuse older protocols (e.g. WAP, OBEX–IrDA)
    - Interaction with applications and phones, as commonly done before

Middleware

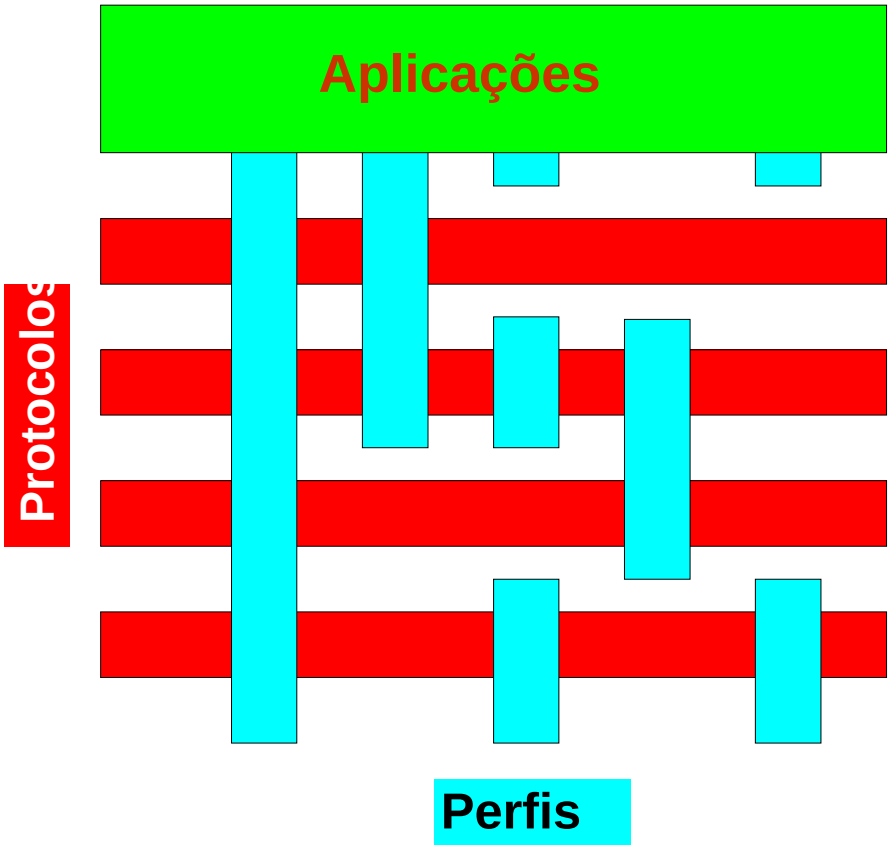
- RFCOMM
  - Based on GSM TS07.10
  - Emulates a serial port, supporting all traditional applications that were able to use a serial port.
  - Supports multiple ports over a single physical channel between two devices.
- Telephony Control Protocol Spec (TCS)
  - Handles call control (setup, release)
  - Group management for gateways, serving multiple devices
    - Audioconference, e.g.

Outline

- Bluetooth networks
- Piconet operation
  - Inquiry
  - Paging
- Bluetooth stack
- Profiles and security
- 802.15.x

Interoperability: Profiles

- Profile: base for BT interoperability (BT too much flexible!)
- “vertical cut” in Bluetooth stack
- A given usage model (typical solution)
- Each BT device supports one or more profiles



Profiles (v.1)

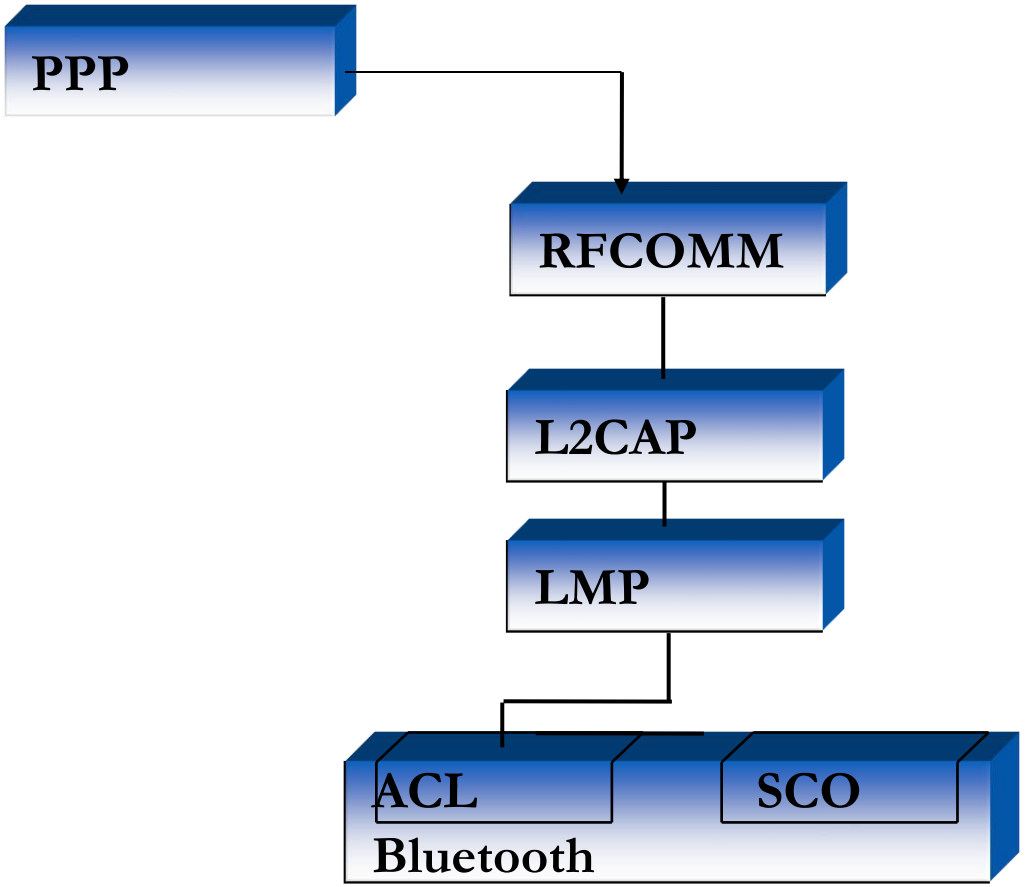
- Generic Access
  - Profile SDA
    (service discovery application)
  - Profiles for serial port, including:
    - Profile Dial-up
    - Profile Fax
    - Profile headset
    - LAN Access (uses PPP)
    - Profile for generic object exchange (OBEX)
      - File transfer
      - Data synchronization
      - Push-pull
- Profile of coordless phone(TCS_BIN)
  - Profile interphone
  - Profile Cordless Telephony

