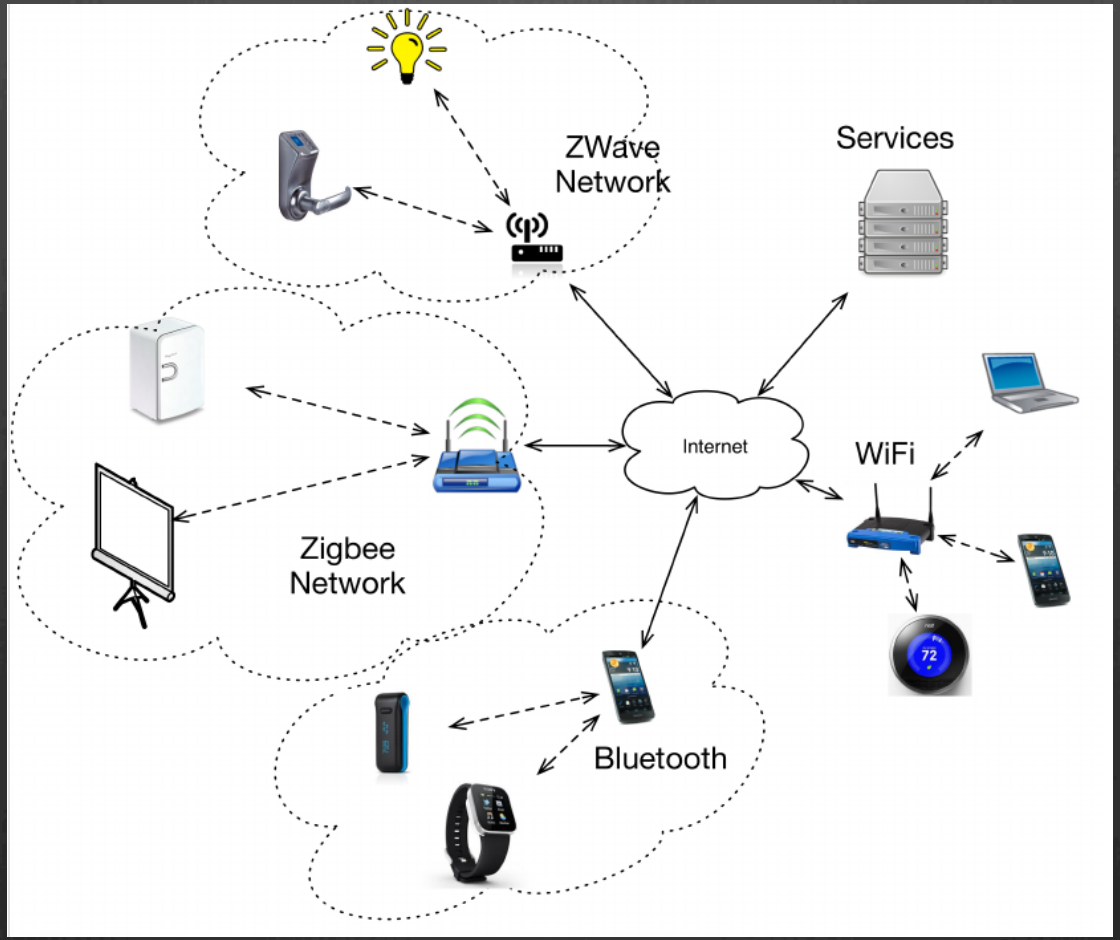
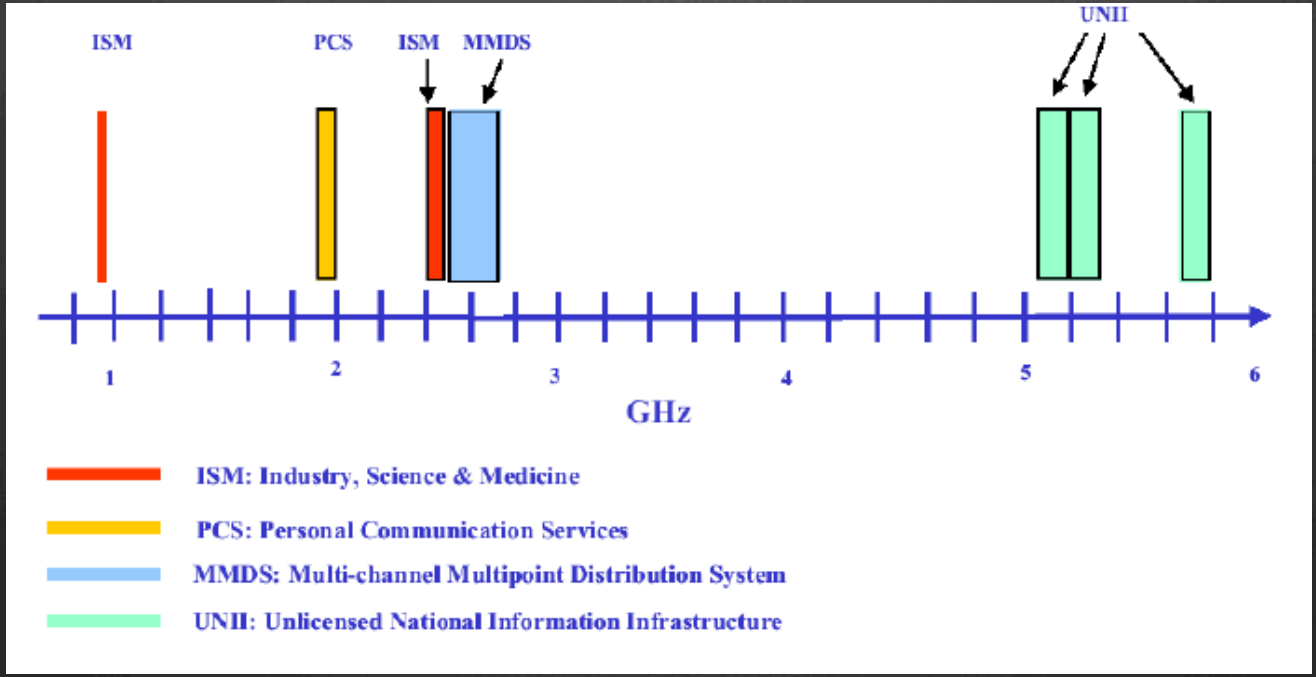
Introduction to Wireless Networks

Objet Communicants
Dino Lopez
dino.lopez@unice.fr
http://www.i3s.unice.fr/~lopezpac/

