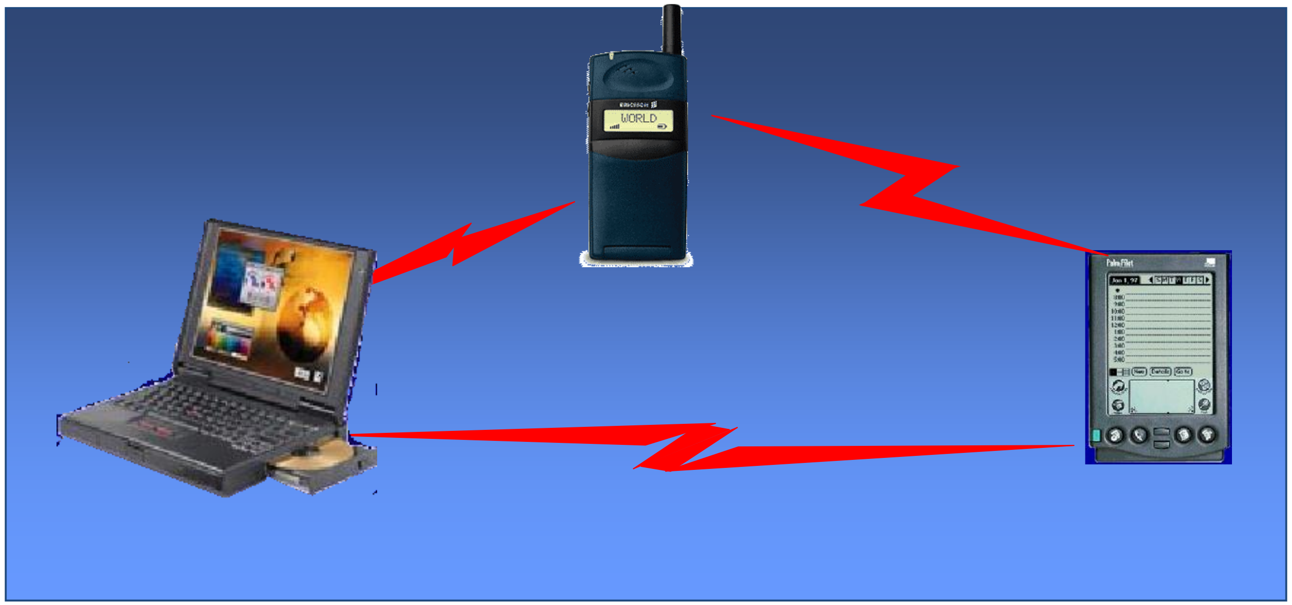
Bluetooth:
Short-range Wireless Communication

- Wide variety of handheld devices
  - Smartphone, palmtop, laptop
- Need compatible data communication interface
    - Complicated cable/config. problem
- Short range wireless comm
    - On demand connectivity
- Inexpensive , application friendly, adopted by vendors

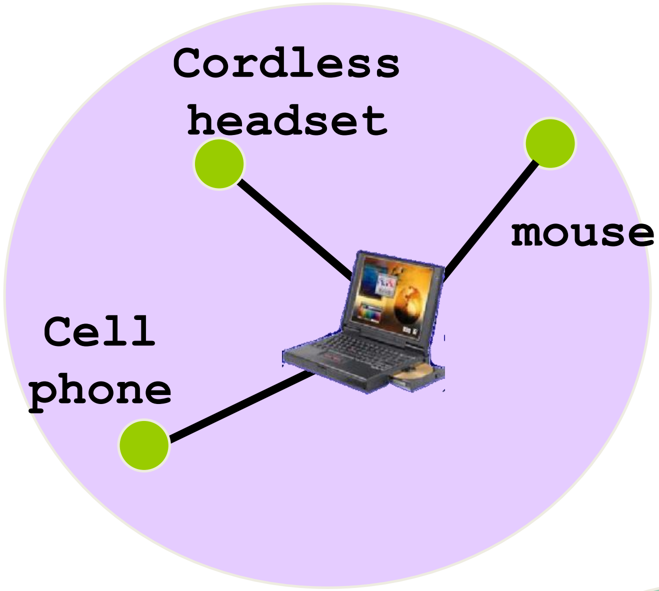
Bluetooth



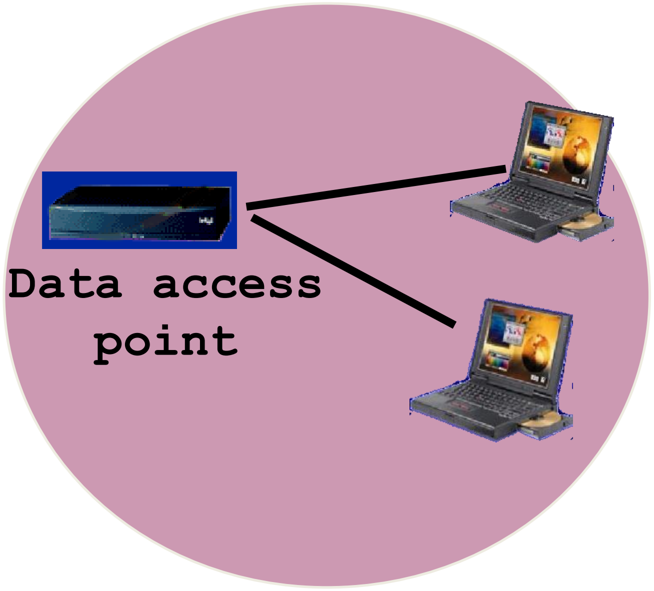
- A cable replacement technology
- 1 Mb/s rate
- Range 10+ meters
- Single chip radio + baseband
  - at low power & low price point

Why not use Wireless LANs?
- power
- cost

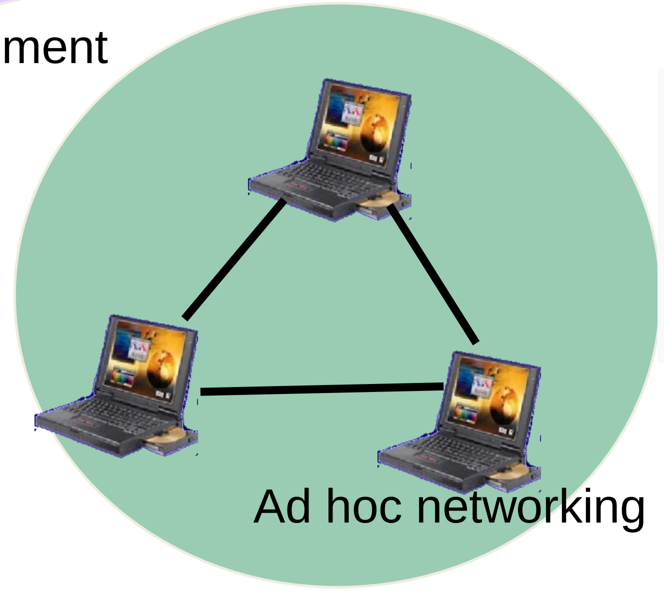
Value proposition of Bluetooth



Cable replacement



Internet access



Ad hoc networking

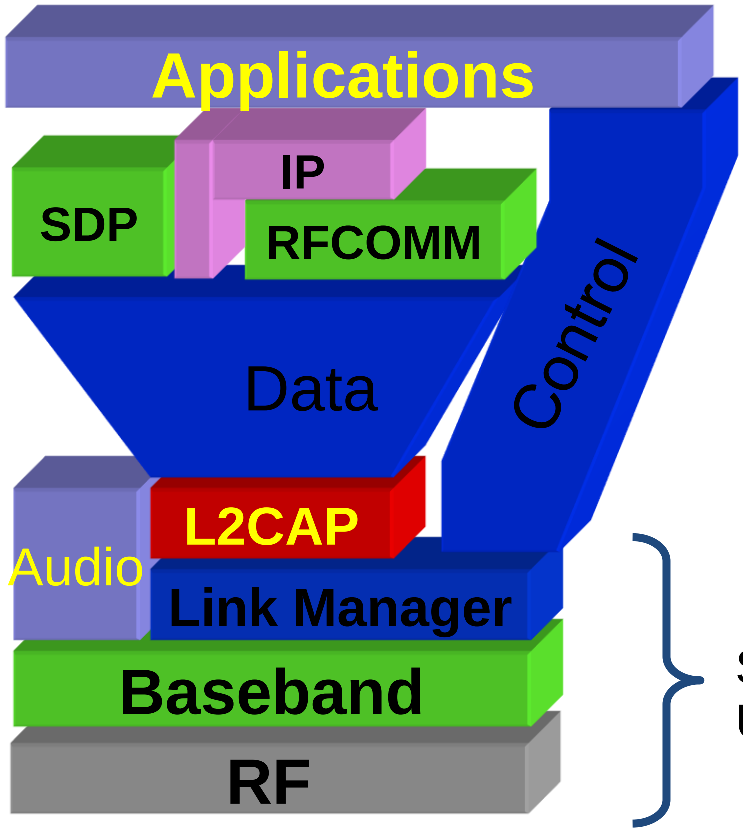
Bluetooth: Initial Days

- February 1998: The Bluetooth SIG is formed
  - promoter company group: Ericsson, IBM, Intel, Nokia, Toshiba
- License free technology
  - Universal wireless connectivity
- May 1998: The Bluetooth SIG goes "public"
- July 1999: 1.0A spec (>1,500 pages) is published
- December 1999: ver. 1.0B is released
- …

- Defines RF wireless communication interface
  - Communication protocols
  - Usage profile
- Link speed, communication range, power level is chosen
  - to support low cost, power efficient, single chip implementation
- Makes single chip radio that works in 2.4 GHz RF band

Bluetooth Specifications

Core spec- Bluetooth protocol Stack



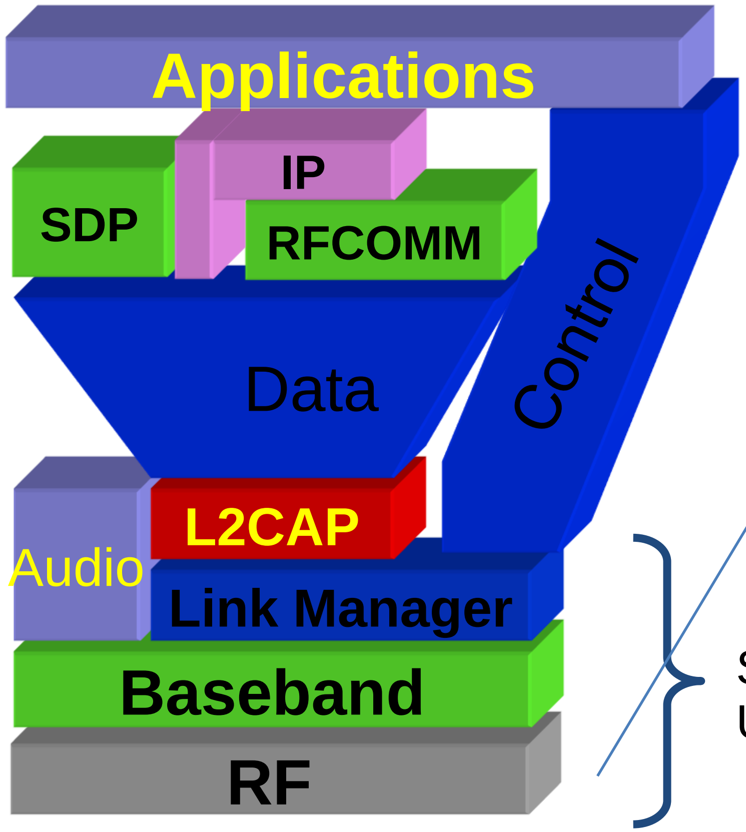
Differs from seven layer model

Supports ad hoc connectivity
Power/resource efficient

Single chip with RS-232, USB, or PC card interface

- A hardware/software/protocol description
- An application framework

Core spec- Bluetooth protocol Stack

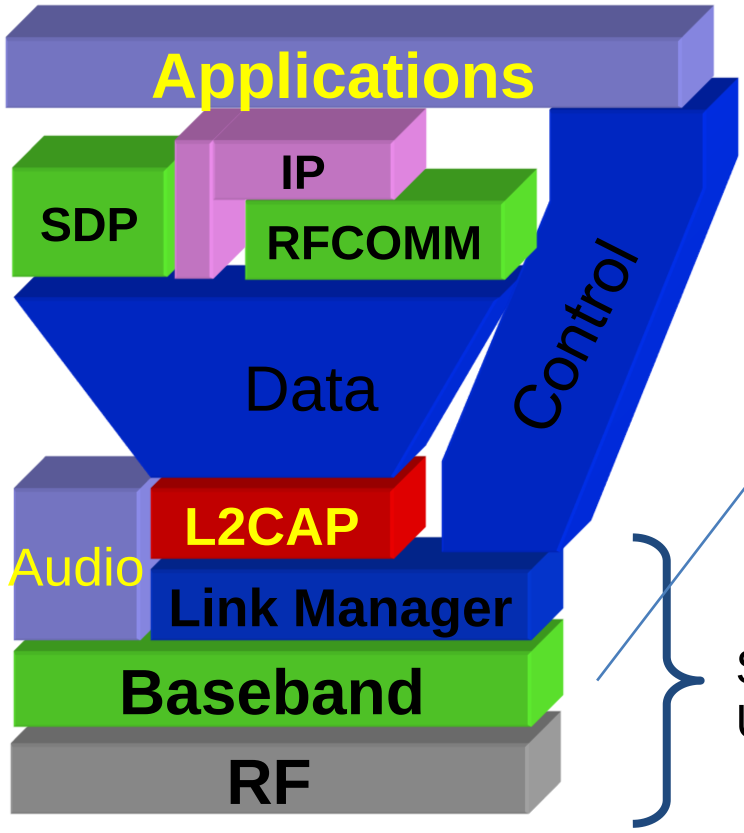


Specifies radio front end, Frequency band, channel, transmit power level

Single chip with RS-232, USB, or PC card interface

- A hardware/software/protocol description
- An application framework

Core spec- Bluetooth protocol Stack

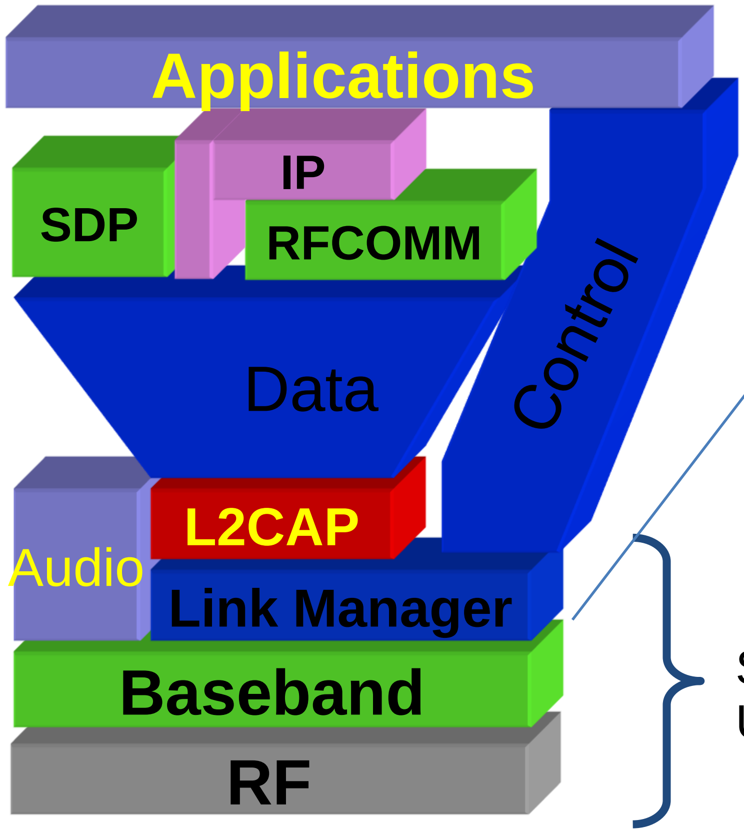


PHY and MAC layer
Device discovery, link
Formation, communication

Single chip with RS-232,
USB, or PC card interface

- A hardware/software/protocol description
- An application framework

Core spec- Bluetooth protocol Stack

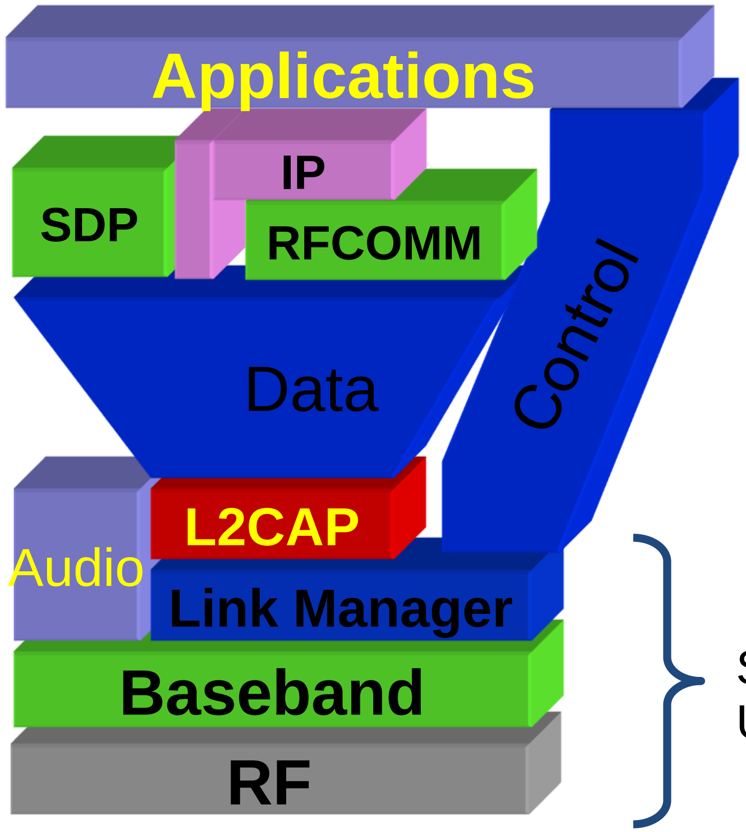


Control messages for Config., managing connections

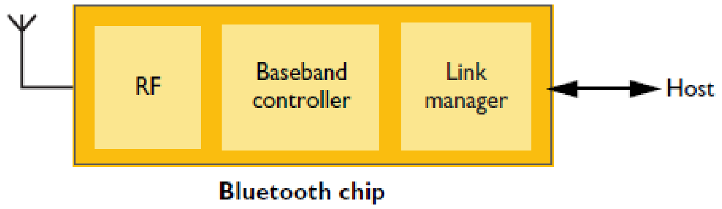
Single chip with RS-232, USB, or PC card interface

- A hardware/software/protocol description
- An application framework

Core spec- Bluetooth protocol Stack



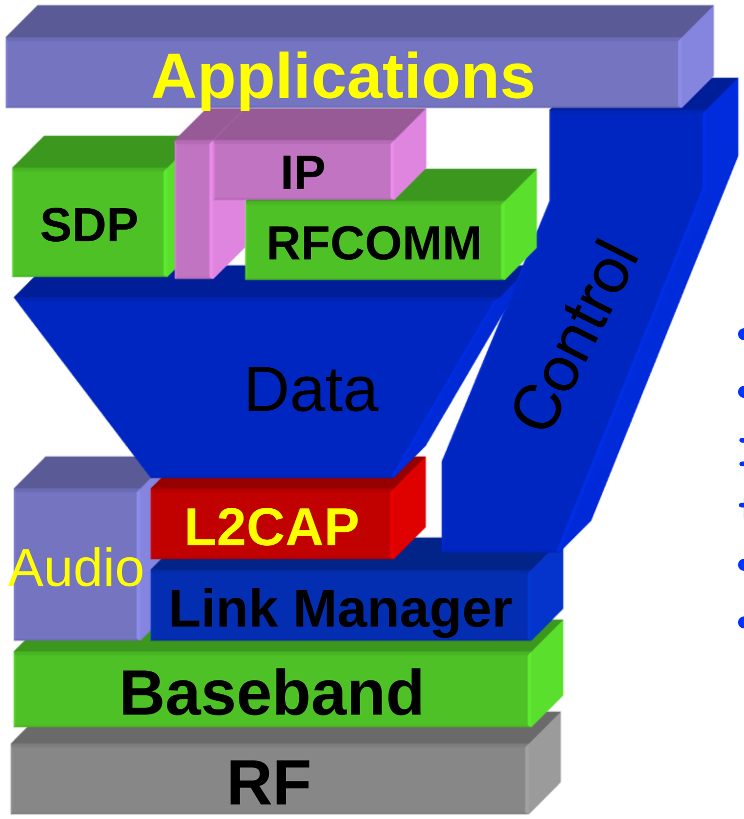
Lowers cost,
Easy to embed



Single chip with RS-232, USB, or PC card interface

- A hardware/software/protocol description
- An application framework

Core spec- Bluetooth protocol Stack



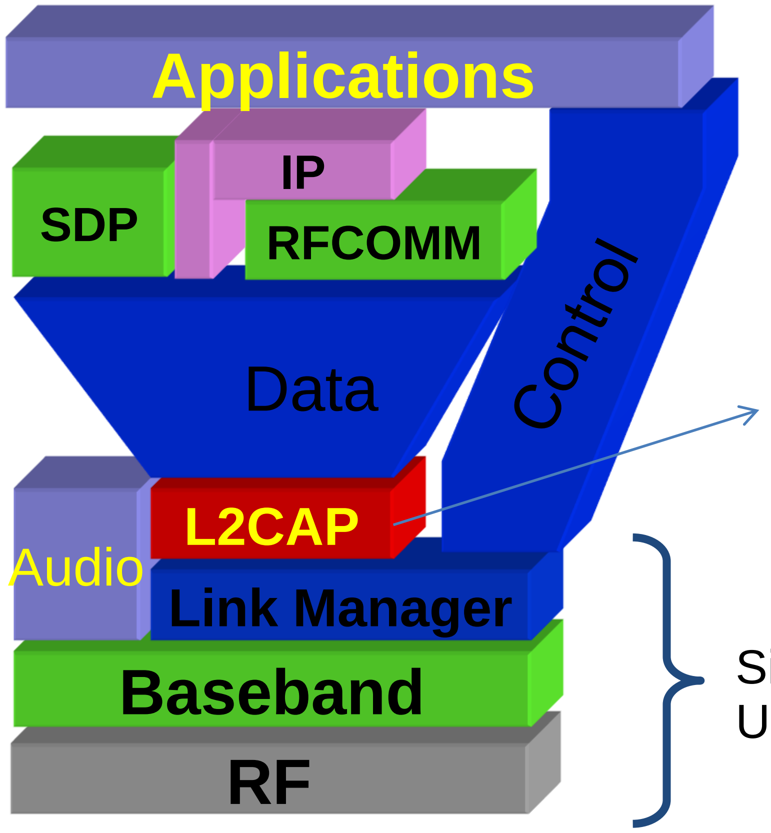
- Host controller interface
- Defines

Interface independent method to communicate with BT chip

- Use HCI commands
- No device-to-device commn.