Profiles (v.2)

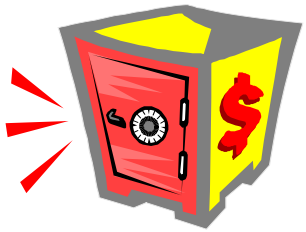
- Radio 2 (next generation radio)
  Compatible with existing systems
- Car Profile
- PAN Profile
- GPS Profile
- Printing Profile
- Still image Profile

(globally better facilities in audio/voice/video)
(better service discovery)
(improved human interfaces)
(improved interoperation with other devices at the 2.4GHz ISM)

Example: LAN Access Point



Bluetooth: security



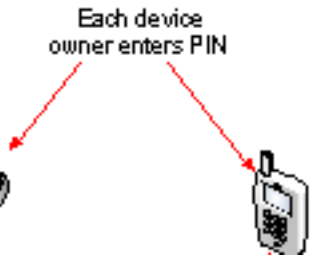
- Devices can be:
  - “Trusted”
  - “Untrusted”
    - Also “unknown” devices
- Services security types:
  - Open services – cypher only
  - Authentication only – machine ID
  - Authentication and authorization (ID+explicit service grant)
- Levels of security:
  - Mode 1
    - No security
  - Mode 2
    - Security guaranteed at service level
  - Mode 3
    - Security guaranteed at link level

Bluetooth: security features

- Mechanisms used in BT for security
  - Fast frequency hopping
  - Low range
  - Authentication
    - Two way challenge/response mechanism
  - Cypher (to ensure privacy)
    - Data between two devices can be encrypted
    - Keys used
      - Cypher size configurable (0-16bytes) by the devices, but there are security constrains (goverment)
      - Keys using standard well-known algorithms
  - Security initialization – device pairing
    - PIN (user input)
    - Shared key

Bluetooth Pairing

Mode 1 – not secure



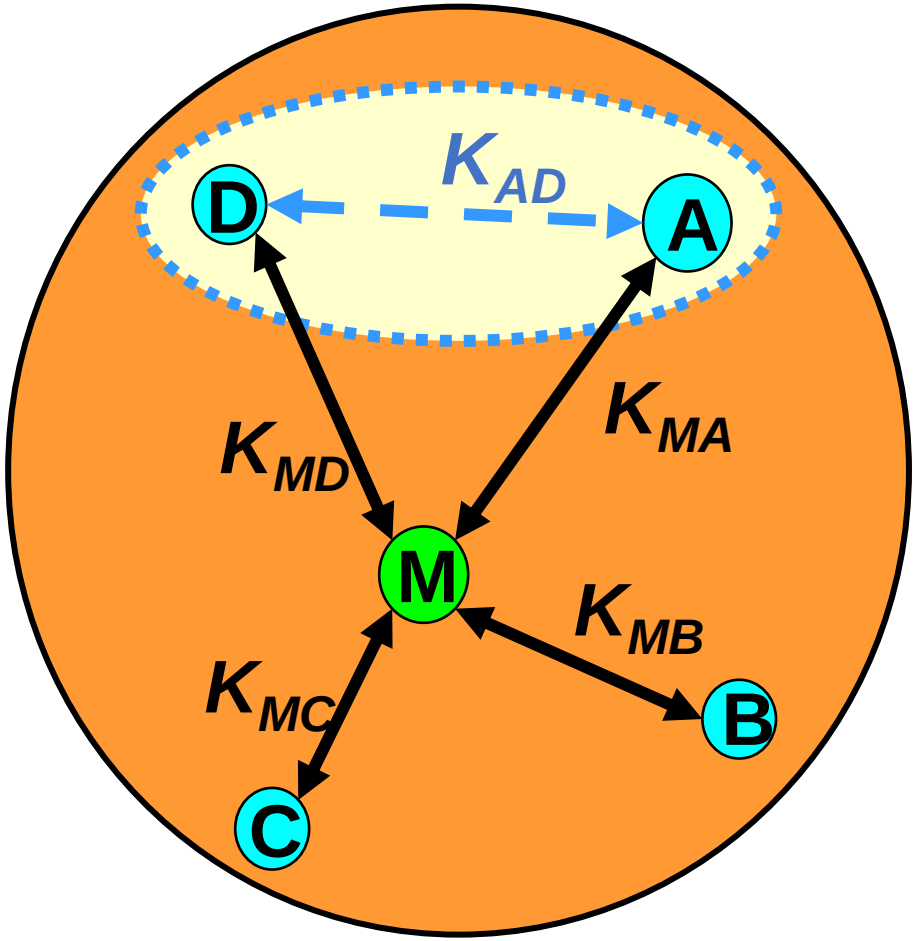
Pin is used when devices pair.
Size: 1 to 8 bytes

