

# WiFi – podstawowe pojęcia

Bezprzewodowa sieć fizyczna...

używa jako medium przestrzeni i fal elektromagnetycznych,  
„fale radiowe”, mikrofale, okolice 2.4GHz lub 5GHz,  
standardy IEEE 802.11, 802.11a/b/g/n/ac (z literką dotyczą warstwy fizycznej)  
WEP, WPA/WPA2, 802.11i – bezpieczeństwo (szyfrowanie, istotne z powodu...)

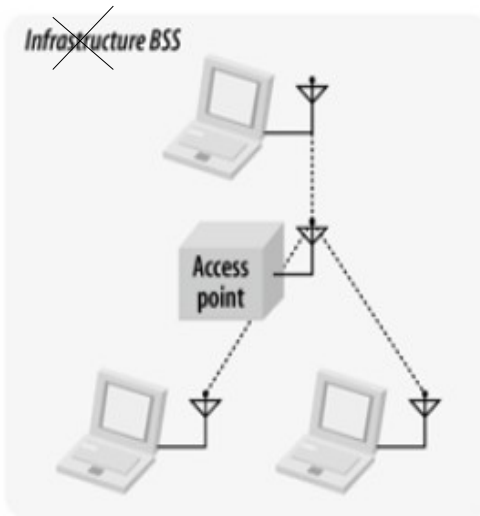
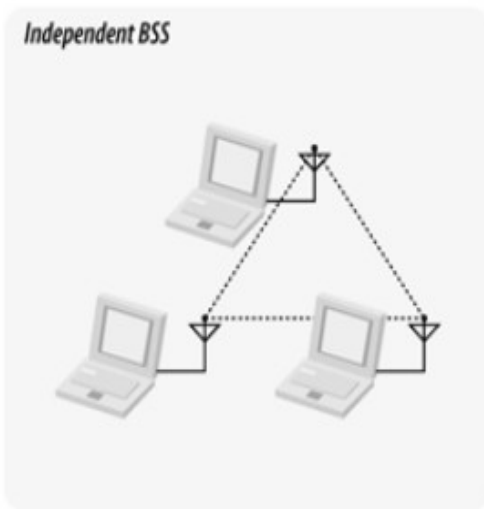
Access point (AP), stacja (STA, klient wifi),

BSS(Basic Service Set, 1x AP + stacje, dane za pośrednictwem AP) ,

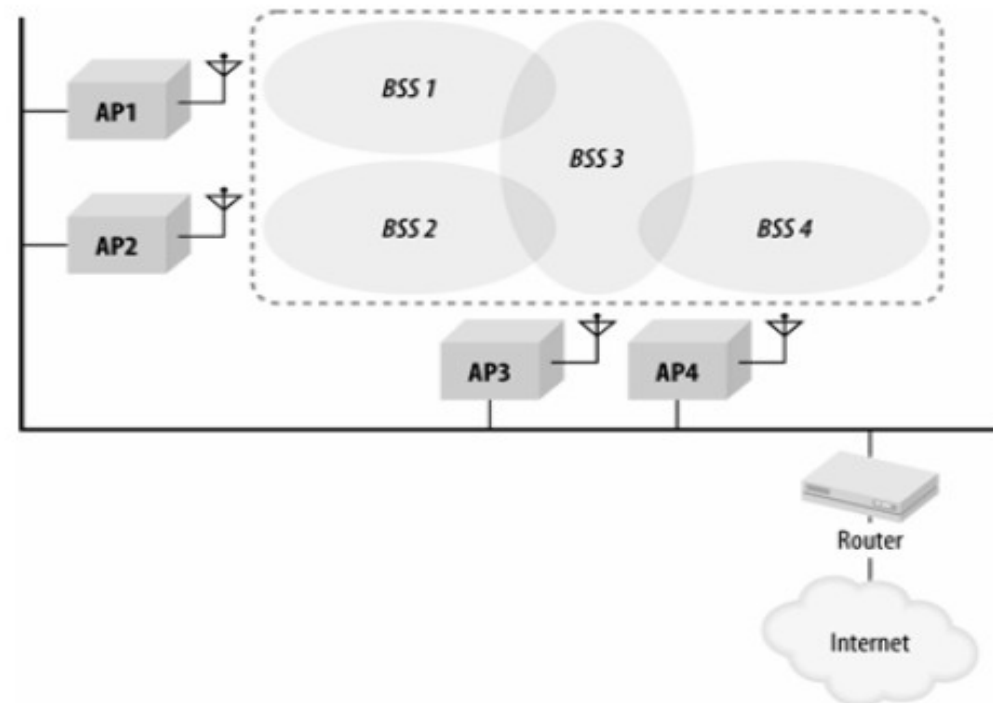
IBSS (Independent BSS, „ad-hoc”, bez AP),

ESS (Extended Service Set, wiele BSS + system dystrybucji, **jeden „nr sieci”**),

System dystrybucji (zazw. eth; most łączący wifi i eth ?),



**Figure 2-5. Extended service set**



# WiFi – podstawowe pojęcia c.d.

(E)SSID - nazwa sieci, ma ją każdy AP, w ESS powinny być identyczne, ramka beacon

BSSID – adr sprzętowy AP, stacje wifi także mają adr sprzętowy

Adr sprzętowy wifi: tak sam jak eth!! 6 bajtów... też możliwy multi/broad-casting...

Kanały wifi, których używa BSS... jest ich 13, w BSS używa się 1 kanału !!,

bliskie BSSy powinny używać innych kanałów !!! kanały nie są odseparowane...

Bezpieczeństwo: stare złe rozwiązanie WEP (4 klucze),

nowe dobre rozwiązanie WPA/WPA2=802.11i (wpa\_supplicant)

**Linux:** interfejsy sieciowe wifi mają nazwy postaci: wlan0, ...

Wyświetlanie widocznych AP: root# iwlist wlan0 scan

Podłączanie się do AP: root# iwconfig wlan0 essid „SSID/nazwa sieci”

Wyświetlanie kanałów: root# iwlist wlan0 chan

```
wlan0    13 channels in total; available frequencies :
```

```
Channel 01 : 2.412 GHz
```

```
Channel 02 : 2.417 GHz
```

```
Channel 03 : 2.422 GHz
```

```
Channel 04 : 2.427 GHz
```

```
Channel 05 : 2.432 GHz
```

```
Channel 06 : 2.437 GHz
```

```
Channel 07 : 2.442 GHz
```

```
Channel 08 : 2.447 GHz
```

```
Channel 09 : 2.452 GHz
```

```
Channel 10 : 2.457 GHz
```