- A hardware/software/protocol description
- An application framework

Core spec- Bluetooth protocol Stack



Logical link control,
Adaptation layer

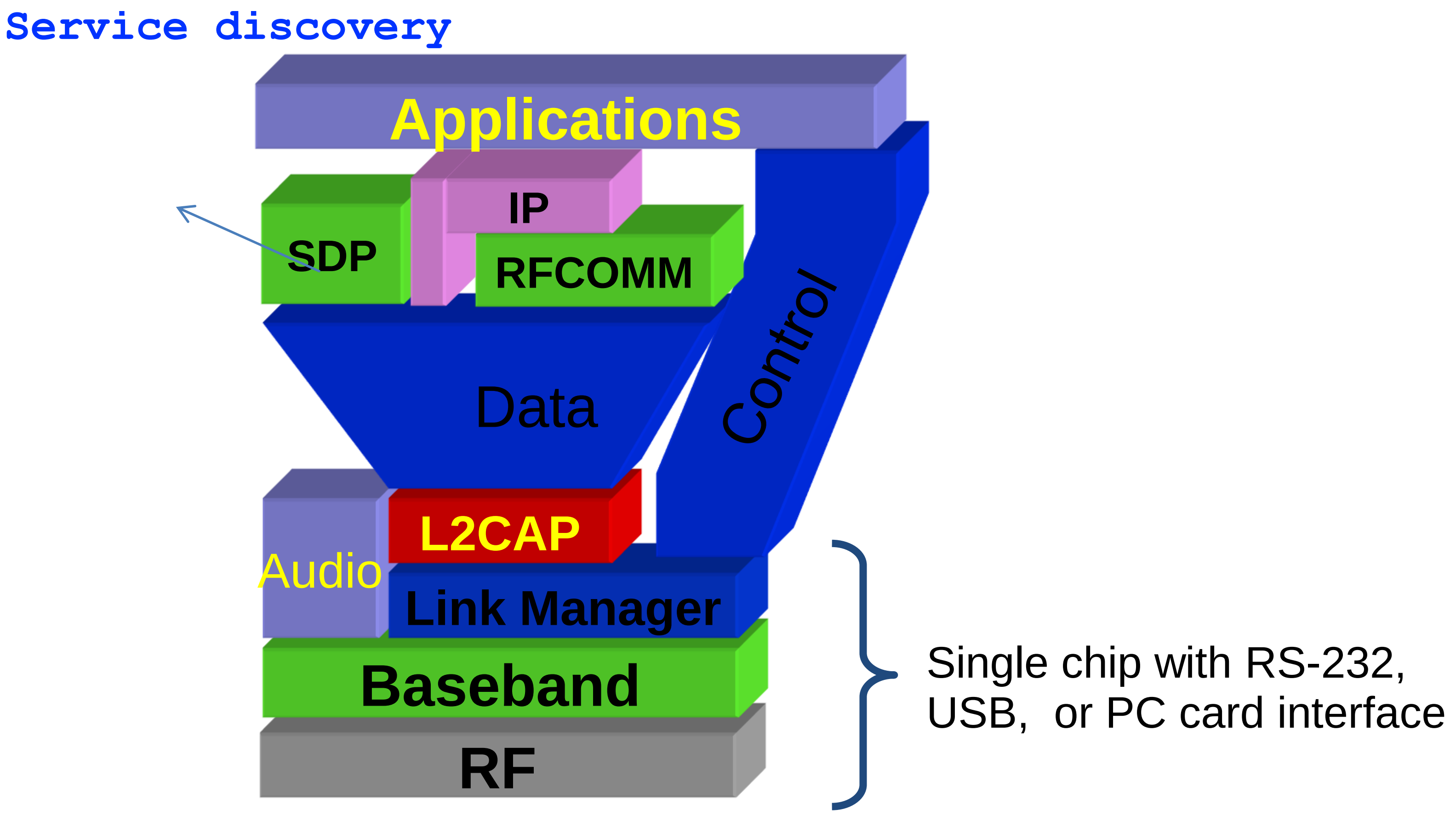
Data link layer

Delivers packet to other end

Single chip with RS-232,
USB, or PC card interface

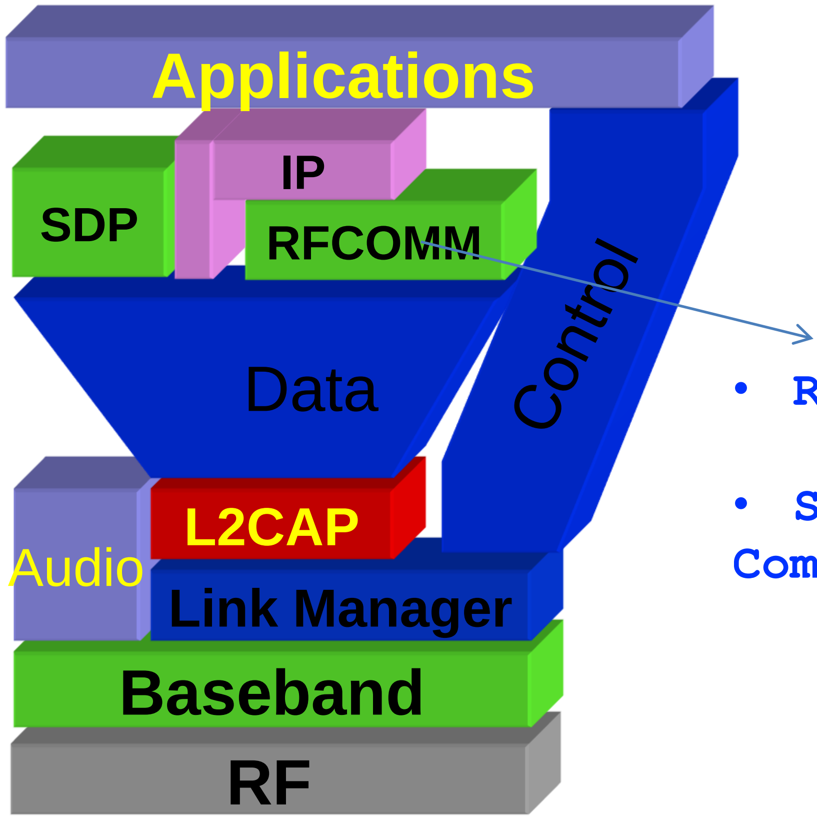
- A hardware/software/protocol description
- An application framework

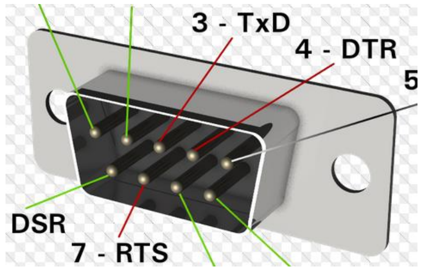
Core spec- Bluetooth protocol Stack



- A hardware/software/protocol description
- An application framework

Core spec- Bluetooth protocol Stack



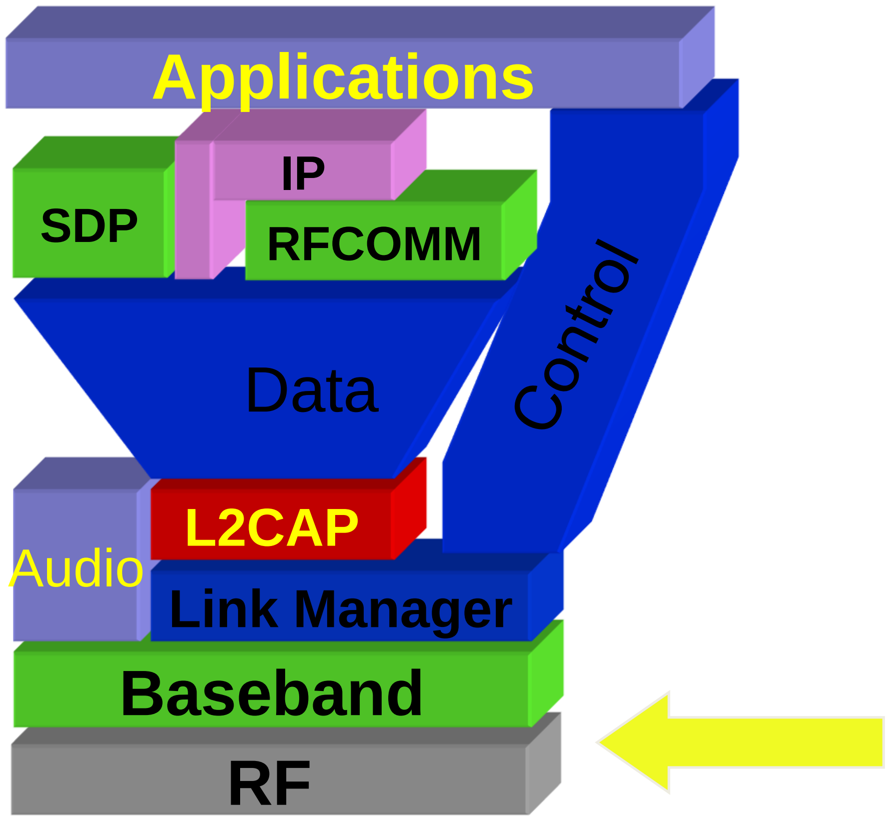


- RS-232 COM port interface
- Supports legacy of COM port Communication

- Emulate RS-232 cable conn. on top of BT
- Supports classical applications---PPP

Technical Overview

Bluetooth Radio Specification

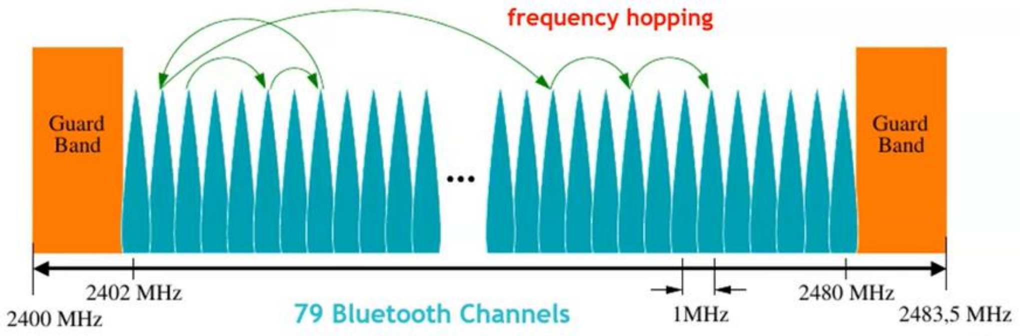


2.4 GHz band --- licence free use

83.5 MHz is allocated

79 channels

Link speed 1Mbps

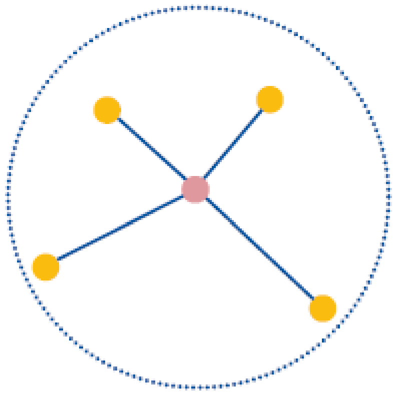


Bluetooth Radio

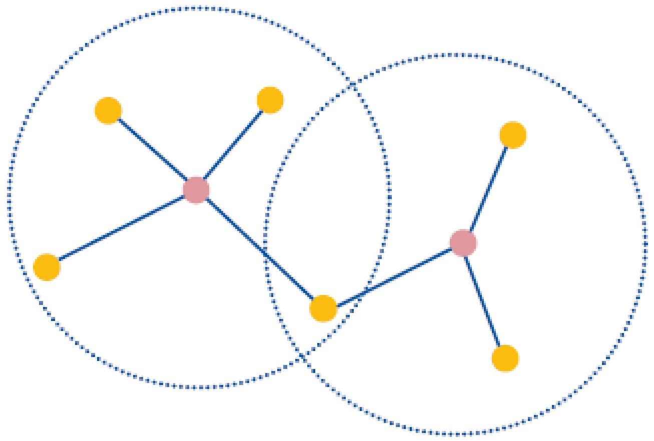
- Uses 2.4 GHz band spread spectrum radio (2400 – 2483.5 MHz)
- Advantages
  - Free
  - Open to everyone worldwide
- Disadvantages
  - Can be noisy (microwaves, cordless phones, garage door openers)

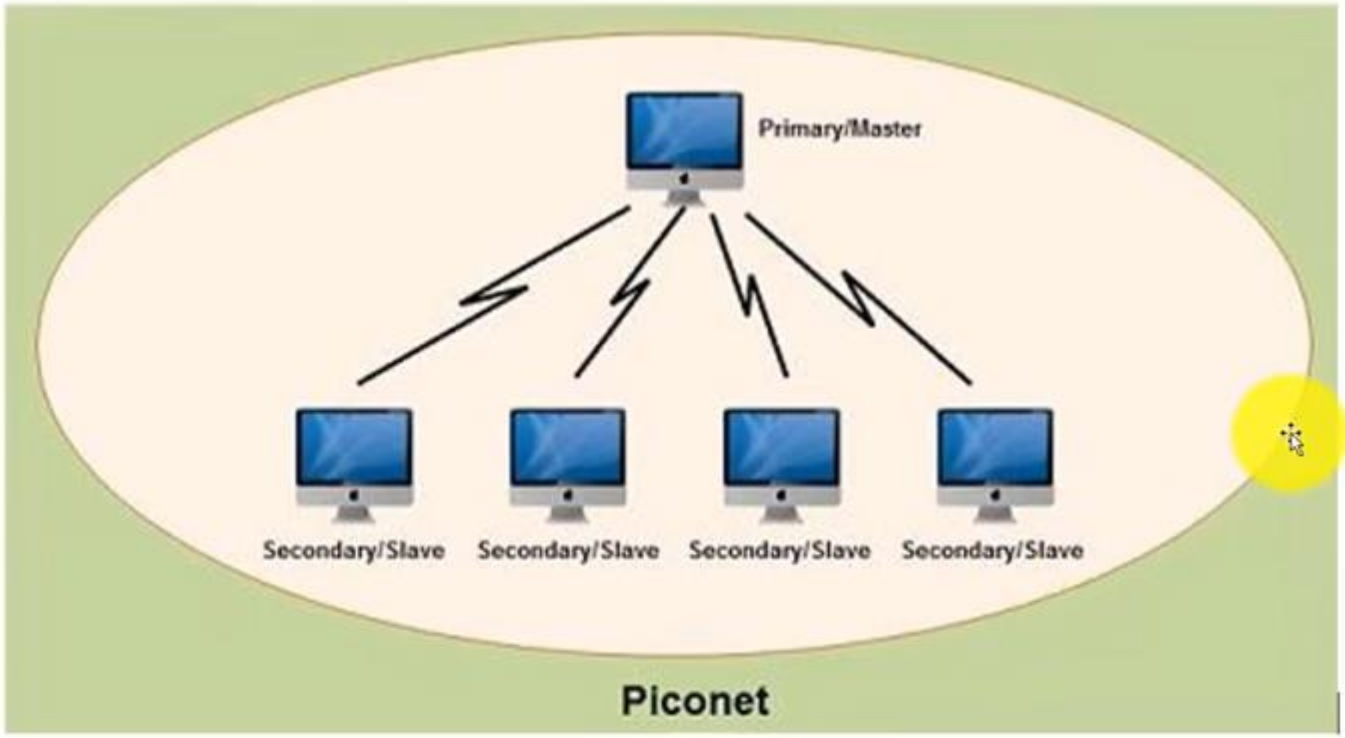
Piconet and Scatternet

- Piconet
  - Set of BT devices sharing common channel
    - One master, up to seven slaves
    - Can serve more than 7
      - Switch to low park

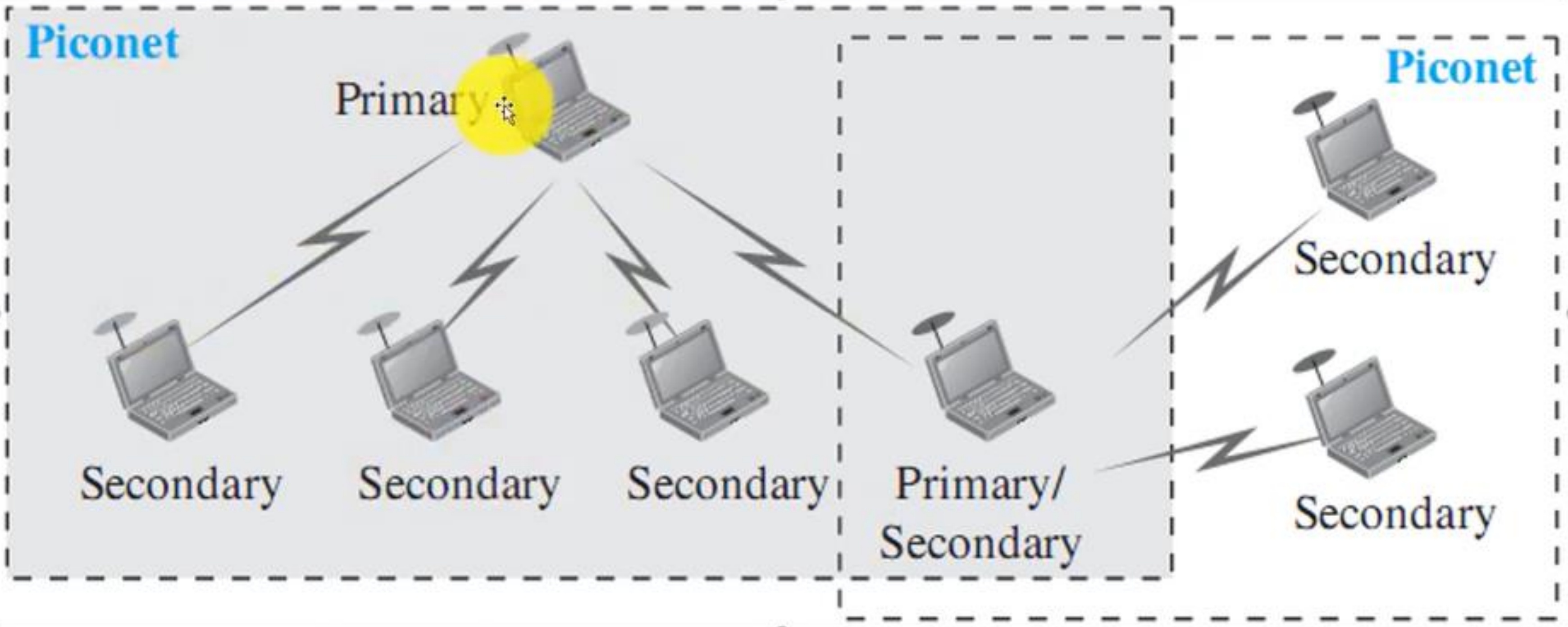

Piconet

- Scatternet
    - Bridge nodes
    - Time share
    - Interconnection of multiple piconets


Scatternet


Primary/Master
Secondary/Slave
Secondary/Slave
Secondary/Slave
Secondary/Slave
Piconet

Scatternet:


Piconet
Primary
Secondary
Secondary
Secondary
Primary/
Secondary
Piconet
Secondary
Secondary