Mode 2 – encryption at the application/service layer

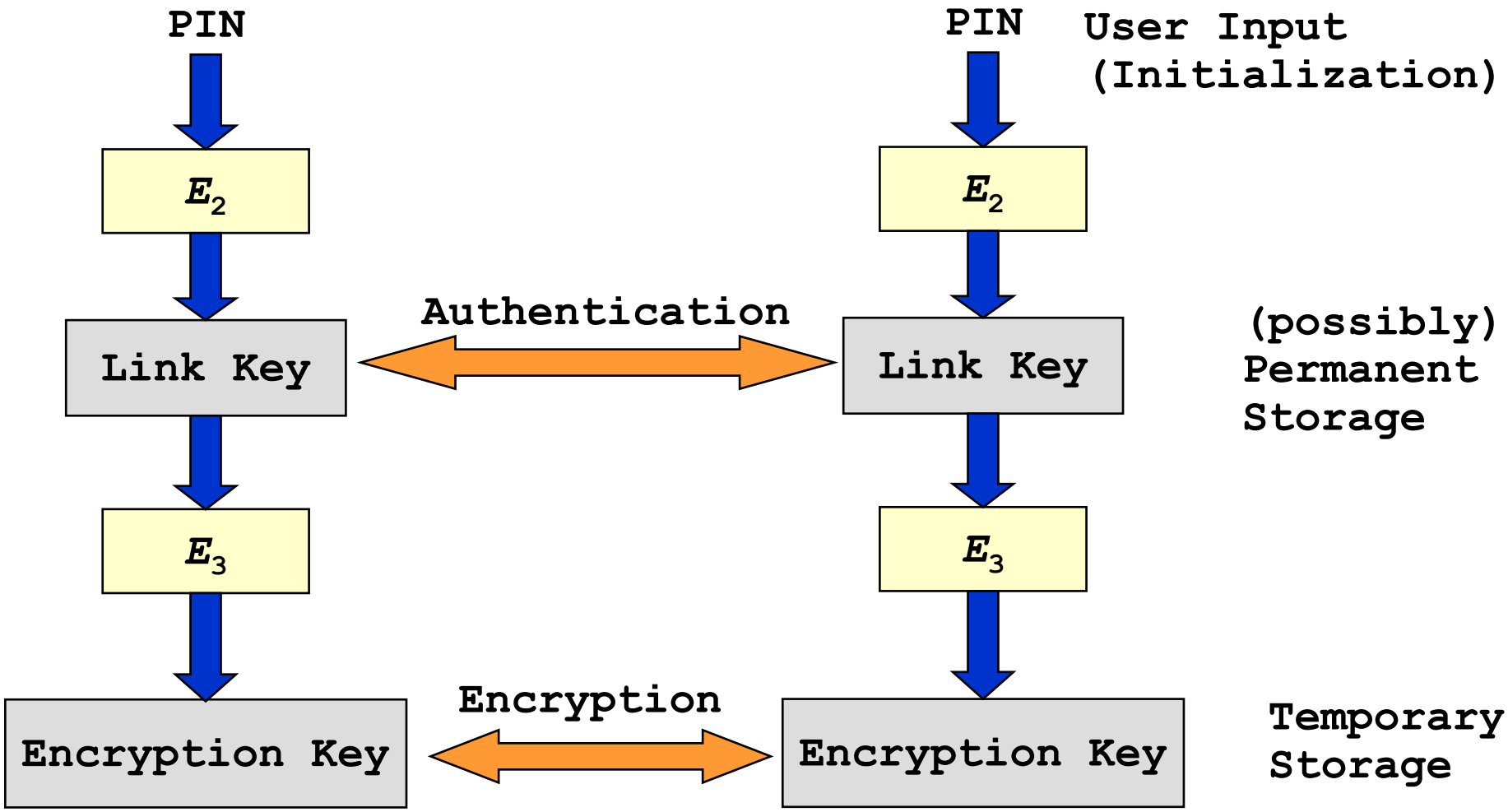
Mode 3 – encryption at the link layer

Link keys in a piconet

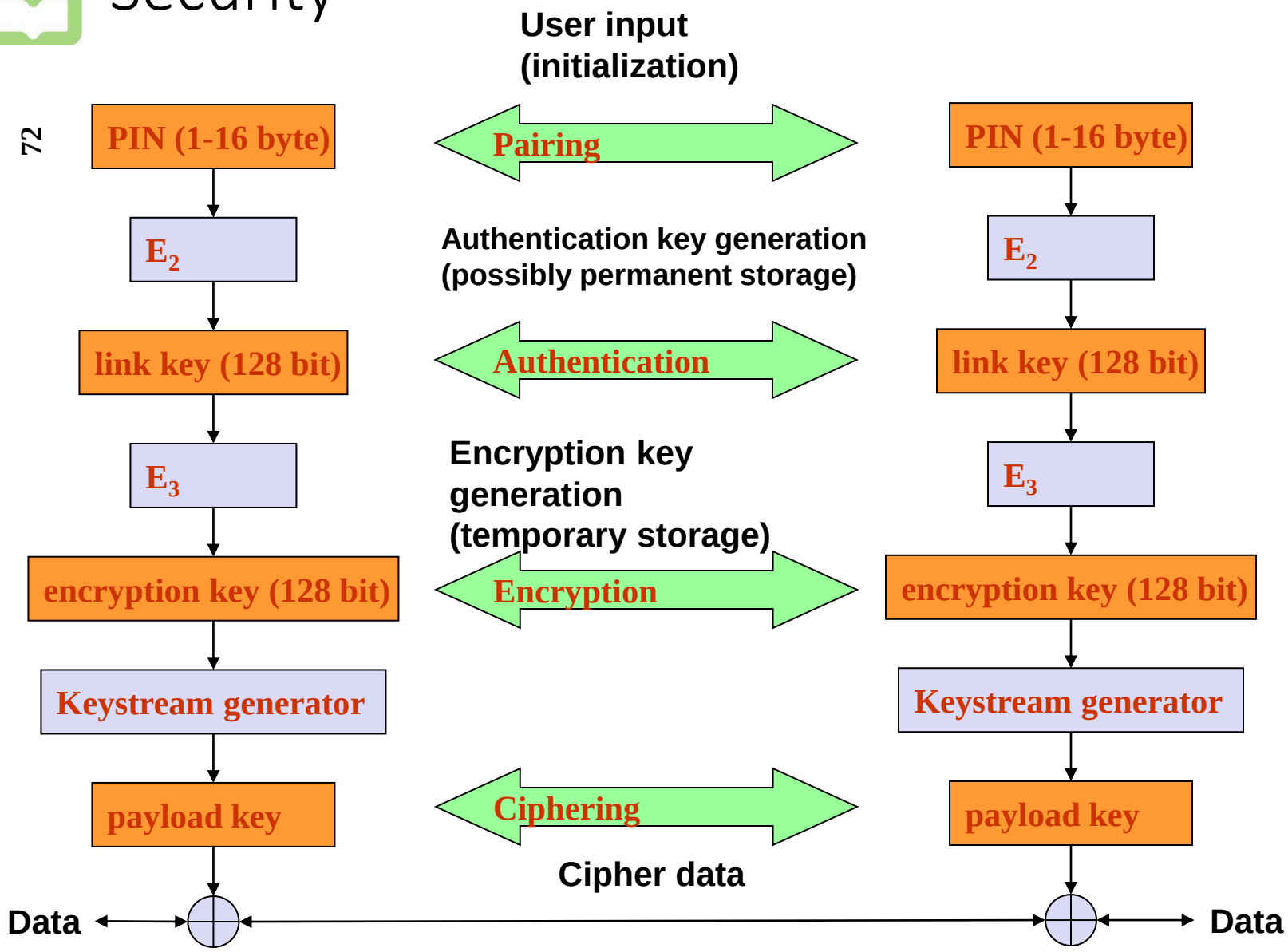
- Link keys are generated via a PIN entry
- A different link key for each pair of devices is allowed
- Authentication:
  - Challenge-Response Scheme
- Permanent storage of link keys



Key generation and usage



Security


User input
(initialization)
PIN (1-16 byte)
Pairing
PIN (1-16 byte)
E2
Authentication key generation
(possibly permanent storage)
E2
link key (128 bit)
Authentication
link key (128 bit)
E3
Encryption key
generation
(temporary storage)
E3
encryption key (128 bit)
Encryption
encryption key (128 bit)
Keystream generator
Keystream generator
payload key
Ciphering
payload key
Cipher data
Data
Data

