LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Praktikum Mobile und Verteilte Systeme

# Connectivity with Android

Prof. Dr. Claudia Linnhoff-Popien
Philipp Marcus, Mirco Schönfeld
http://www.mobile.ifi.lmu.de

Sommersemester 2015

# Connectivity with Android

**Today:**

- Bluetooth
- NFC
- 802.11 & Wifi Direct / P2P

**Further Reading:**

- Android Developer Guide (http://developer.android.com/guide/topics/connectivity/)

**Next week:**

- Client-Server Communication

# Bluetooth: Introduction (I)

**Bluetooth**

- Low-cost, radio-based wireless network technology
- Standardized by the Bluetooth Special Interest Group (SIG), a consortium founded in 1998 by Ericsson, Intel, IBM, Nokia, and Toshiba
- Uses the license-free 2.4 GHz band

- Use Cases
  - Wireless internet access
  - Synchronization
  - Wireless headset
  - Data exchange

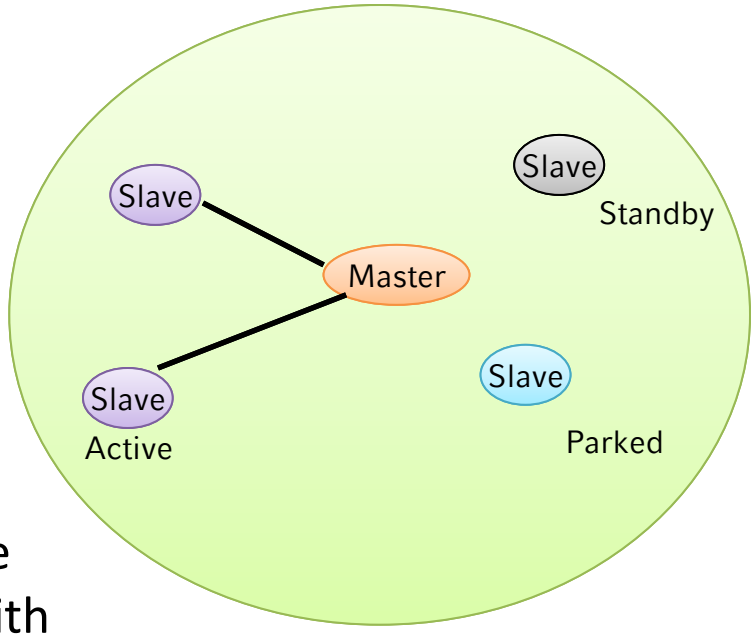
# Bluetooth: Introduction (II)

- 2.4 GHz ISM band (2400-2483,5MHz), 79 RF channels, 1 MHz carrier spacing
  - Channel 0: 2402 MHz … channel 78: 2480 MHz
  - Guard bands: Lower Guard Band 2MHz, Upper Guard Band 3,5MHZ
  - G-FSK modulation (1 MSymbol/s), 1-100 mW transmit power
- FHSS and TDD
  - **Frequency hopping** with 1600 hops/s
  - Hopping sequence in a pseudo random fashion, determined by a master
  - **Time division duplex** for send/receive separation
- Voice link – SCO (Synchronous Connection Oriented)
  - FEC (forward error correction), no retransmission, 64 kbit/s duplex, point-to-point, circuit switched
- Data link – ACL (Asynchronous Connection Less)
  - Asynchronous, fast acknowledge, point-to-multipoint, up to 433.9 kbit/s symmetric or 723.2/57.6 kbit/s asymmetric, packet switched
- Topology
  - Overlapping piconets (stars) forming a scatternet

# Bluetooth: Piconets

**Piconet**

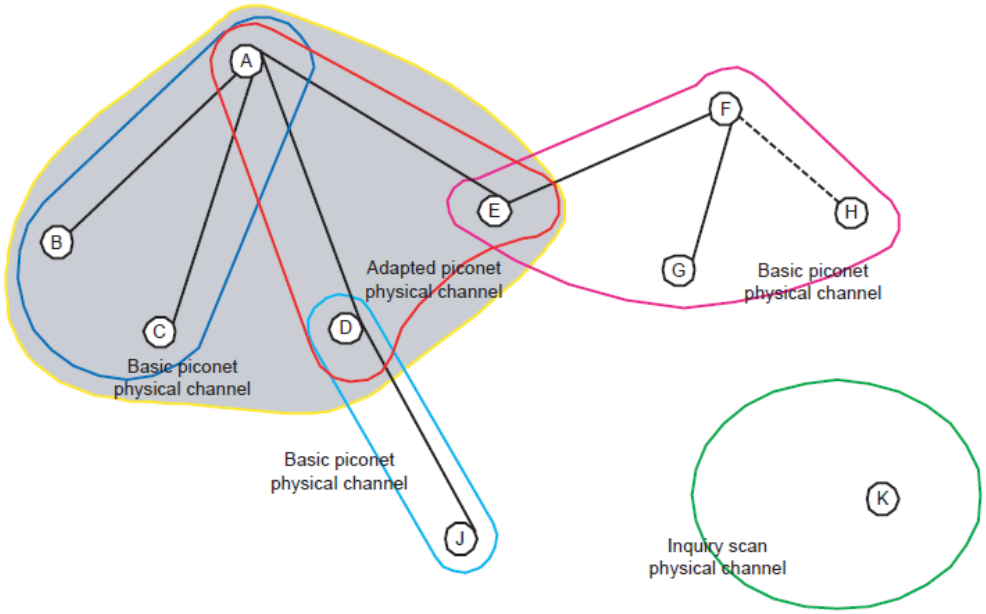
- Basic unit of networking in Bluetooth
- Consists of a master device and up to seven slaves
- Master coordinates medium access
- Slave may only communicate with the master and may only communicate when granted permission by the master
- Only units that have to exchange data share the same Piconet, so that many Piconets with overlapping coverage can exist simultaneously



# Bluetooth: Scatternets

## Scatternet

- Group of linked Piconets joined by common devices
- Devices linking the Piconets can be slaves on both Piconets, or a master of one Piconet and a slave of another



# Bluetooth: Multiplexing (I)

**Medium access among <u>different Piconets</u>:**

FH (Frequency Hopping)-CDMA

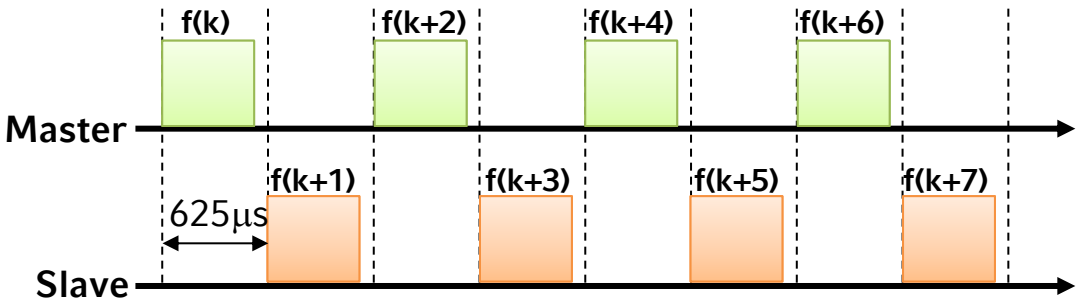
(cp. FHSS in 802.11)

- Master coordinates medium access
- Slave can only communicate with the master and can only communicate when granted permission by the master
- Total bandwidth divided into 79 physical channels, each of bandwidth 1 MHz
- **Overlapping Piconets are separated from one another by frequency hopping**
- Frequency hopping occurs by jumping from one physical channel to another in a pseudorandom sequence

- The same hopping sequence is shared by all devices on a single Piconet
- 1,600 hops per second
- Slot length of 625µs (625 bits at 1 Mbps)
- **Hopping sequence is a function of the master's Bluetooth address and clock**
- As the master's address is known to all terminals in a Piconet, all terminals can derive the hopping sequence
- Because different Piconets have different masters, they use different hopping sequences

# Bluetooth: Multiplexing (II)

## Access within a Piconet: TDMA/TDD




TDD (Time Division Duplex)