Objectives of this course


ZWave
Network
Services
Internet
WiFi
Zigbee
Network
Bluetooth

The ISM frequency band

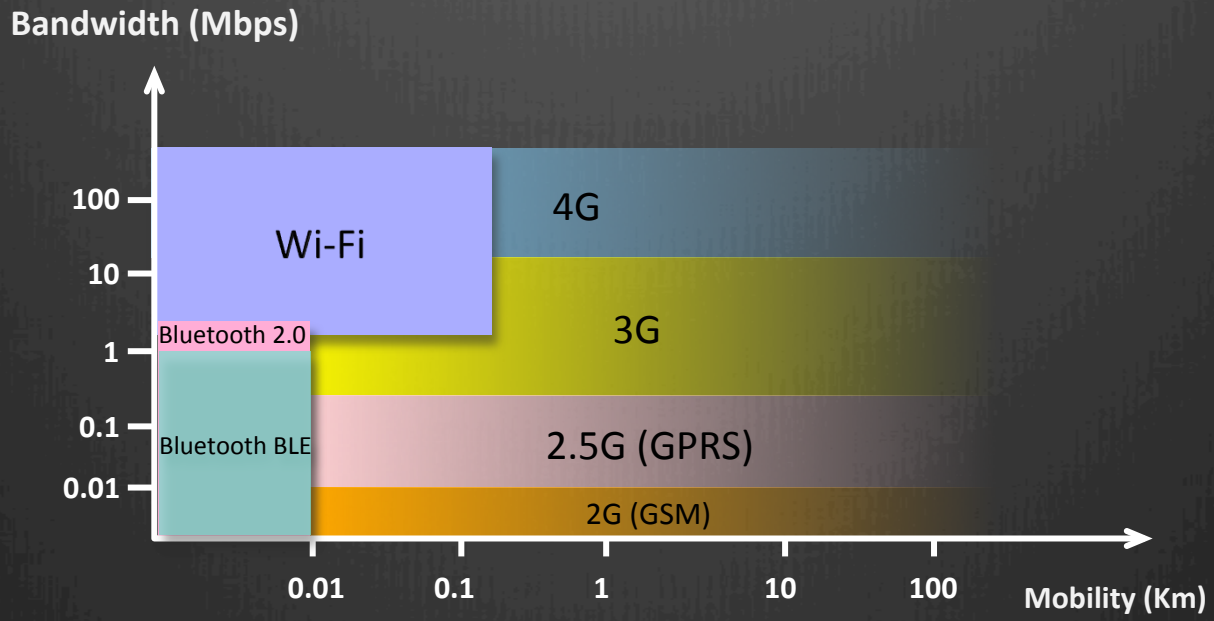


http://www.art-telecom.fr/

Wireless standards – coverage range classification

- TAGGING
  - RFID
- WPAN
  - IEEE 802.15
    - ✓ IEEE 802.15.1 – Bluetooth
    - ✓ IEEE 802.15.3 – UWB (Ultra Wide Band)
    - ✓ IEEE 802.15.4 – ZigBee
- WLAN
  - IEEE 802.11 (Wi-Fi)
  - IEEE 802.11b, a, g
  - IEEE 802.11n
  - IEEE 802.11s
- WMAN
  - IEEE 802.16
  - IEEE 802.16-2004
  - IEEE 802.16e/IEEE 802.20 (Wi-Mobile)
- WRAN
  - IEEE 802.22 Utilisation des bandes TV 54-698 (Wi-RAN)