Outline

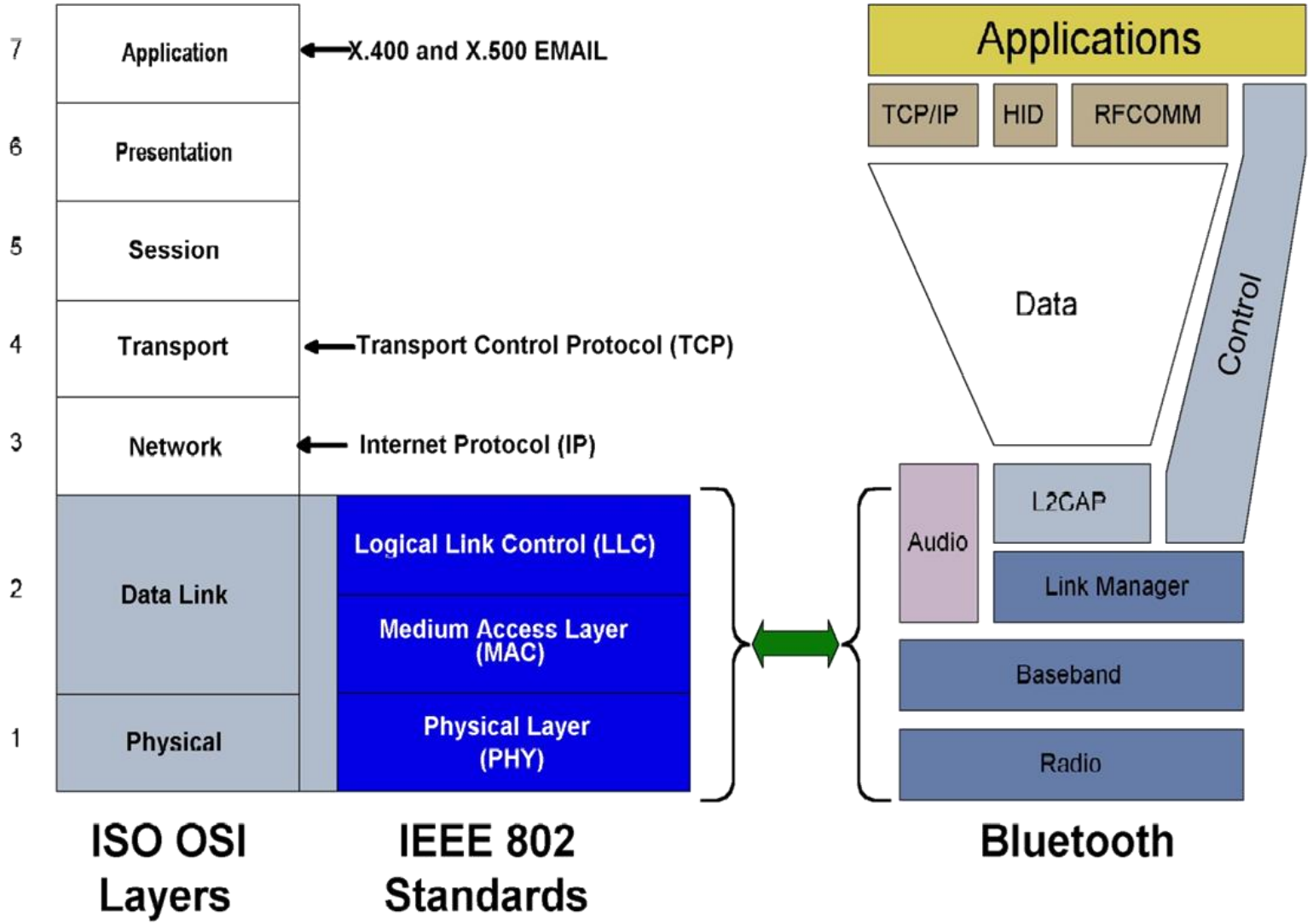
- Bluetooth networks
- Piconet operation
  - Inquiry
  - Paging
- Bluetooth stack
- Profiles and security
- 802.15.x

IEEE 802.15.1

- Adopted the Bluetooth MAC and PHY specifications
  - IEEE 802.15.1 and Bluetooth are almost identical regarding physical layer, baseband, link manager, logical link control and adaption protocol, and host control interface
- Range of up to 10 meters, uses FH-SS
- Data transfer rates of up to 1 Mbps
  - And higher for newer versions
- Not designed to carry heavy traffic loads
- Defines:
  - PAN Profile
  - PAN Testing profile
  - New stack layer

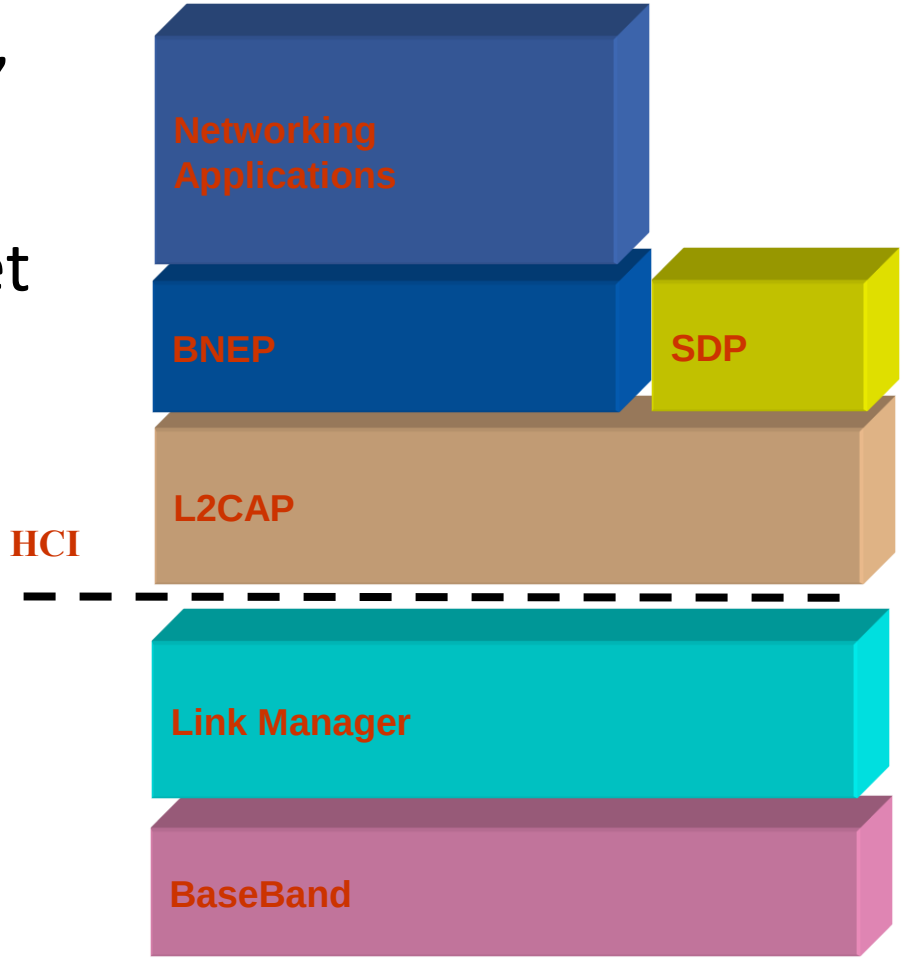
Bluetooth Network Encapsulation Protocol (BNEP)

IEEE 802.15 and Bluetooth SIG



Bluetooth Networking Encapsulation Protocol

- Provides an “Ethernet alike” environment
- Supports all type of Ethernet communications
- Deploys header compressions
- Implements packet filtering
- Implements an extension header



BNEP – Why?