Low power modes

- Different low power modes for improving battery life
- Piconets are formed on demand when devices are ready for communication
- All other times, devices can be turned off
- Three kinds of low power modes are supported
- Hold: device should be put to sleep for a specified time duration------------master searches for new node
- Sniff: put slave in low duty cycle mode --- wake up periodically
- Park: Similar to sniff-----stay synchronized with master without being an active member
  - Admits more than seven slaves

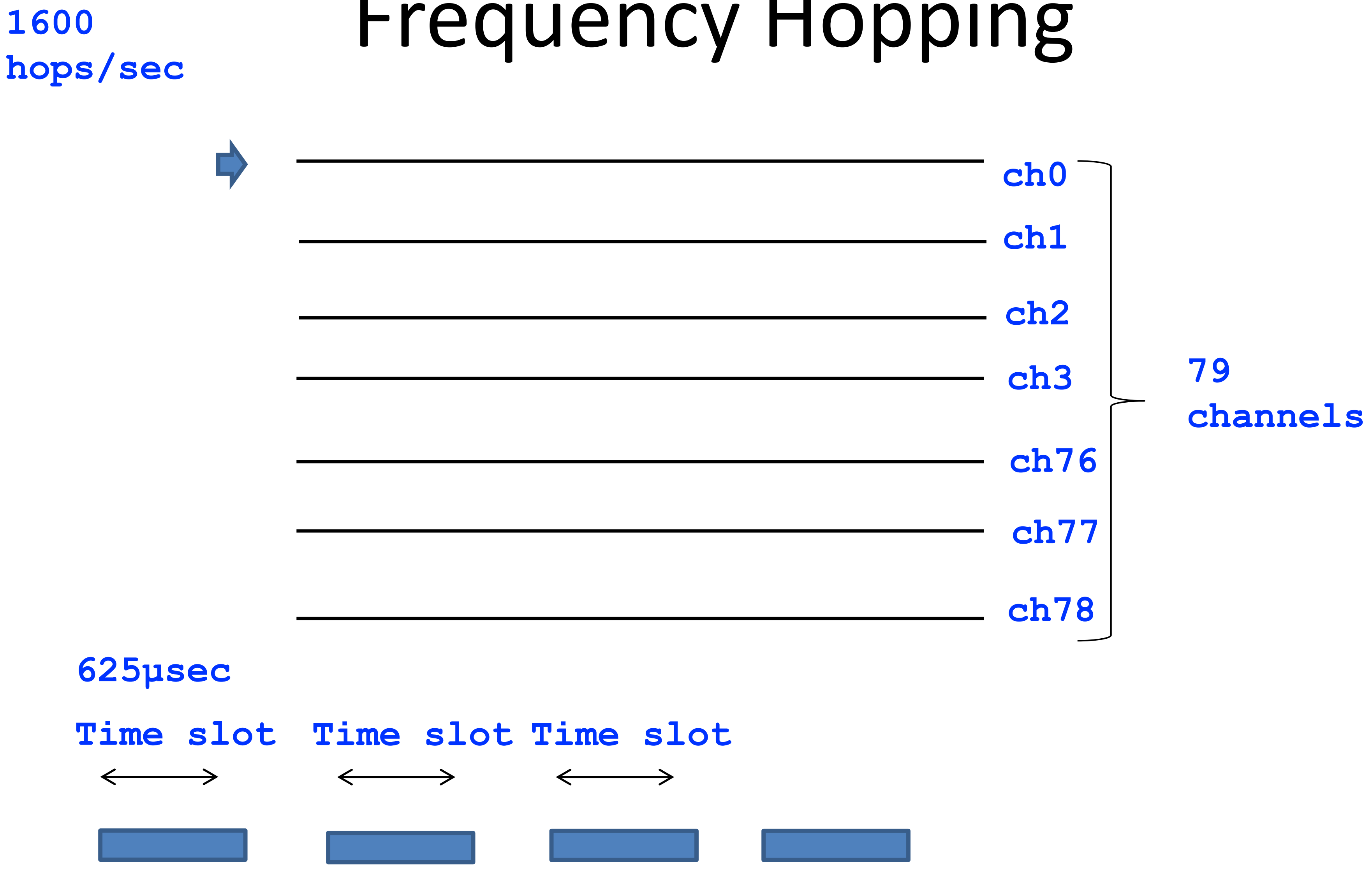
Radio

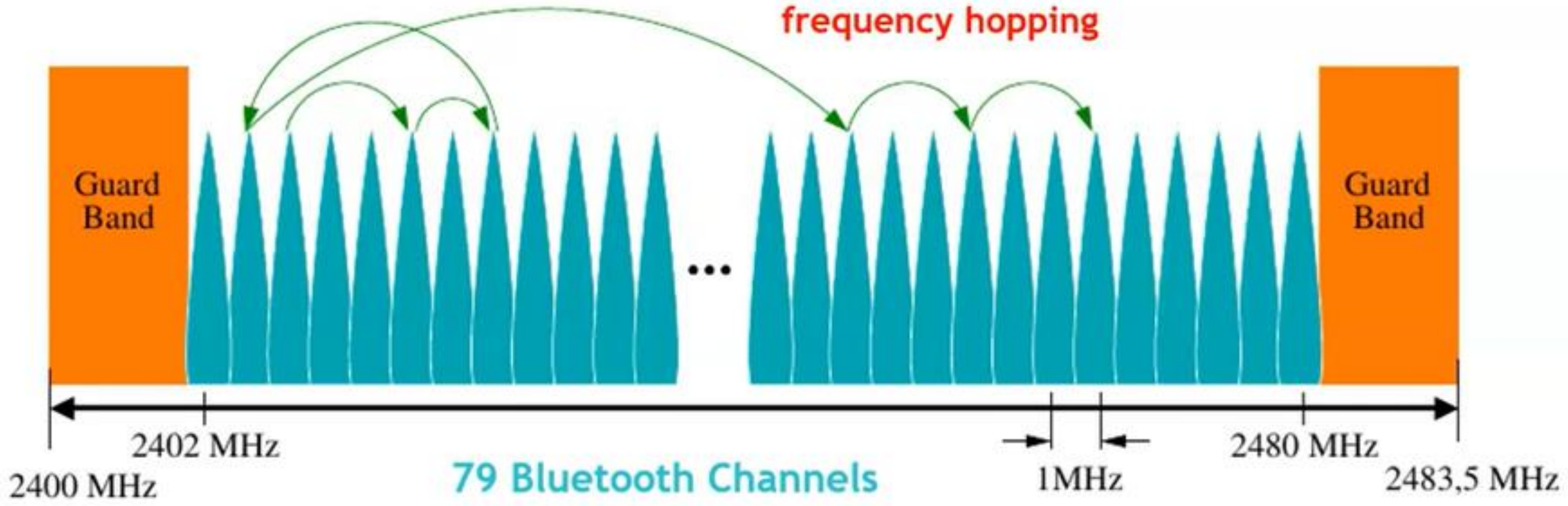
- Low Cost
  - Single chip radio (minimize external components)
  - Time division duplex

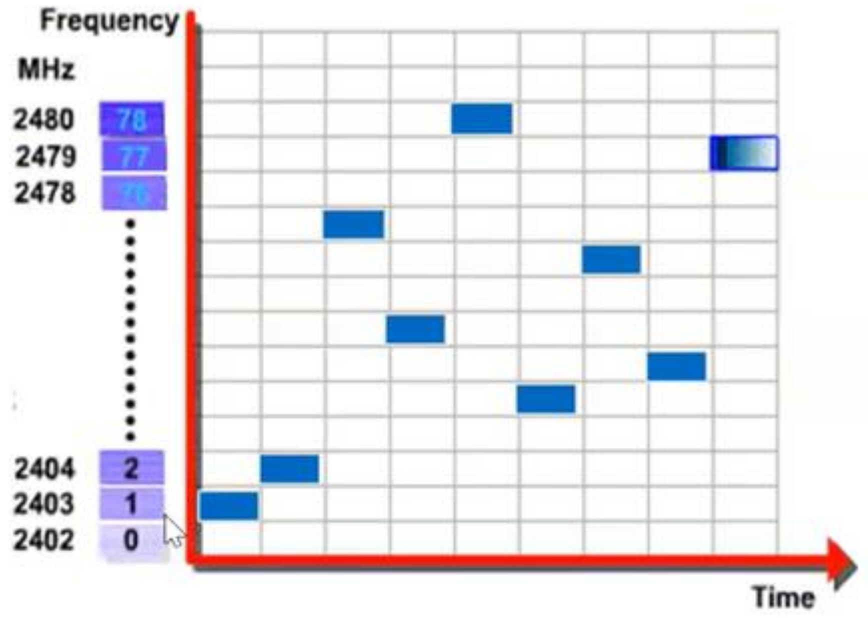
- Low Power
  - Standby modes [Hold, Sniff, Park]
  - Low power oscillator (reduces receiver sensitivity)

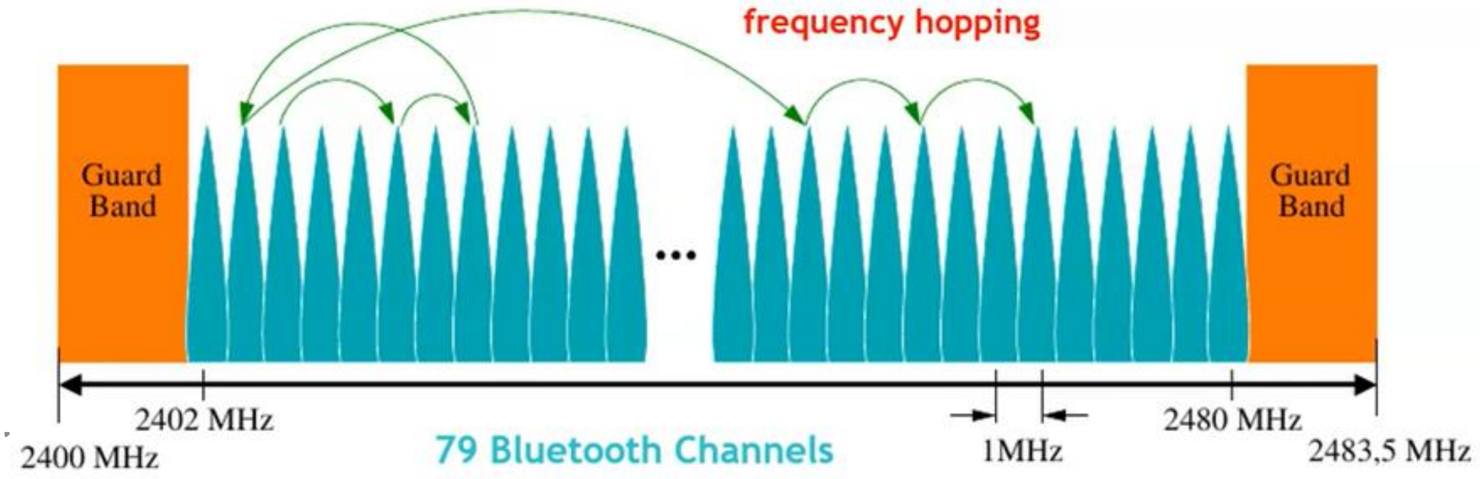
- Robust Operation
  - Fast frequency hopping 1600 hops/sec
  - Strong interference protection

Frequency Hopping



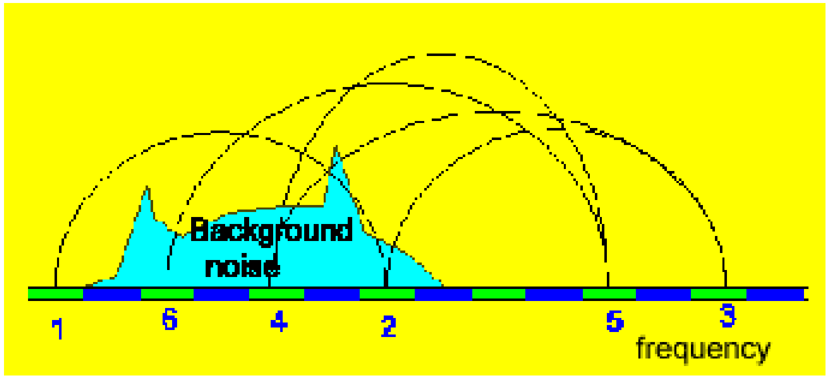

frequency hopping
Guard Band
Guard Band
...
2402 MHz
2400 MHz
79 Bluetooth Channels
1MHz
2480 MHz
2483,5 MHz


Frequency
MHz
2480
2479
2478
2404
2403
2402
78
77
76
2
1
0
Time


frequency hopping
Guard Band
Guard Band
...
2402 MHz
2400 MHz
79 Bluetooth Channels
1MHz
2480 MHz
2483,5 MHz

Frequency Hopping

- In order to mitigate interference, Bluetooth implements frequency hopping
- 1600 hops per second through 79, 1MHz channels
- Spreads Bluetooth traffic over the entire band
- All slaves in piconet follow the master for frequency hop sequence

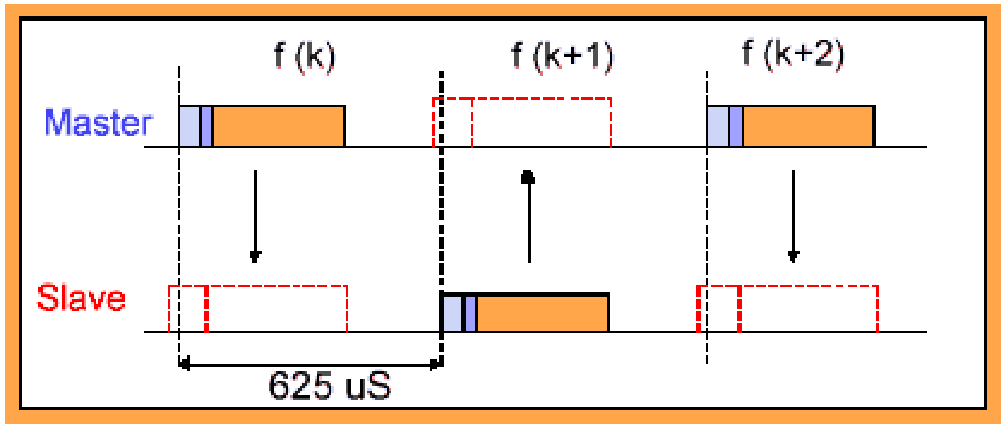


Radio & Modulation

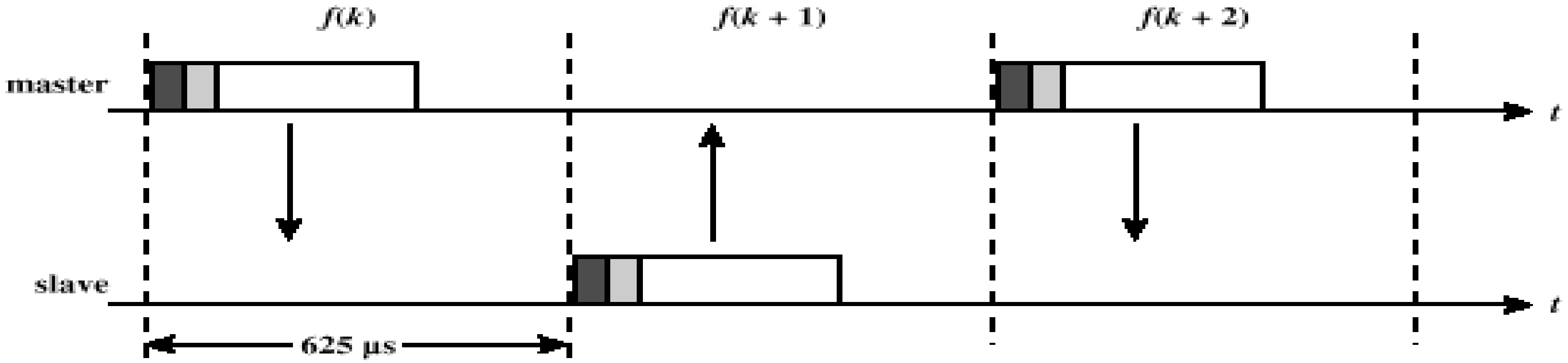
- frequency synthesis: frequency hopping
  - 2.400-2.4835 GHz
  - 2.402 + k MHz, k=0, …, 78
  - 1,600 hops per second
- conversion bits into symbols: modulation
  - GFSK ($BT = 0.5$; $0.28 < h < 0.35$);
  - 1 MSymbols/s
- transmit power
  - up to 20dbm with power control

Frequency Hopping

- Hops every packet
- Packets can be 1, 3, or 5 slots long (a slot is 625μs)
- Packets are pretty short (366 bits)

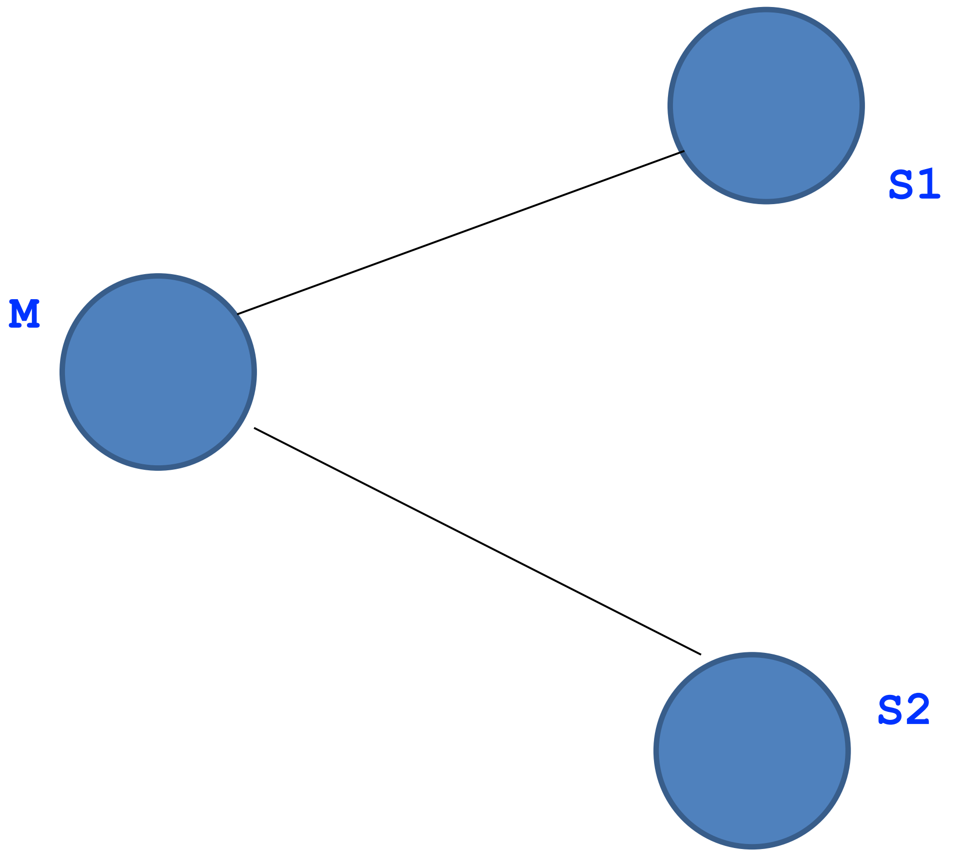


Frequency Hopping



- Each frame uses a single hop frequency for its duration

Piconet channel comm.