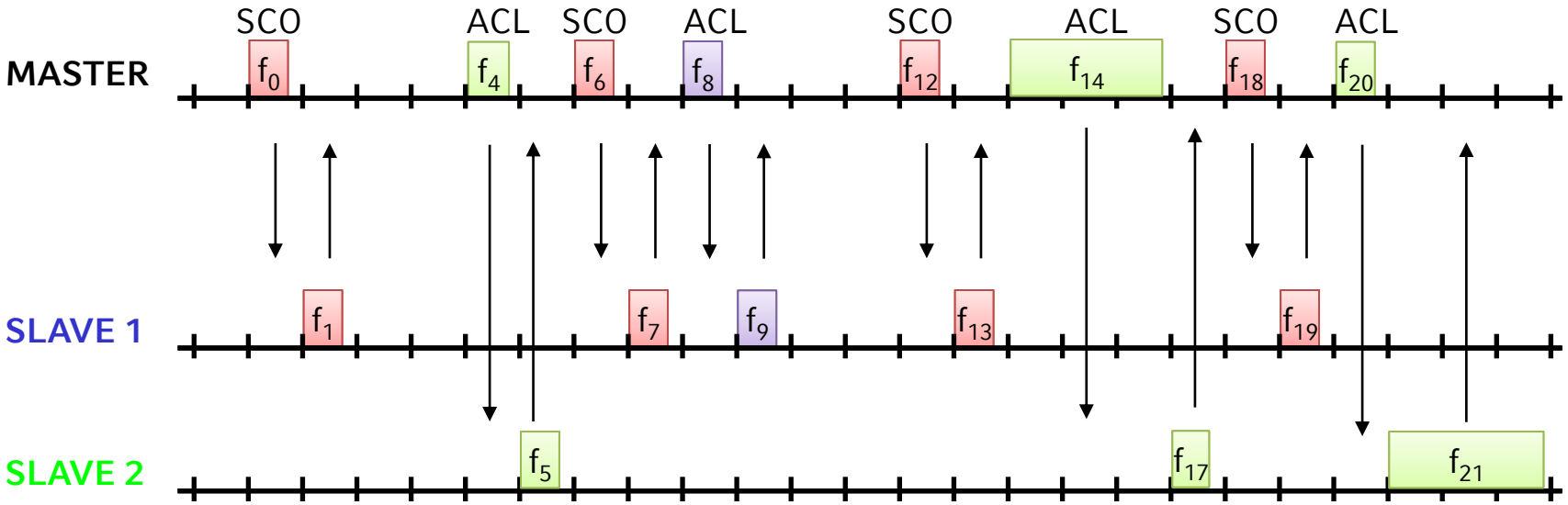
- Multi-slot operation: 1, 3, or 5 slots may be combined to a packet
- Master only transmits in even slots
- Slaves only transmit in odd slots
- Bandwidth of 1 MHz shared between all devices of a Piconet

Polling

- At each slot the master alternately polls a slave
- A slave is only allowed to “talk”, if the master has granted permission before
- Master alternately grants access permission to all slave devices of a Piconet

# Bluetooth: Baseband link types & robustness (I)

- **SCO (Synchronous Connection Oriented) – Voice**
  - Periodic single slot packet assignment, 64 kbit/s full-duplex, point-to-point
- **ACL (Asynchronous Connection Less) – Data**
  - Variable packet size (1,3,5 slots), asymmetric bandwidth, point-to-multipoint



# Bluetooth: Packet types

## Asynchronous packets (ACL):

| Type | Payload Header (bytes) | User Payload (bytes) | FEC | CRC | Symmetric Max. Rate (kb/s) | Asymmetric Max. Rate (kb/s) | |
|---|---|---|---|---|---|---|---|
| | | | | | | Forward | Reverse |
| DM1 | 1 | 0-17 | 2/3 | yes | 108.8 | 108.8 | 108.8 |
| DH1 | 1 | 0-27 | no | yes | 172.8 | 172.8 | 172.8 |
| DM3 | 2 | 0-121 | 2/3 | yes | 258.1 | 387.2 | 54.4 |
| DH3 | 2 | 0-183 | no | yes | 390.4 | 585.6 | 86.4 |
| DM5 | 2 | 0-224 | 2/3 | yes | 286.7 | 477.8 | 36.3 |
| DH5 | 2 | 0-339 | no | yes | 433.9 | 723.2 | 57.6 |
| AUX1 | 1 | 0-29 | no | no | 185.6 | 185.6 | 185.6 |
| 2-DH1 | 2 | 0-54 | no | yes | 345.6 | 345.6 | 345.6 |
| 2-DH3 | 2 | 0-367 | no | yes | 782.9 | 1174.4 | 172.8 |
| 2-DH5 | 2 | 0-679 | no | yes | 869.1 | 1448.5 | 115.2 |
| 3-DH1 | 2 | 0-83 | no | yes | 531.2 | 531.2 | 531.2 |
| 3-DH3 | 2 | 0-552 | no | yes | 1177.6 | 1766.4 | 235.6 |
| 3-DH5 | 2 | 0-1021 | no | yes | 1306.9 | 2178.1 | 177.1 |

DM = Data Medium Rate
DH = Data High Rate
2/3-DH = DH with 2/3 MBits/s

## Synchronous packets (SCO/eSCO):

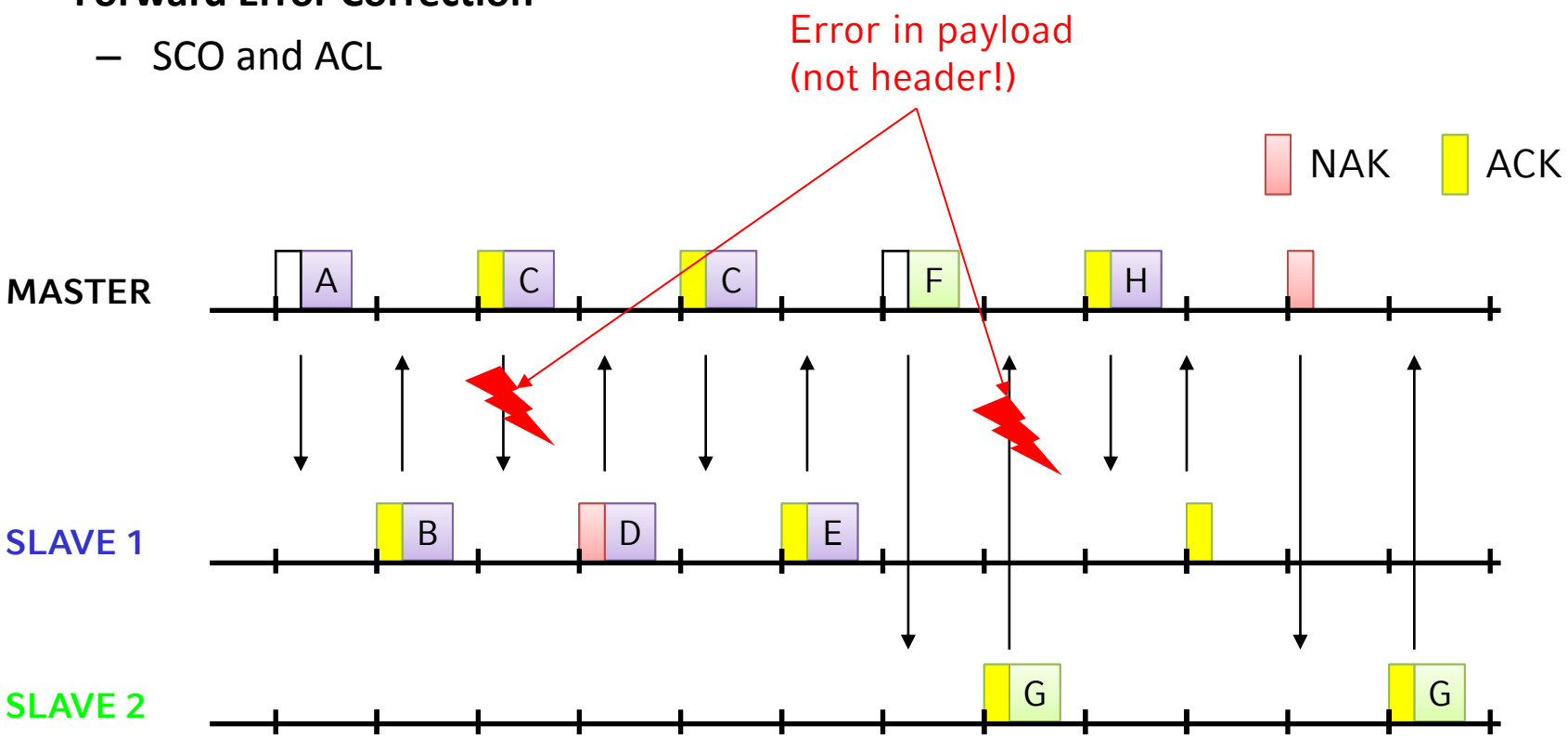
| Type | Payload Header (bytes) | User Payload (bytes) | FEC | CRC | Symmetric Max. Rate (kb/s) |
|---|---|---|---|---|---|
| HV1 | na | 10 | 1/3 | no | 64.0 |
| HV2 | na | 20 | 2/3 | no | 64.0 |
| HV3 | na | 30 | no | no | 64.0 |
| DV[1] | 1 D | 10+(0-9) D | 2/3 D | yes D | 64.0+57.6 D |
| EV3 | na | 1-30 | No | Yes | 96 |
| EV4 | na | 1-120 | 2/3 | Yes | 192 |
| EV5 | na | 1-180 | No | Yes | 288 |
| 2-EV3 | na | 1-60 | No | Yes | 192 |
| 2-EV5 | na | 1-360 | No | Yes | 576 |
| 3-EV3 | na | 1-90 | No | Yes | 288 |
| 3-EV5 | na | 1-540 | No | Yes | 864 |