- Supports all network protocols (IPV4, v6, e.g.)
  - LAN access from SIG (PPP) does not scale...
- Allows establishment of peer-to-peer, hiding the notion Master-Slave of Bluetooth
- All IETF protocols should work with 802.15.1 with BNEP
- Re-use all existing network applications
- Solution similar to those in other networks (802.11)
- Keep uniform bridge support inside IEEE
- **Cost**:
  - Small overhead, balanced by header compression

WiFi vs. Bluetooth

	Bluetooth	Wifi
Specifications authority	Bluetooth SIG	IEEE, WECA
Year of development	1994	1991
Bandwidth	Low (800 Kbps)	High (11 Mbps)
Hardware requirement	Bluetooth adaptor on all the devices connecting with each other	Wireless adaptors on all the devices of the network, a wireless router and/or wireless access points
Cost	Low	High
Power Consumption	Low	High
Frequency	2.4 GHz	2.4 GHz
Security	It is less secure	It is more secure
Range	10 meters	100 meters
Primary Devices	Mobile phones, mouse, keyboards,office and industrial automation devices	Notebook computers, desktopcomputers, servers
Ease of Use	Fairly simple to use. Can be used to connect upto seven devices at a time. It is easy to switch between devices or find and connect to any device.	It is more complex and requires configuration of hardware and software.

Bluetooth Spec Evolution (BT classic)

Specifications	1.1	1.2	2.0 + EDR	2.1 + EDR	3.0 +HS	4.0
Adopted	**2002**	**2005**	**2004**	**2007**	**2009**	**2010**
Transmission Rate	723.1 kbps	723.1 kbps	2.1 Mbps	3 Mbps	24 Mbps	25 Mbps
Standard PAN Range	10 m	10 m	10 m	10 m	10 m	50 m
Improved Pairing (without a PIN)				Yes	Yes	Yes
Improved Security		Yes	Yes	Yes	Yes	Yes
NFC Support			Yes	Yes	Yes	Yes
Voice Dialing	Yes	Yes	Yes	Yes	Yes	Yes
Call Mute	Yes	Yes	Yes	Yes	Yes	Yes
Last-Number Redial	Yes	Yes	Yes	Yes	Yes	Yes
Fast Transmission Speeds			Yes	Yes	Yes	Yes
Lower Power Consumption			Yes	Yes	Yes	Yes
Bluetooth Low Energy						Yes

Bluetooth 4.0: Low Energy

What are the USE CASES for BT 4.0?

- Proximity
- Time
- Emergency
- Network availability
- Personal User Interface
- Simple remote control
- Browse over Bluetooth
- Temperature Sensor
- Humidity Sensor
- HVAC
- Generic I/O (automation)
- Battery status
- Heart rate monitor
- Physical activity monitor
- Blood glucose monitor
- Cycling sensors
- Pulse Oximeter
- Body thermometer

Short range wireless application areas

	Voice	Data	Audio	Video	State
Bluetooth ACL/HS	x	Y	Y	x	x
Bluetooth SCO/eSCO	Y	x	x	x	x
Bluetooth low energy (BLE)	x	x	x	x	Y
Wi-Fi	(VoIP)	Y	Y	Y	x
Wi-Fi Direct	Y	Y	Y	x	x
ZigBee	x	x	x	x	Y

How much energy does traditional Bluetooth use?

- Traditional Bluetooth is *connection oriented*. When a device is connected, a link is maintained, even if there is no data flowing.

- Sniff modes allow devices to sleep, reducing power consumption to give months of battery life

- Peak transmit current is typically around 25mA

- Even though it has been independently shown to be lower power than other radio standards, it is still not low enough power for coin cells and energy harvesting applications

What is Bluetooth Low Energy?

- Bluetooth low energy is a open, short range radio technology
  - Blank sheet of paper design
  - Different to Bluetooth classic (BR/EDR)
  - Optimized for ultra low power
  - Enable coin cell battery use cases
    - < 20mA peak current
    - < 5 uA average current