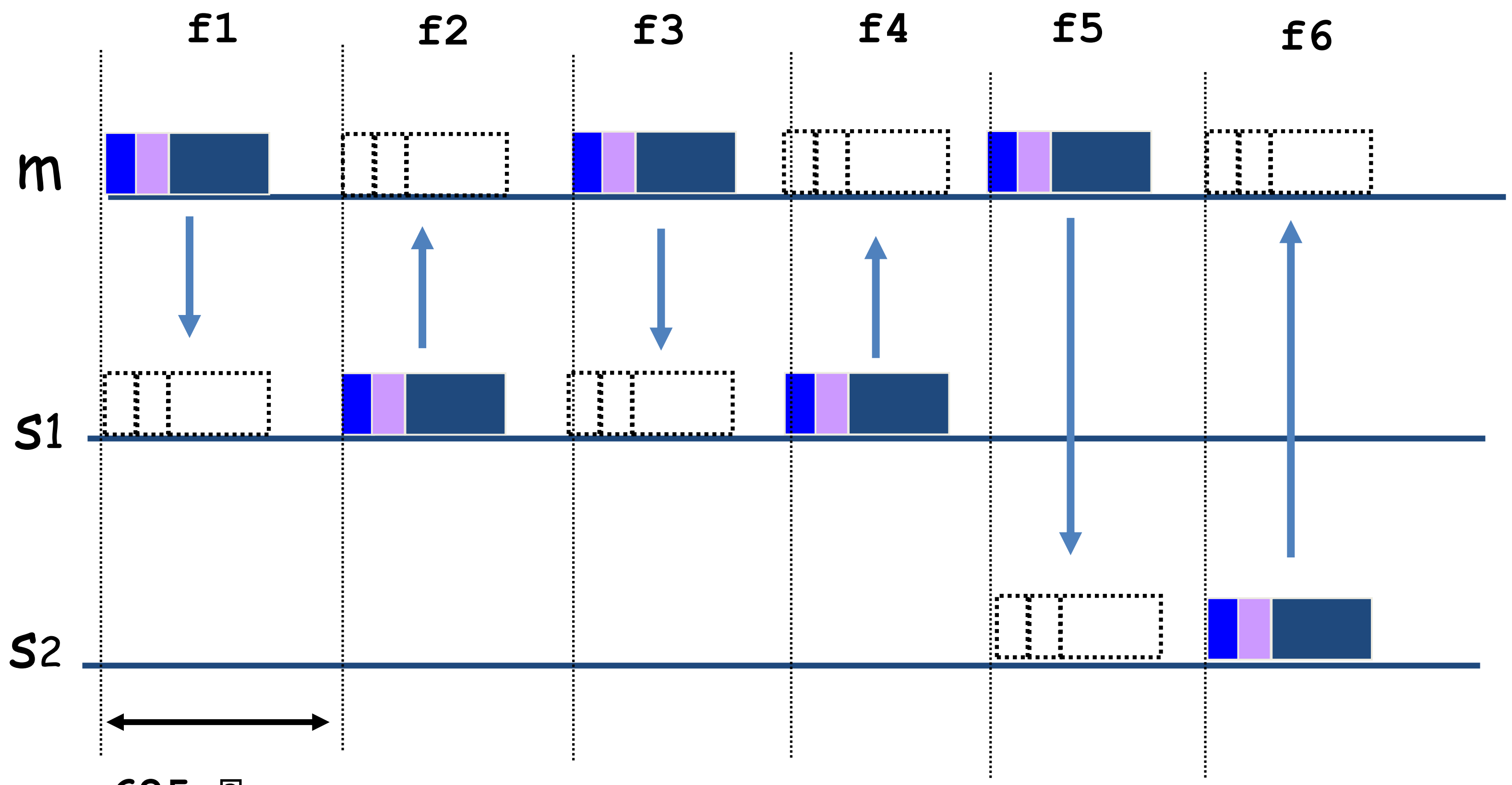


Piconet channel is divided into 625µsec Slots

Different freq. used

Piconet channel

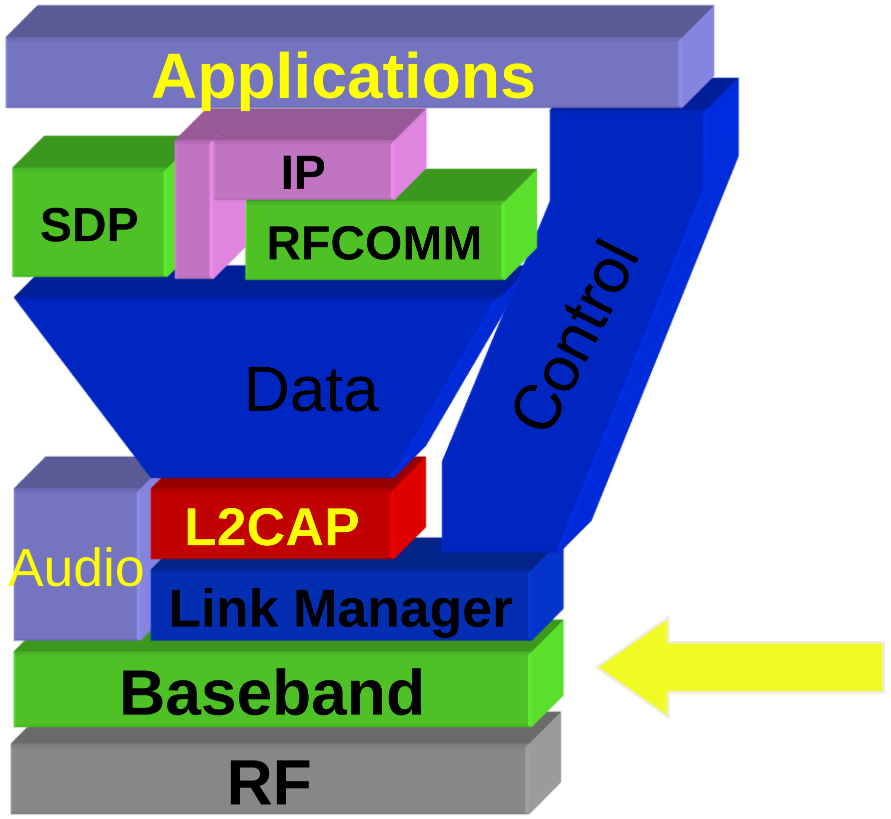
FH/TDD



625 µsec

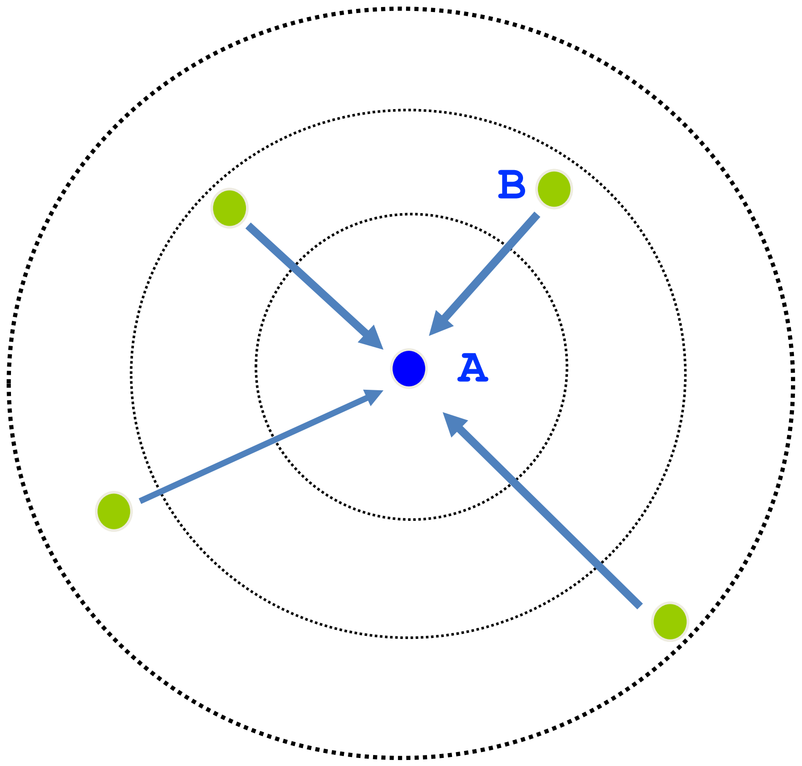
1600 hops/sec

Baseband


Applications
IP
SDP
RFCOMM
Data
Control
L2CAP
Audio
Link Manager
Baseband
RF

Piconet Connection Setup

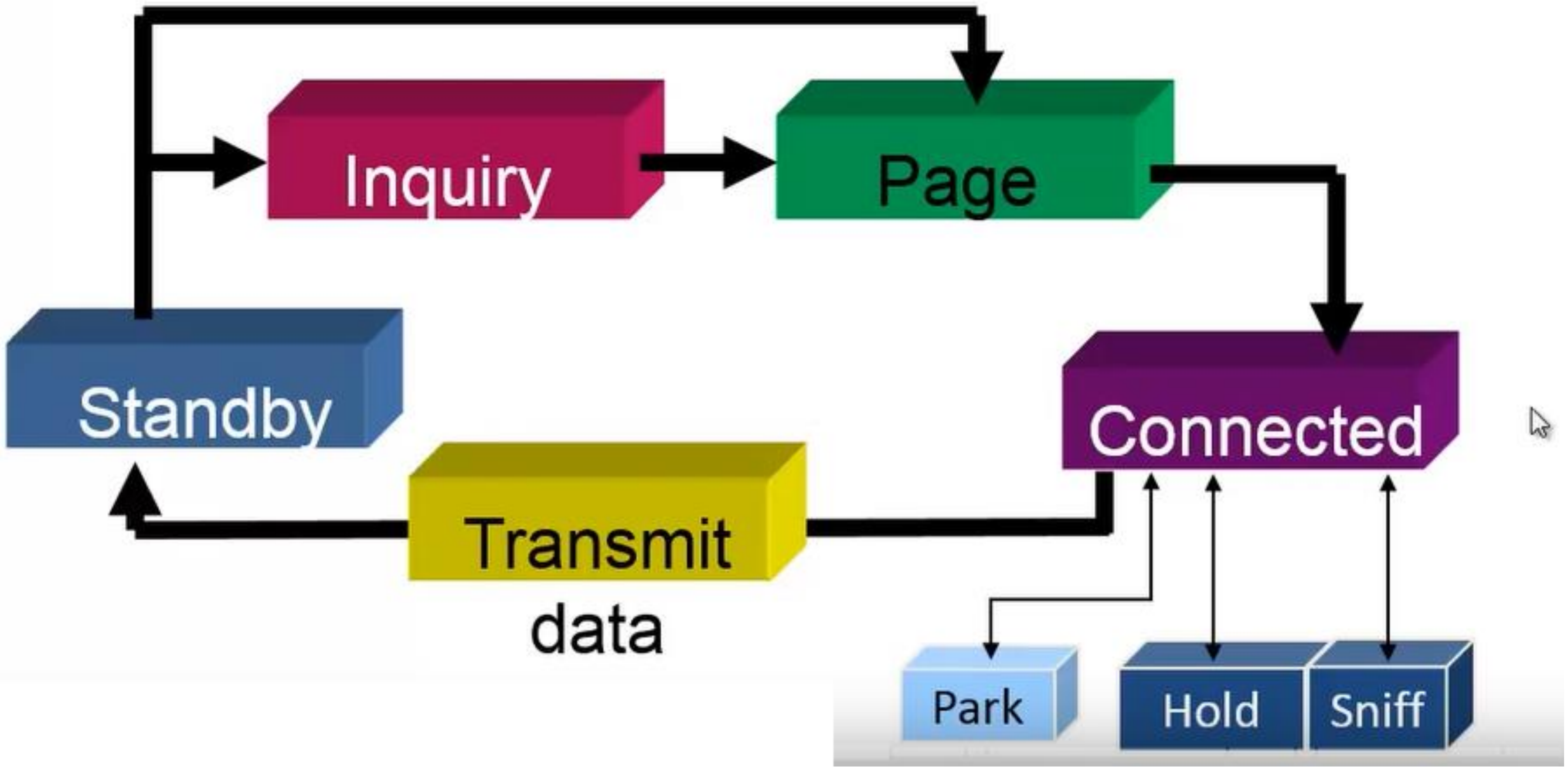
- Inquiry - scan protocol
  - To discover nodes in proximity
- Paging
  - Establish connections



- **Two nodes cannot exchange messages**
  - **Until they agree to a common channel hop sequence**

- **Mandate the use of a known inquiry hopping sequence**

Connection State Machine



Bluetooth: Hello, Anyone Around?

- Inquiry Procedure
  - Sends out an inquire, which is a request for nearby devices (within 10 meters)
  - Devices that allow themselves to be discoverable issue an inquiry response
  - Process can take up to 10.24 seconds, after which the inquiring device should know everyone within 10 meters of itself

Inquiry Procedure

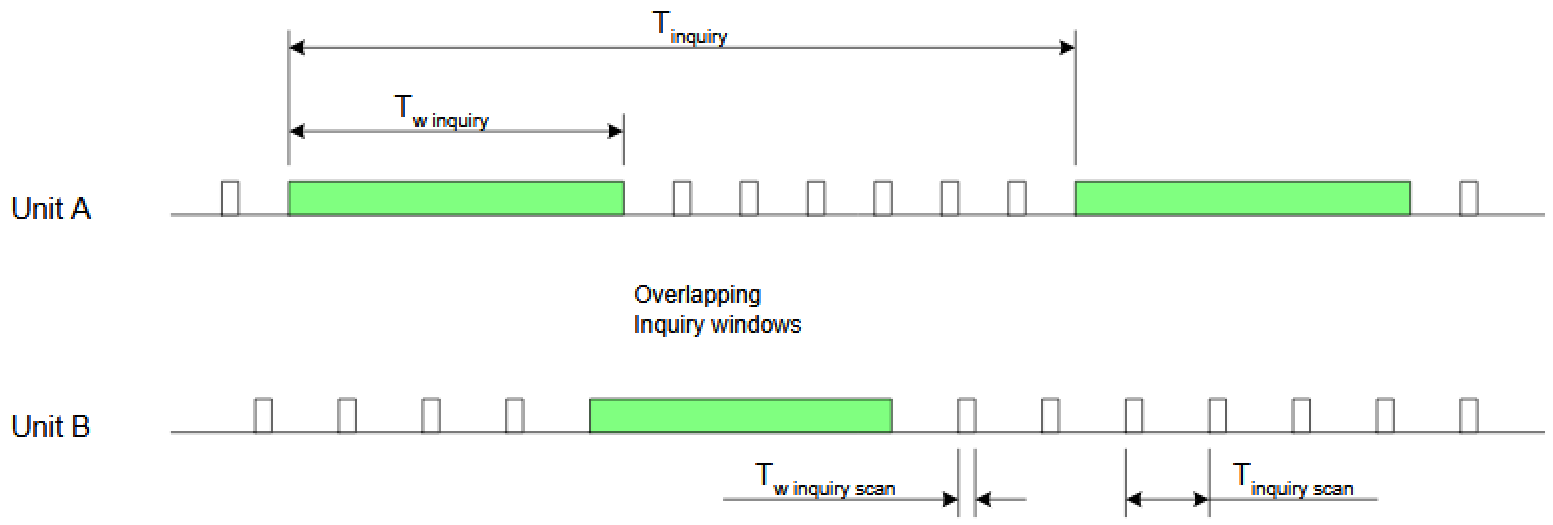
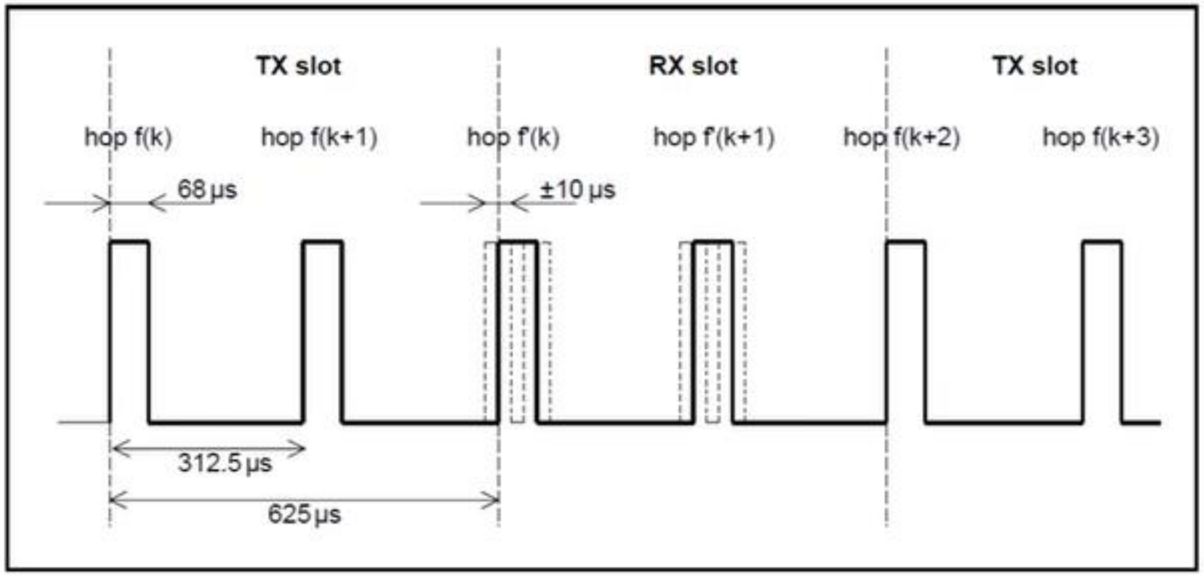
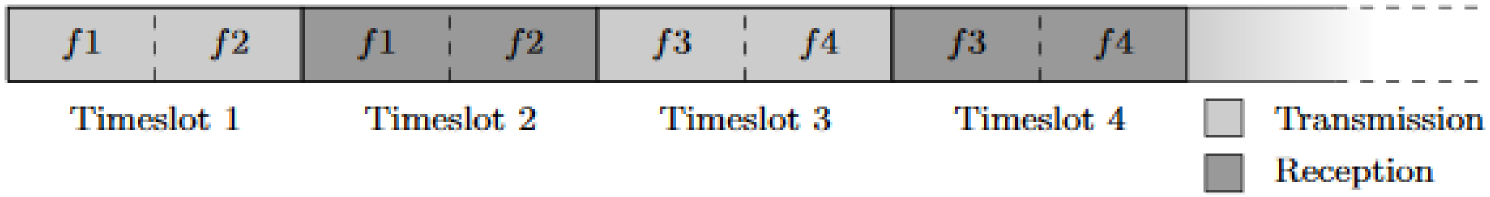


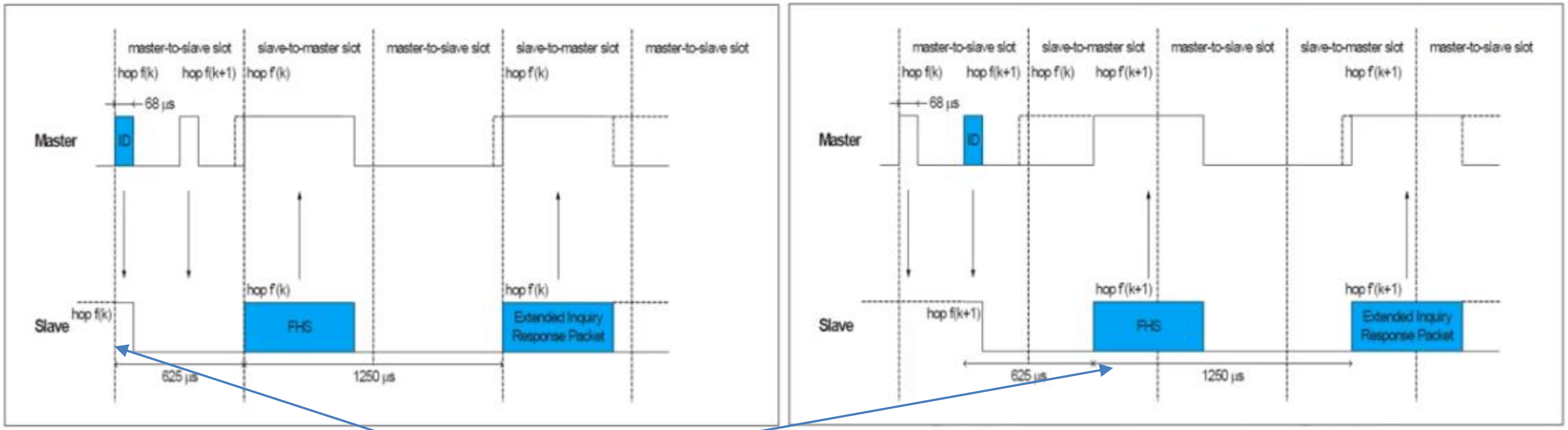
Figure 3.2: *Periodic inquiry and inquiry scan.*

Inquiry Procedure



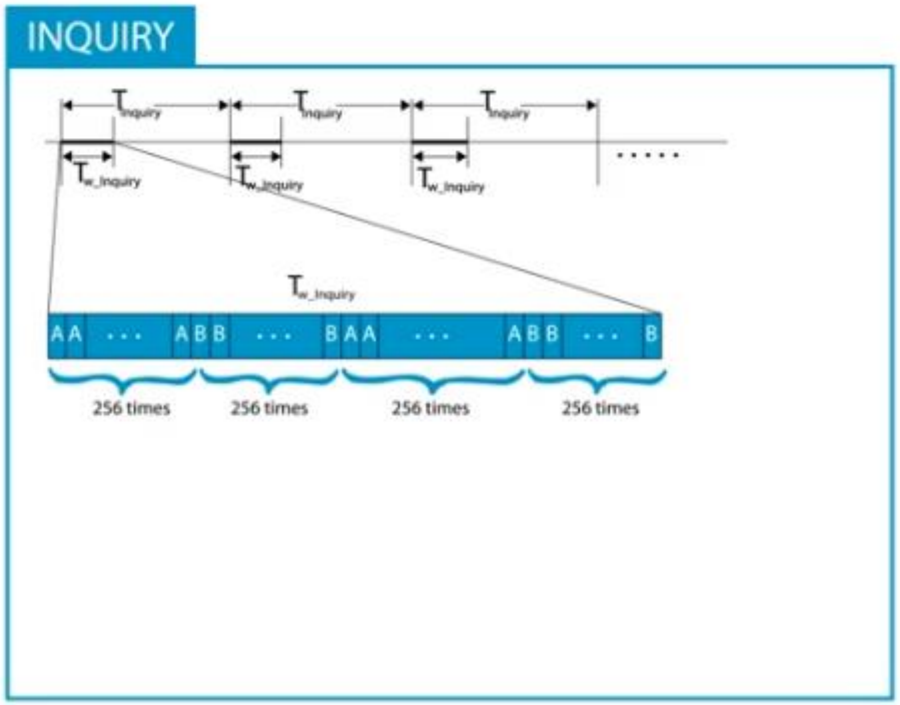


Inquiry Procedure



Slaves open the window for discovery (11.25msec)