HV = High quality Voice
DV = Data Voice (Voice + Audio)
EV = Extended Voice

Source: Bluetooth Specification Version 4.0, www.bluetooth.com

# Bluetooth: Baseband link types & robustness (II)

- **Retransmission**
  - ACL only, very fast
- **Forward Error Correction**
  - SCO and ACL



# Bluetooth: Specifications

- Bluetooth 1.1:

  IEEE 802.15, Non-encrypted Channels

- Bluetooth 1.2:

  Adaptive Frequency Hopping (AFH), Enhanced SCO (eSCO), November 2003

- Bluetooth 2.0 + EDR:

  Enhanced Data Rate (EDR), October 2004

- Bluetooth 2.1 + EDR:

  Secure Simple Pairing, Quality of Service, July 2007

- Bluetooth 3.0 + HS:

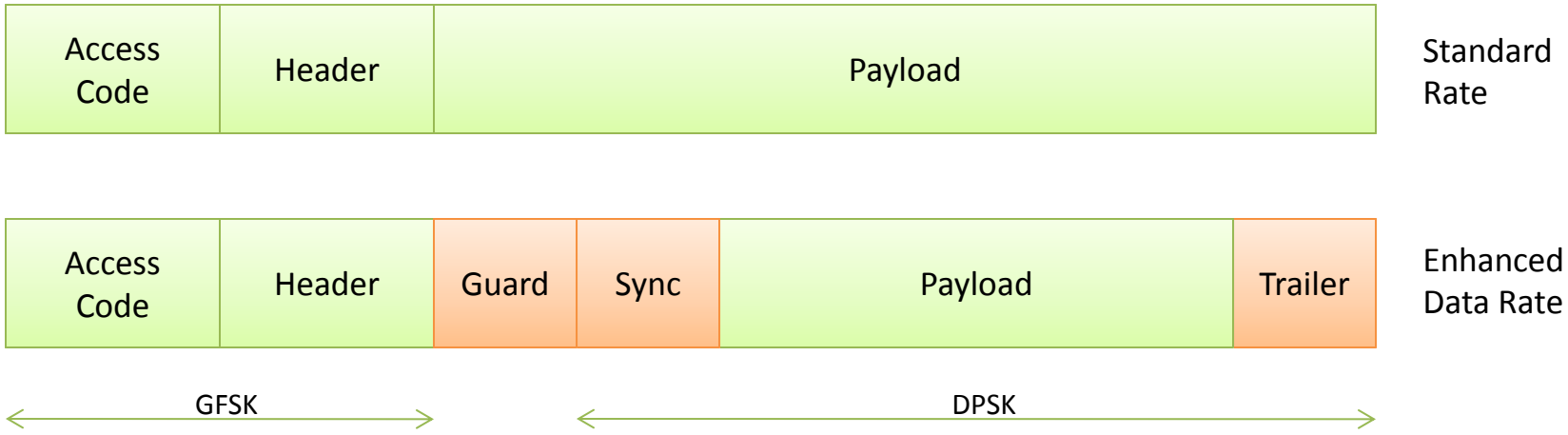
  Alternate MAC/PHY (AMP) Controller, April 2009

- Bluetooth 4.0:

  Low Energy Technology, June 2010

# Bluetooth: Enhanced Data Rate

- Symbol rate: 1 MSymbol/s
- 2 MBit/s with π/4DQPSK modulation scheme
- 3 MBit/s with 8DPSK modulation scheme
- Fully backwards compatible with Standard Rate
- Modulation scheme is changed within the packet
  - Access Code & Header: GFSK
  - Payload: π/4DQPSK or 8DPSK



# Excursus - Signals & Modulation (I)

## Frequency Shift Keying (FSK)

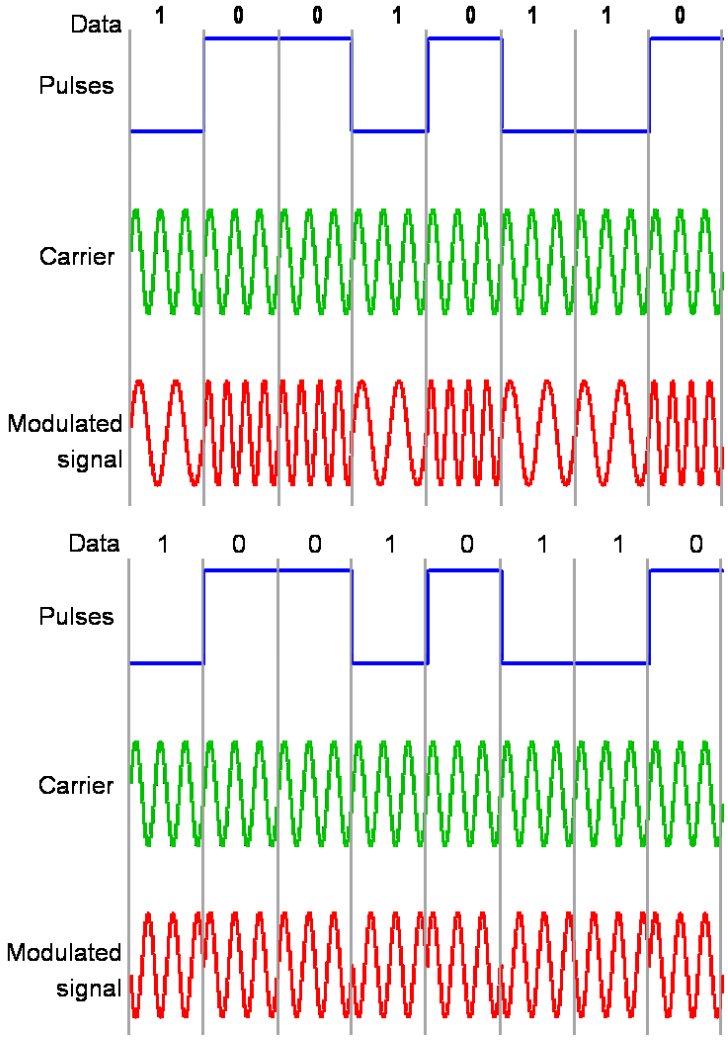
- Frequency of the carrier is changed according to the data
- Transmitter sends different frequencies for a "1" than for a "0”

## Phase Shift Keying (PSK)

- Data is represented by shift in the phase of a signal
- Simplest scheme uses two phases to represent the two binary digits

## Differential PSK

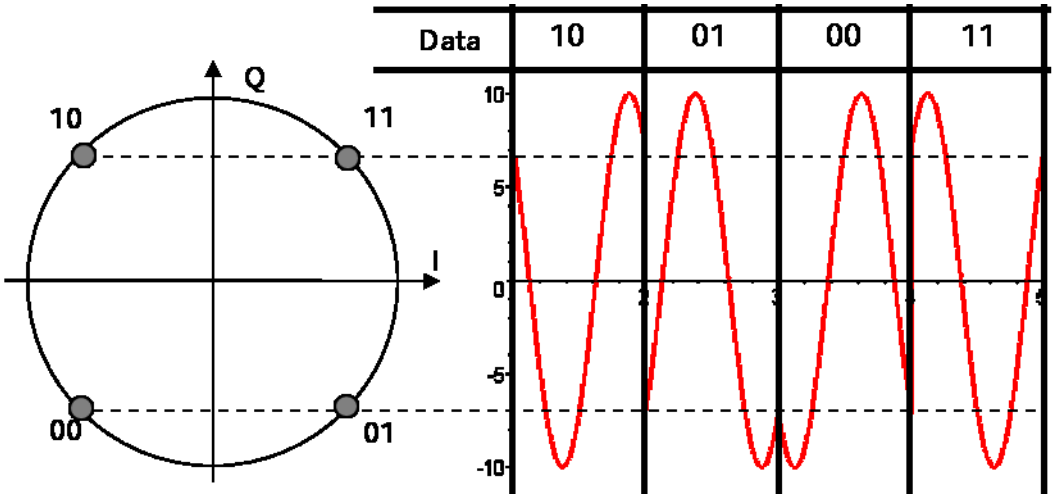
- Data is represented by change in signal phase
- Eliminates (to a certain degree) need for reference signal at receiver