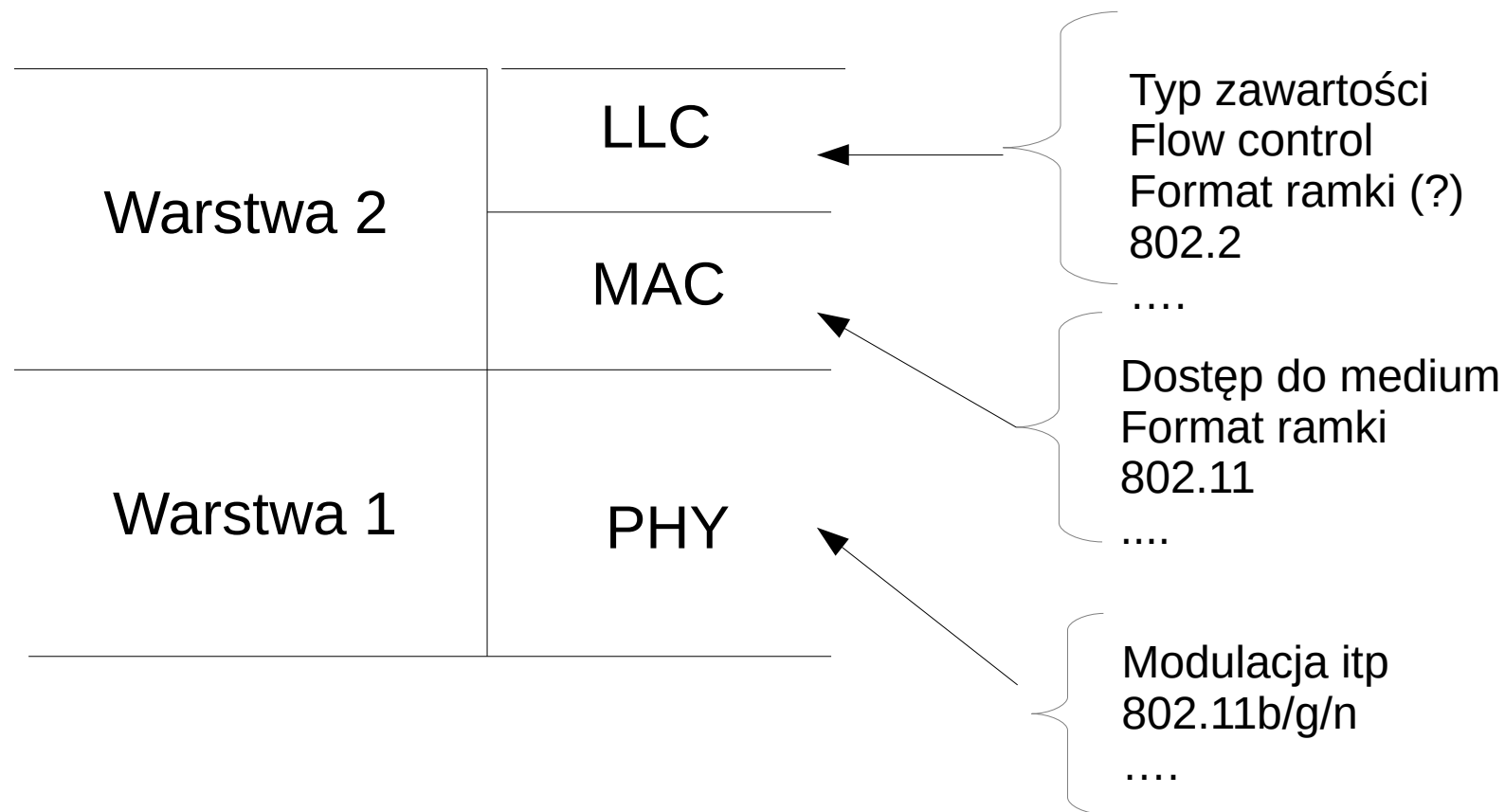
```
Channel 11 : 2.462 GHz
```

```
Channel 12 : 2.467 GHz
```

```
Channel 13 : 2.472 GHz
```

```
Current Frequency:2.427 GHz (Channel 4)
```

# Podział warstw 1 i 2 na podwarstwy...



# Wifi, warstwa 2, podwarstwa MAC

Dostęp do medium:

DCF = Distributed Coordination Function = CSMA/CA

PCF = Point -"- bez rywalizacji o dostęp, HCF = posiada elem QoS

**CSMA/CA** = Carrier Sense Multiple Access with Collision Avoidance

w eth było CSMA/CD ! w wifi **NIE MA** wykrywania kolizji w czasie nadawania !!

są pozytywne potwierdzenia ACK otrzymania ramki

jest okno Backoff losowej rywalizacji o dostęp do łącza (contention window),

stacje losowo wybierają slot z „contention window” i czekają odp czas,

okno zwiększa długość przy każdej nieudanej próbie wysłania ramki

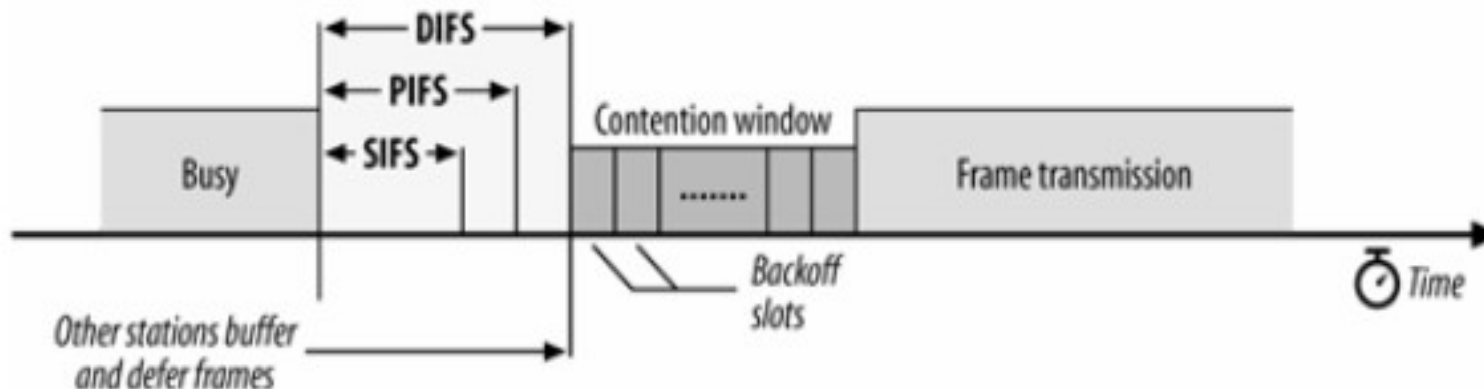
są odstępy czasowe między ramkami o różnej długości: DIFS, SIFS, EIFS, ...

