IEEE 802.11 PHY Layer Part 2: From bits to signals

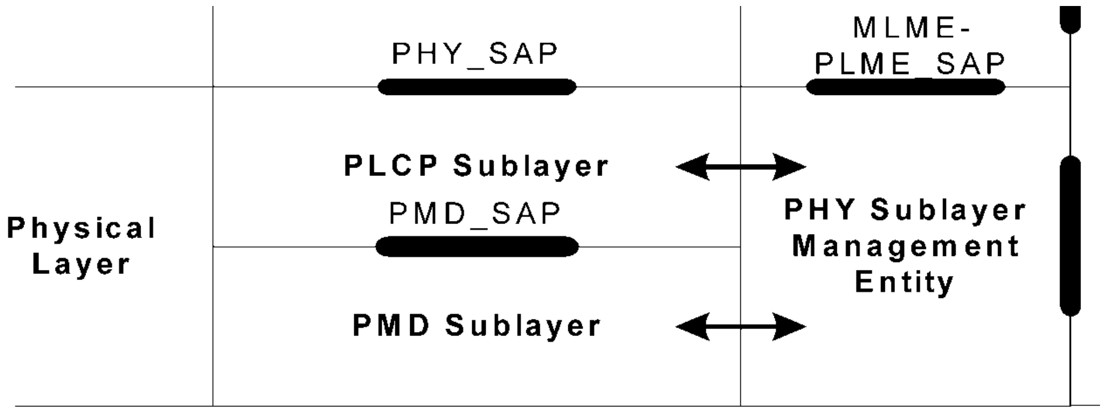
Based on IEEE Std 802.11-2012:

- Clause 18: Orthogonal frequency division multiplexing (OFDM) PHY specification
- Annex L: Examples of encoding a frame for OFDM PHYs. Example 1 - BCC encoding
- C. Beard and W. Stallings, Wireless Communication Networks and Systems, Chapter 8 –Orthogonal Frequency Division Multiplexing

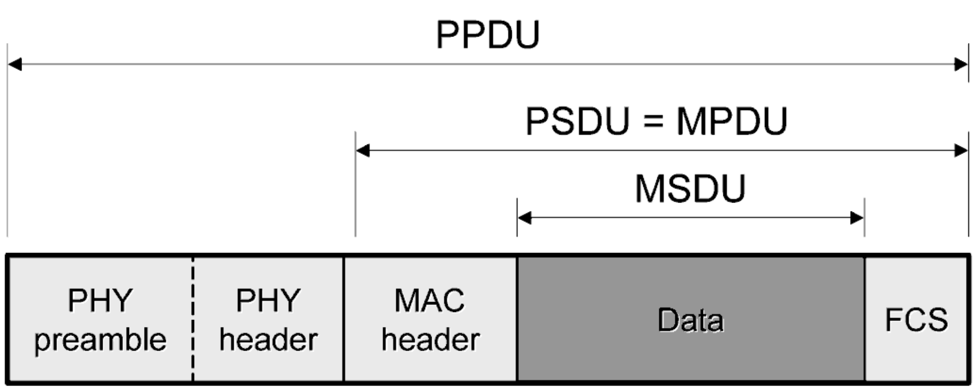
Learning Outcomes

- Explain how digital modulation techniques (QPSK, QAMs, etc) are used to convert from bits to signals
- Explain the principle used in OFDM and how multiple access is achieved using OFDM and SC-FDMA.
- Understand PPDU encoding process

PHY – Physical Layer



- Physical layer (PHY) consists of two sublayers
  - PLCP (Convergence Procedure) -- processing bits of PSDU
  - PMD (Medium Dependent) -- converting bits into the OFDM (Orthogonal Frequency Division Multiplexing) symbols



Key:

PPDU = PLCP protocol data unit
PSDU = PLCP service data unit
MPDU = MAC protocol data unit
MSDU = MAC service data unit

PLCP = physical (PHY) layer convergence procedure
MAC = medium access control

From bits to signals: PLCP

- In the first part of the PHY layer operations (PLCP – Convergence Procedure) the bits of the MPDU undergo the following operations:

Transmitter (Tx)

- Scrambling
- Convolutional encoding
- Puncturing
- Interleaving

Receiver (Rx)

- De-scrambling
- De-convolution (Viterbi algorithm)
- Inserting dummy bits
- De-interleaving

Tx

Signals

Rx

From bits to signals: PMD

- In the second part of the PHY layer operations, In the PMD – Medium Dependent sublayer, the bits of the message are converted into signals following two main steps:
  – Mapping groups of bits into a complex (frequency) domain using (typically) the Quadrature Amplitude Modulation (QAM)
  – Formation of the OFDM signals
- The receiver converts the received signals back to bits