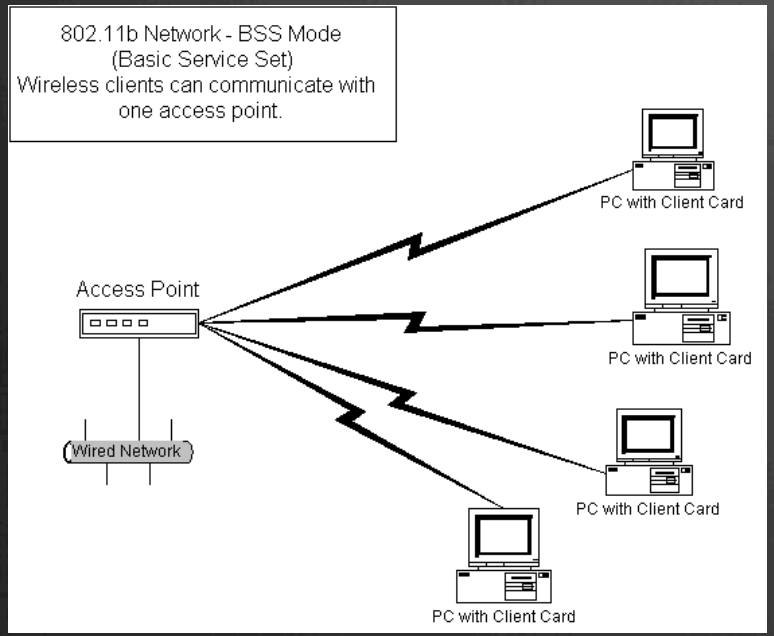
Wireless Spectrum Sharing

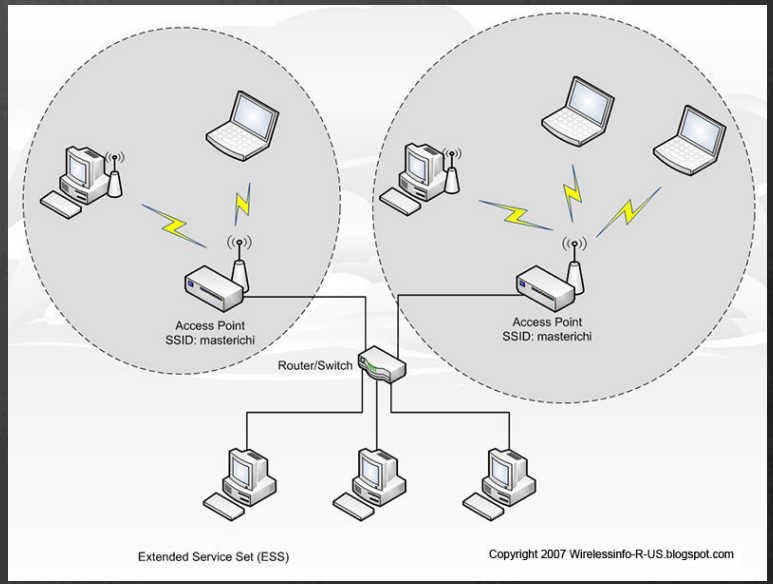


Wi-Fi – IEEE802.11

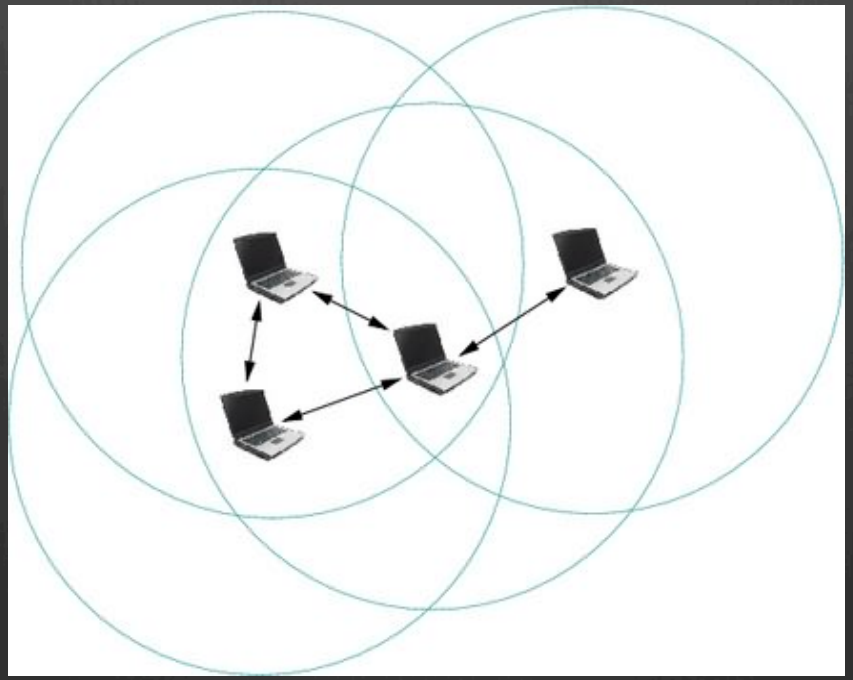
- Completely compatible with IP networks
  - Only the two lowest layers are modified. i.e. the Data Link and the Physical Layers
- 2 working modes
  - Ad hoc mode. Mobile nodes (MN) communicates with neighbors (MNs also) in the coverage area
  - Infrastructure mode. Nodes communicates with a non mobile node (the Access Point – AP)
- 3 physical layers
  - Direct Sequence Spread Spectrum - DSSS
  - Frequency Hopping Spread Spectrum - FHSS
  - Infrared

Wi-Fi – The Infrastructure Mode

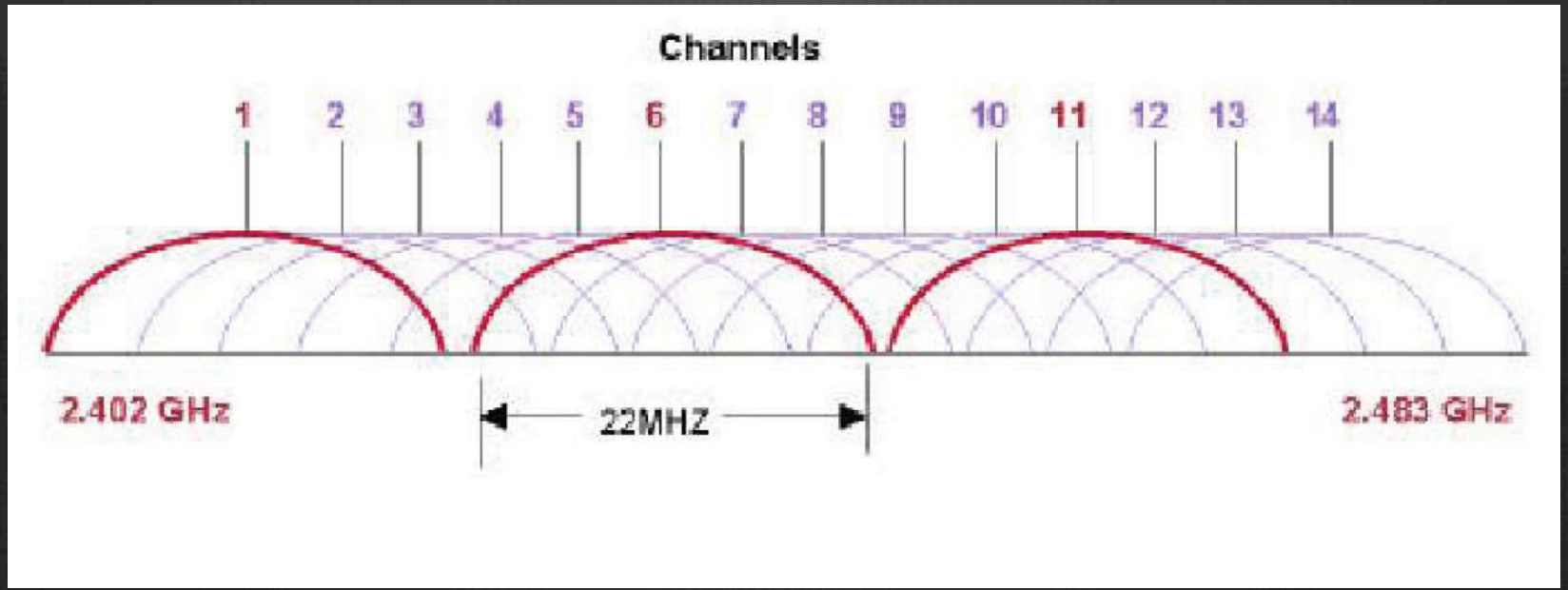




Wi-Fi – The Ad hoc mode

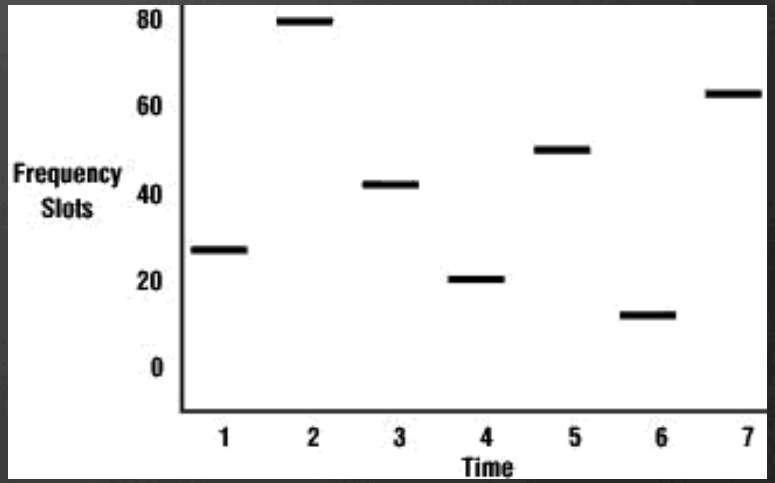


The DSSS Physical Layer



The FHSS Physical Layer

- 75 channels of 1MHz each
- Switching channel every 400ms
- Rate = 2Mbps



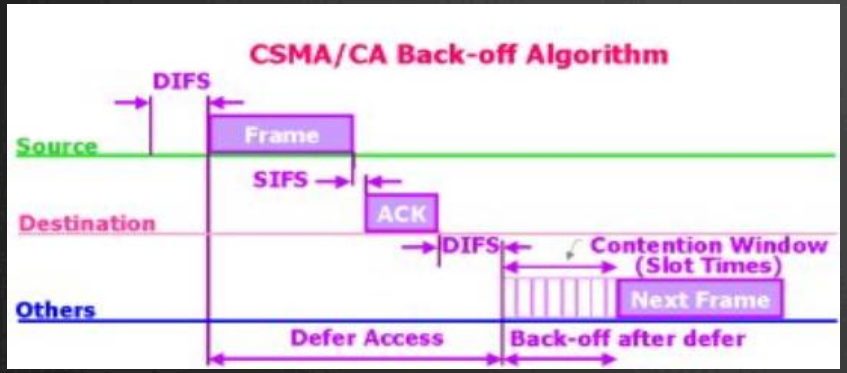
Some Common Problems/ Questions in Wireless Networks