jeśli nośnik był wolny przez czas DIFS + czas w oknie rywalizacji

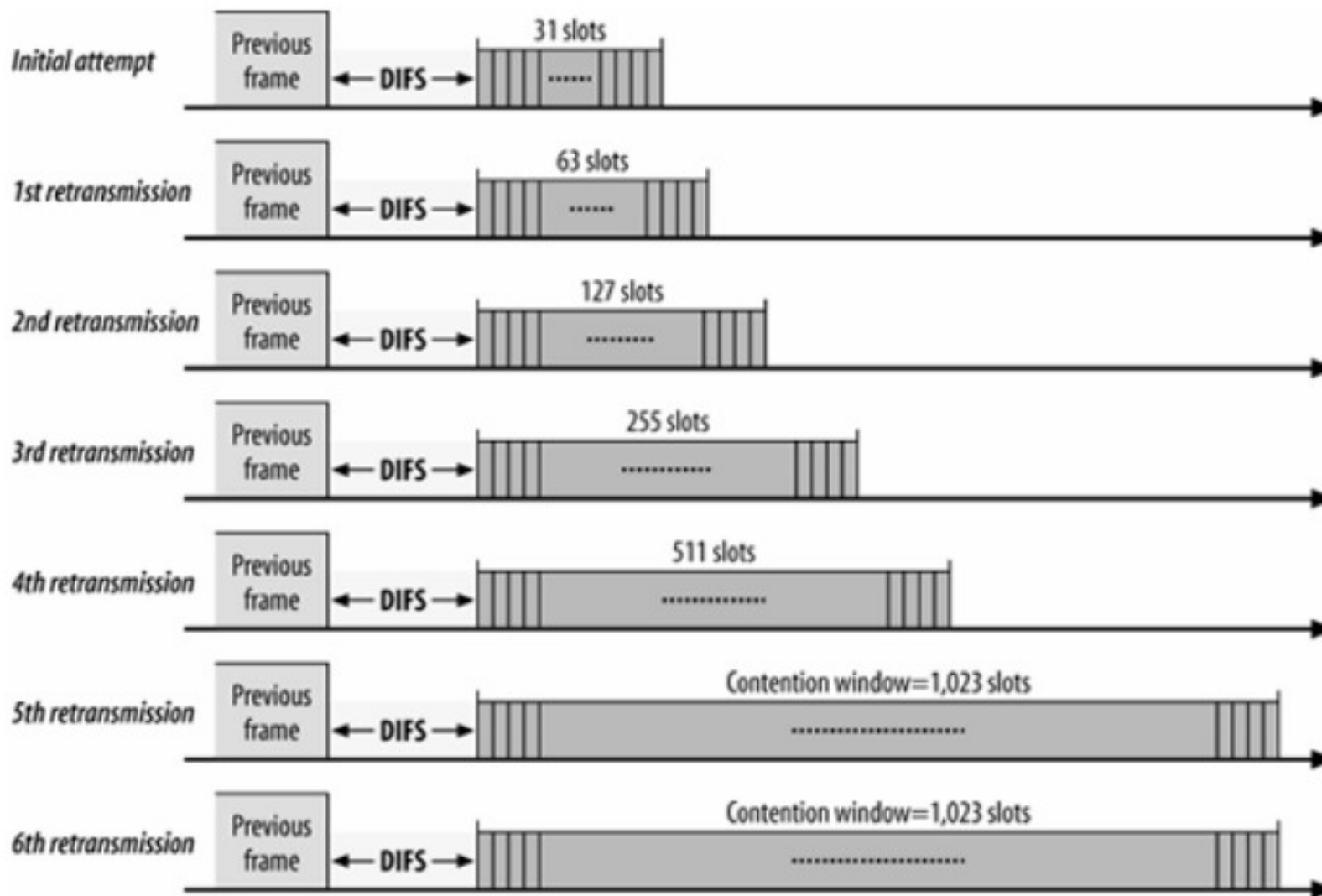
to można nadawać (jeśli był błąd to EIFS zamiast DIFS)

chodzi o to aby 1 stacja zaczęła nadawać z wielu próbujących...

są dodatkowe mechanizmy „rezerwowania” medium: NAV i RTS/CTS



# Zmiana długości okna rywalizacji przy retransmisji (brak ack)



# Wifi, warstwa 2, podwarstwa MAC, c.d.

Inne rozwiązania pozwalające uniknąć błędnych ramek (rezerwowanie medium):

**prot RTS/CTS** (Request To Send, Clear To Send), krótkie ramki do rezerw łącza inaczej: do „uciszania” innych stacji, znane z łącza szeregowego rs232...

*Linux*: próg RTS, chodzi o długość ramki, `iwconfig wlan0 rts`; jest też próg fragm!

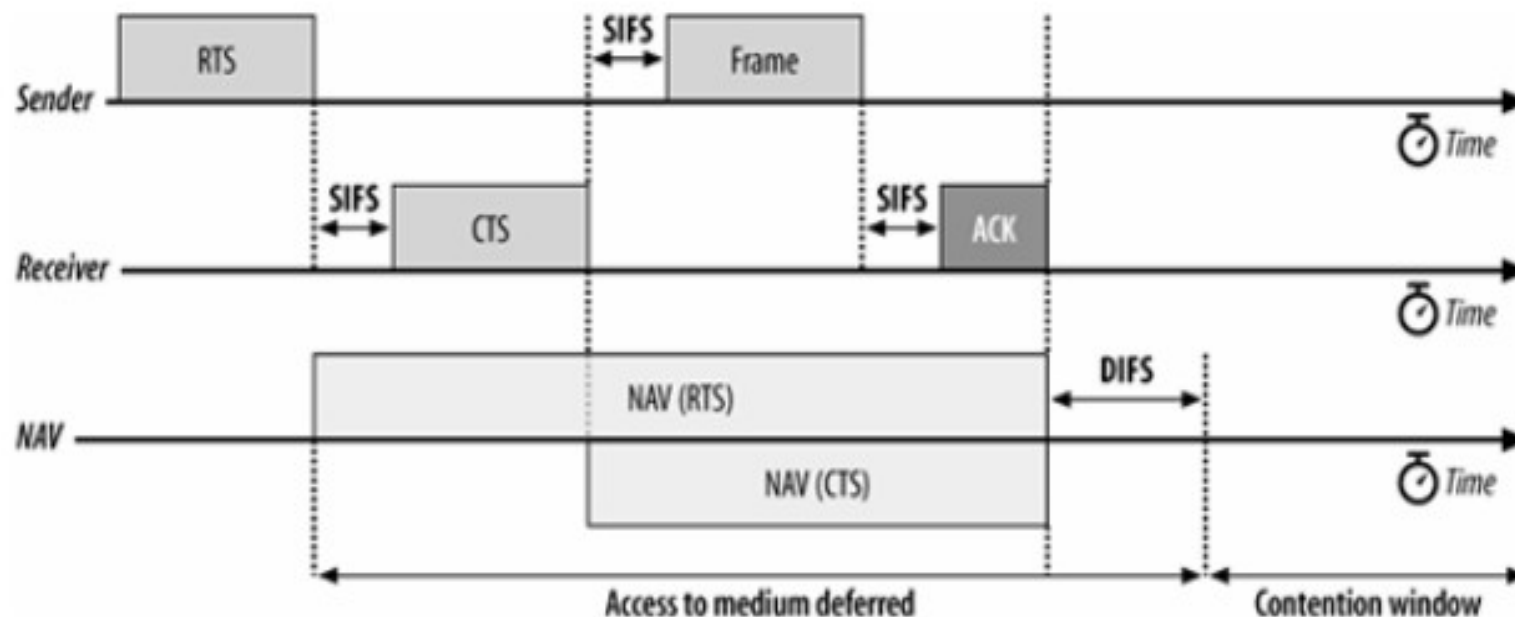
**wektor NAV**: ramki mają pole Duration, w którym informują o czasie nadawania... wszystkie stacje które widzą taką ramkę modyfikują swój wektor NAV który mówi jak długo medium będzie zajęte przez inną stację...