# Excursus - Signals & Modulation (II)

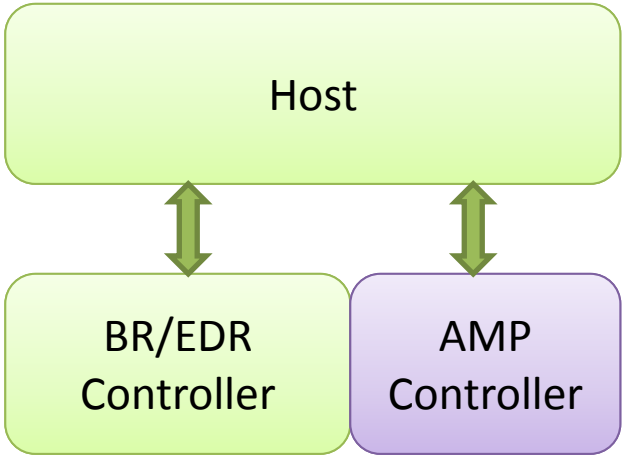
**Quadrature Phase Shift Keying**

- So far: each kind of shift keying defines two signal states for representing binary 0 or 1
- But: a shift keying scheme can fix an arbitrary number of signal states (in theory) to increase the number of bits transferred in a single modulation step
- Example: Quadrature Phase Shift Keying (QPSK)
  - Four signal states → two bits are transmitted in a single step



# Bluetooth: Bluetooth AMP (Alternate MAC/PHY)

- Uses 802.11 (WLAN) to transmit at higher data rates (up to 24 Mbit/s)
- Bluetooth radio still used for device discovery
  - AMP Managers can discover the AMPs that are available on the other device
  - If available on both devices: Data traffic is moved from BR/DER Controller to AMP Controller
- AMP consists of Protocol Adaptation Layer (PAL) on top of a MAC and PHY
  - PAL responsible for mapping the Bluetooth protocols to specific protocols of underlying MAC/PHY

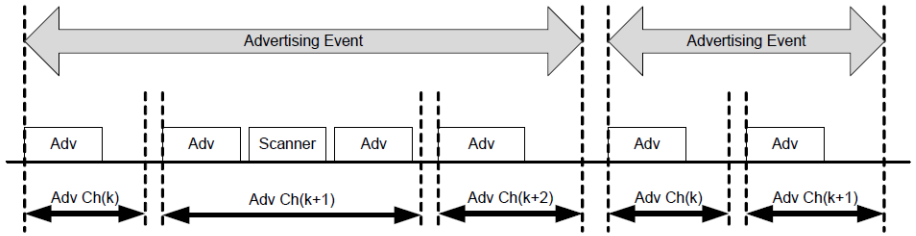


# Bluetooth: Low Energy (v4.0)

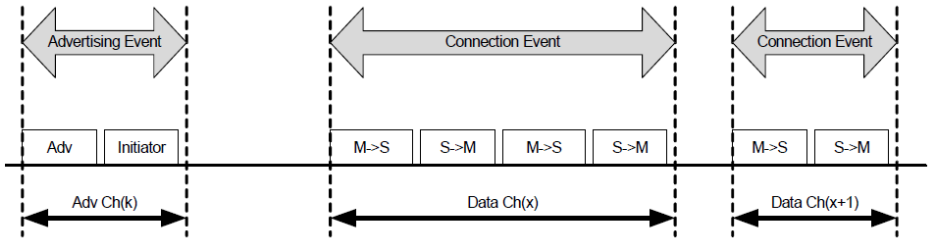
- Bluetooth Core Specification Version 4.0:
  - Formal adoption in July 2010
  - Hallmark feature: Bluetooth low energy technology
- Key features:
  - Ultra-low peak, average and idle mode power consumption
  - Ability to run for years on standard, coin-cell batteries
  - Low cost
  - Low complexity
  - Multi-vendor interoperability
  - Enhanced range

# Bluetooth: BLE Operation

- Operates in 2,4 GHz ISM band (like BR/EDR radio)
- FDMA:
  - 40 channels separated by 2 MHz
  - 3 used as advertising channels, 37 as data channels
  - Frequency hopping pattering: pseudo-random order of 37 frequency
- TDMA:
  - Advertising and connection events
  - Devices: Advertisers, Scanners, Initiators




Advertising Events




Connection Events

# Bluetooth: BLE Comparison

| Technical Specification | Classic *Bluetooth* technology | *Bluetooth* low energy technology |
|---|---|---|
| Radio frequency | 2.4 GHz | 2.4 GHz |
| Distance/Range | 10 meters | 10 meters |
| Over the air data rate | 1-3Mbps | 1Mbps |
| Application throughput | 0.7-2.1 Mbps | 0.2 Mbps |
| Nodes/Active slaves | 7- 16,777,184 | Unlimited |
| Security | 64b/128b and application layer user defined | 128b AES and application layer user defined |
| Robustness | Adaptive fast frequency hopping, FEC, fast ACK | Adaptive fast frequency hopping |
| Latency (from a non connected state) | | |
| Total time to send data (det.battery life) | 100ms | <6ms |
| Government regulation | Worldwide | Worldwide |
| Certification body | Bluetooth SIG | Bluetooth SIG |
| Voice capable | Yes | No |
| Network topology | Scatternet | Star-bus |
| Power consumption | 1 as the reference | 0.01 to 0.5(depending on use case) |
| Peak current consumption | <30 mA | <15 mA (max 15 mA to run on coin cell battery) |
| Service discovery | Yes | Yes |
| Profile concept | Yes | Yes |
| Primary use cases | Mobile phones, gaming, headsets, stereo audio streaming, automotive, PCs, etc. | Mobile phones, gaming, PCs, watches, sports & fitness, healthcare, automotive, home electronics, automation, Industrial, etc. |

Source: Bluetooth Low Energy-Technical Facts, http://www.bluetooth.com/SiteCollectionDocuments/Bluetoothlowenergyfactsheet.pdf

# Bluetooth: Android

- Android supports classical Bluetooth and Bluetooth Low Energy (since API Level 18)
  - Reference: `android.bluetooth`

- Using the Bluetooth API, apps can
  - Scan for other Bluetooth devices
  - Query the local Bluetooth adapter for paired Bluetooth devices
  - Establish RFCOMM channels
  - Connect to other devices through service discovery
  - Transfer data to and from other devices
  - Manage multiple connections

# Bluetooth: First steps with Android

- **Bluetooth Permissions**

```
<manifest ... >
  <uses-permission android:name="android.permission.BLUETOOTH" />
  <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
  ...
</manifest>
```

- **Setting Up Bluetooth**
  - Check if Bluetooth is supported:

```
BluetoothAdapter mBluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();
if (mBluetoothAdapter == null) {
    // Device does not support Bluetooth
}
```

  - Enable Bluetooth:

```
if (!mBluetoothAdapter.isEnabled()) {
     Intent enableBtIntent = new
     Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
     startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
}
```

# Bluetooth: Summary

- Bluetooth
  - Low-cost, radio-based wireless network technology
  - Uses the license-free 2.4 GHz band
- Multiplexing
  - Frequency hopping , Time division duplex
- Link types
  - SCO, ACL
- Topology
  - Piconet, Scatternet
- Specifications
  - Classical, EDR, HS, BLE
- Android Bluetooth API

# NFC – Introduction (I)

*„Near field communication (NFC) is a set of standards for smartphones and similar devices to establish radio communication with each other by touching them together or bringing them into close proximity, usually no more than a few inches."*

*Source: http://en.wikipedia.org/wiki/Near_field_communication*

- Wireless Short Range Communication Technology
- Based on RFID technology at 13,56 MHz
- Operating distance: up to 10 cm
- Data exchange rate: up to 424 kilobits/s