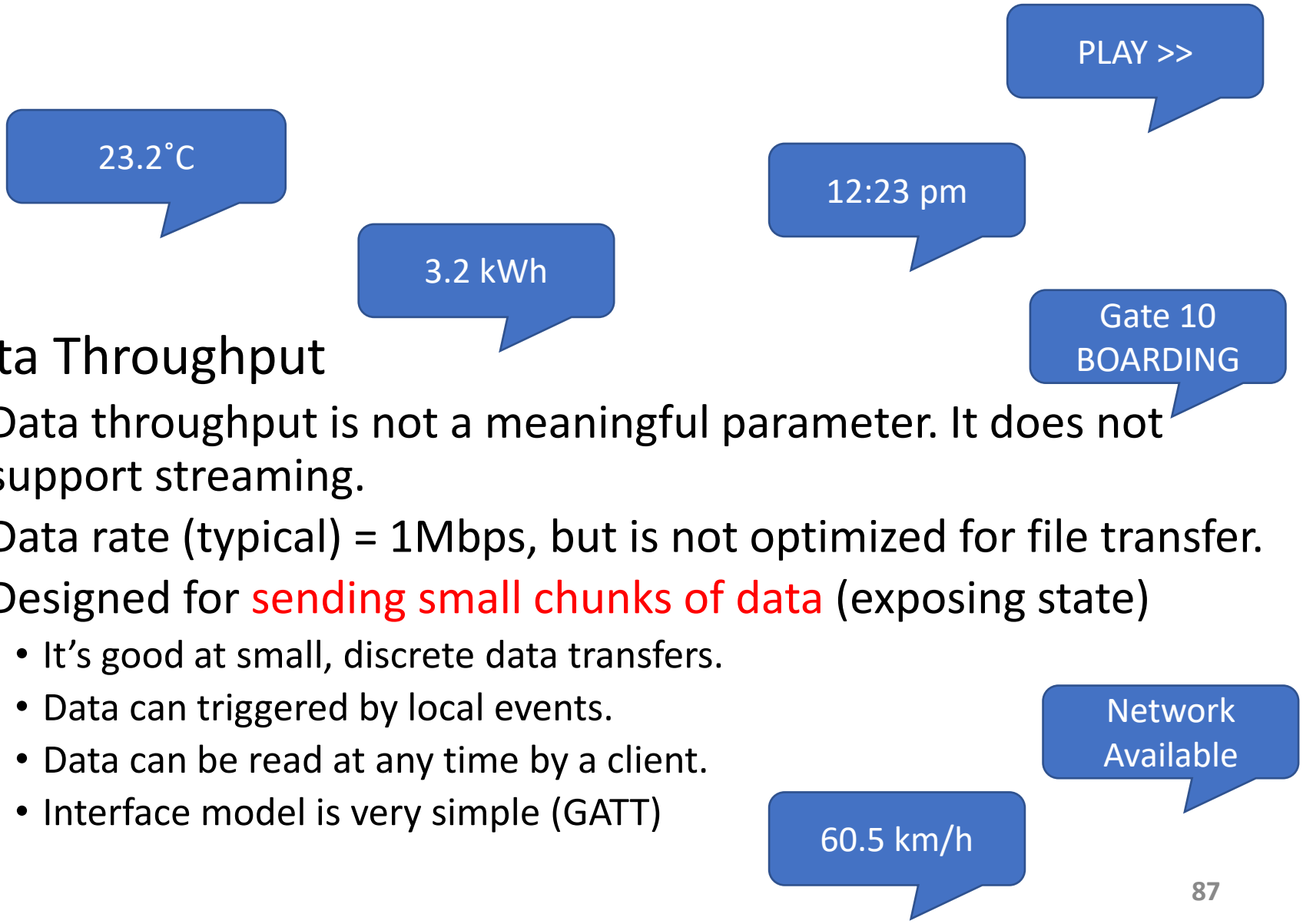
Basic Concepts of Bluetooth 4.0

- Everything is optimized for lowest power consumption
  - Short packets reduce TX peak current
  - Short packets reduce RX time
  - Less RF channels to improve discovery and connection time
  - Simple state machine
  - Single protocol
  - Etc.

Bluetooth low energy factsheet

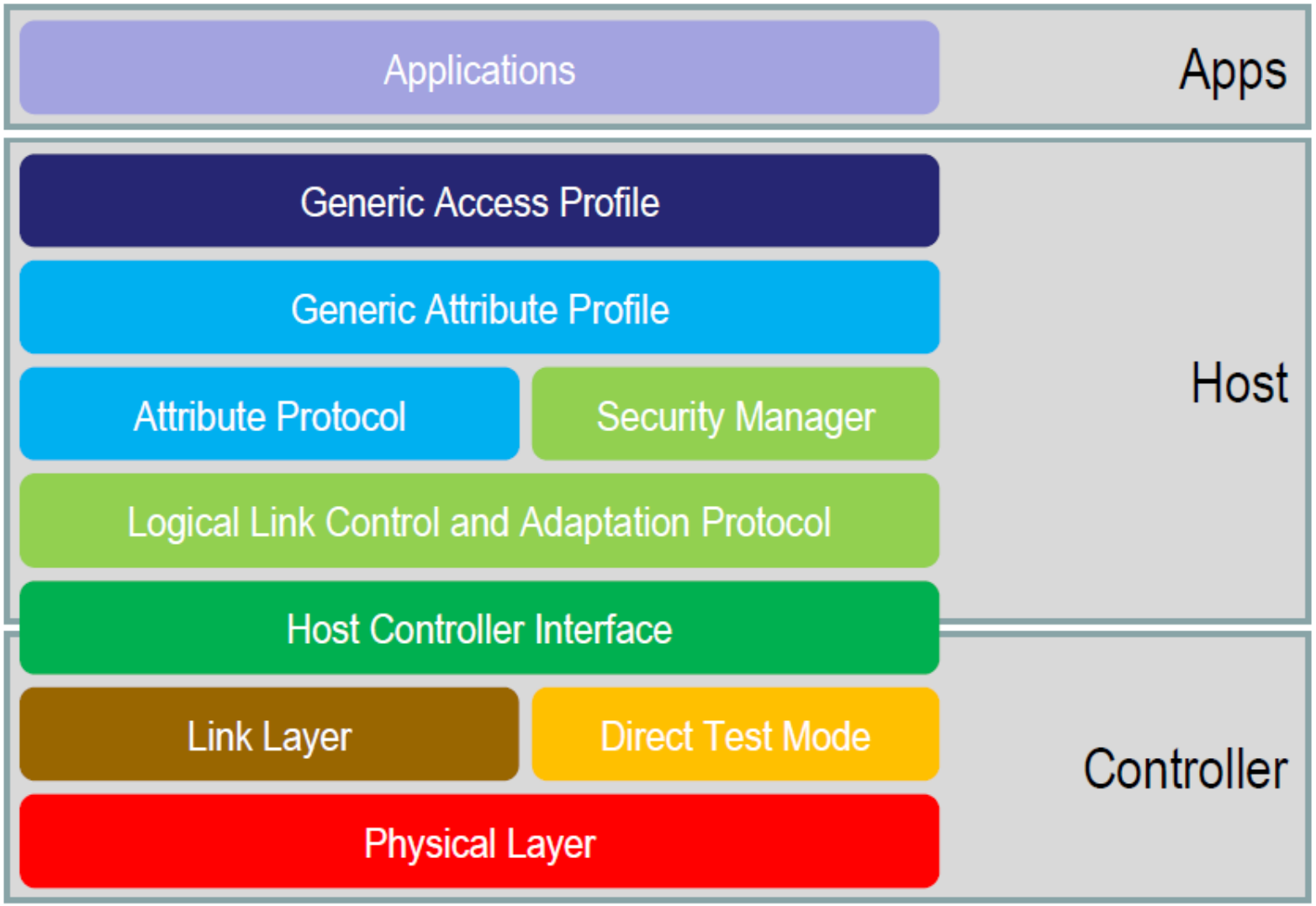
Range:	**~ 150 meters open field**
Output Power:	**~ 10 mW (10dBm)**
Max Current:	**~ 15 mA**
Latency:	**3 ms**
Topology:	**Star**
Connections:	**> 2 billion**
Modulation:	**GFSK @ 2.4 GHz**
Robustness:	**Adaptive Frequency Hopping, 24 bit CRC**
Security:	**128bit AES CCM**
Sleep current:	**~ 1µA**
Modes:	**Broadcast, Connection, Event Data Models, Reads, Writes**