komunikacja nie tylko stacja – AP, ale także stacja – stacja !!!

(choć dane są zawsze przesyłane za pośrednictwem AP)

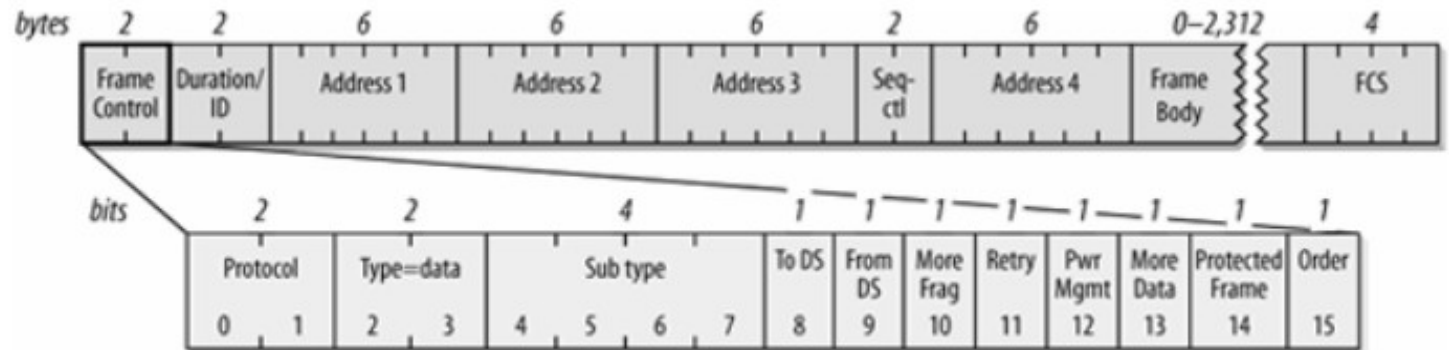
RTS/CTS rozwiązuje problem dalekich stacji:

1 stacja wysła rts, AP wysyła cts, 2 stacja widzi cts (choć nie widzi rts...)



# Wifi, warstwa 2, podwarstwa MAC, c.d.

Format ramki wifi:



Co się znajduje w ramce ?

Typ i podtyp, b. dużo podtypów, typy ramek:

mgmt (beacon, auth, assoc, ...)

control (rts/cts, ack, ...),

data (max dług danych= 2304, 2296 z powodu nagł LLC/802.2)

Adresy:

dst, src, transmitter(?), bssid; interpret zależy od typu ramki !!

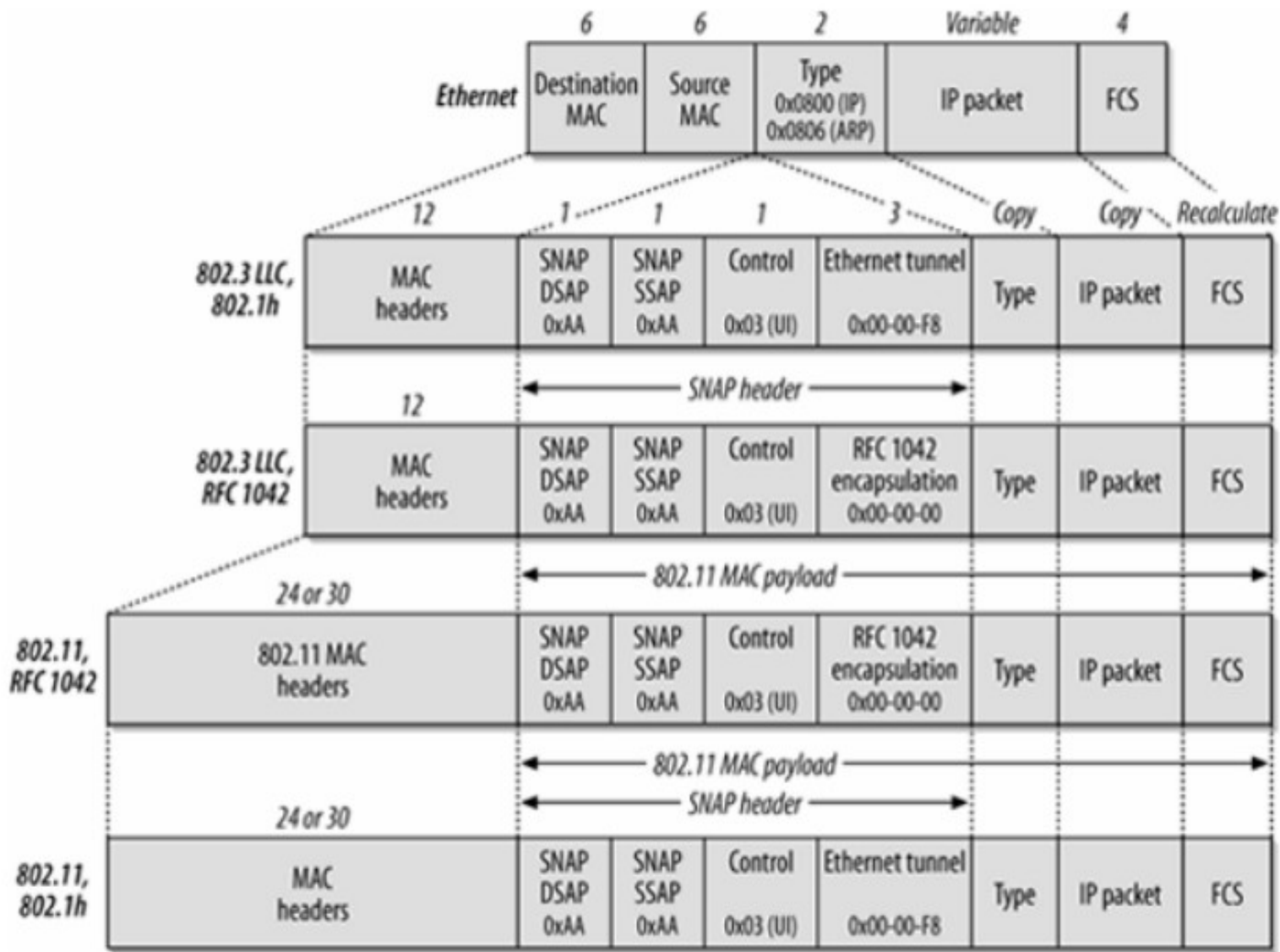
dst może być w tym samym BSS lub innym w ramach ESS