Inquiry Procedure



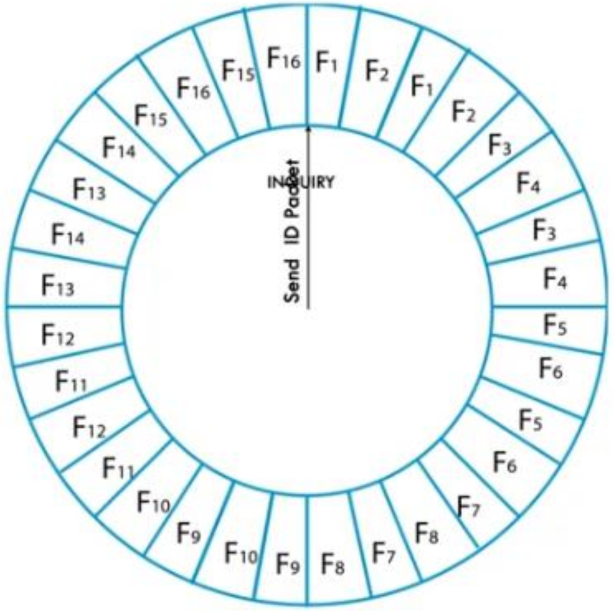
Uses 32 inquire channels to send out inquiry messages

Send out inquiry on 32 channels, broken up into 2 inquiry hop trains (16 different channels to transmit packets)

Intended to catch a device in inquiry scan mode on one of the 32 inquire channels

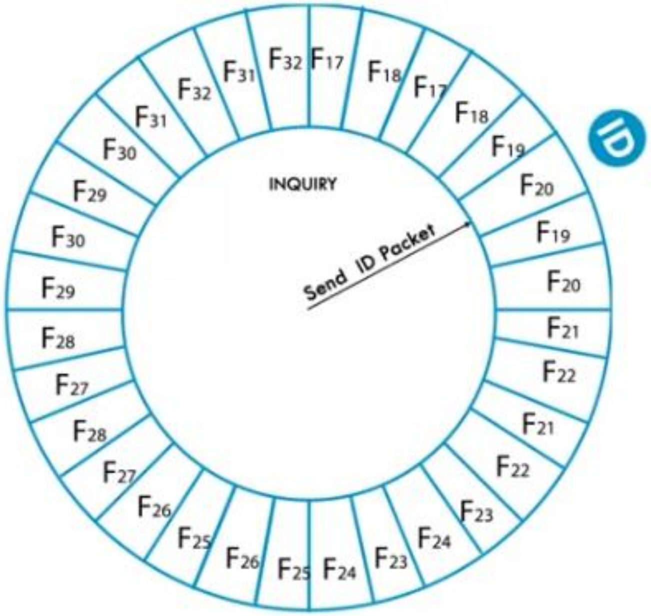
Inquiry Procedure

switches frequency for every 0.325 us
Total Time taken is 312.5us x 32 = 10 ms


INQUIRY
Send ID Packet
F16 F1 F2 F1 F2 F3 F4 F3 F4 F5 F6 F5 F6 F7 F8 F7 F8 F9 F10 F9 F10 F11 F12 F11 F12 F13 F14 F13 F14 F15 F16 F15

TrainA : 1 Train takes (32 * 0.312.5micro sec) = 10 milli sec
256 Trains take (10 milli sec * 256) = 2.56sec.

switches frequency for every 0.325 us
Total Time taken is 312.5us x 32 = 10 ms


INQUIRY
Send ID Packet
ID
F32 F17 F18 F17 F18 F19 F20 F19 F20 F21 F22 F21 F22 F23 F24 F23 F24 F25 F26 F25 F26 F27 F28 F27 F28 F29 F30 F29 F30 F31 F32 F31

TrainB : 1 Train takes (32 * 0.312.5micro sec) = 10 milli sec
256 Trains take (10 milli sec * 256) = 2.56sec.

Few numbers

- 10.24sec inquiry window
- 32 channels are used
- Two trains A and B of 16 freq.
- Slaves open the window for discovery

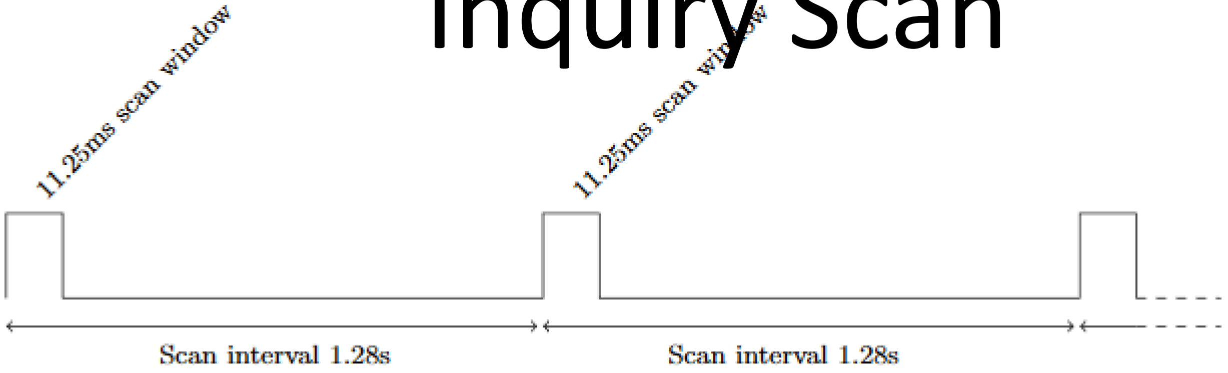
(11.25msec)

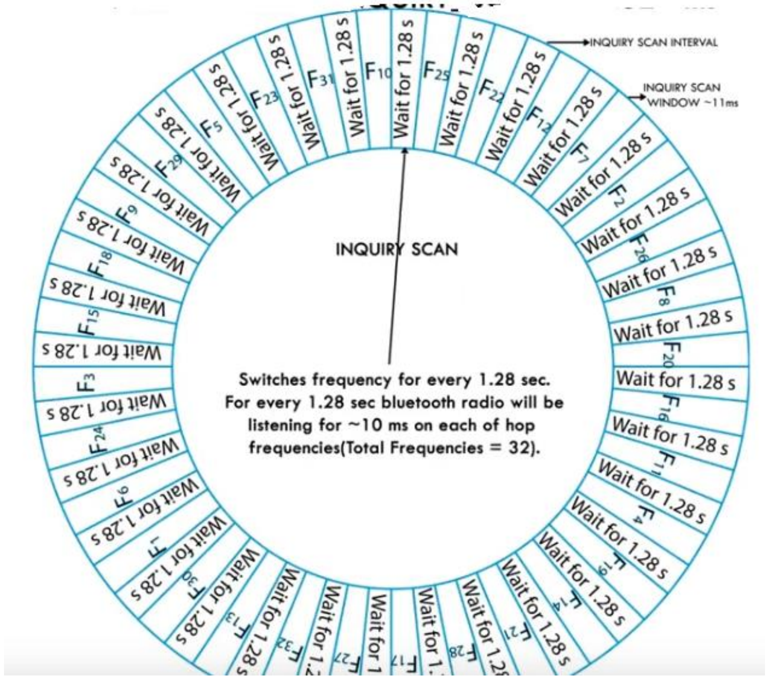
Scanning by master takes

One train takes 312.5ms*32=10msec

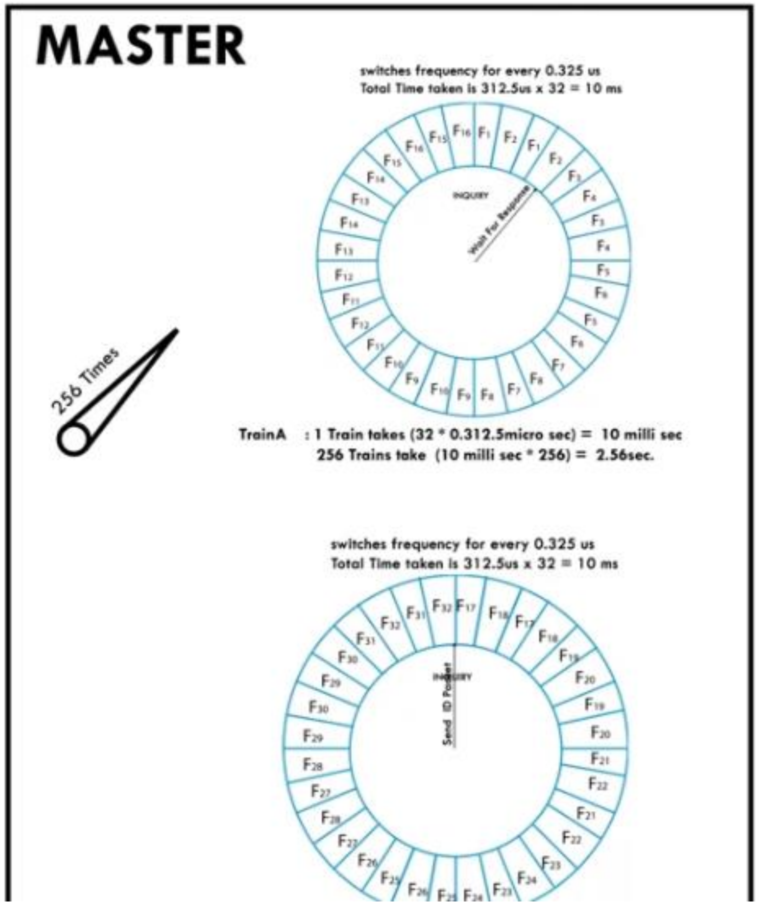
$$2\ \mathit{trains} \times 2\ \mathit{iterations} \times 256\ \mathit{times} \times 0.01\ s = 10.24\, s$$

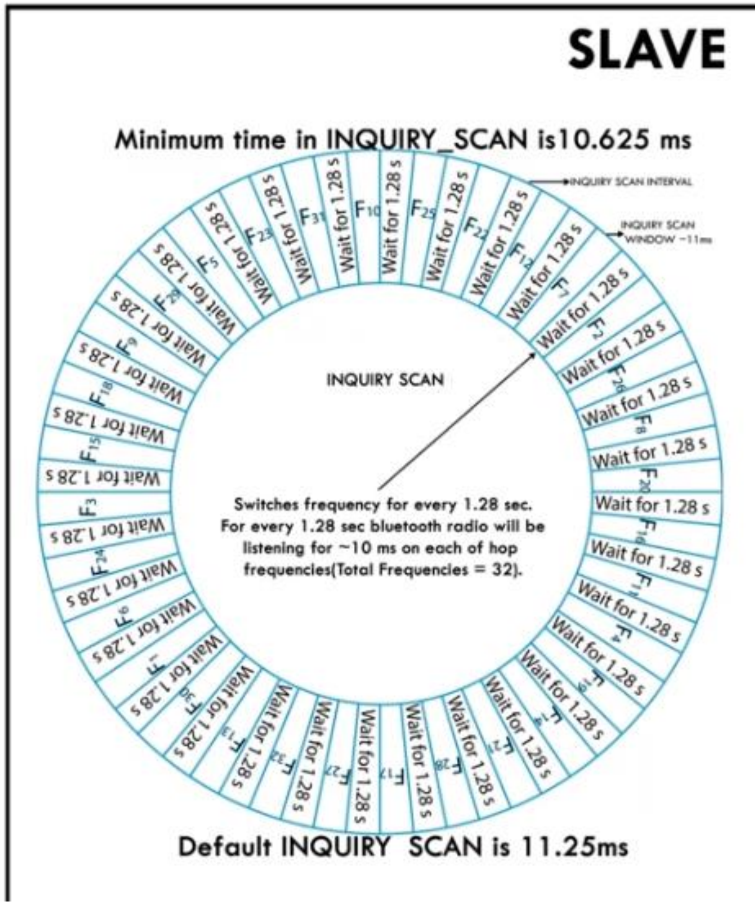
Inquiry Scan




INQUIRY SCAN INTERVAL
INQUIRY SCAN WINDOW ~11ms
INQUIRY SCAN
Switches frequency for every 1.28 sec. For every 1.28 sec bluetooth radio will be listening for ~10 ms on each of hop frequencies(Total Frequencies = 32).
Wait for 1.28 s
F10
F25
F22
F12
F7
F2
F26
F8
F20
F16
F11
F4
F19
F14
F21
F28
F17
F27
F32
F13
F30
F1
F6
F24
F3
F15
F18
F9
F29
F5
F23
F31

Inquiry Procedure & Scan


MASTER
switches frequency for every 0.325 us
Total Time taken is 312.5us x 32 = 10 ms
INQUIRY
Wait For Response
256 Times
TrainA : 1 Train takes (32 * 0.312.5micro sec) = 10 milli sec
256 Trains take (10 milli sec * 256) = 2.56sec.
switches frequency for every 0.325 us
Total Time taken is 312.5us x 32 = 10 ms
INQUIRY
Send ID Packet


SLAVE
Minimum time in INQUIRY_SCAN is10.625 ms
INQUIRY SCAN INTERVAL
INQUIRY SCAN WINDOW ~11ms
INQUIRY SCAN
Wait for 1.28 s
Switches frequency for every 1.28 sec.
For every 1.28 sec bluetooth radio will be listening for ~10 ms on each of hop frequencies(Total Frequencies = 32).
Default INQUIRY SCAN is 11.25ms

Inquiry Scan

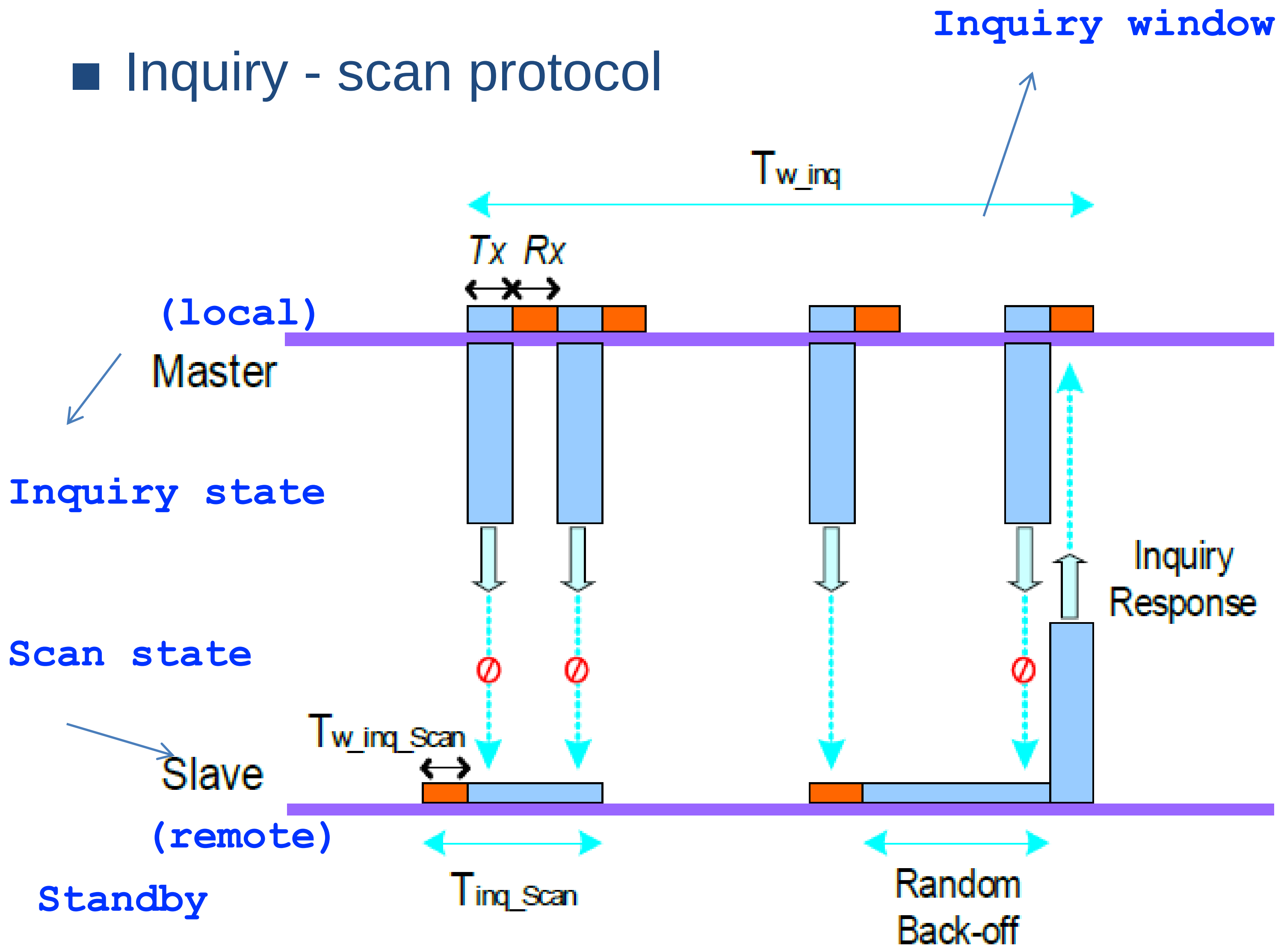
- A device periodically listens for inquiry packets at a single frequency – chosen out of 16 frequencies
- Stays in the state long enough for a inquiring device to cover 16 frequencies

Issues with Inquire Messages

- Are the inquirer transmitting and the receiver listening on the same frequency?
  - Since they are not yet connected, they are on totally different hop sequences, and most likely on different channels
  - Known hop sequence
- If they are on the same frequency, what if they are on a noisy channel?
  - Bluetooth provides the capability for receivers to issue multiple inquiry responses

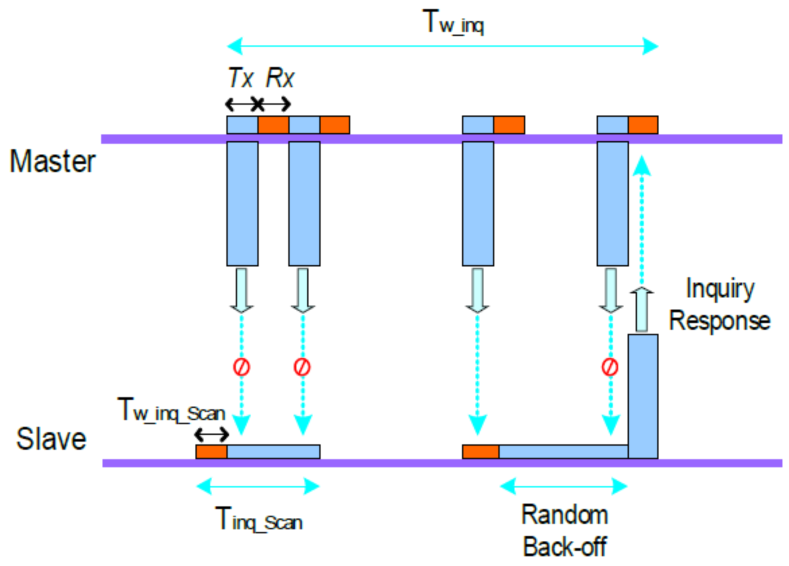
Connection Setup

- Inquiry - scan protocol



Connection Setup

- Inquiry - scan protocol
  - to learn about the clock offset and device address of other nodes in proximity

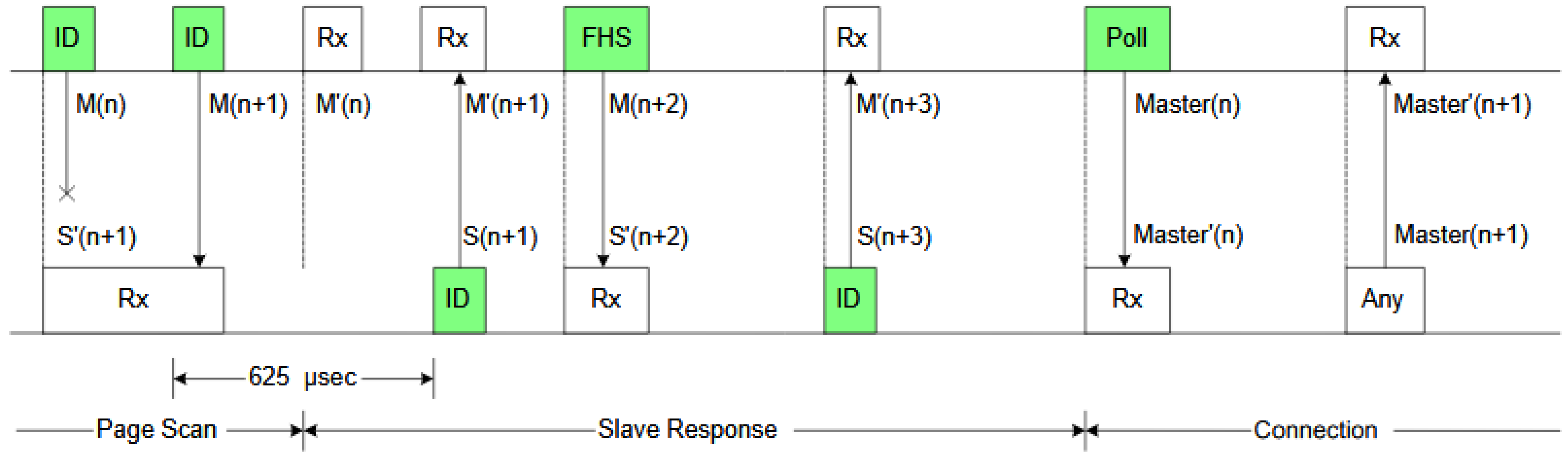


During this time the device listens to a single frequency of the InquiryHopping Sequence