Bit-Mapping by Digital Modulation

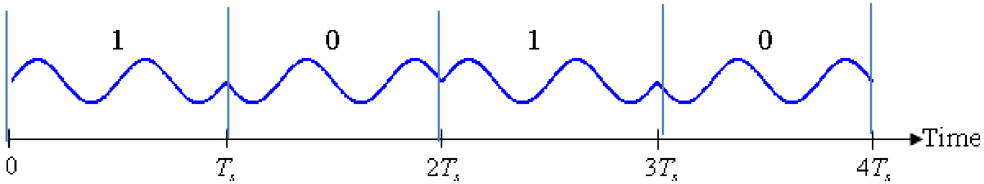
Read in week 3 Lecture Notes: Introduction to digital modulation.

- The process of **converting bits into signals**, that is sinusoids of different frequency and phases is called **Digital Modulation** or [Keying] or bit encoding.
- The simplest modulation technique is the Binary Phase Shift Keying (BPSK)
- In **BPSK** a single bit, $b \in (0,1)$, is converted into a sinusoid with the phase of either 0 or π:

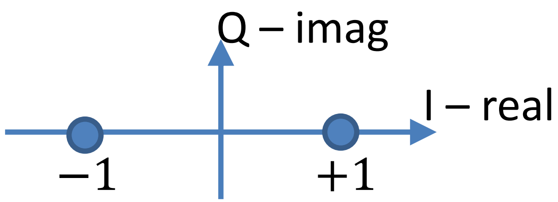
$$s(t) = (2b-1)\cos(\omega_c t) = \cos(\omega_c t + b\pi)$$

where $\omega_c = 2\pi f_c$, and f_c is the carrier frequency

In time domain:



The constellation diagram:



Quadrature Phase Shift Keying – QPSK

Encoding table:

$$\varphi_k = \frac{\pi}{4} + k\frac{\pi}{2} \text{ for } k = 0, 1, 2, 3$$

bits	11	01	00	10
k	0	1	2	3

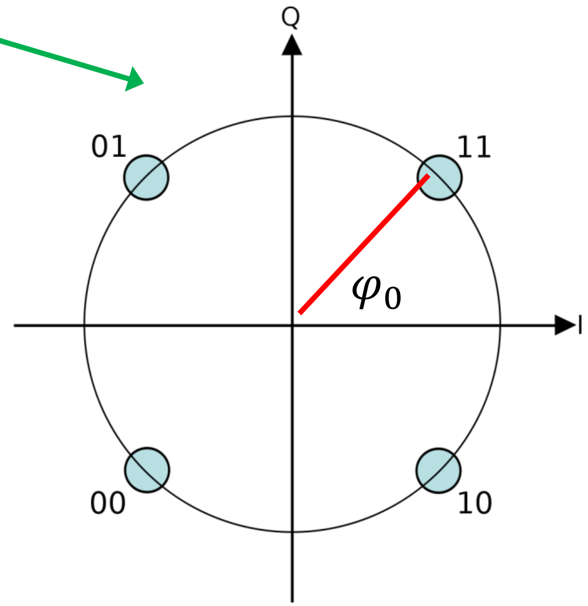
A **pair of bits** is represented by a sinusoidal signal of a fixed amplitude and the phase depending on the values of the bits

QPSK signal(s) in the time domain:

$$s_k(t) = e^{j(\omega_c t + \varphi_k)} = e^{j\omega_c t} e^{j\varphi_k}$$

$\omega_c = 2\pi f_c$, f_c – the carrier frequency

The **constellation diagram**



The quadrature (imaginary) component:

$$Q_k(t) = \sin(\omega_c t + \varphi_k) = \mathrm{imag}(s_k(t))$$

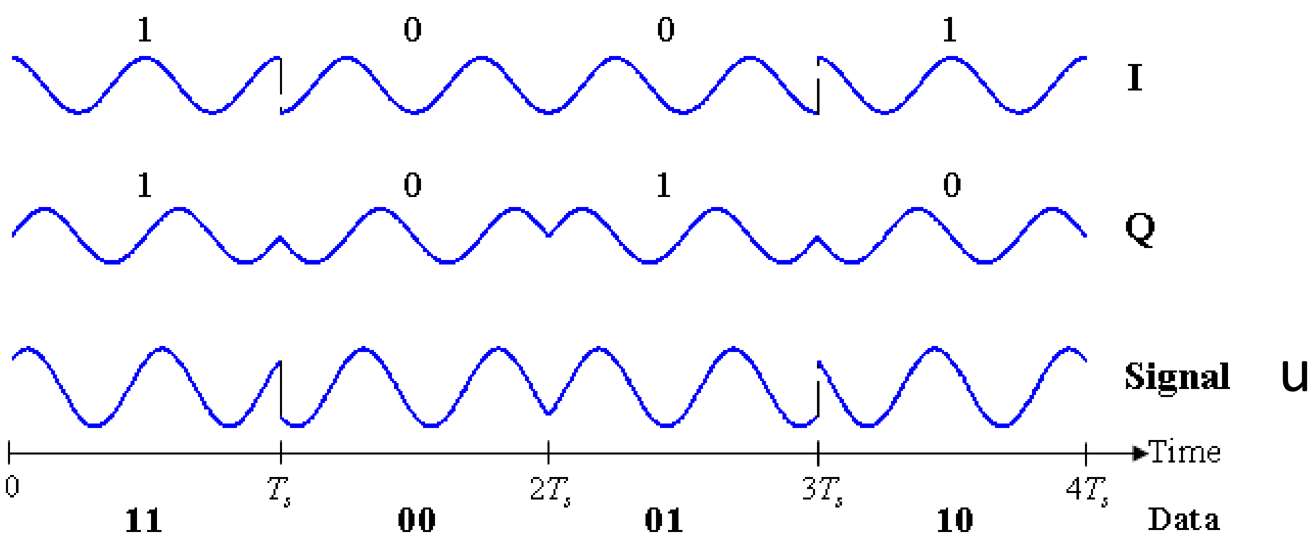
The in-phase (real) component:

$$I_k(t) = \cos(\omega_c t + \varphi_k) = \mathrm{real}(s_k(t))$$

QPSK signal in time domain

- The modulated signal consists of two components:
- a cosine wave (I – in-phase) and a sine wave (Q – quadrature).
- The total signal (in the bottom) is the sum of the two components
- The data consists of the binary stream:

1 1 0 0 0 1 1 0



Detecting phase from the sum $I_k + Q_k$

- If the total signal transmitted is:

$$u_k(t) = I_k(t) + Q_k(t)$$

$$u_k(t) = \cos(\omega_c t + \varphi_k) + \sin(\omega_c t + \varphi_k)$$

- The question is how we can extract at the receiver the phase φ_k so that we can get the bits back.
- Can be done with a little bit of trigonometry. Calculate:

➢ $cps = u_k \cos(\omega_c t) = \cos\varphi_k + \sin\varphi_k + f(2\omega_c t)$

➢ $cms = u_k \sin(\omega_c t) = \cos\varphi_k - \sin\varphi_k + f(2\omega_c t)$

Filter it out!