# NFC – Introduction (II)

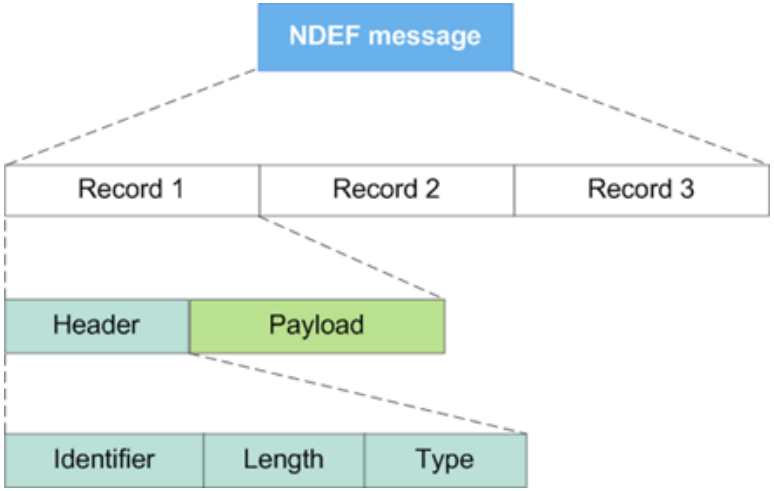
- Advantages
  - Intuitive usage; Easy and simple connection method
  - Complementary to Bluetooth and 802.11 with their long distance capabilities
  - Works in dirty environment
  - Provides communication method to non-self powered devices


- Use Cases
  - Mobile Ticketing
  - Mobile Payment
  - Smart Posters
  - …

# NFC: Android & NDEF

- Android supports NFC since API level 9
  - Since API level 10:  comprehensive reader/writer support + foreground NDEF pushing
  - Since API level 14: push NDEF messages to other devices with Android Beam + convenience methods to create NDEF records

- **NDEF** (NFC Data Exchange Format)
  - Standard data format to send and receive data in Android
  - Reference: `android.nfc.NdefMessage`, `android.nfc.NdefRecord`
  - Other formats: `android.nfc.tech`

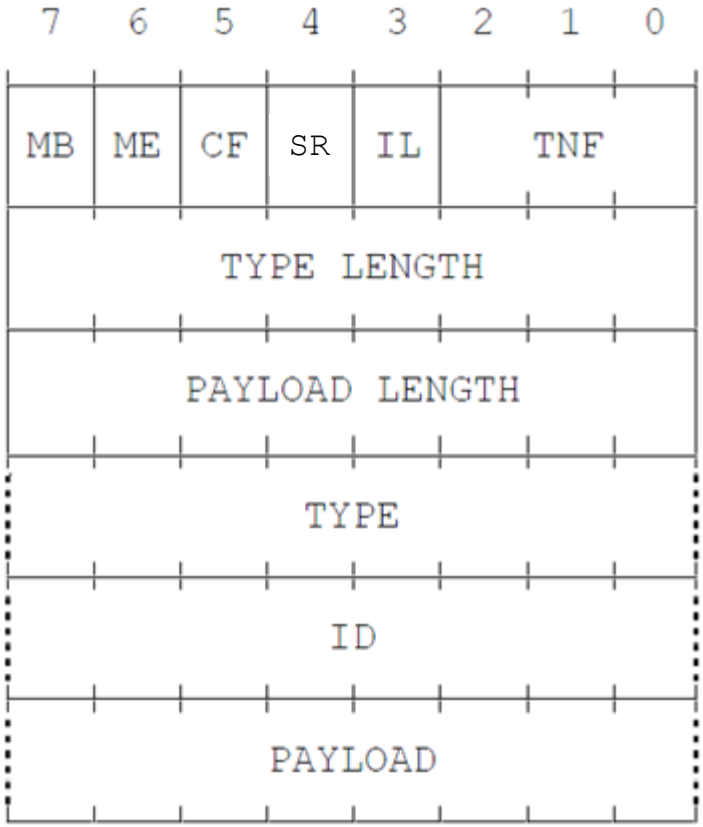


# NFC: TLV Blocks

- Tag
  - Single byte to identify the type of TLV block
- Length
  - Contains the size (in bytes) of the value field (either one or three byte format)
  - One byte format: a single byte value from 0x00..0xFF
- Value
  - Only present if the Length Field (described above) is present and not equal to 0x00.
  - Value field contains payload (e.g., an NDEF Message)
- Terminator TLV
  - Last TLV block in the data area
  - Consist of a single byte: 0x0FE

# NFC: NDEF example

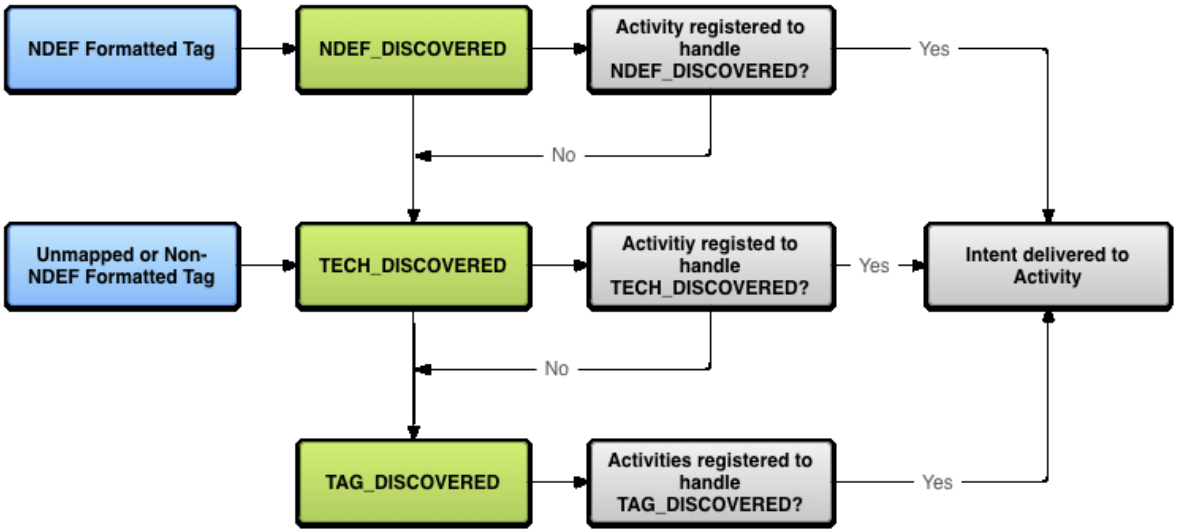
- Payload : „nokia.com“
  - 6e 6f 6b 69 61 2e 63 6f 6d
- Type:
  - URI Record: 55,
  - „http://“: 0x03
- Length:
  - Payload: 0A, Type: 01
- Flags/Identifier:
  - MB: 1, ME: 1, CF: 0,
    SR: 1, IL: 0, TNF: 001
- TLV: Tag: 03, Length: 0e, Terminator: FE




- **Gesamt:** 03 0e d1 01 0a 55 03 6e 6f 6b 69 61 2e 63 6f 6d fe

# NFC: Tag Dispatch System

1. Parsing the NFC tag and figuring out the MIME type or a URI that identifies the data payload in the tag.
2. Encapsulating the MIME type or URI and the payload into an intent.
3. Starts an activity based on the intent.



# NFC: Manifest

- Requesting NFC access:

```
<uses-permission android:name="android.permission.NFC" />

<uses-feature android:name="android.hardware.nfc"
              android:required="true" />

<uses-sdk android:minSdkVersion="10"/>
```

- Filtering for NFC intents:

```
<intent-filter>
  <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
  <category android:name="android.intent.category.DEFAULT"/>
  <data android:scheme="http"
        android:host="developer.android.com"
        android:pathPrefix="/index.html" />
</intent-filter>
```

# NFC: Reading tags

1. Check to see if activity was launched with a NFC intent to ensure that a tag was scanned
2. Obtain the extras out of the intent

```
public void onResume() {
  super.onResume();
  ...
  if (NfcAdapter.ACTION_NDEF_DISCOVERED.equals(getIntent().getAction())) {

    Parcelable[] rawMsgs =
      intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);

    if (rawMsgs != null) {
      msgs = new NdefMessage[rawMsgs.length];
      for (int i = 0; i < rawMsgs.length; i++) {
        msgs[i] = (NdefMessage) rawMsgs[i];
      }
    }
  }
  //process the msgs array
}
```

# NFC: Summary

- NFC
  - Wireless Short Range Communication Technology
  - Based on RFID technology at 13,56 MHz
  - Operating distance: up to 10 cm
  - Data exchange rate: up to 424 kilobits/s
- NDEF (NFC Data Exchange Format)
- Android Bluetooth API
  - Tag Dispatch System
  - Reading tags