Designed for exposing state



- Data Throughput
  - Data throughput is not a meaningful parameter. It does not support streaming.
  - Data rate (typical) = 1Mbps, but is not optimized for file transfer.
  - Designed for sending small chunks of data (exposing state)
    - It's good at small, discrete data transfers.
    - Data can triggered by local events.
    - Data can be read at any time by a client.
    - Interface model is very simple (GATT)

Bluetooth Low Energy Architecture



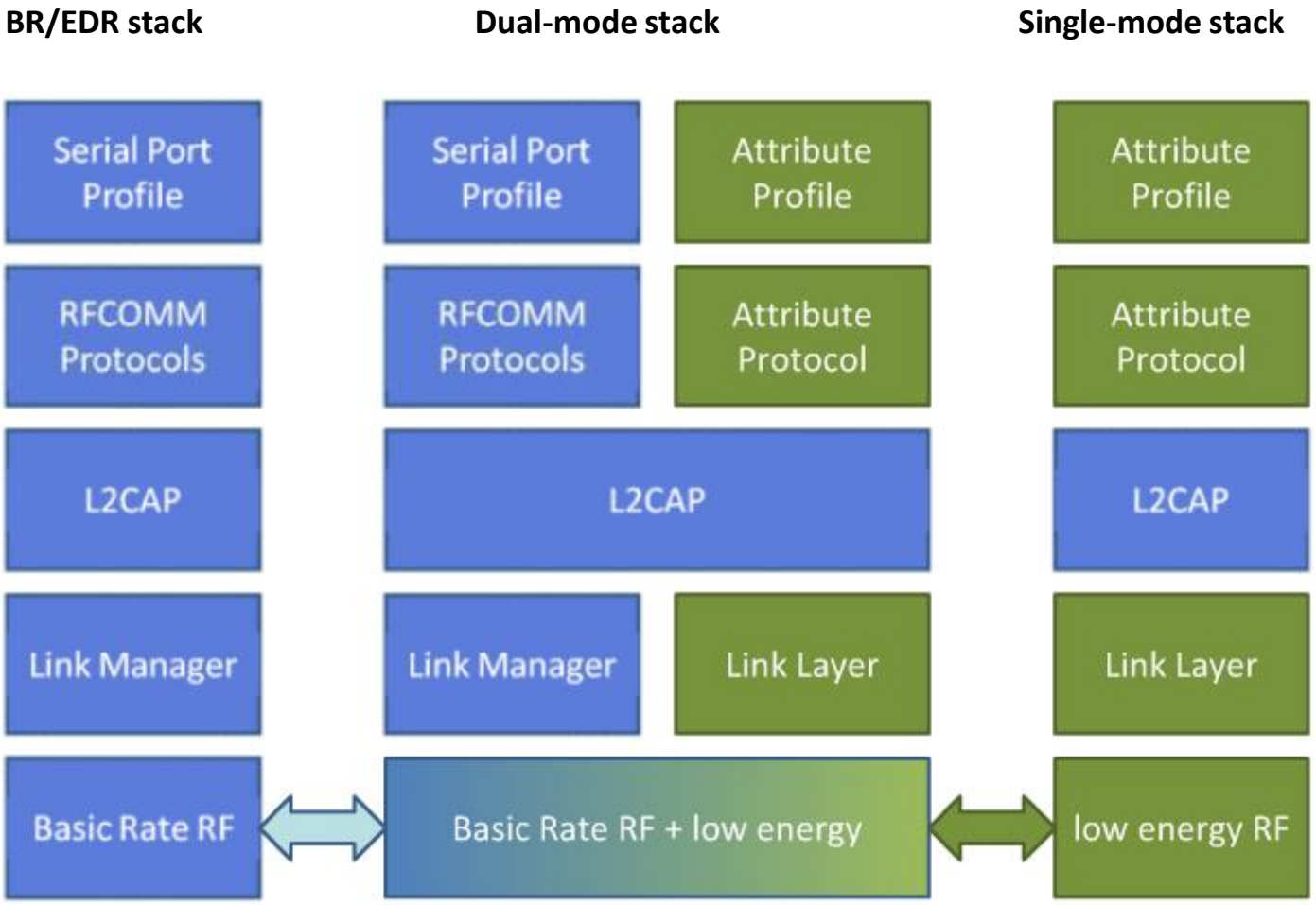
Device Modes

- Dual Mode
  - Bluetooth BR/EDR and LE
    - BR – Basic Rate
    - EDR – Enhanced Data Rate
  - Used anywhere that BR/EDR is used today

- Single Mode
  - Implements only Bluetooth low energy
  - Will be used in new devices / applications