bssid jest po to aby odróżniać ramki naszego i cudzego BSS...

ramka może przeskakiwać między: wifi → eth → wifi (zmienia się adr3)

Zmiana ramki przy przeskakiwaniu wifi → eth → wifi w ESS...

**Figure 3-13. IP encapsulation in 802.11**





# Wifi, warstwa 1 = PHY

Uzyskaliśmy dostęp do mediu, jak przesłać ramkę ???

**Table 17.4** IEEE 802.11 Physical Layer Standards

	<b>802.11</b>	<b>802.11a</b>	<b>802.11b</b>	<b>802.11g</b>
<b>Available bandwidth</b>	83.5 MHz	300 MHz	83.5 MHz	83.5 MHz
<b>Unlicensed frequency of operation</b>	2.4–2.4835 GHz DSSS, FHSS	5.15–5.35 GHz OFDM 5.725–5.825 GHz OFDM	2.4–2.4835 GHz DSSS	2.4–2.4835 GHz DSSS, OFDM
<b>Number of non-overlapping channels</b>	3 (indoor/outdoor)	4 indoor 4 (indoor/outdoor) 4 outdoor	3 (indoor/outdoor)	3 (indoor/outdoor)
<b>Data rate per channel</b>	1, 2 Mbps	6, 9, 12, 18, 24, 36, 48, 54 Mbps	1, 2, 5.5, 11 Mbps	1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54 Mbps
<b>Compatibility</b>	802.11	Wi-Fi5	Wi-Fi	Wi-Fi at 11 Mbps and below

# Wifi, warstwa 1 = PHY

„SS” = Spread Spectrum, rozpraszanie widma,  
użycie szerokiego zakresu częstotliwości fal radiowych (vs 1 częstotliwość nośna...)  
w celu 1. uniknięcia zakłóceń, 2. zwiększenia przepustowości

FHSS = Frequency Hopping SS

skakanie po częstotliwościach wg pewnego wzorca, nadal używane przez Bluetooth !!

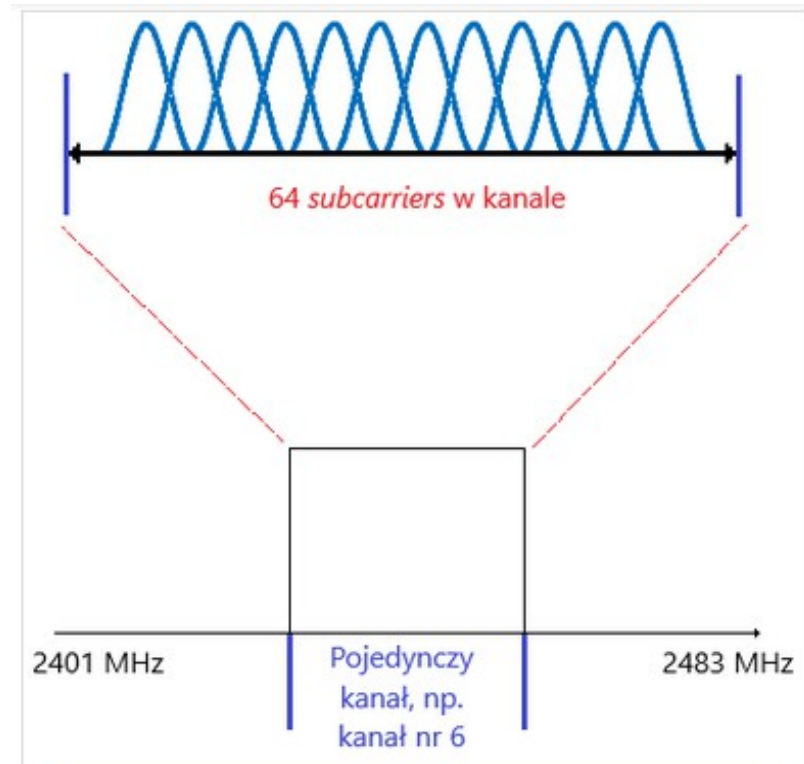
DSSS = Direct Sequence SS

bit zamieniamy na ciąg 11 bitów, kod Barkera „0” = 1 0 1 1 0 1 1 1 0 0 0, „1” negatyw  
przesyłamy je równoległe 11 pod-kanalami (dlatego kanały DS są szerokie...)

OFDM = jak FDM (Frequency Division Multiplexing),  
ale częstotliwości nośne są ortogonalne...

mogą być bliżej siebie! jest ich więcej !!  
mamy więcej kanałów,  
którymi możemy przesyłać dane...

OFDM →



Podział kanału na 64 subcarriers, każda o szerokości 31,25 KHz

# Wifi, warstwa 1 = PHY, c.d.

Jak przesłać ciąg danych (bitów) przez pod-kanal?