# WLAN: IEEE 802.11 Introduction

- Standards differ with respect to frequency, multiplexing and modulation scheme

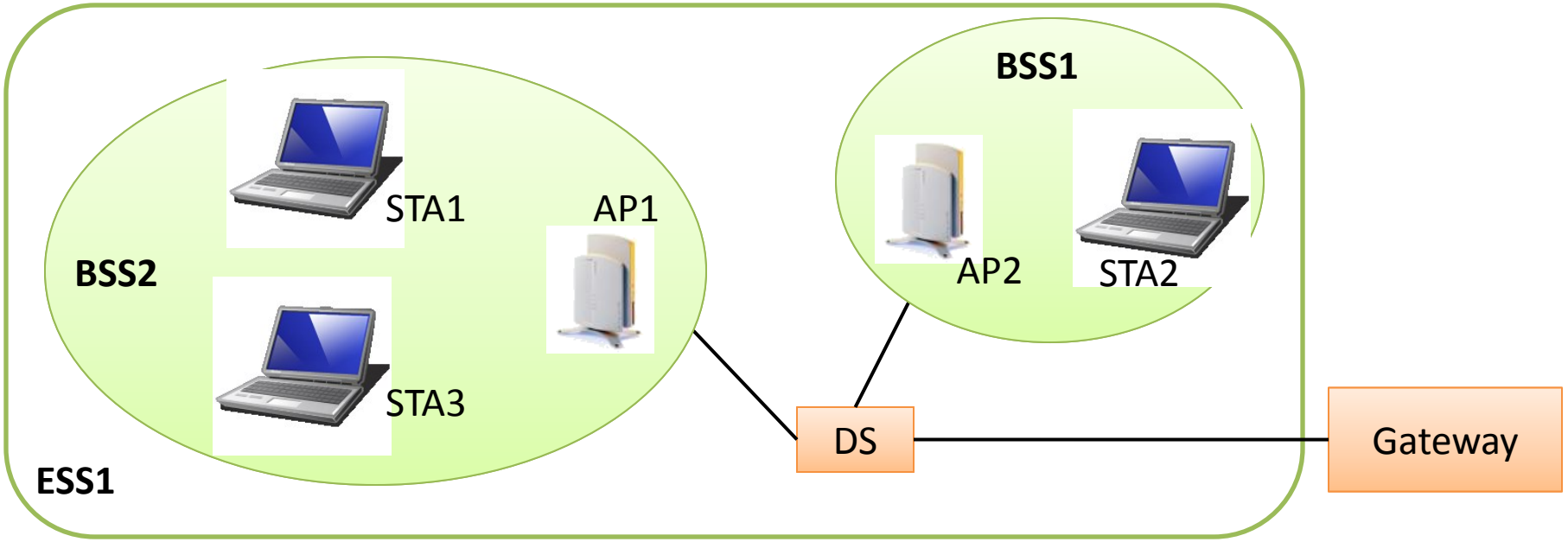
| Standard | Speed (MBit/s) | Frequency (Ghz) | Multiplexing | Modulation |
|---|---|---|---|---|
| 802.11 | 1 - 2 | 2.4 | FHSS/DSSS | FSK |
| 802.11b | 1 – 11 | 2.4 | DSSS/CCK | PSK |
| 802.11g | 6 – 54 | 2.4 | OFDM DSSS/CCK | PSK |
| 802.11a | 6 – 54 | 5.0 | OFDM | PSK |
| 802.11n | 6 - 600 | 2.4 / 5.0 | SDM/OFDM | PSK or QAM |

- Additional Extensions (e.g.):
  - 802.11e: Quality of Service (QoS), Direct Link Protocol (DLP)
  - 802.11f: Information exchange among Access Points (APs)
  - 802.11i: Extensions to security and authentication mechanisms

FHSS: Frequency Hopping Spread Spectrum
DSSS: Direct Sequence Spread Spectrum
OFDM: Orthogonal Frequency Division Multiplexing
CCK: Complementary Code Keying
SDM: Space Division Multiplexing
FSK: Frequency Shift Keying
PSK: Phase Shift Keying
QAM: Quadrature Amplitude Modulation

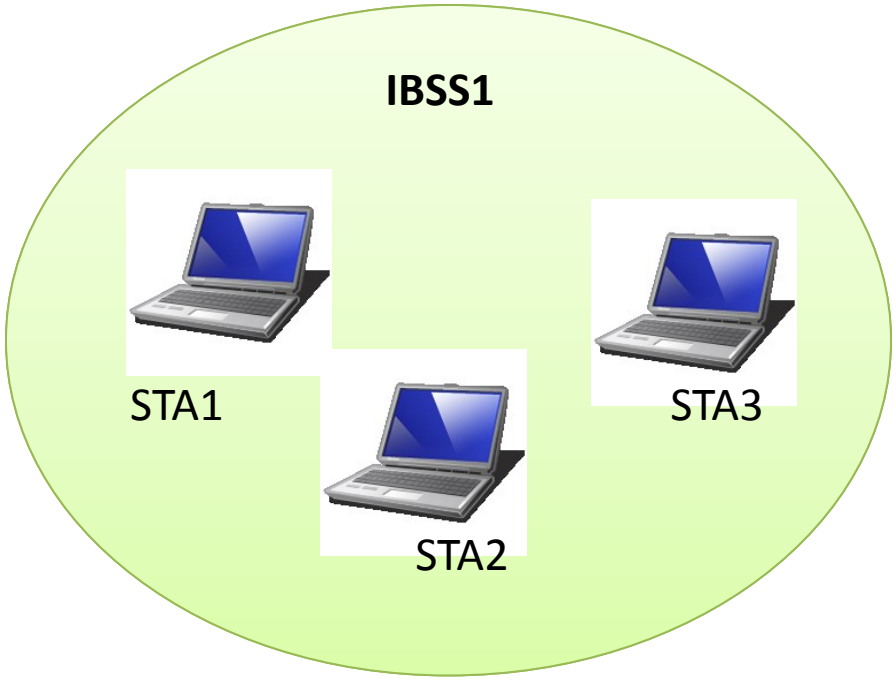
# WLAN: 802.11 System Architecture (I)

- Basic Service Set (BSS): Incorporates an Access Point (AP) and at least one Station (STA)
- Extended Service Set (ESS): One or more BSSs connected through a Distribution System (DS)



# WLAN: 802.11 System Architecture (II)

- Independent Basic Service Set (IBSS):
  Wireless Ad-Hoc Network (without AP)



# WLAN: Medium Access Control (MAC)

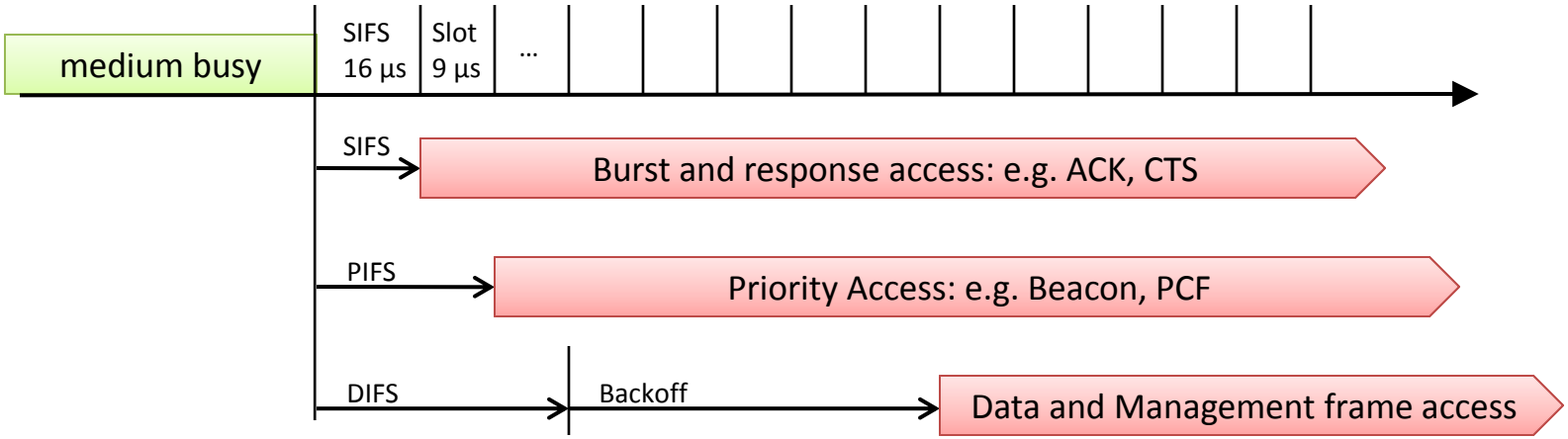
- Which device has access to the transmission medium at any time?