Inquiry Response

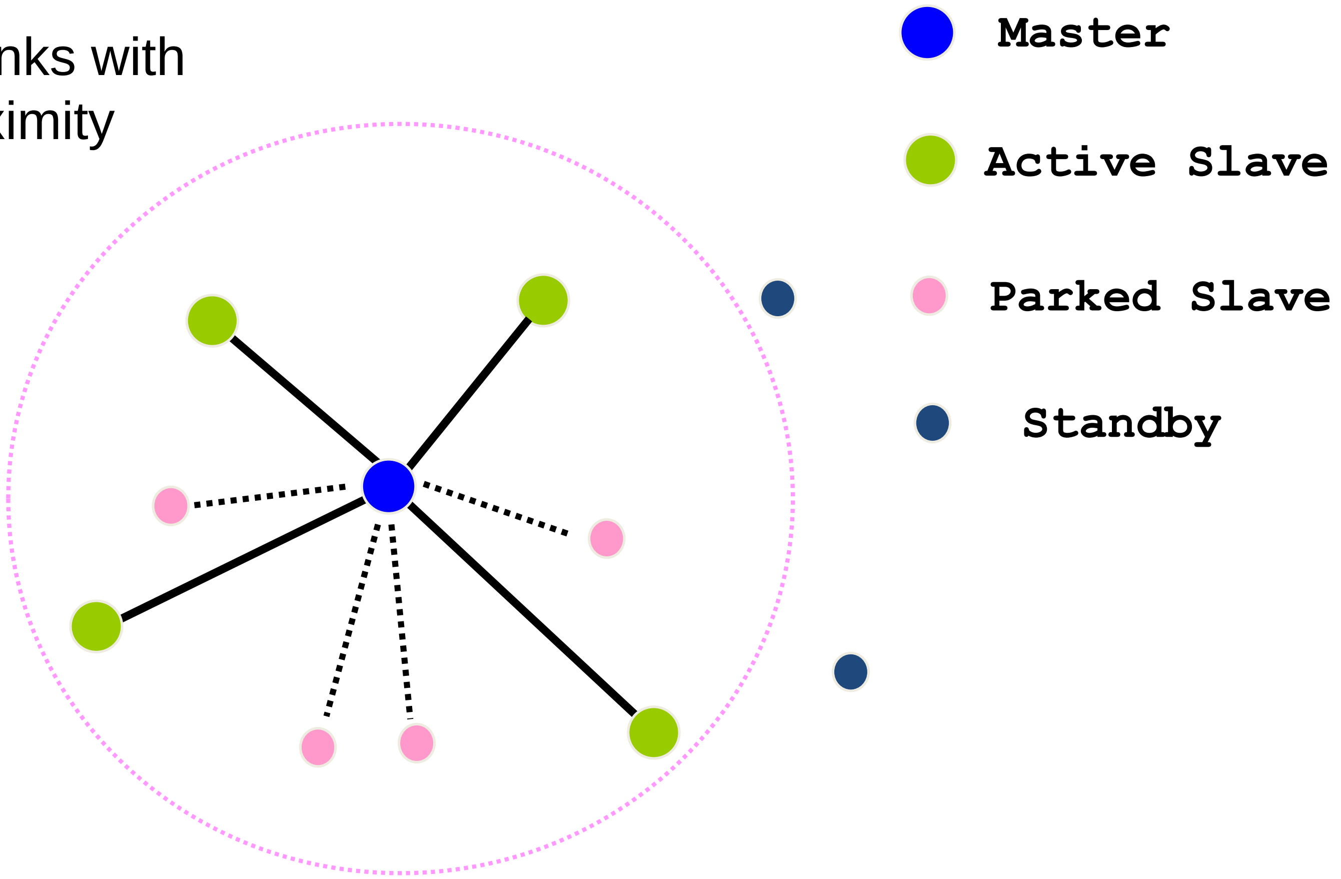
- When radio receives inquire, it will wait before sending an FHS packet as a response
  - Exponential backoff
  - This is done to avoid collision with another radio that also wants to send an FHS packet
- FHS Packet contains:
  - Device ID
  - Clock
- After inquiring radio is done with inquiring procedure, it knows all of the radios (that are discoverable) within range

Paging


ID
ID
Rx
Rx
FHS
Rx
Poll
Rx
M(n)
M(n+1)
M'(n)
M'(n+1)
M(n+2)
M'(n+3)
Master(n)
Master'(n+1)
S'(n+1)
S(n+1)
S'(n+2)
S(n+3)
Master'(n)
Master(n+1)
Rx
ID
Rx
ID
Rx
Any
625 µsec
Page Scan
Slave Response
Connection

Piconet formation

- Page - scan protocol
  - to establish links with nodes in proximity



Paging

Similar process

Unicast massage to selected listener

Master and slave get formed

Admitting new device

- Master can start discovering
- Wait to be scanned
- Original communication gets suspended
- Latency

Baseband Layer

- Provides functionality to determine nearby Bluetooth devices
- Provides in-order delivery of byte streams
- Handles Frequency Hop Sequences for Synchronization and Transmission
- Establishes Links
  - Synchronous Connection Oriented (SCO)
  - Asynchronous Connection-Link (ACL)

Piconet connection

- Link speed 1Mbps
- 625 µsec slot time
  - Transmission of 625 bits

- Single slot packet size 366 bits (30 bytes payload)
  - Guard time for feq. hop

- Two different kinds of links
  - SCO
  - ACL
  - On each link, 16 types of packet can be used

Synchronous link (SCO)

- Transmits real time voice
  - Master establishes the link
  - Master reserves slot
- Three kinds of voice packets

(1) HV3 : 30 bytes of voice data- no error correction code

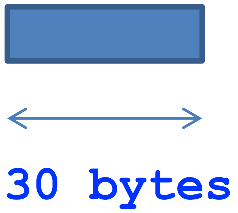
(2) HV2: 20bytes of voice + 10 bytes of FEC code

(2) HV1: 10 bytes of voice + 20 bytes of FEC code

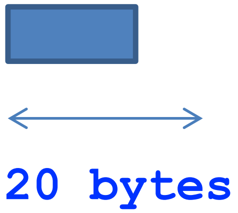
No retransmission

Synchronous link (SCO)

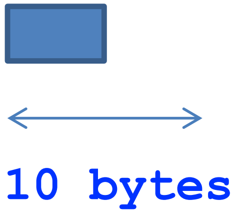
- Transmits real time voice
  - Reserve slot
- Speech coder generates 10 bytes every 1.25ms



(HV3) One slot is needed (reserved) every 3.75ms (every 6th slot)



(HV2) One slot is needed every 2.5ms (every 4th slot)



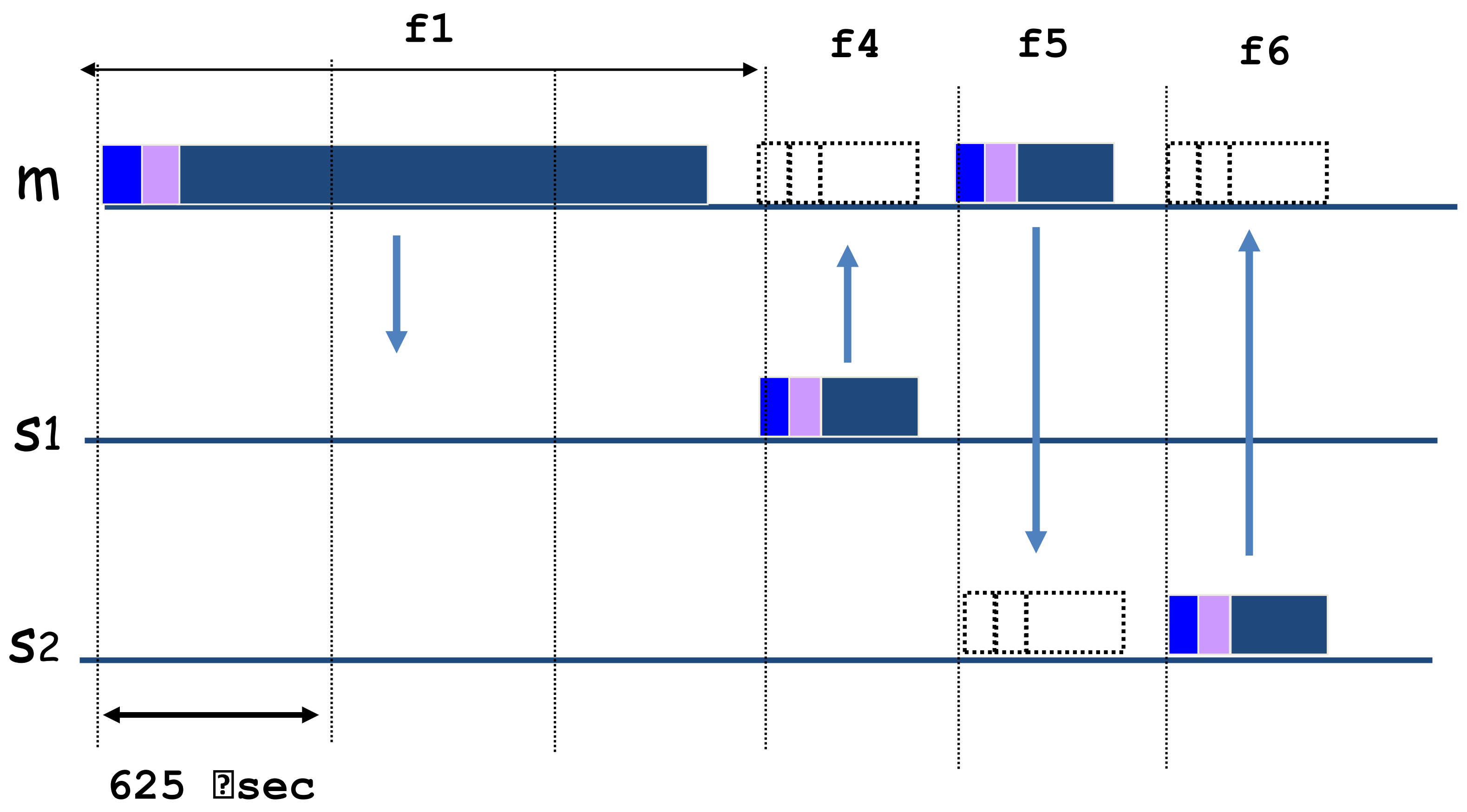
(HV1) One slot is needed every 1.25ms (every 2nd slot) ⇐ Noisy channel.

Asynchronous link

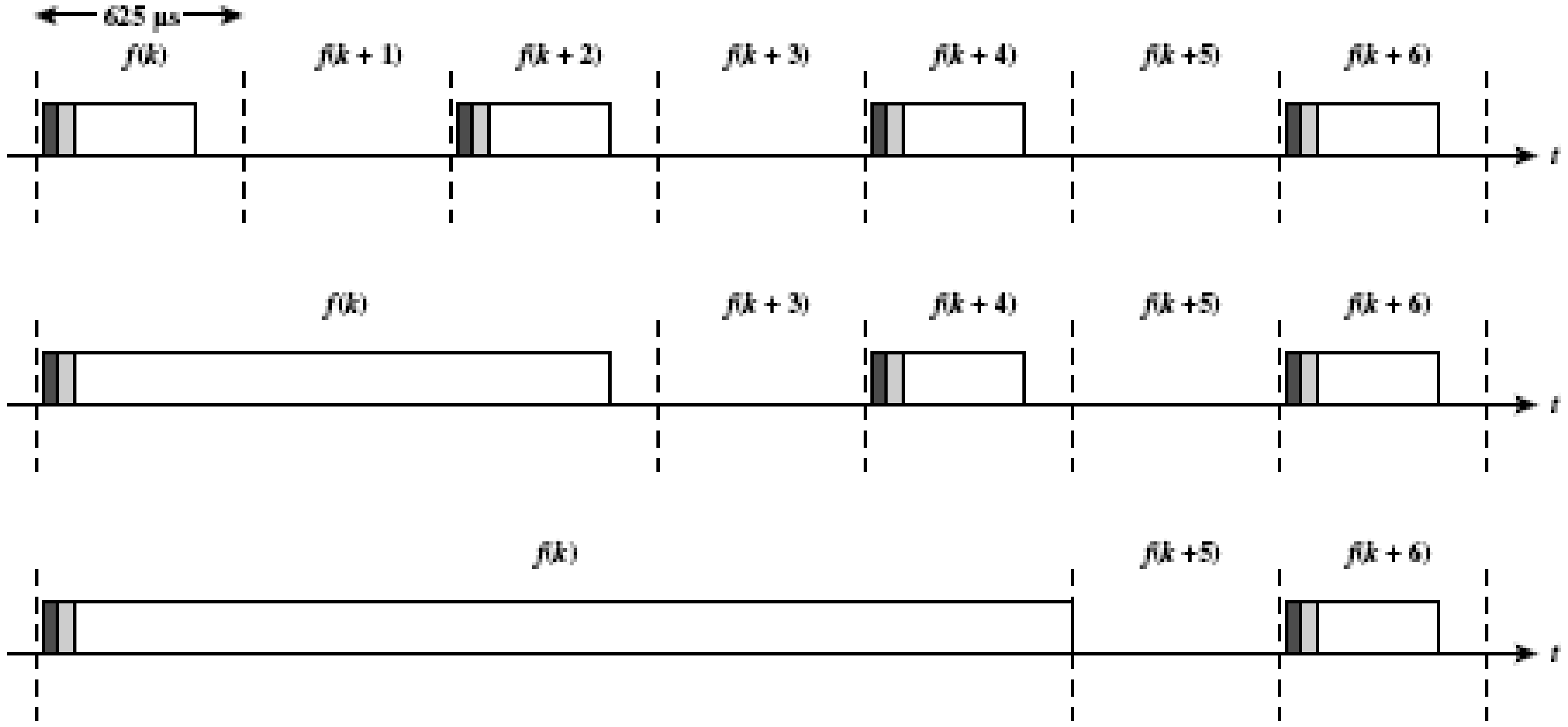
- Data communication
  - Momentary connection M and S for one frame
- Protected by cyclic redundancy code (CRC)
- Retransmission of data
  - For error, data loss
- Demand based slot allocation
  - SCO has higher priority on slots
- Master is responsible for distributing slots among ACL links
- Allow multislot packet transmission
  - 3, 5
  - Transmitter stays fixed on a hop frequency

FH/TDD

Multi slot packets



Multislot Frames

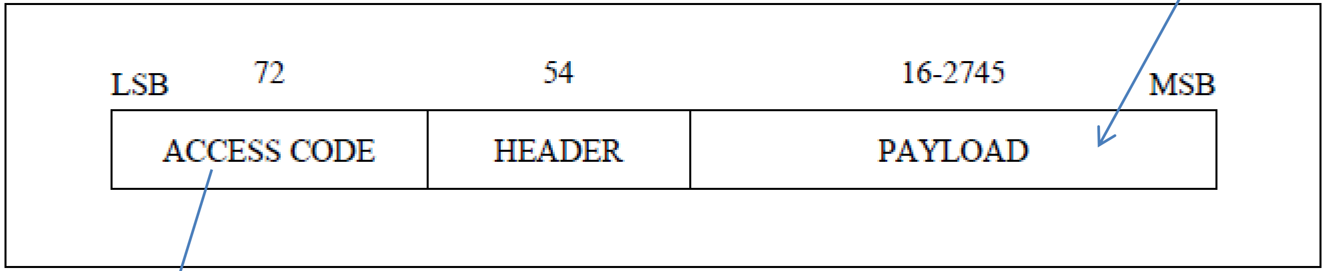


Addressing

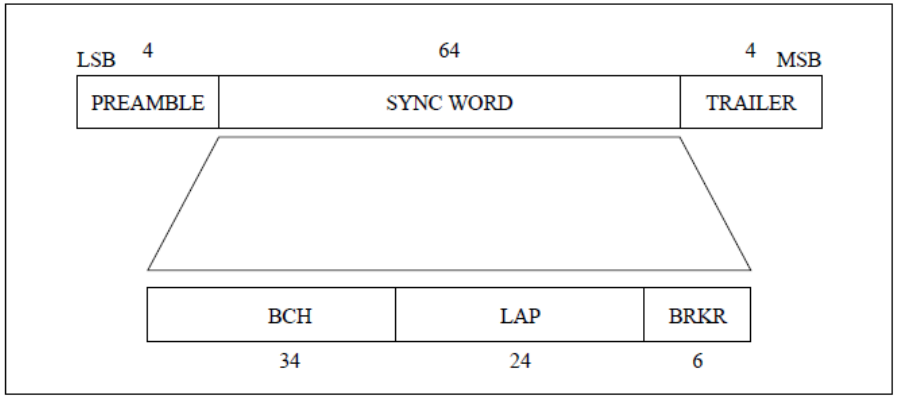
- Bluetooth device address (BD_ADDR)
  - 48 bit IEEE MAC address
- Active Member address (AM_ADDR)
  - 3 bits active slave address
  - all zero broadcast address
  - Temporary
- Parked Member address (PM_ADDR)
  - 8 bit parked slave address

Packet format

2-342 bytes

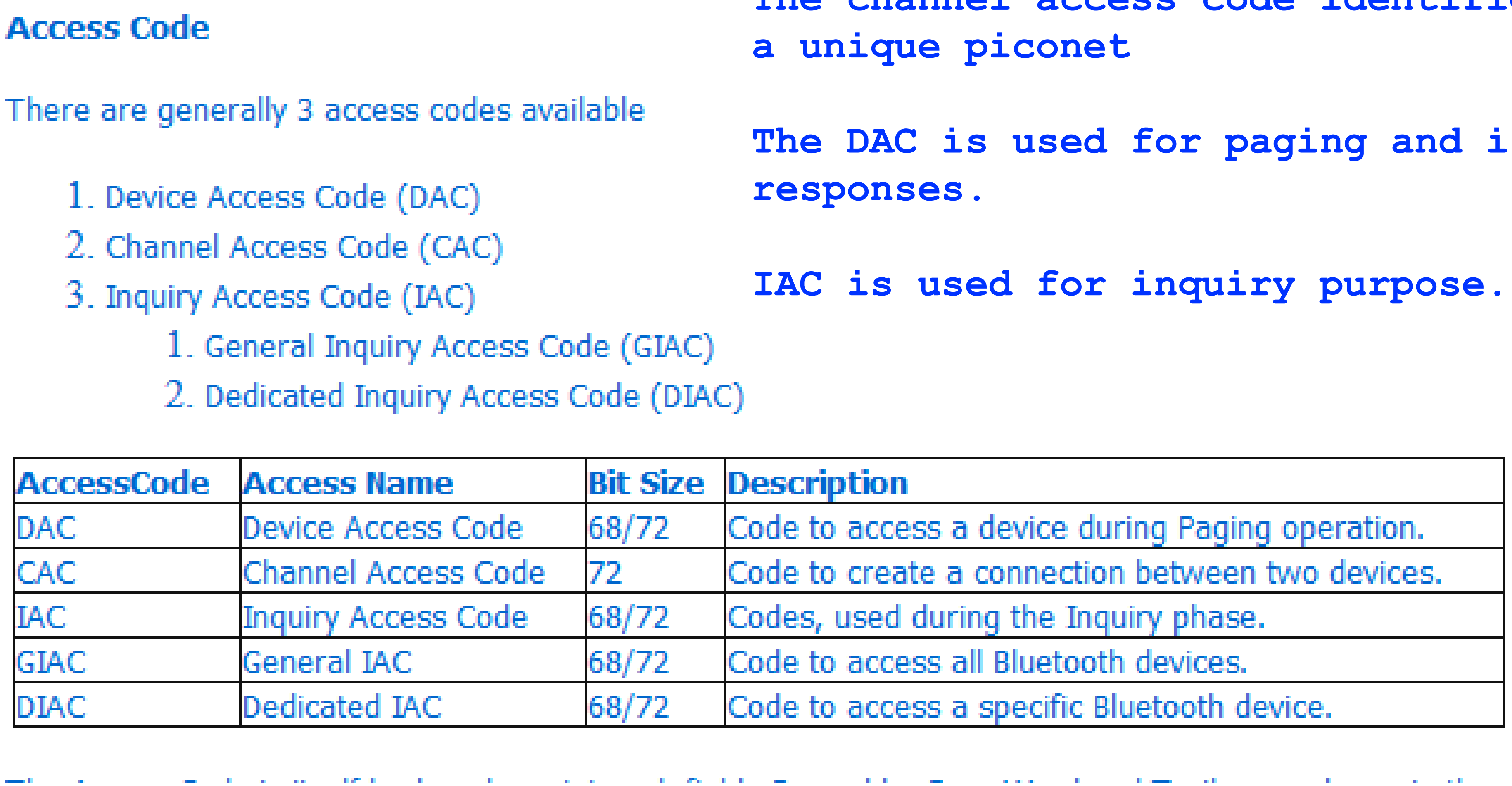


Used for synch, identification of the piconet



Access code identifies all the packets exchanged in a piconet

Access code



Access Code

There are generally 3 access codes available

1. Device Access Code (DAC)
2. Channel Access Code (CAC)
3. Inquiry Access Code (IAC)
    1. General Inquiry Access Code (GIAC)
    2. Dedicated Inquiry Access Code (DIAC)

AccessCode	Access Name	Bit Size	Description
DAC	Device Access Code	68/72	Code to access a device during Paging operation.
CAC	Channel Access Code	72	Code to create a connection between two devices.
IAC	Inquiry Access Code	68/72	Codes, used during the Inquiry phase.
GIAC	General IAC	68/72	Code to access all Bluetooth devices.
DIAC	Dedicated IAC	68/72	Code to access a specific Bluetooth device.