Channel Access Method

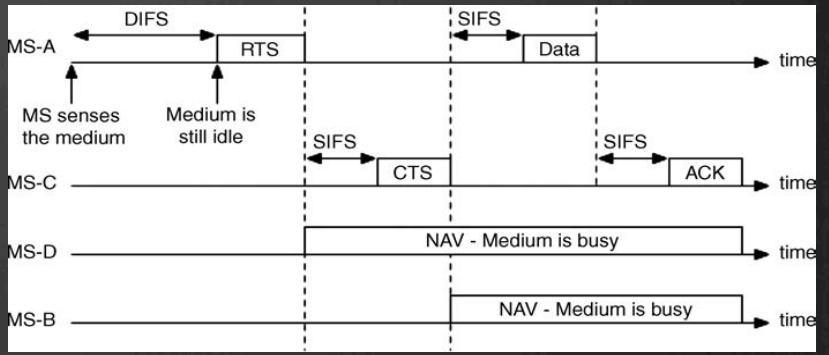
- Distributed coordination
- Single point controlled

Distributed Coordination Function in 802.11

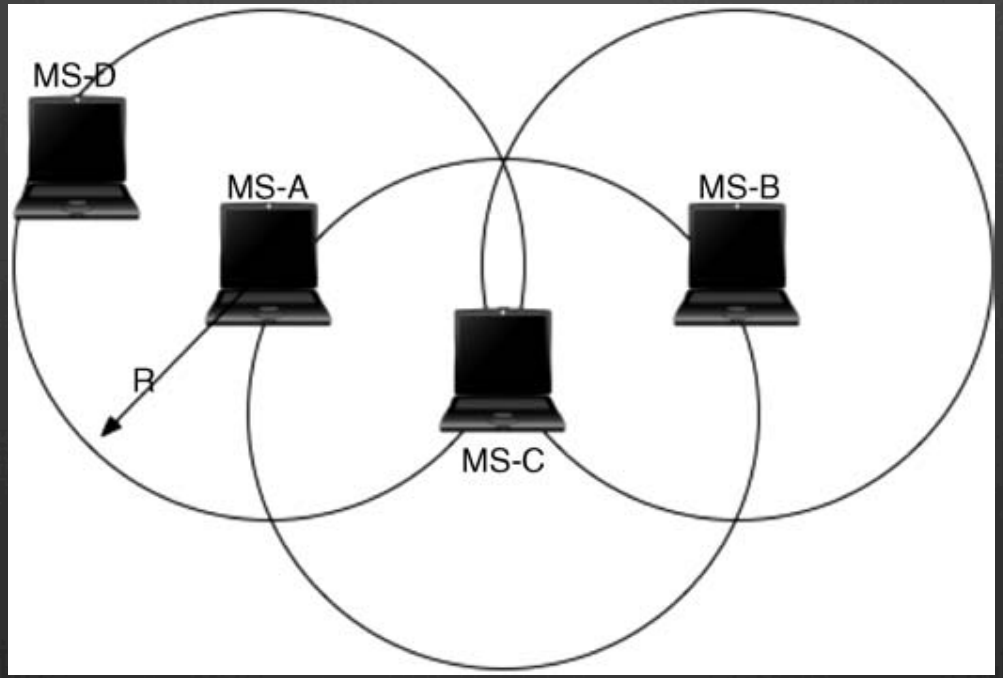
CSMA/CA



CSMA/CA w RTS/CTS



The Hidden Node and the Exposed Node Problems

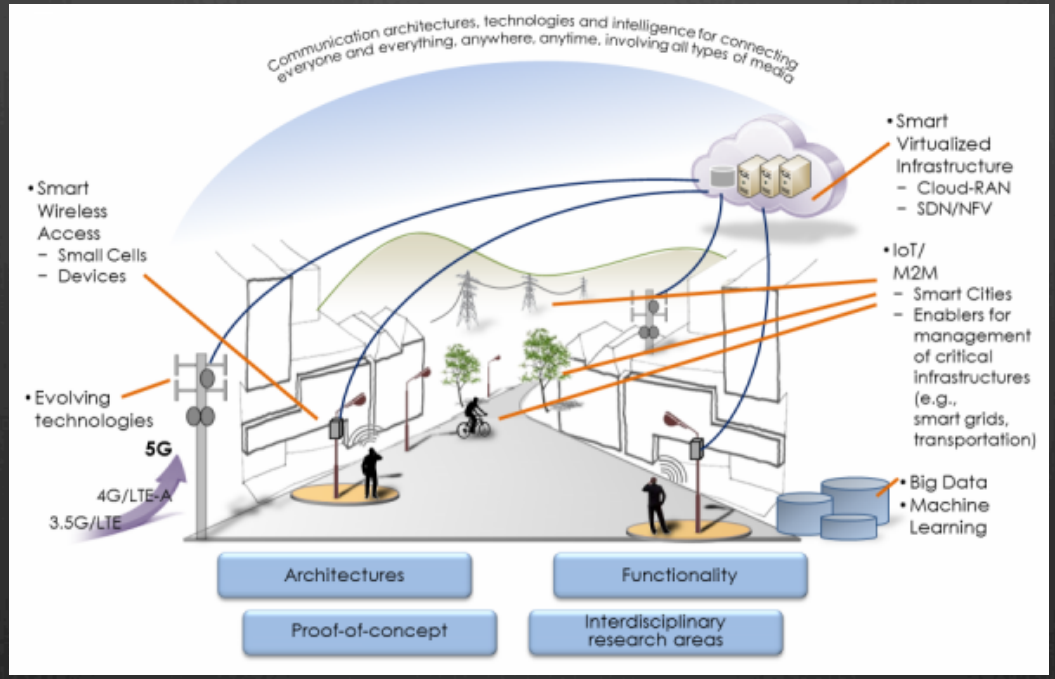


The Wireless World Research Forum

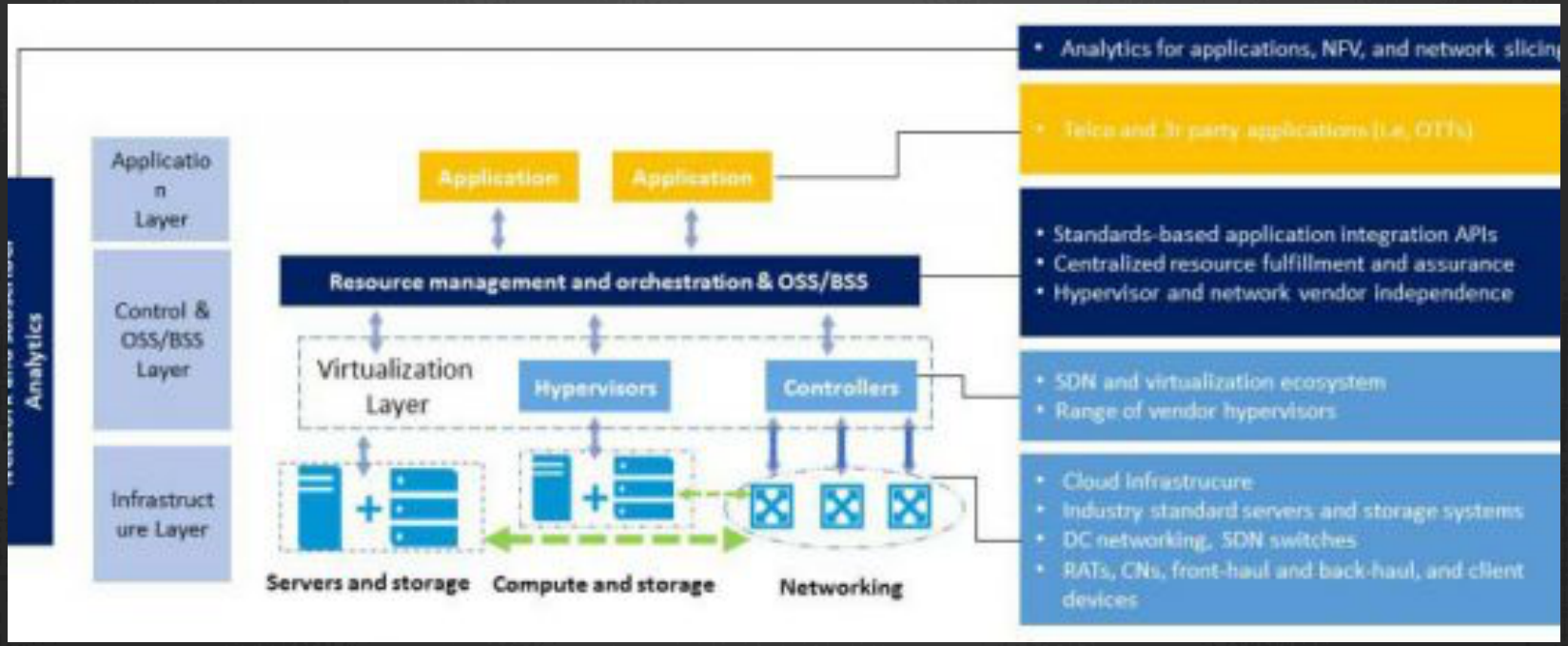
Working Group C

Communication Architectures and Technologies

Scope and Network Vision – up to 2015



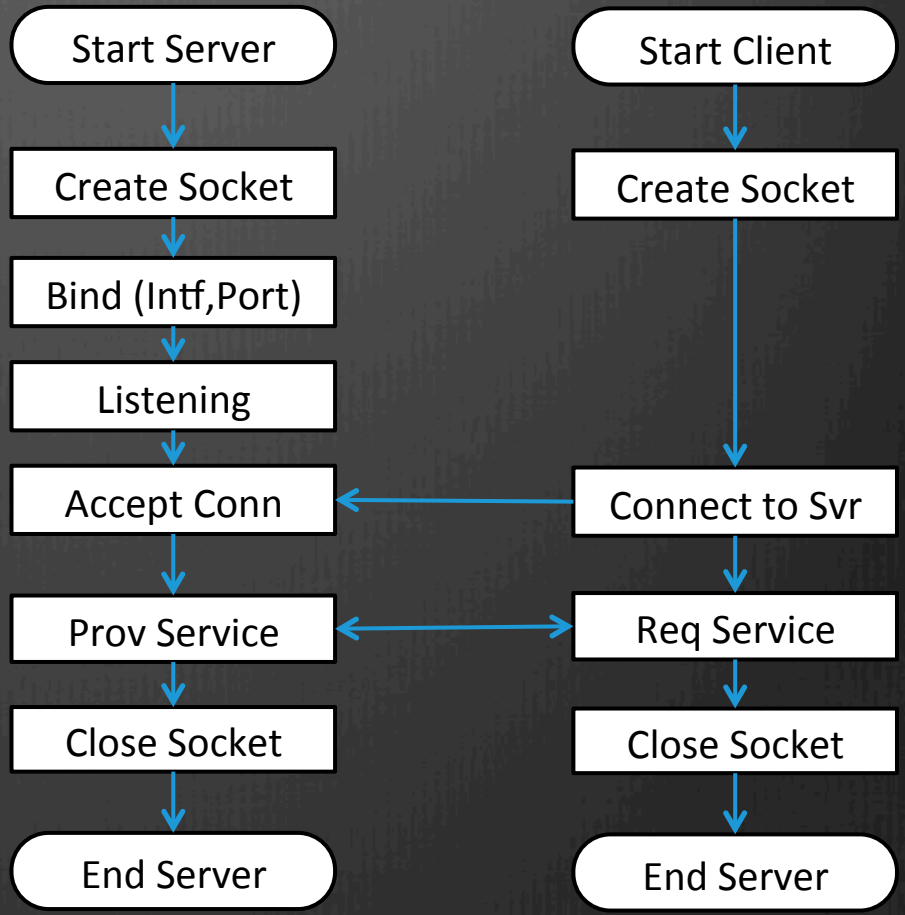
Scope and Network Vision – 2016



The Sockets

Connected Sockets

- Bluetooth connected sockets follows the same principle of IP connected sockets
- The flow-chart of a connected socket



The example of the TCP socket

Server

```
1.    import socket

2.    HOST = ''    # any available interf
3.    PORT = 5000  # Arbitrary non-priv port
4.    s = socket.socket(socket.AF_INET,
      socket.SOCK_STREAM)
5.    s.bind((HOST, PORT))
6.    s.listen(1)
7.    conn, addr = s.accept()
8.    while 1:
9.        data = conn.recv(1024)
10.       if not data: break
11.       conn.sendall("Hi!")
12.   conn.close()
13.   s.close()
```