- Problems:
  - What happens if two stations want to transmit at the same time?
  - How to synchronize stations and access points?
  - How to reduce energy consumption?
  - How to detect a „hidden station“?

- The air interface is a single medium, shared among all participating senders and receivers
- The concept of „Carrier Sense Multiple Access with Collision Detection“, used e.g. for wired Ethernet, is not applicable for the air interface
- „Carrier Sense Multiple Access with Collision Avoidance“ is required

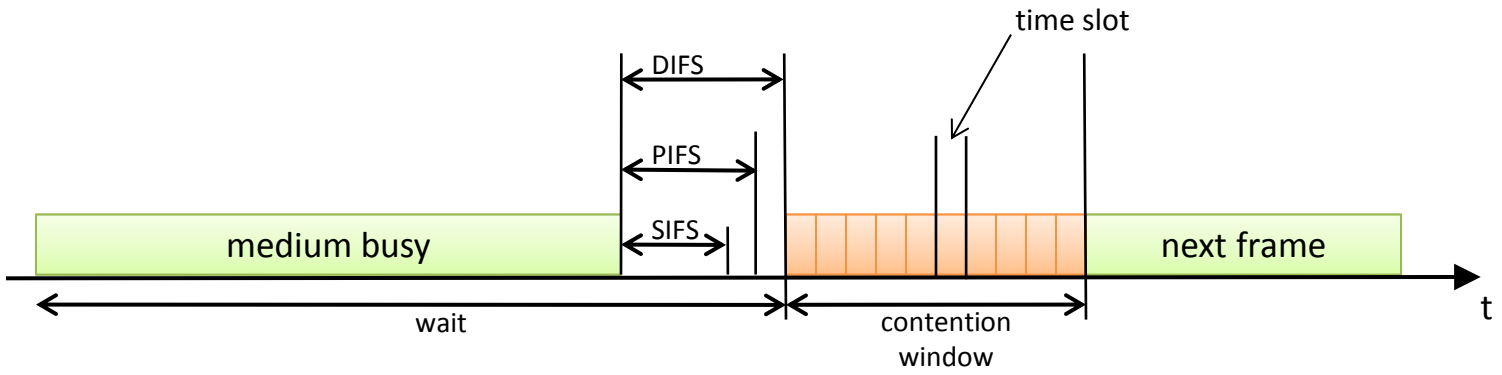
# WLAN: 802.11 Distributed channel access

- Different inter-frame spaces (IFS) durations provide access to the wireless medium at different priority levels
- Short IFS (SIFS): as short as possible but still accomodate PHY and MAC latencies. Used e.g. for data frame acknowlegements (ACKs)
- PointCoordinationFunction IFS (PIFS): Beacons, Contention-free period
- DistributedCoordinationFunction IFS (DIFS): „normal“ data frames

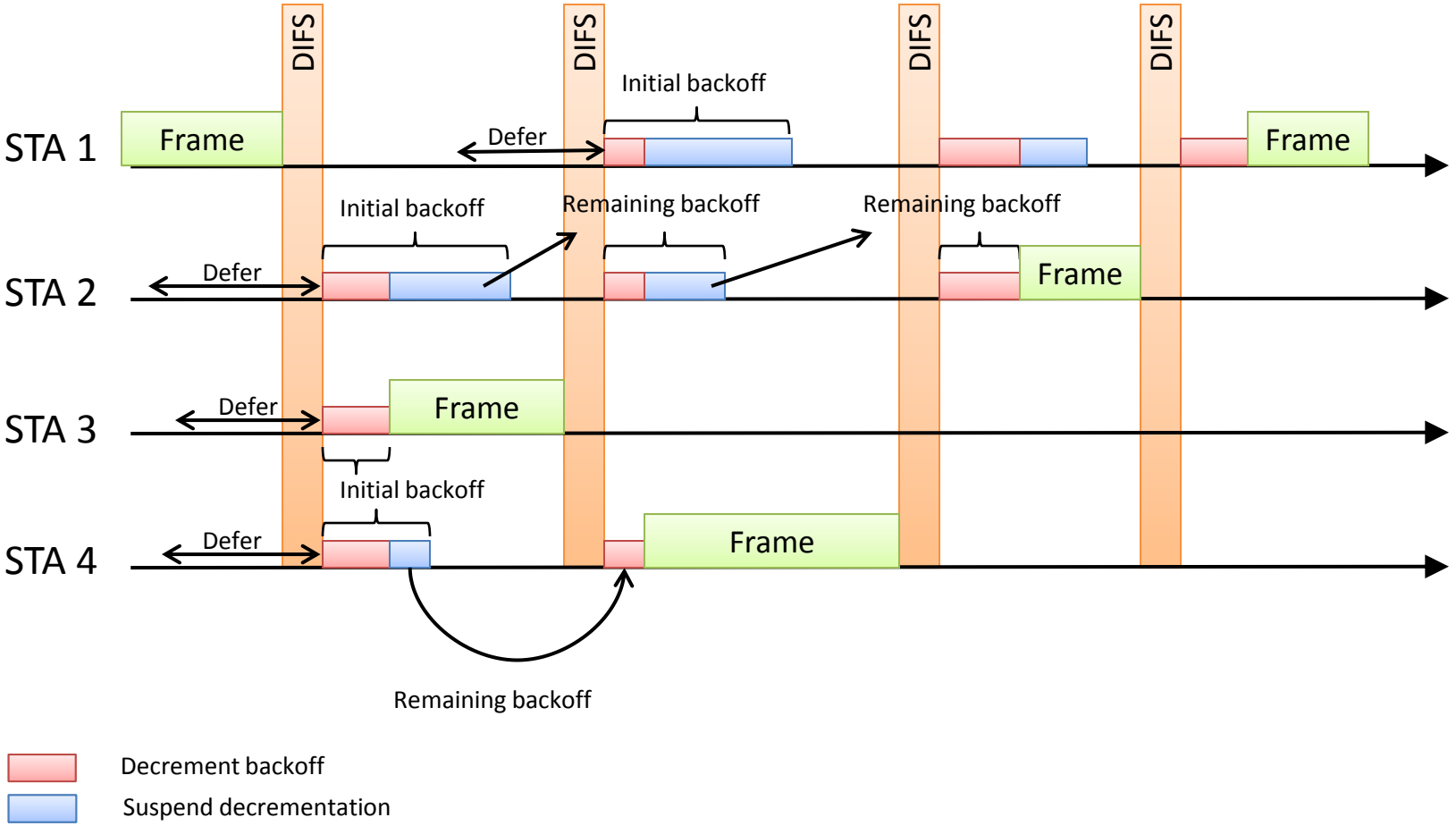


# WLAN: 802.11 Backoff

- When medium transitions from busy to idle, multiple stations may be ready to send data → Each STA maintains a **backoff counter** (BC)
- The backoff counter is set to a random number from the interval [0, CW] (contention windows, CW) which is initially set to CWmin = 15
- Each STA senses the medium
  - Medium gets idle: STA waits for a DIFS, then starts decrementing BC
  - Counter reaches zero AND medium still idle → start transmission and if successful set CW to CWmin
  - When transmission fails (no ACK) → double CW (max. 1023)