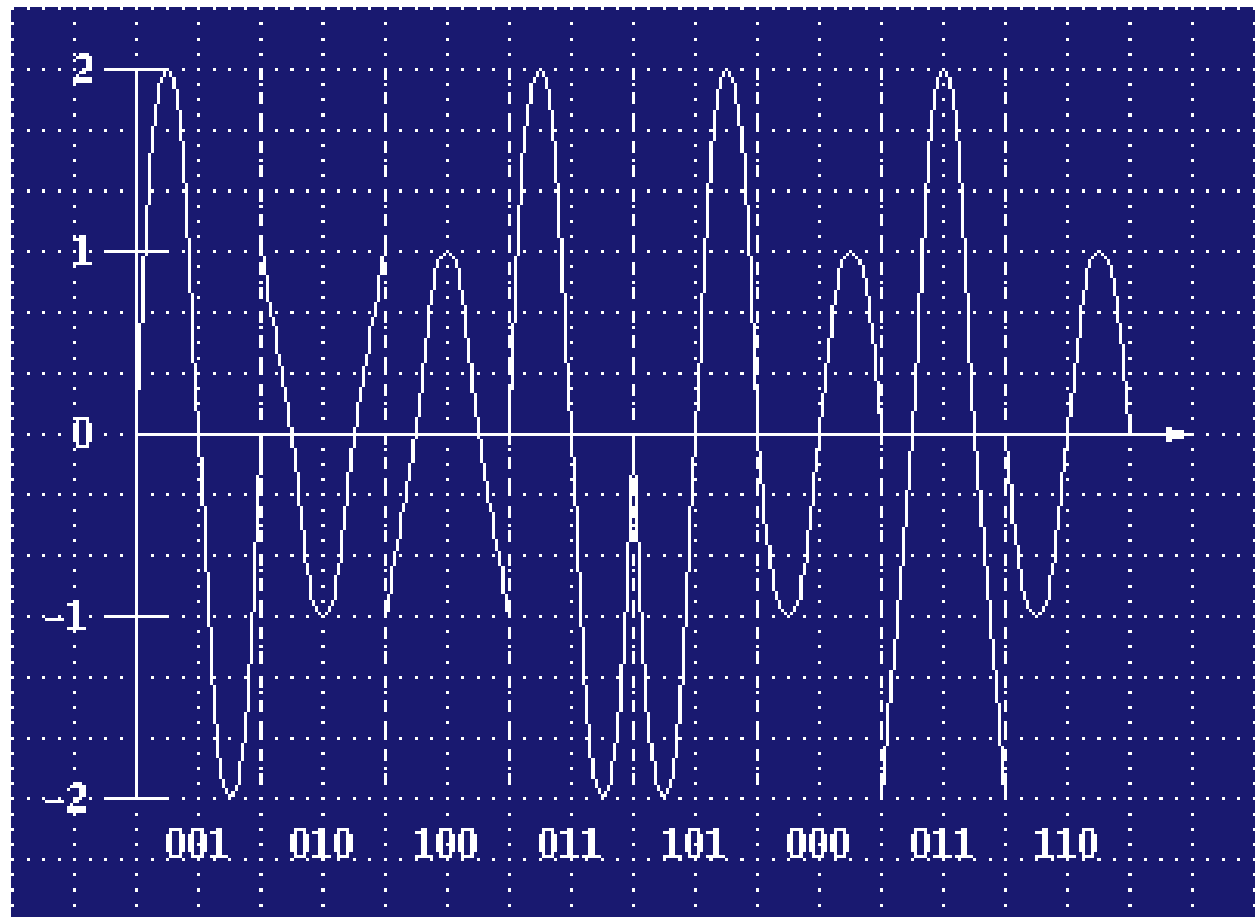
czyli modulowanie częstotliwości nośnej przez ciąg bitów...

są różne rozwiązania: AM (modulacja amplitudowa), FM (częstotliwościowa)

PSK (fazowa), mieszane, np. QAM (amplitudowo/fazowa, patrz tzw „konstelacje”):

Bit value	Amplitude	Phase shift
000	1	None
001	2	None
010	1	1/4
011	2	1/4
100	1	1/2
101	2	1/2
110	1	3/4
111	2	3/4

8-QAM



# Wifi, bezpieczeństwo

Zawsze na początku:

1. autentykacja (open, shared key/ **tego nie używać**)
2. asocjacja (powiązanie z jednym AP)

WEP (**tego nie używać**), Wired Equivalent Privacy

szyfr strumieniowy RC4, zbyt dużo danych się ujawnia w ramkach !!!  
4 klucze, nr klucza jest w ramce !