Client

```
1.    import socket

2.    HOST = '10.0.0.2'    # The remote
      host
3.    PORT = 5000           # The remote
      port
4.    s = socket.socket(socket.AF_INET,
      socket.SOCK_STREAM)
5.    s.connect((HOST, PORT))
6.    val = "Hello!"
7.    s.sendall(val)
8.    data = s.recv(1024)
9.    s.close()
10.   print "Received: %s" %(data)
```

Bluetooth

Origin of the name

- Harald I Bleutooth (in Danish, Harald Blåtand) (b. c. 910—d. c. 987), king of Denmark was credited with the first unification of Denmark and Norway
- Ericsson, inspired on the history of Harald I, proposed a new technology which aims at linking different devices from different manufacturers, but at low-cost

Objectives of the Bluetooth Technology

- Cable replacement
  - Ericsson decided to investigate the feasibility of a low-power and low cost radio interface between mobile phones and their accessories
  - Today, the Bluetooth technology is supported by the Bluetooth SIG (Special Interest Group), founded by Ericsson, IBM, Intel, Nokia and Toshiba in 1998
- Personal Area Networks
- Ad hoc networks

Advantages and Disadvantages

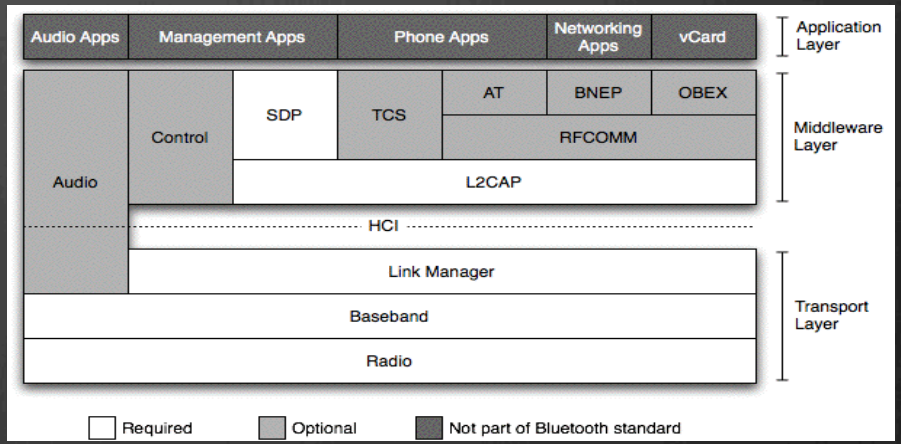
- Advantages
  - Eliminate wires and cables between stationary and mobile devices
  - Facilitates both data and voice communication
  - It is inexpensive
  - Possibility of automatically establish communication
  - Automatically create ad hoc networks
- Disadvantages
  - Low rate
  - Low range
  - Security

Bluetooth Specifications

- Bluetooth 1.0 and 1.0B
- Bluetooth 1.1
  - IEEE 802.15.1–2002
- Bluetooth 1.2
  - IEEE 802.15.1–2005
- Bluetooth 2.0 + EDR and 2.1 + EDR
- Bluetooth 3.0 + HS
  - Relies on 802.11
- Bluetooth 4.0 + LE
- Bluetooth 4.1 and 4.2
- Bluetooth 5
  - Higher rate → 2Mbps

The Bluetooth Architecture

- The horizontal approach
  - Similar to the OSI model
- The vertical approach
  - The application drives the Bluetooth layered implementation
  - Implementation of profiles
- Mandatory and optional protocols

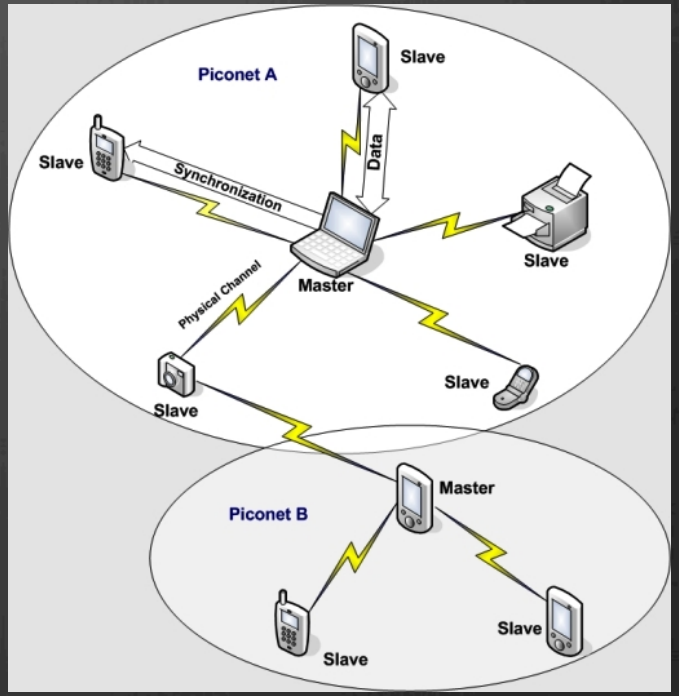


The Radio Layer – Power Classes

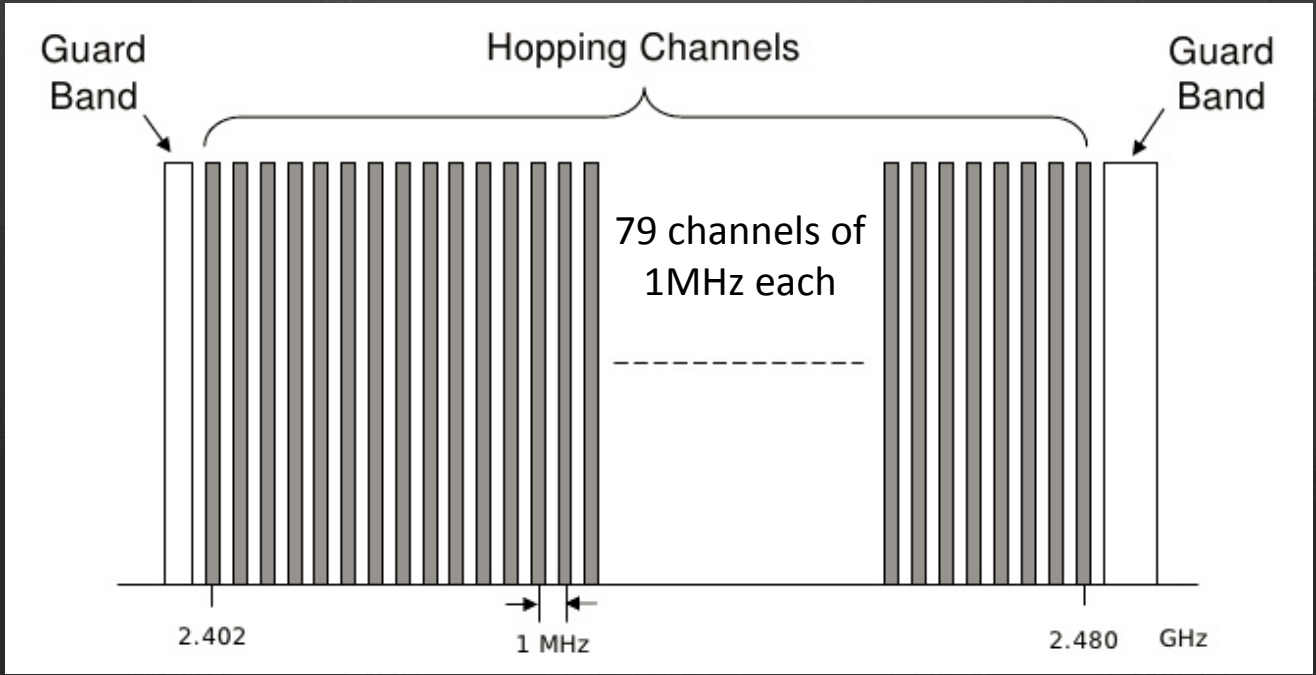
- Class 1
  - maximum of 100mW and a minimum of 1mW
  - Expected range of 100m
  - This device is capable of controlling power, in steps of 2 to 8dB
- Class 2
  - Maximum of 2.5mW and a minimum of 0.25 mW
  - Expected range 10m
  - Power control is optional
- Class 3
  - Maximum of 1 mW
  - Expected range 10cm
  - Power control is optional