[illegible]

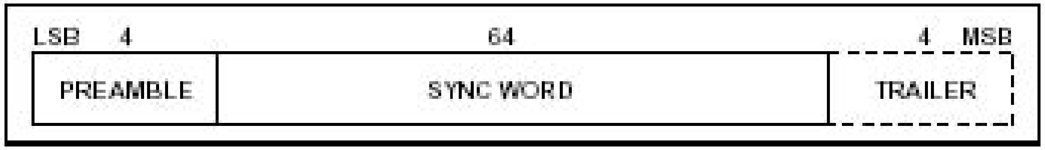
The channel access code identifies a unique piconet

The DAC is used for paging and its responses.

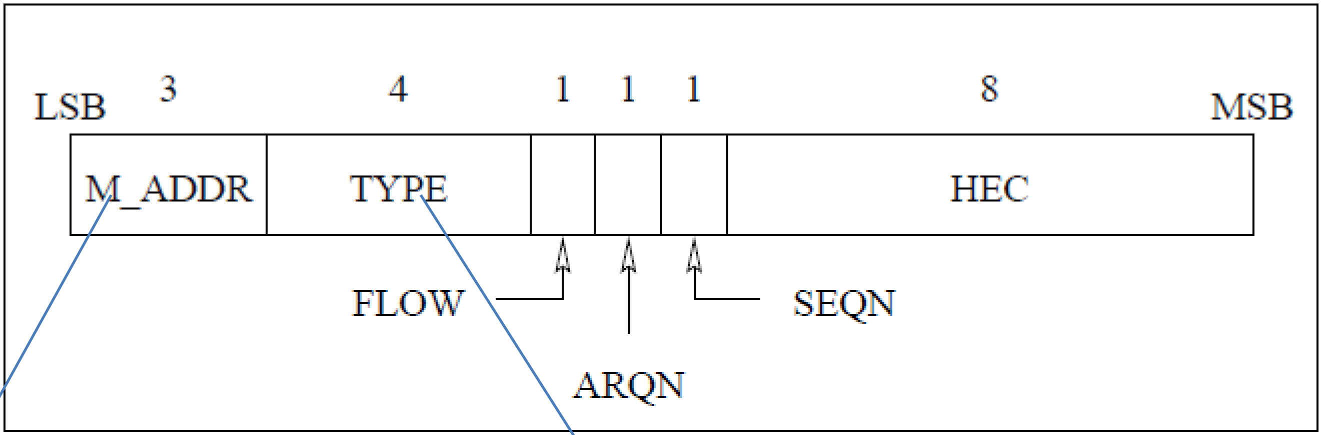
IAC is used for inquiry purpose.

Access code

The Access Code is itself broken down into sub fields-Preamble, Sync Word and Trailer, as shown in the figure below



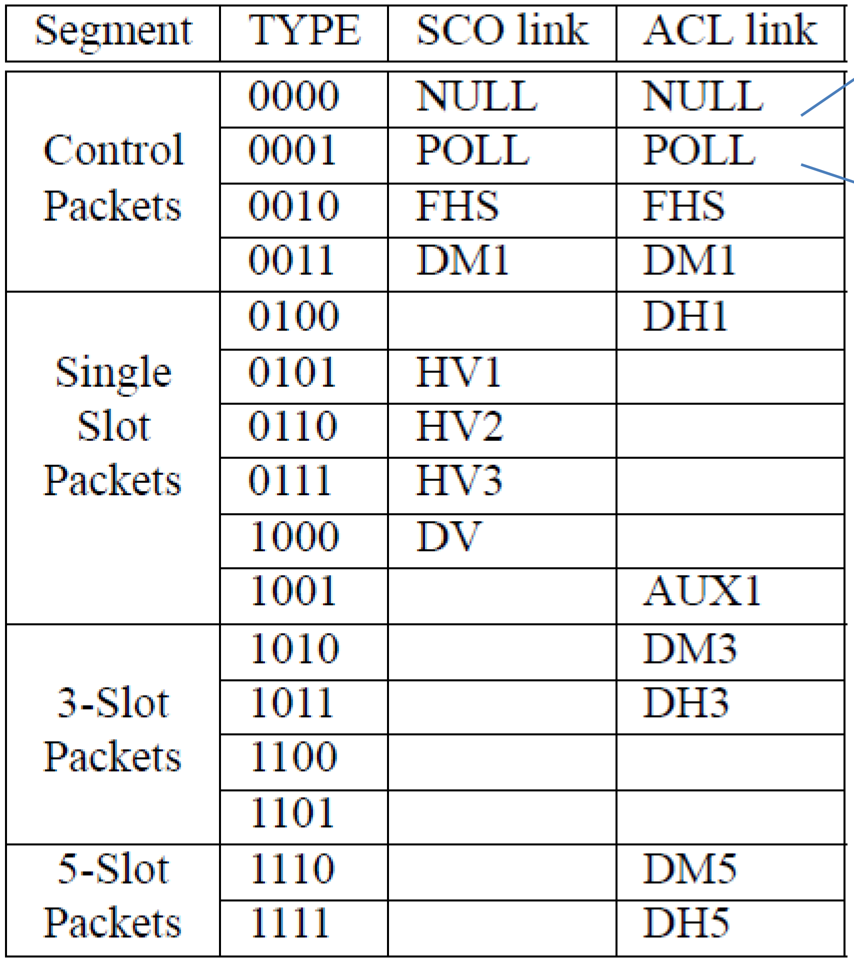
The Preamble is a 4 bit sequence of alternate 0 and 1 depending on the first bit of the subsequent Sync Word. There can be two values of 0101 or 1010 depending on the first bit of the Sync Word is 0 or 1. The Sync Word is a 64 bit code that is derived from the 24 bit LAP. The Sync Word is derived from the Master BD_ADDR's LAP if the packet is for Channel Access. IF the packet is for Device Access, the LAP of the Slave is used and if the packet is for Inquiry Access, a standard LAP value is used. The Trailer is a sequence of 4 bits of alternating 0 and 1 depending on the last bit of the Sync Word. The Trailer is used if the Sync Word is followed by the Header Field, but may also be used in other cases.



Address of the active participant: Slave id

Packet type

SCO, ACL?
Type?



Segment	TYPE	SCO link	ACL link
Control Packets	0000	NULL	NULL
	0001	POLL	POLL
	0010	FHS	FHS
	0011	DM1	DM1
Single Slot Packets	0100		DH1
	0101	HV1	
	0110	HV2	
	0111	HV3	
	1000	DV	
	1001		AUX1
3-Slot Packets	1010		DM3
	1011		DH3
	1100		
	1101		
5-Slot Packets	1110		DM5
	1111		DH5

Used for acknowledgements or flow control.

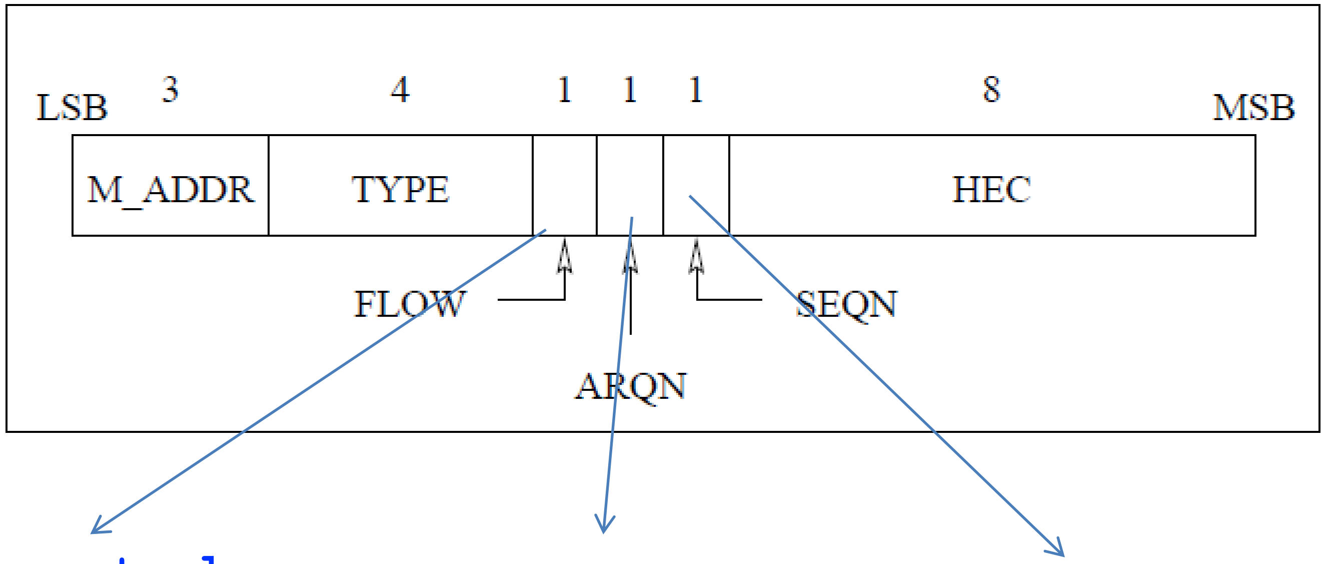
Used by the master to poll slaves. Requires acknowledgement.

Presence /absence of FEC, ARQ etc

Error correction

- **FEC, ARQ**
- **FEC=> reduce retransmission**
- **Overhead**
- **Flexible to protect payload: DM, DH**

$$\begin{tikzcd}
\text{DM} \arrow[d] & \text{DH} \arrow[d] \\
\text{With FEC} & \text{Without FEC}
\end{tikzcd}$$



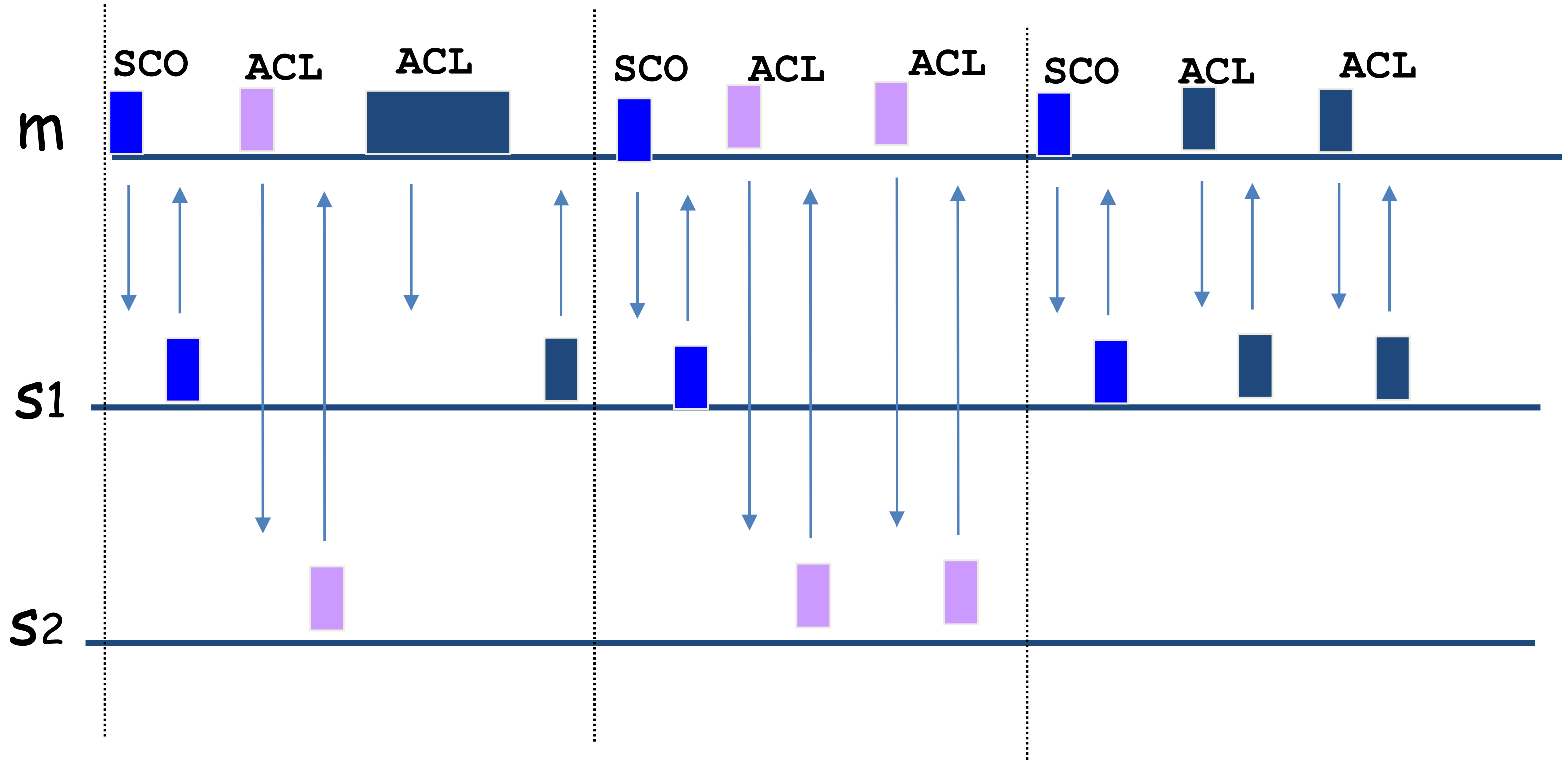
Flow control (ACL)
Rx is full (0, stop tx)
Rx empty (1, GO)

ACK
ARQN=1 (successful)
ARQN=0 (unsuccessful)

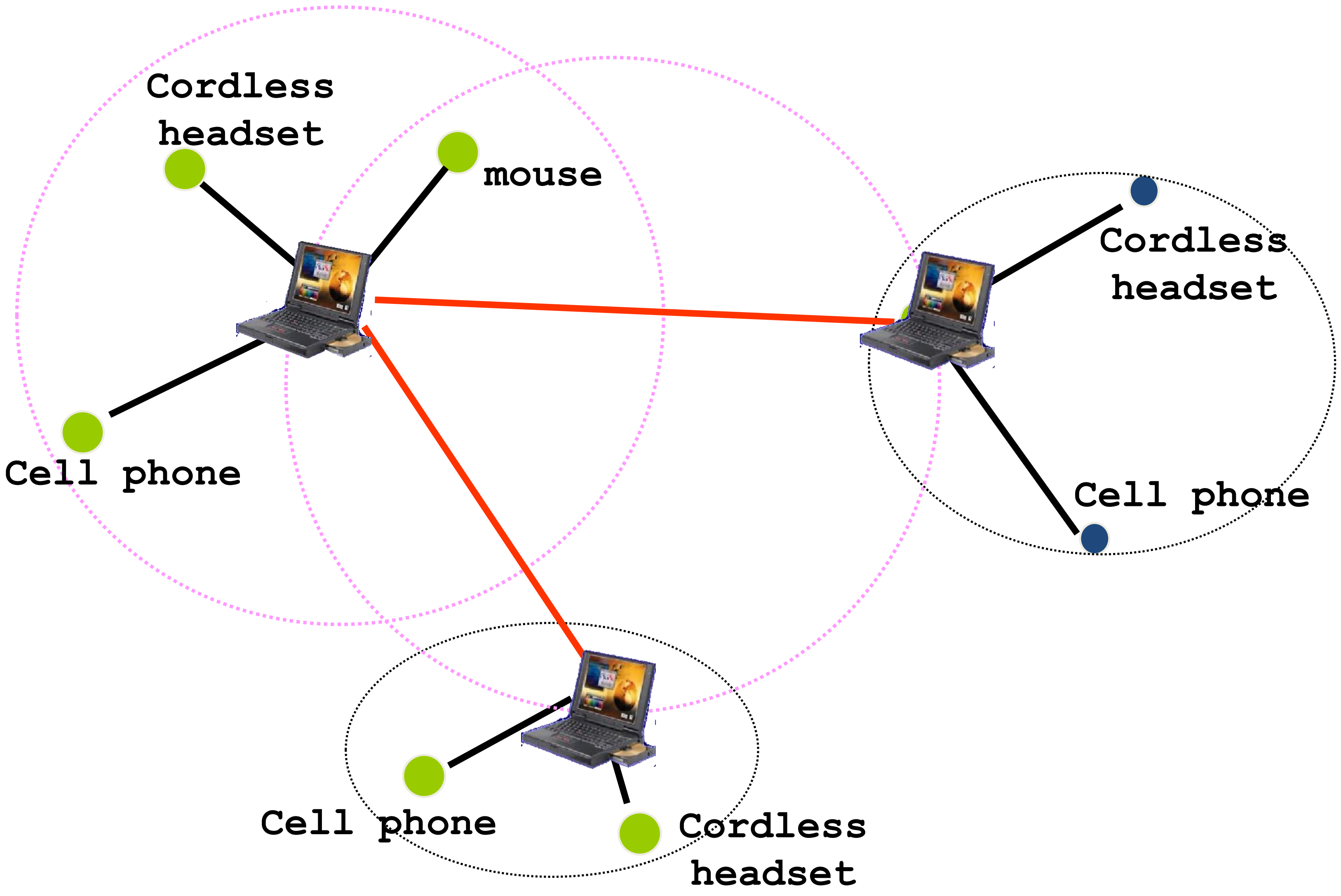
Distinguish new and retransmitted packet