WPA/WPA2, 802.11i (tego używać), Wifi Protected Access

„Personal” WPA/WPA2-PSK PreShared Key

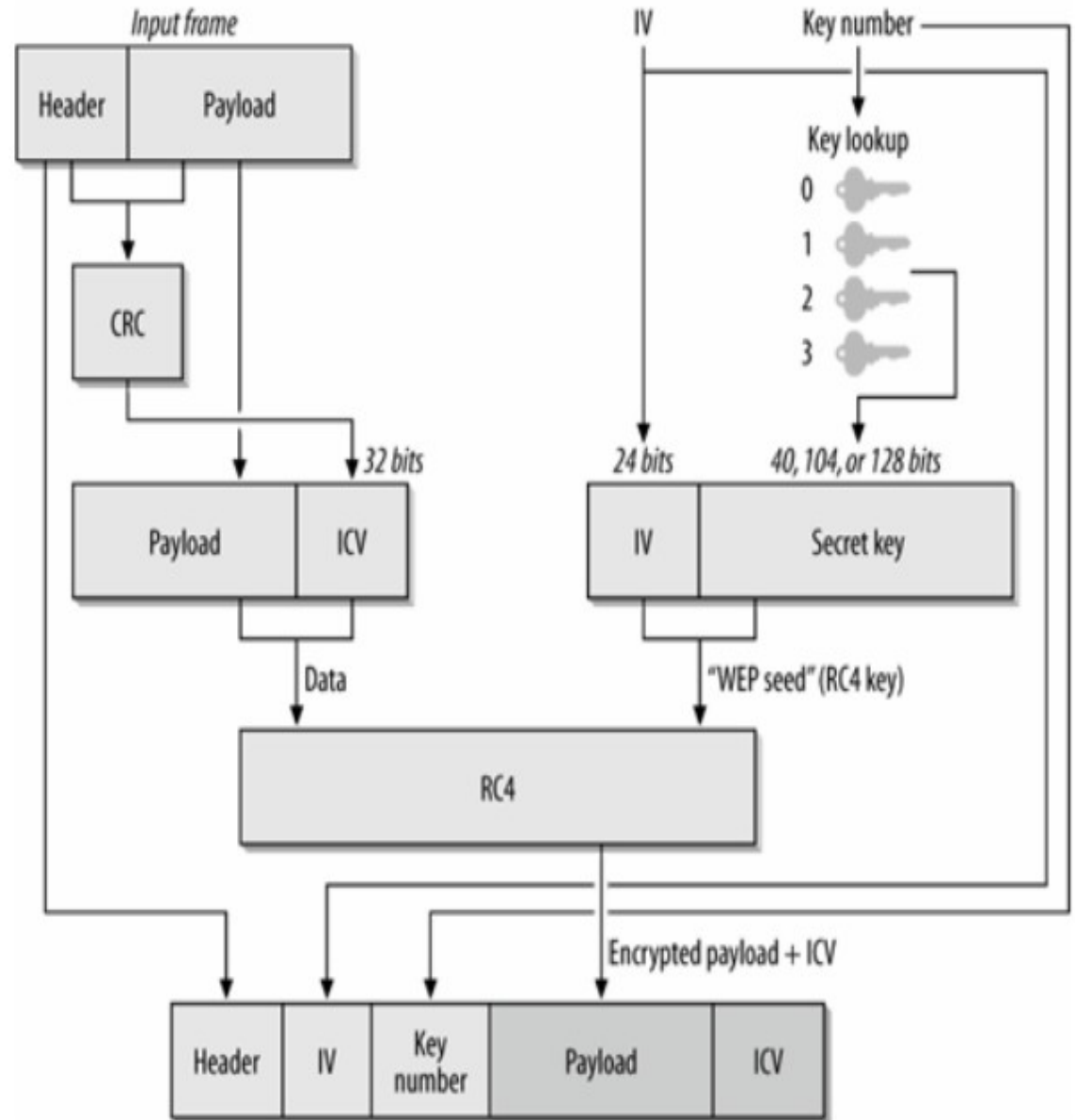
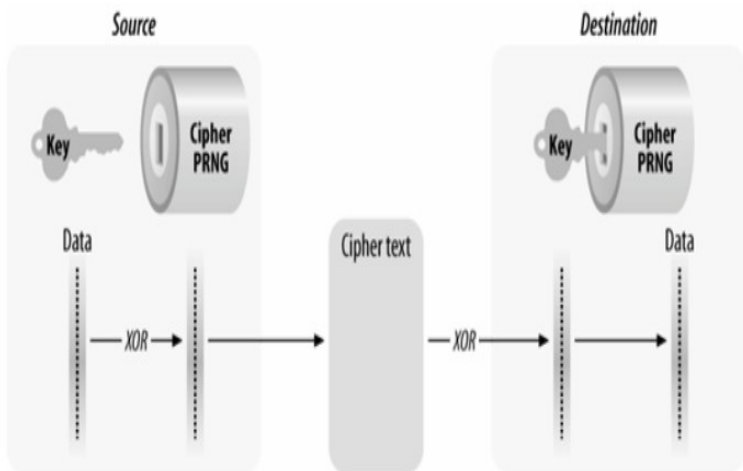
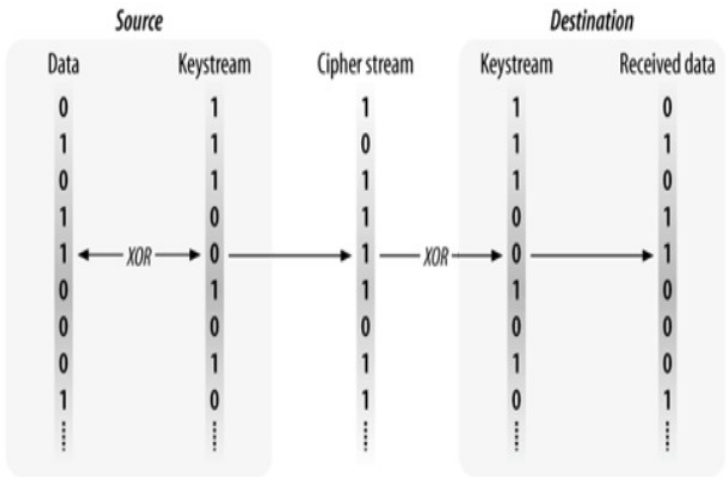
pojedynczy klucz używany przez AP i stacje

„Enterprise” używa się EAP (znanego też z PPP), 802.1X

komunikaty EAP, negocjacja sposobu uwierzytelnienia użytkowników,  
wpuszczanie użytkowników do sieci wifi na podstawie danych z ser aut,  
3 składniki:

1. suplikant (wpa\_supPLICant),
2. autentykator (AP),
3. serwer autentykacji (RADIUS)

# Wifi, WEP



# Wifi w Linux-ie

## *Polecenia linuxowe:*

ifconfig wlan0 up

iwconfig wlan0

iwconfig wlan0 mode managed/ad-hoc/monitor

Iwconfig wlan0 essid <nazwa sieci>

Iwconfig wlan0 chan <nr kanału>

iwlist wlan0 X/ X=scan, chan, keys, ...

wpa\_supplicant, wpa\_passwd (łatwa obsługa wpa/personal)

## *Jak się używa trybu ad-hoc ?*

wł mode=ad-hoc, nadać nazwę 1 węzłow przez subcmd essid,

pozostałe węzły podłączają się też przez subcmd essid...

można używać WEP... (nie można WPA ?!)

## *Jeśli chodzi o implementacje w linuxie...*

Wext (stare), nl80211, moduły cfg80211 mac80211 (nowe),

używa się gniazdek „netlink” (komunikacja user space - kernel)

# Bluetooth

Spec IEEE 802.15, RF 2.4GHz, zasięg 10m (klasa 2), zastępuje kabel rs232...

Przepustowość: ok 1Mb/s (721kb/s, w nowszych spec więcej...)

Warstwa fiz/radiowa: FHSS (skakanie po kanałach, 79 kanałów)

Zabezpieczenia? Parowanie urządzeń bt (*linux*: bluetooth-agent <kod>)

Master/Slave: 1 master  $\leq 7$  slaves, komunikacja wyłącznie M-S (nie S-S),

adr sprzętowe 6 bajtów (jak eth/wifi), adr M determinuje schemat FH !!!

wymiana komunikatów między M a S (kilka slotów czasowych, TDM)