The Baseband Layer – The Network Topologies

- The piconet
  - The piconet is a collection of two or more devices sharing the same physical channel
  - One master can control up to seven slaves
- The scatternet
  - Interconnection of several piconets
  - Bluetooth does not natively support routing

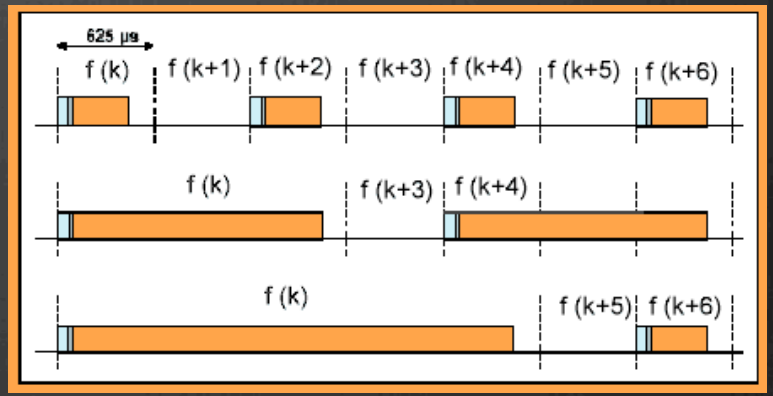


The Frequency Band Division



Physical Channel Construction

- Physical channels: pseudo-random code driving the frequency hopping sequence
  - Time Division Duplex (TDD) scheme
  - 1600 hops/s in the connected mode
  - 3200 hops/s in inquiry and page mode
- The channel is divided in timeslots (TS) of 625microsecs
- The Master sends data packets in an odd-numbered TS and the slave replies in the following even-numbered TS
- Packets can take 1, 3 or 5 TSs



The 4 Physical Channels

- The basic piconet channel
  - Used by connected devices during normal operation
- The adapted piconet channel
  - Used to avoid interference and easy coexistence with other systems working in the same band
  - A subset of the 79 available channels is used (minimum 20)
  - Slaves use the same frequency as the master in its preceding transmission
- The inquiry scan channel
  - Used in inquiry mode in order for a device to be discovered
- The paging scan channel
  - Used to page a connectable device

The 4 physical links

- Active Physical Link
- Parked Physical Link
- Inquiry Physical Link
- Active Physical Link

Logical Links

- Synchronous Connection-Oriented (SCO) logical transport
- Extended Synchronous Connection-Oriented (eSCO) logical transport
- Asynchronous Connection-Oriented (ACL) logical transport
  - Transmission employs an ARQ protocol
- Active Slave Broadcast (ASB) logical transport
  - Control message for the whole piconet
- Parked Slave Broadcast (PSB) logical transport
  - Control message for the whole piconet

Low Power Modes

- Hold mode
  - Physical link is only active during slots reserved for the operation of synchronous links. Either, the master or the slave can require to establish a hold mode
- Sniff mode
  - Device wakes up periodically to communicate with the master or engage any activity on another physical channel
  - SCO and eSCO are not affected
  - Reserved for slaves
- Parked state
  - Used to stay synchronized with the master without being an active member of the piconet
  - All logical links are disable, except the PSB link

Logical Link Control and Adaptation Protocol (L2CAP)

- L2CAP provides
  - Multiplexing through Channel Identifiers (CID)
  - Segmentation and reassembly operations
    - ✓ Up to 64 kB service data unit (payload)
  - Per-channel flow control and retransmission
- L2CAP does not transport voice or synchronous data, although VoIP would be carried by L2CAP
- L2CAP provides asynchronous connection-oriented and connectionless channels, with some QoS level
  1. Basic flow control - default
  2. Flow control mode
  3. Retransmission mode

Service Discovery Protocol

- SDP enables a Bluetooth device to
  - Inquire what services are available in its environment and their characteristics.
  - Inform the list of services provided by itself
- Each Bluetooth device maintains a list of Service Records. Each record contains the list of attributes of a given application that characterize the provided service
- Interoperability between Bluetooth devices is enabled with the introduction of Profiles
- Services are identified by UUIDs (128 bits)
  - uuid32 – assigned or reserved
  - uuid16 – currently assigned numbers

Generic Access Profile

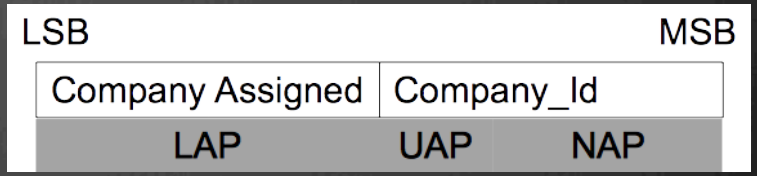
- GAP defines the generic procedures to
  - Discover Bluetooth devices
  - Connect Bluetooth devices
  - Use different levels of security
- GAP describes the use of the lower layers of the Bluetooth protocol stack, as well as some higher layer protocols
- GAP defines the generic terms that can be presented to the customers

GAP

- The Bluetooth device address (BD_ADDR)
  - The baseband address coded on 48 bits
- The Bluetooth device name
  - Coded on 248 bytes maximum
- The Bluetooth passkey (Bluetooth PIN)
  - Used to authenticate two Bluetooth devices
  - Maximum length = 16 bytes
- The class of device
  - Types of services supported by the device