Piggyback

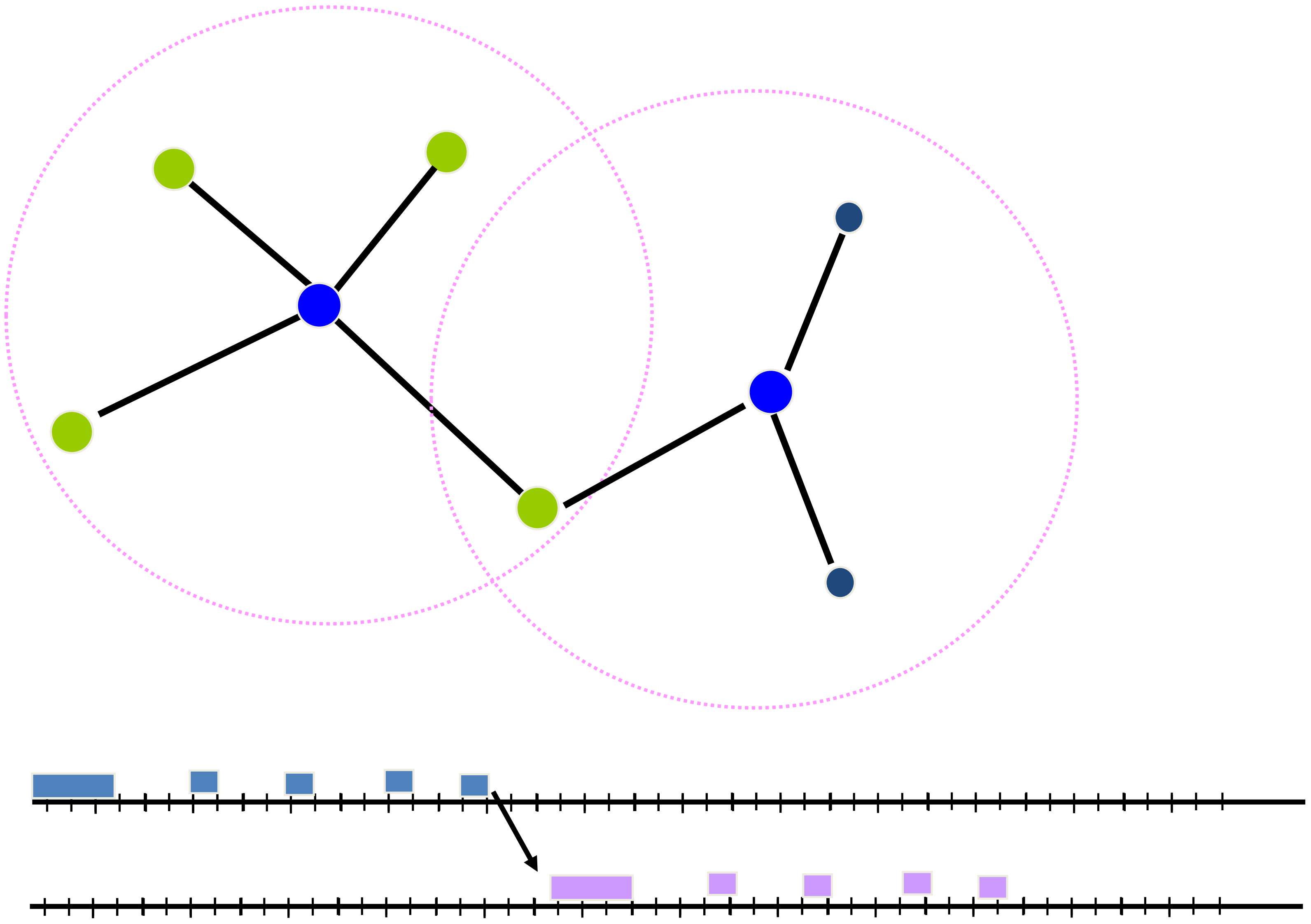
Mixed Link Example



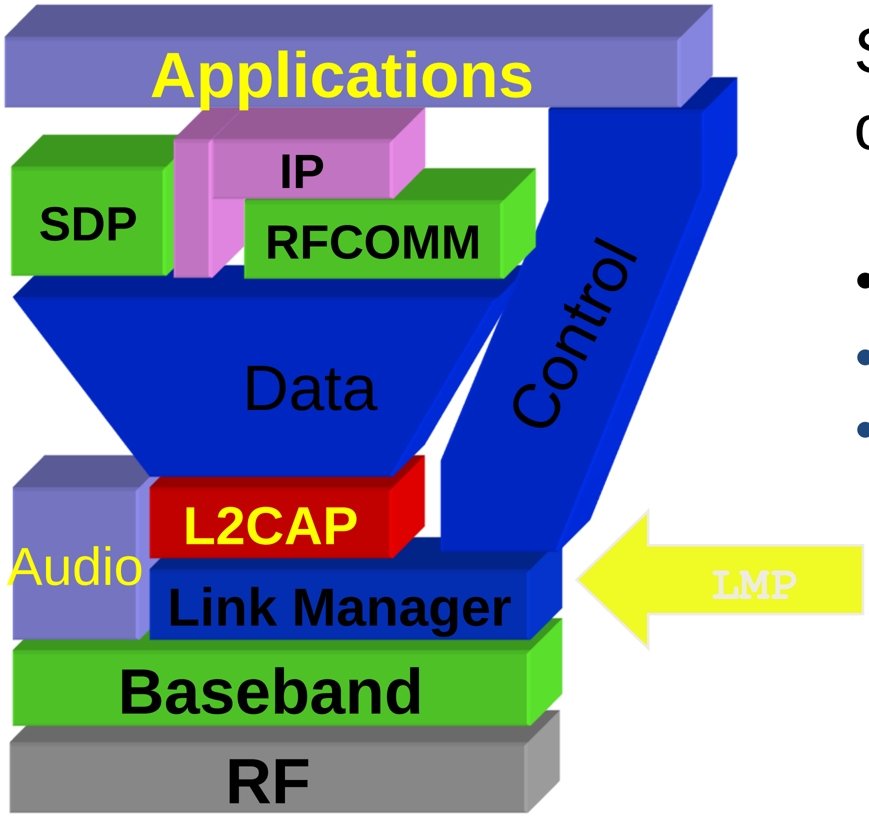
Inter piconet communication



Scatternet



Link Manager Protocol



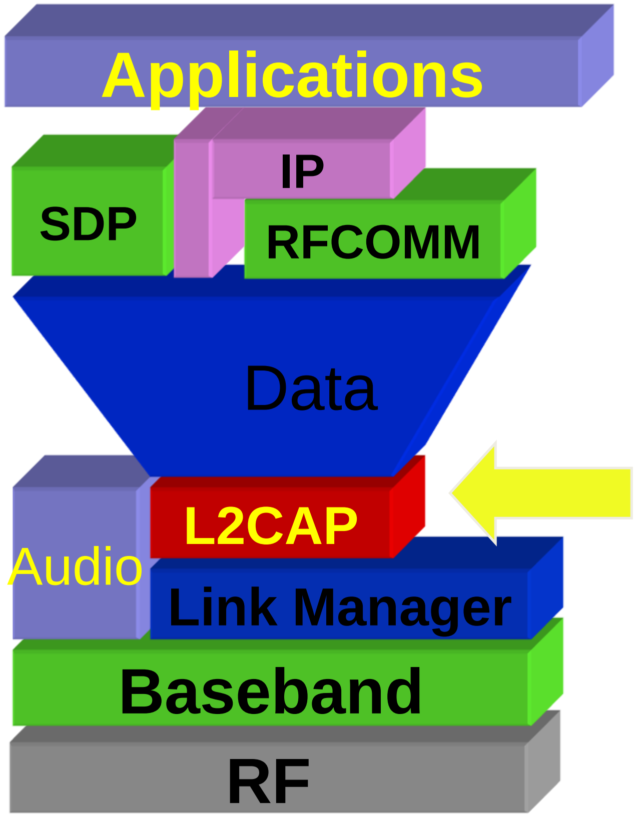
Setup and Management of Baseband connections

- Piconet Management
- Link Configuration
- Security

Link Manager Protocol

- **Piconet Management**
  - Piconet creation
  - Attach and detach slaves
  - Master-slave switch
  - Establishing SCO and ACL links
  - Handling of low power modes (Sniff, Hold, Park)
- **Link Configuration**
  - packet type negotiation
  - power control
- **Security functions**
  - Authentication
  - Encryption

L2CAP



Logical Link Control and Adaptation Protocol

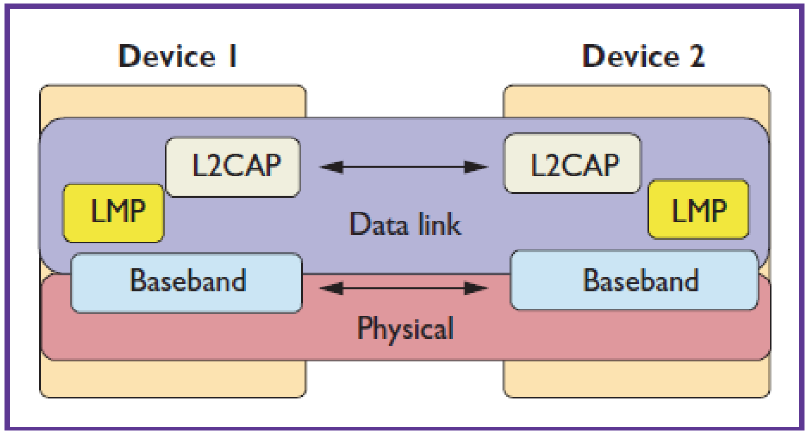
- L2CAP provides
  - Protocol multiplexing
  - Segmentation and Re-assembly
  -

L2CAP

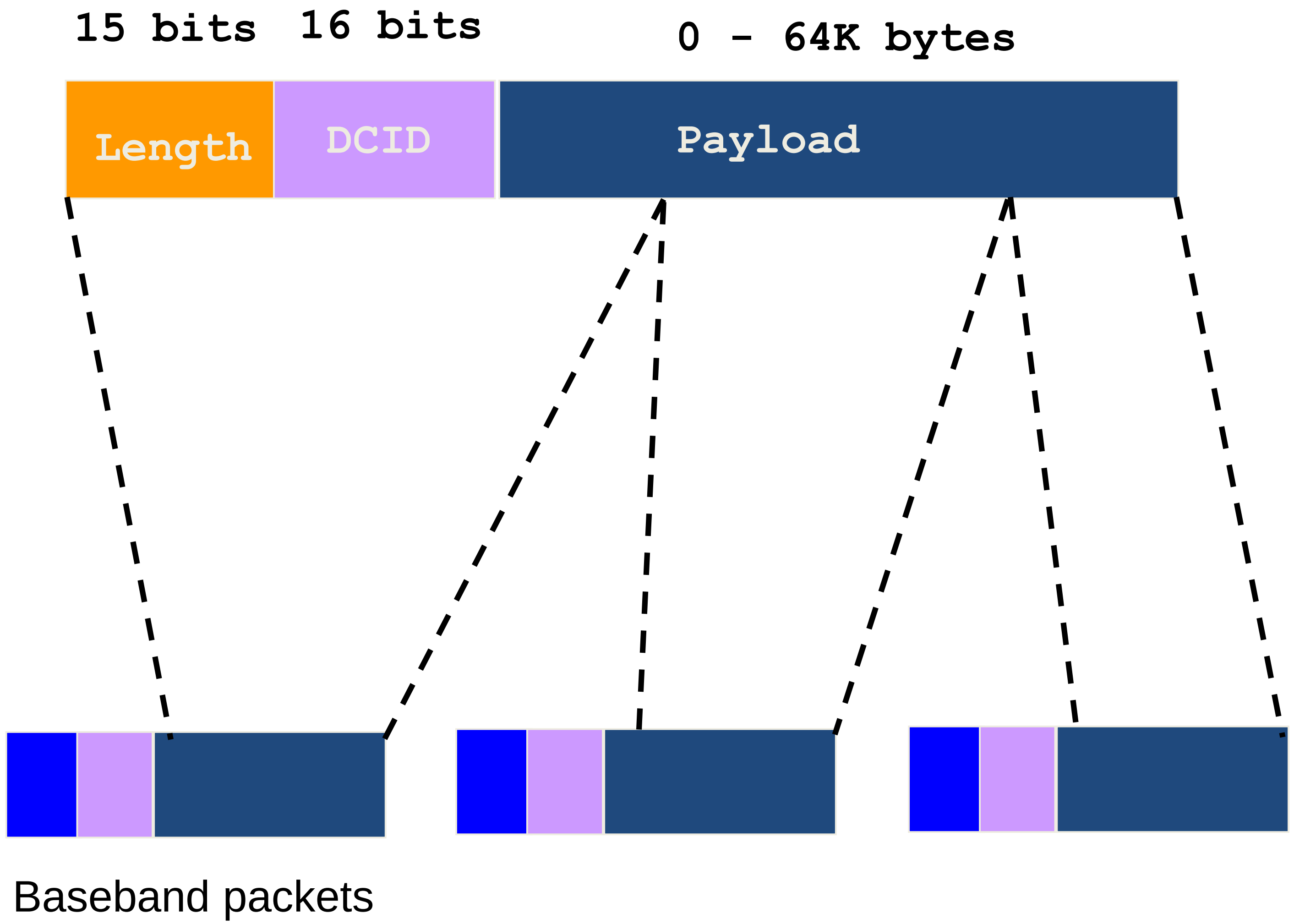
Does not support integrity, reliability checks.
Protocol multiplexing

- Mux/Demux of higher layer protocols is supported using channels- each higher layer protocol is carried in a different channel
- L2CAP must be able to distinguish between upper layer protocols such as the Service Discovery Protocol, RFCOMM, and Telephony Control.

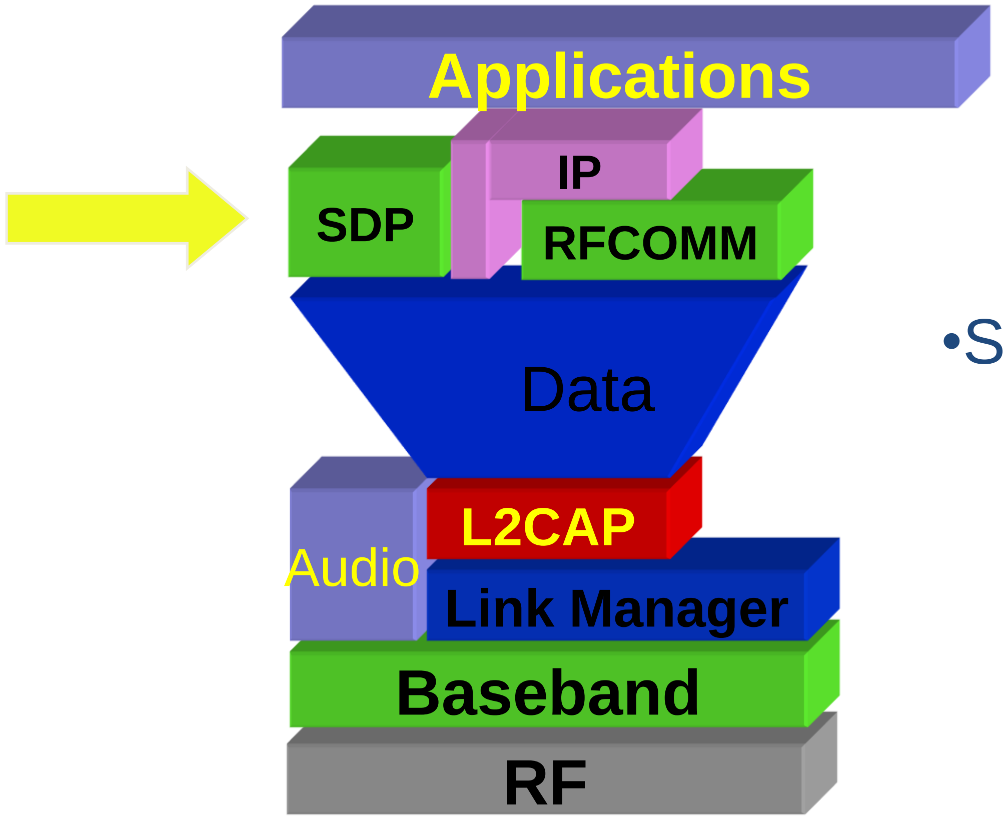
- Segmentation and Re-assembly
  - Data packets defined by the Baseband Protocol are limited in size
  - Large L2CAP packets must be segmented into multiple smaller Baseband packets prior to their transmission over the air
  - multiple received Baseband packets may be reassembled into a single larger L2CAP packet following a simple integrity check
  - Segmentation and Reassembly (SAR) functionality is absolutely necessary to support protocols using packets larger than those supported by the Baseband.



L2CAP Packet Format



Bluetooth Service Discovery Protocol



- SDP provides
  - Standard means for a BT device to query and discover services offered by a peer BT device
  - It’s a client-server protocol

Example usage of SDP

- Establish L2CAP connection to remote device
- Query for services
  - Search [if it knows UUID of the service] for specific class of service, or
  - browse for services
- Retrieve attributes that detail how to connect to the service
- Establish a separate (non-SDP) connection to user the service

Interoperability & Profiles

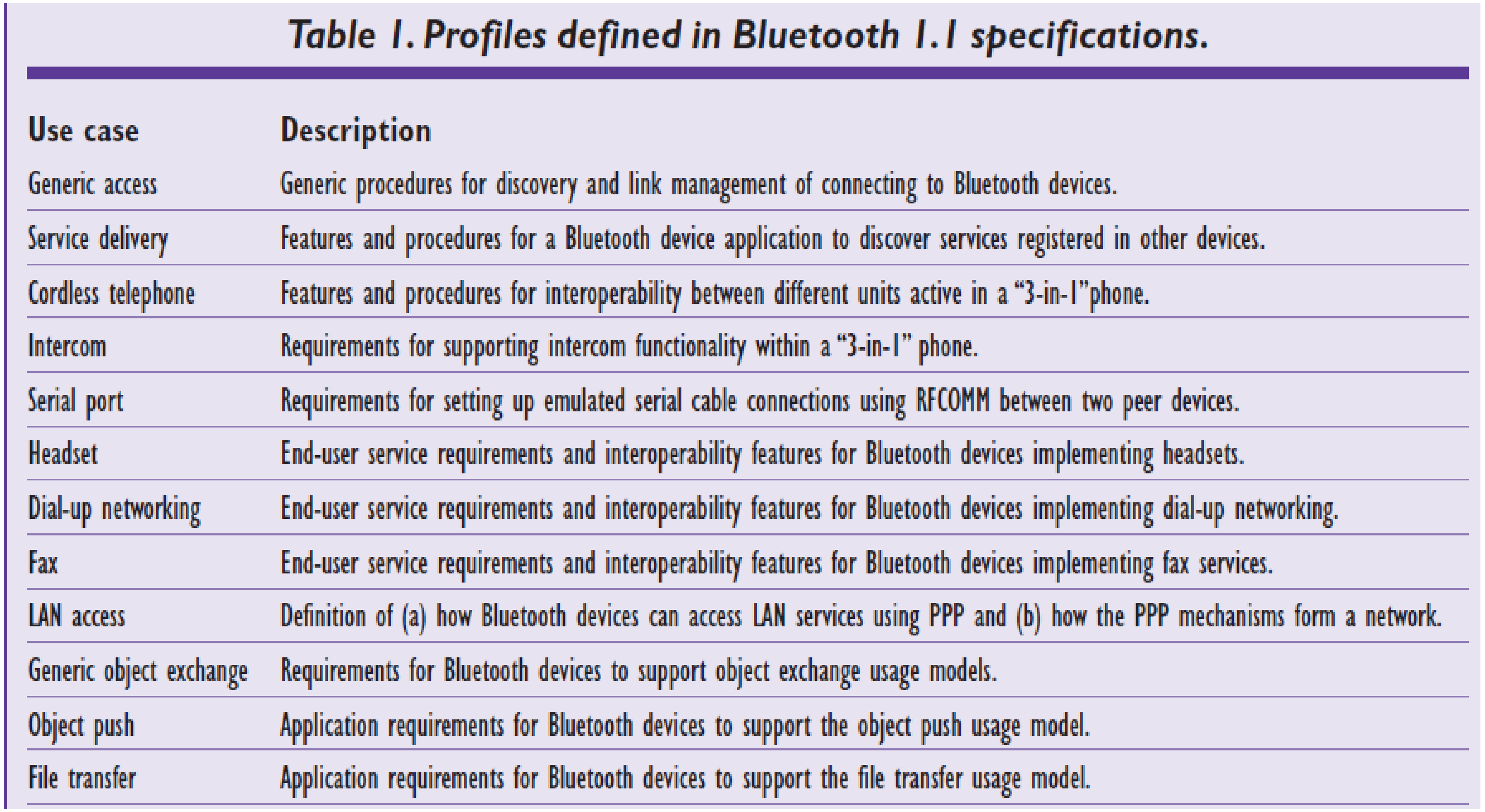
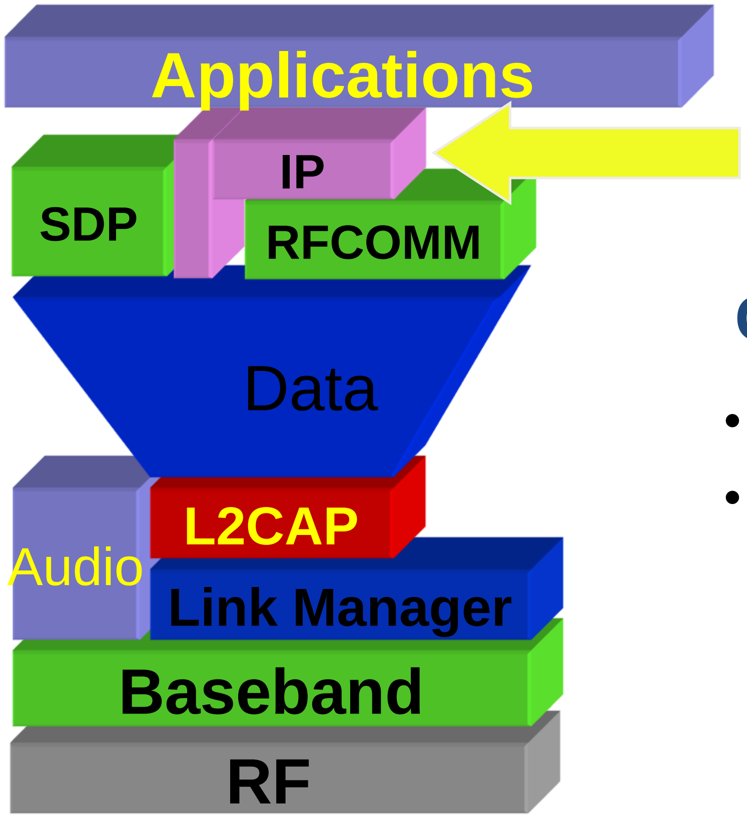


Table 1. Profiles defined in Bluetooth 1.1 specifications.

Use case	Description
Generic access	Generic procedures for discovery and link management of connecting to Bluetooth devices.
Service delivery	Features and procedures for a Bluetooth device application to discover services registered in other devices.
Cordless telephone	Features and procedures for interoperability between different units active in a "3-in-1"phone.
Intercom	Requirements for supporting intercom functionality within a "3-in-1" phone.
Serial port	Requirements for setting up emulated serial cable connections using RFCOMM between two peer devices.
Headset	End-user service requirements and interoperability features for Bluetooth devices implementing headsets.
Dial-up networking	End-user service requirements and interoperability features for Bluetooth devices implementing dial-up networking.
Fax	End-user service requirements and interoperability features for Bluetooth devices implementing fax services.
LAN access	Definition of (a) how Bluetooth devices can access LAN services using PPP and (b) how the PPP mechanisms form a network.
Generic object exchange	Requirements for Bluetooth devices to support object exchange usage models.
Object push	Application requirements for Bluetooth devices to support the object push usage model.
File transfer	Application requirements for Bluetooth devices to support the file transfer usage model.

IP over Bluetooth V 1.0



GOALS

- Internet access using cell phones
- Connect PDA devices & laptop computers to the Internet via LAN access points