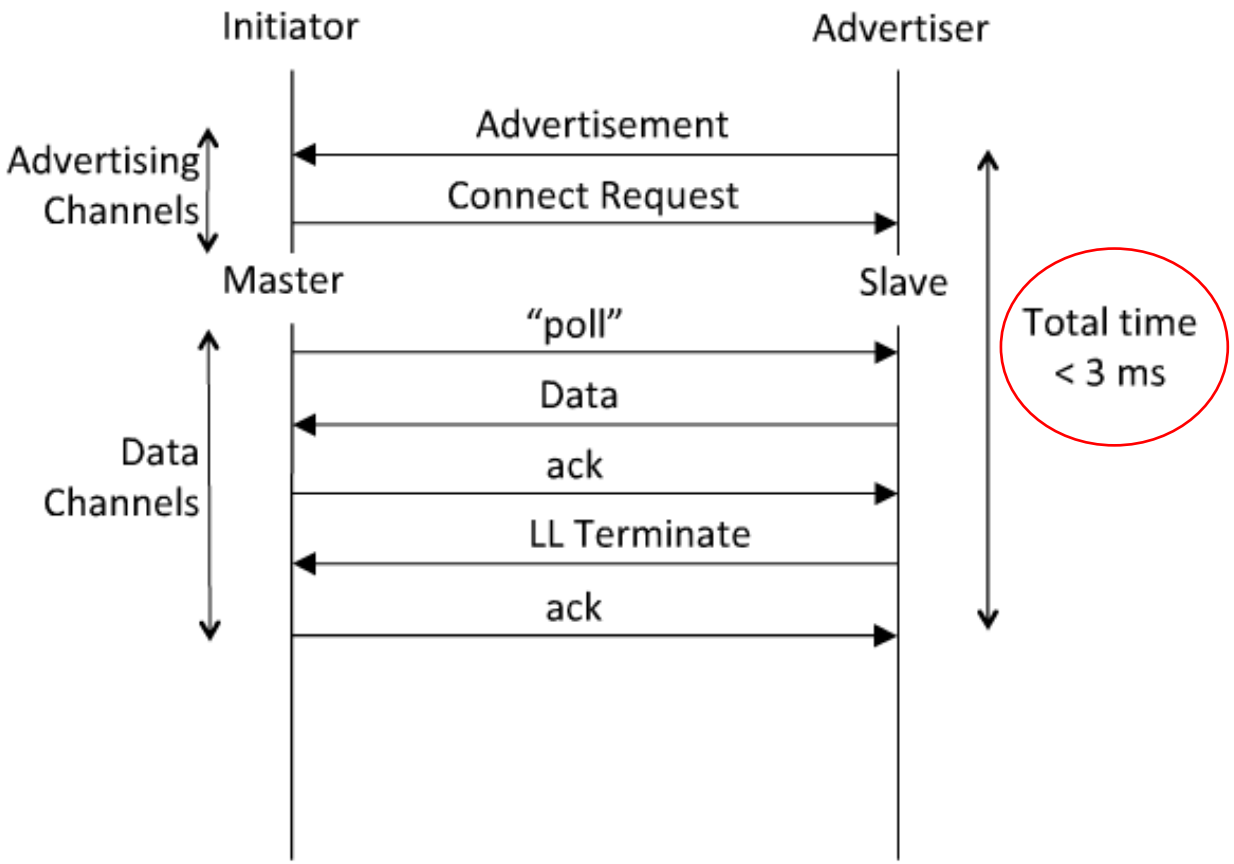
Device Modes

- Dual mode + single modes



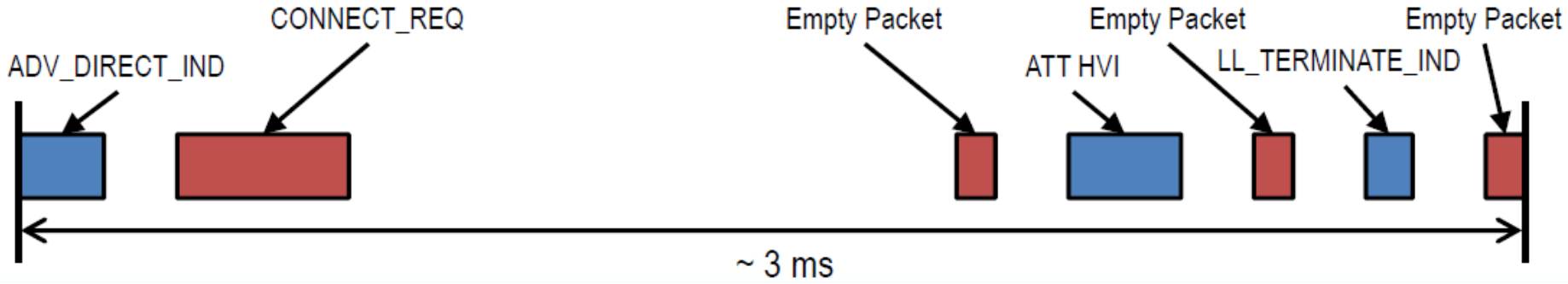
Link Layer Connection

- Very low latency connection



Time From Disconnected to Data ~ 3ms

Time (us)	Master Tx	Radio Active (us)	Slave Tx
0		176	ADV_DIRECT_IND
326	CONNECT_REQ	352	
1928	Empty Packet	80	
2158		144	Attribute Protocol Handle Value Indication
2452	Empty Packet (Acknowledgement)	80	
2682		96	LL_TERMINATE_IND
2928	Empty Packet (Acknowledgement)	80	



How low can the energy get?

- From the previous slide, calculate energy per transaction
  - Assume an upper bound of 3ms per minimal transaction
  - Estimated TX power is 15mW (mostly TX power amp for 65nm chips)
  - For 1.5v battery, this is 10mA. 0.015W * 0.003 sec = 45 micro Joule
- How long could a sensor last on a battery?
  - An example battery: Lenmar WC357, 1.55v, 180mAh, $2-5
  - 180mAh/10mA = 18Hr = 64,800 seconds = 21.6M transactions
  - Suppose this sensor sends a report every minute = 1440/day
  - For just the BT LE transactions, this is 15,000 days, or > 40 years
  - This far exceeds the life of the battery and/or the product
- This means that battery will cost more than the electronics
  - This sensor could run on scavenged power, e.g. ambient light

BLE and GAP

- Generic Access Profile (GAP)
  - GAP defines a base profile which all Bluetooth devices implement, which ties all the various layers together to form the basic requirements for a Bluetooth device
  - GAP also defines generic procedures for connection-related services:
    - Device Discovery
    - Link Establishment
    - Link Management
    - Link Termination
    - Initiation of security features

BLE and GAP

- The GAP layer works in one of **four** profile roles:
  - **Broadcaster**: an advertiser that is non-connectable
  - **Observer**: scans for advertisements, but cannot initiate connections
  - **Peripheral**: an advertiser that is connectable and can operate as a slave in a single link layer connection
  - **Central**: scans for advertisements and initiates connections; operates as a master in a single or multiple link layer connections

BLE and GAP

Temperature Sensor (Broadcaster) → Temperature Display (Observer)

Figure 1 – Temperature Sensor (Broadcaster)

Figure 2 – Temperature Display (Observer)

Watch (Peripheral) ← → Mobile Phone (Central)

Figure 3 - Watch (Peripheral)

Figure 4 – Mobile Phone (Central)

BLE and GAP – Discoverable Modes

- GAP supports **three** different discoverable modes:
  - **Non-discoverable Mode**: No advertisements
  - **Limited Discoverable Mode**: Device advertises for a limited amount of time before returning to the standby state
  - **General Discoverable Mode**: Devices advertises continuously
- GAP manages the data that is sent out in advertisement and scan response packets

BLE and GAP - Pairing

- Pairing can be initiated by either the central or peripheral device
- The two devices generate and exchange short-term keys (STK) which can be used to decrypt data packets
- Either device can request to enable “bonding” to create a long-term relationship between the two devices
  - A long-term key (LTK) is generated, exchanged, and stored allowing device to re-encrypt the link quickly upon re-connection, **without going through the complete pairing process once again**
  - Profile / Service configuration data is remembered, so that the user does not need to re-configure the device every time they re- connect

BLE and GAP - Pairing

- Each device also states its input/output capabilities from among these options:
  - **DisplayOnly** – no way user can input anything into device, but it can output data
  - **DisplayYesNo** – user can input “yes” or “no” but nothing else; can also display data
  - **KeyboardOnl**y – user can input a password or PIN, but no display
  - **NoInputNoOutput** – device has no means for user input, and has no display
  - **KeyboardDisplay** – device has a means for display as well as for input