Addresses of Bluetooth Devices

- Bluetooth Device Address (BD_ADDR)
  - Lower Address Part (LAP) – 24 bits
  - Upper Address Part (UAP) – 8 bits
  - Non-significant Address Part (NAP) – 16 bits
- Active Member Address (AM_ADDR)
  - Identify active piconet members (3 bits)
  - Broadcast address = 000
- Parked Member Address (PM_ADDR)
  - Identify parked devices (8 bits)
- Access Request Address (AR_ADDR)
  - The parked device use this address to become an active member

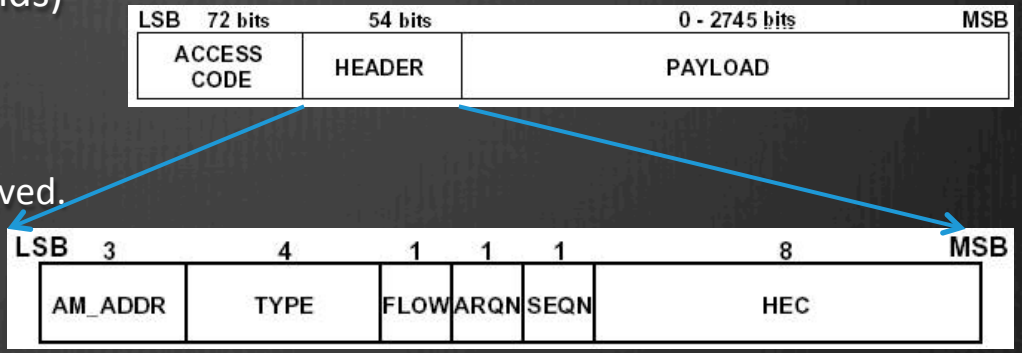


Packet Format

- 3 types of access code
  - Device AC (page, page scan)
  - Channel AC (identify the piconet)
  - Inquiry AC
    - ✓ General IAC (GIAC) 0x9E8B33
    - ✓ Dedicated IAC (DIAC) – from 0x9E8B00 to 0x9E8B3F

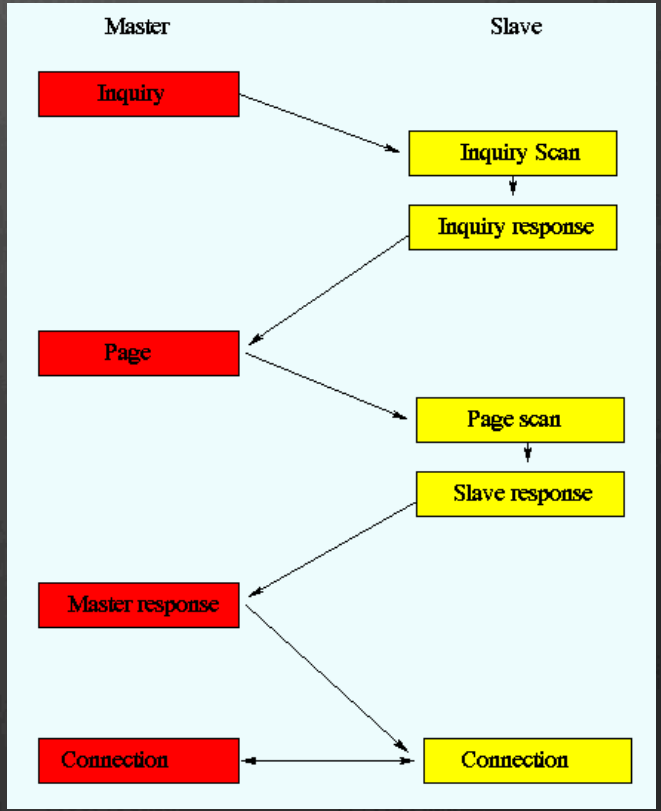
Packet Format

- Header
  - AM_ADDR. Active Member ADDR
  - TYPE. (SCO, ACL or commands)
  - FLOW (flow control)
    - ✓ 0 = stop; 1 = continue
  - ARQ
    - ✓ 1 Packet successfully received. 0 otherwise
  - SEQN
    - ✓ Sequence bit
  - Header Error Control (HEC)
    - ✓ G(X) = X^8 + X^7 + X^5 + X^2 + X + 1



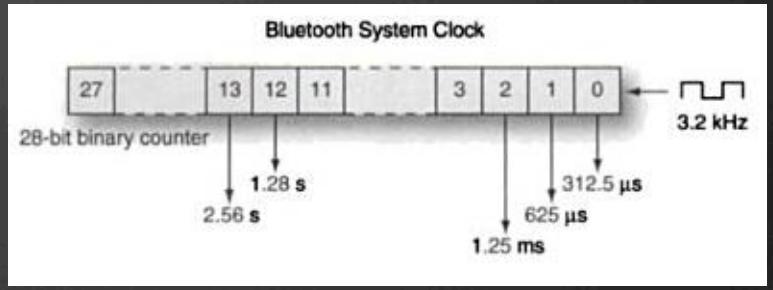
Discovery and Connection

- Inquiring and paging
  - Master: 3200 hops/s
  - Slave: 1.28 hops/s



Bluetooth Synchronization

- Every device has its own native clock (CLKN)
- The clock is implemented with a 28-bit counter
- driven by a low power oscillator when in STANDBY, Park, Hold and Sniff mode
- Driven by a crystal oscillator



Bluetooth and Linux

Bluez

- Bluez provides support for the core Bluetooth layer and protocols
  - Standard Bluetooth
  - Bluetooth LE
- Initially developed by Qualcomm, is now an open source project under the terms of GPL (http://www.bluez.org/)
  - Available in several Linux distributions
- Bluez provides several command-line tools to configure Bluetooth devices and debug the applications

Some Bluez Commands

- hciconfig
  - Configure the basic properties of Bluetooth adapters
  - If invoked without arguments, it will display the status of the current adapters attached to the computer
  - Bluetooth devices are usually identified by “hciX”, where X is the number of the device
- hcitool
  - Search and detect nearby Bluetooth devices
  - Test and show information about low-level Bluetooth connections
- hcidump
  - For low-level debugging of connection setup and data transfer

Some Bluez Commands

- sdptool
  - Browsing and searching services advertised by nearby devices
  - Basic configuration of the SDP services offered by the local machine
- l2ping
  - The “ping” tool for Bluetooth devices
- uuidgen
  - Generates UUIDs
  - Useful to advertise applications with non standard UUID
- gatttool
  - Client implementation of the GATT protocol. Manipulates the BLE attributes

Package Installation

- We will need the following packages (to install with the APT package manager)
  - bluez-*
  - libbluetooth-dev
  - python-bluez
- For the labs, you will need to stop the bluetooth daemon, and lunch it again in compatibility mode (# bluetoothd -C)