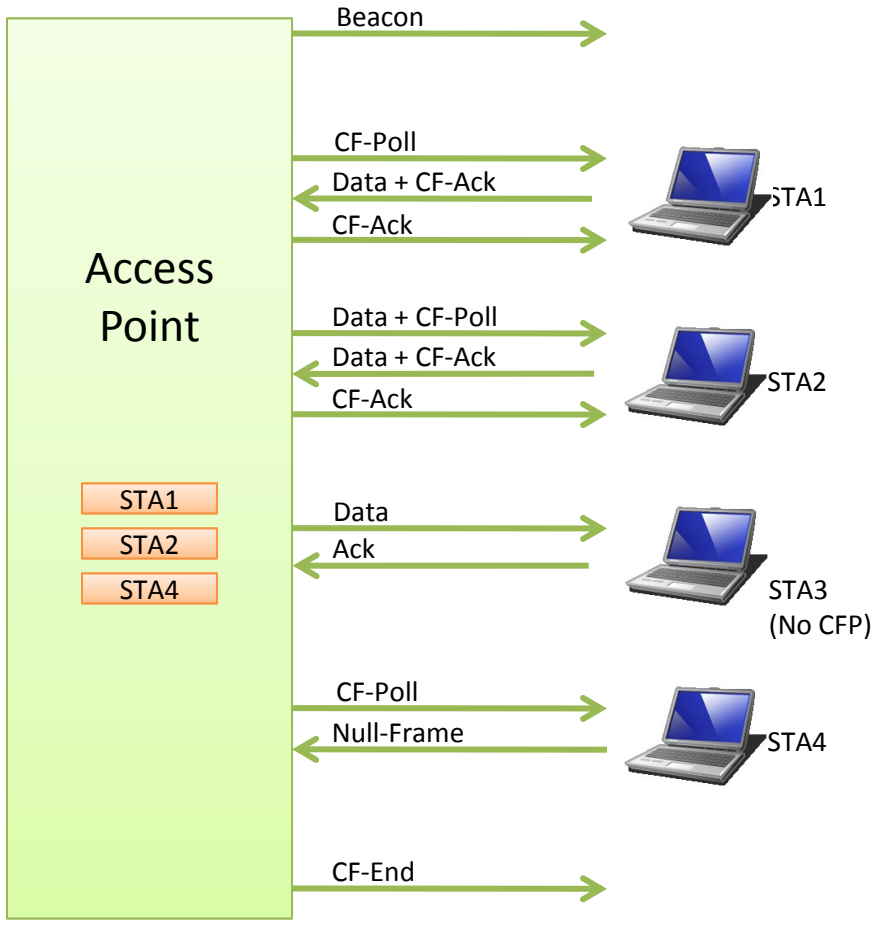


# WLAN: 802.11 Backoff example

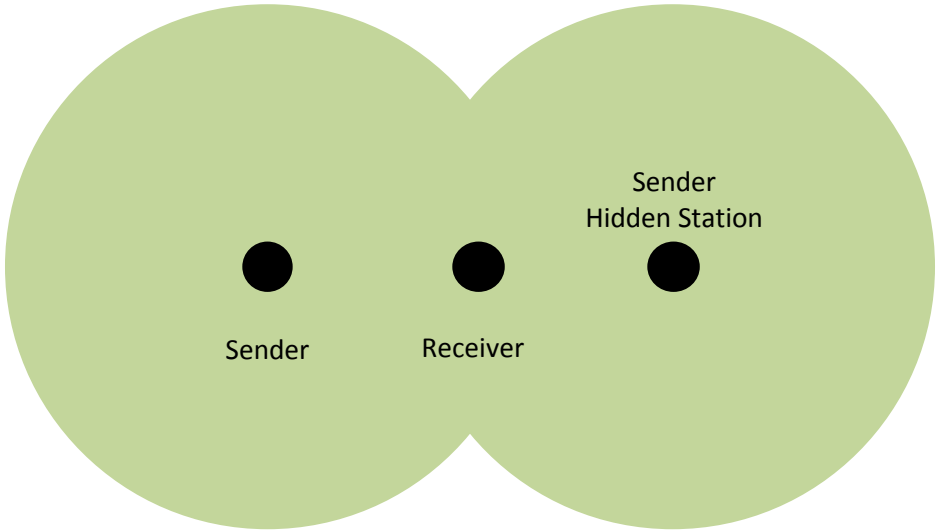


# WLAN: 802.11 Point Coordination Function (PCF)

- Time-critical traffic
- Optional
- Based on Distributed Coordination Function (DCF)
- Contention-Free Period (CFP)
- IFS = PIFS < DIFS
- PCF even works with STAs that do not support CFP



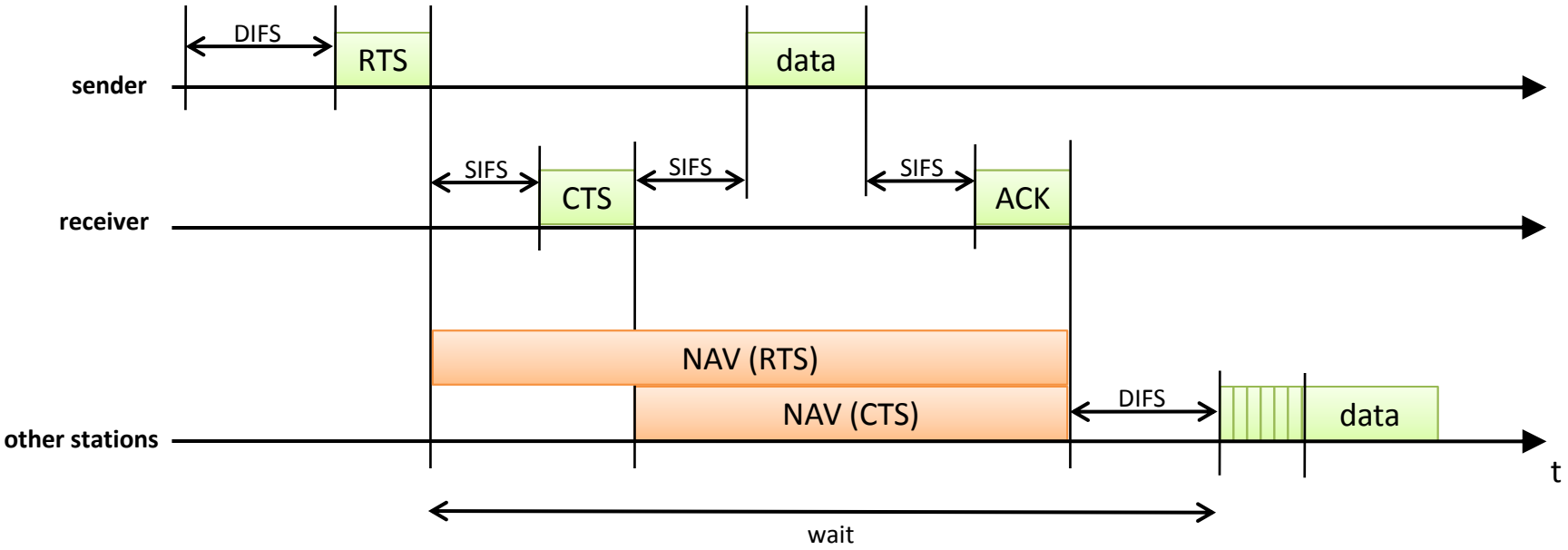
# WLAN: 802.11 Hidden Station Problem




- „Hidden Station Problem“
- Solution in 802.11
  - Sender: „Request to Send“ (RTS)
  - Receiver: „Clear to Send“ (CTS)
  - Hidden Station now knows about transmission
  - However, collisions of RTS/CTS frames are possible, frames are just smaller

# WLAN: 802.11 Network Allocation Vector

- Station transmits a „Request to Send“ (RTS) before transmitting data
- Receiver transmits a „Clear to Send“ (CTS)
- Other stations set their Network Allocation Vector (NAV) accordingly



# WLAN: Wi-Fi P2P

- **Wi-Fi Direct**
  - Wi-Fi technology enabling Wi-Fi devices to connect directly (without an intermediate access point)


- **In Android**: Wi-Fi peer-to-peer (P2P)
  - Since API level 14: `android.net.wifi.p2p`
  - Android's Wi-Fi P2P framework complies with the Wi-Fi Alliance's Wi-Fi Direct™ certification program
- Three main components:
  - **Methods**: discover, request, and connect to peers (`android.net.wifi.p2p.WifiP2pManager`)
  - **Listeners**: Notification of the success or failure of method calls
  - **Intents**: Notification of specific events detected by the Wi-Fi P2P framework

# WLAN: Wi-Fi P2P Applications

- Permissions

```
<uses-sdk android:minSdkVersion="14" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

- Create and register a broadcast receiver
  - `WIFI_P2P_STATE_CHANGED_ACTION`, `WIFI_P2P_PEERS_CHANGED_ACTION`, `WIFI_P2P_CONNECTION_CHANGED_ACTION`, `WIFI_P2P_THIS_DEVICE_CHANGED_ACTION`
- Discover and connect to peers
  - `android.net.wifi.p2p.WifiP2pManager.discoverPeers`, `android.net.wifi.p2p.WifiP2pManager.connect`
- Transfer data
  - ServerSocket/Socket

# WLAN: Summary

- IEEE 802.11
  - System Architecture
  - Medium Access Control
    - Distributed Channel Access / Inter Frame Spacing
    - Point Coordination Function
    - Hidden Station Problem: Request-to-Send / Clear-to-Send
- Wi-Fi P2P