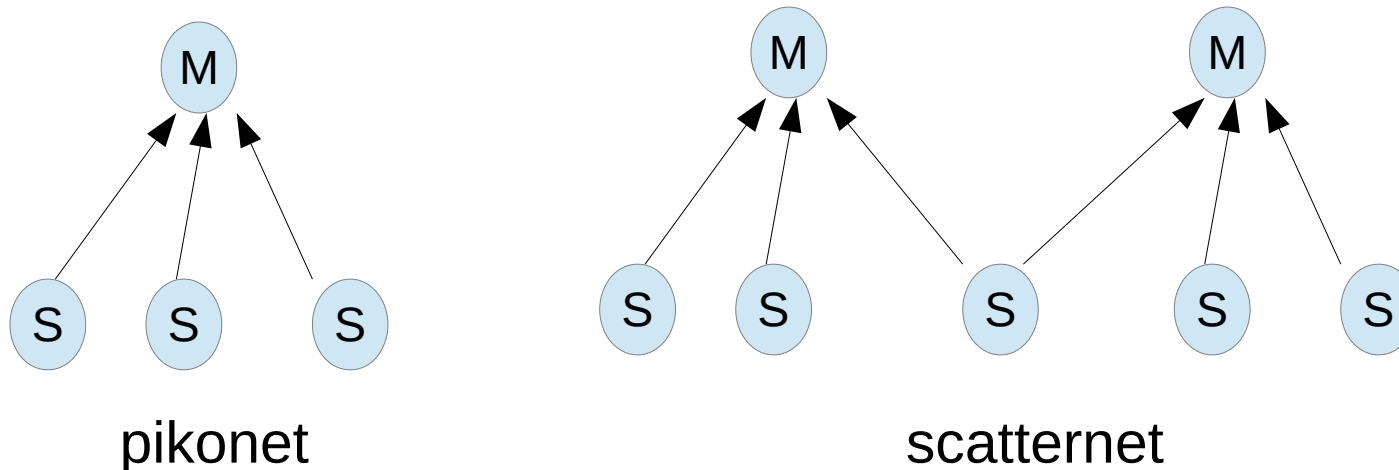
nadaje M lub jeden z S (wszystkim rządzi M)

Podstawowa sieć: **piconet**, wiele piconet-ów to **scatternet**

1 węzeł może być w wielu piconetach!!

problem formowania/obsługi scatternet nierozw. !!

dzięki FH (freq hop) jest szansa, że bliskie piconety używają innej częstotl w chwili t



# Bluetooth c.d.

Stos proto (nietypowy!!), profile (=usługi)

RF – warstwa radiowa, 2.4 GHz, 79 kanałów,  
FHSS,

Baseband – odp MAC, ramki, *nazwa myląca!!*

sloty czasowe 625mikrosek,  
ramki mogą być wielokr slotów,

wymiana ramek między M i S (runda?),

2 rodzaje logicznych kanałów:

ACL (asynch datagramy) i SCO (synch, audio)

LMP ???

L2CAP = Logical Link Control Adaptation Prot  
pkg o rozmiarze do 64kB, podobne do UDP??

RFCOMM – emul łącza szeregowego nad bt  
nad poł rfcmm można uruchomić prot PPP  
podobne do TCP??

Profile – rodzaj usług...

DUN dial-up networking

OBEX FTP przesył plików, prot OBEX

SDP ogłaszanie dostępnych profili

PAN, PANU siec nad bt...

Audio – np. słuchawki,

obsługiwane bez l2cap, bezp nad baseband...

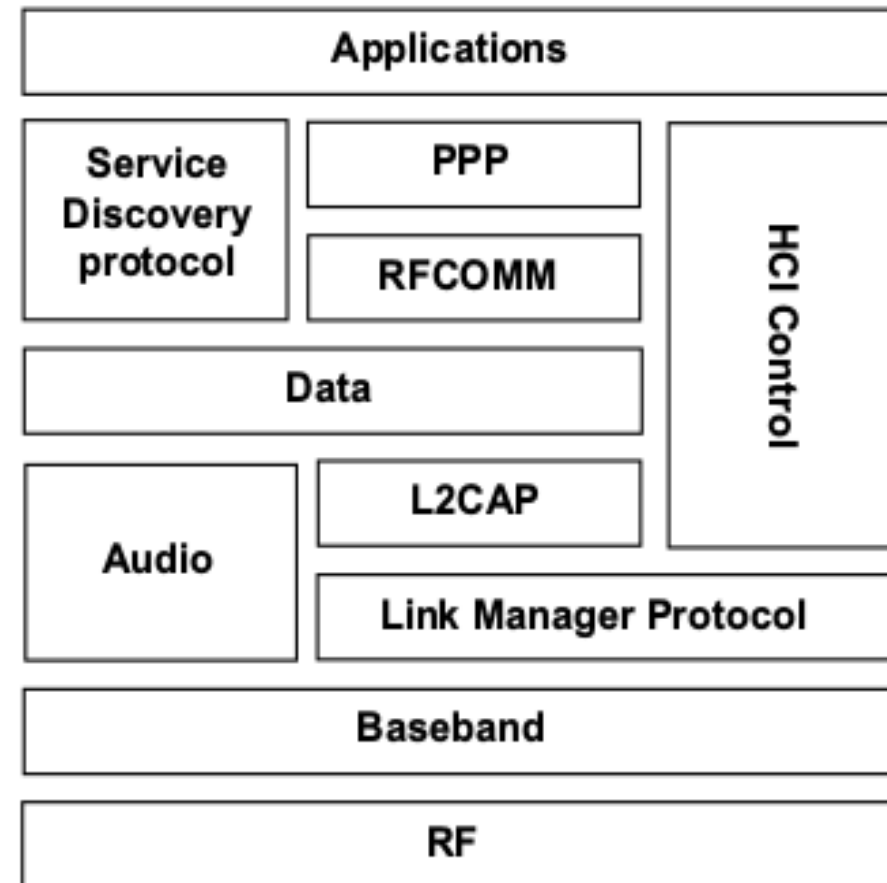


Fig. 1. Organisation of the BT stack.



# Bluetooth c.d.

Impl linuxowa prot bt: BlueZ...

*Komendy linuxowe:*

bluetoothd

bluetooth-agent <kod>

hcitool dev; # adr\_BT naszego urządzenia

hcitool scan; # wykrywanie sąsiednich urządzeń

hcitool con; # pokazuje połączenia M-S i typ kanału (ACL/SCO)

hcitool sr; # zamiana M/S

rfcomm connect <X> <adr\_BT> <kanał>; # powstanie plik /dev/rfcommX

rfcomm listen rfcommX <kanał>

sdptool browse <adr\_BT>; # pokazuje profile/usługi dostępne na urządzeniu...

Service Name: Dial-Up Networking

Service RecHandle: 0x10026

Service Class ID List:

"Dialup Networking" (0x1103)

Protocol Descriptor List:

"L2CAP" (0x0100)

"RFCOMM" (0x0003)

Channel: 22; # MH: rfcomm/ kanał 22 – to jest modem GPRS !!!

Language Base Attr List:

code\_ISO639: 0x454e

encoding: 0x6a

base\_offset: 0x100

Profile Descriptor List:

"Dialup Networking" (0x1103)

Version: 0x0100