➢ $\cos\varphi_k = cps + cms$

➢ $\sin\varphi_k = cps - cms$

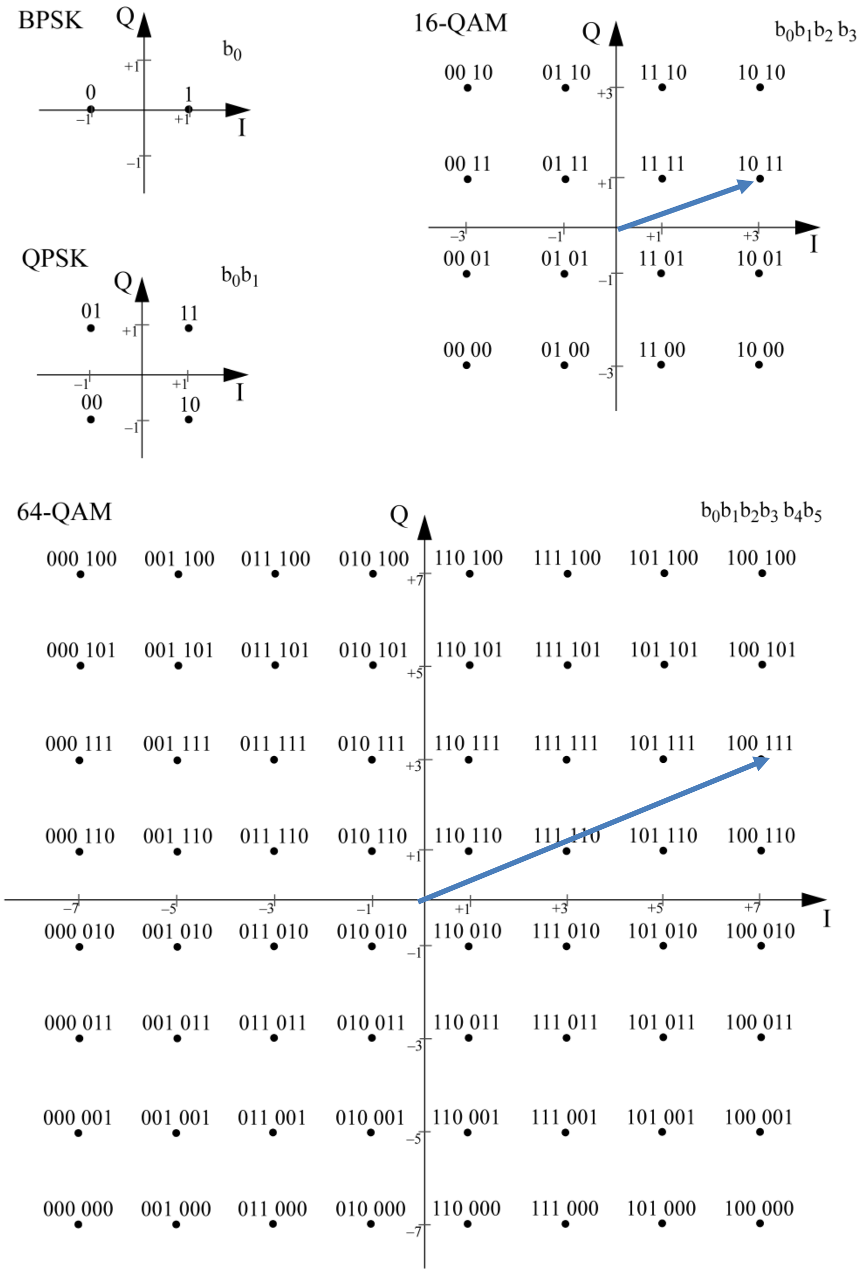


Figure 18-10—BPSK, QPSK, 16-QAM, and 64-QAM constellation bit encoding

In addition to **BPSK** and **QPSK** the 802.11 standard defines also:

- 16-QAM maps a group of **four** bits into a complex number representing a sinusoidal signal
- 64-QAM maps **six** bits into a complex number representing a sinusoidal signal
- The relative amplitudes of the Q and I components are

$$-7, -5, -3, -1, 1, 3, 5, 7$$

- A scaling factor is used to achieve the same average power for all mappings.

Encoding Tables

Table 18-9—QPSK encoding table

Input bit (b_0)	I-out
0	–1
1	1

Input bit (b_1)	Q-out
0	–1
1	1

Table 18-10—16-QAM encoding table

Input bits (b_0 b_1)	I-out
00	–3
01	–1
11	1
10	3

Input bits (b_2 b_3)	Q-out
00	–3
01	–1
11	1
10	3

Table 18-11—64-QAM encoding table

Input bits (b_0 b_1 b_2)	I-out
000	–7
001	–5
011	–3
010	–1
110	1
111	3
101	5
100	7

Input bits (b_3 b_4 b_5)	Q-out
000	–7
001	–5
011	–3
010	–1
110	1
111	3
101	5
100	7

Encoding tables links bits to points in the constellation diagram

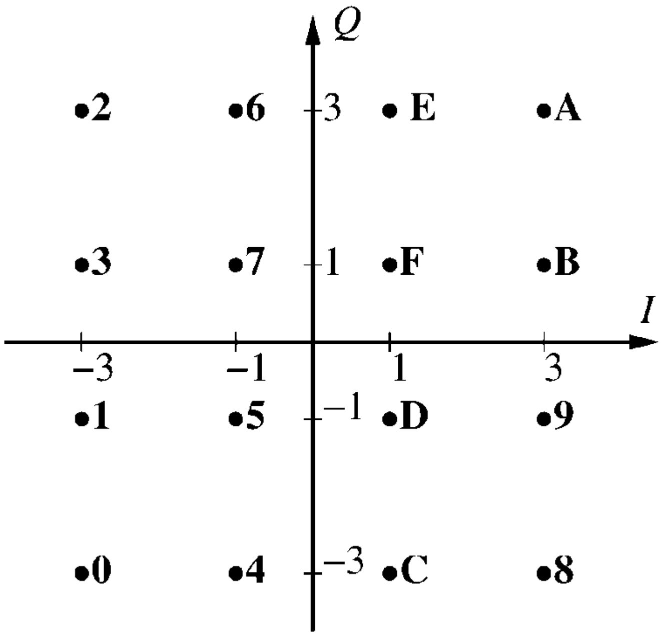
Note the order of groups of bits in the Gray code (aka reflected code)

Try the MATLAB code for 16QAM:

```
Iout = [-3 -1 3 1]   %  bb = [-3 -1 3 1]
Qout = Iout‘     %[Iout, Qout] = meshgrid(bb)
enT = Iout([1 1 1 1],:)+1i*Qout(:,[1 1 1 1])
                         % entT = Iout+1i*Qout ;
encT = enT(:)      % this is the encoding table
bits_d = (0:15)'
bits_h = dec2hex(bits_d)
[bits_h  num2str(bits_d, ‘| %3d |')
num2str(encT)]
```

(watch out for the MATLAB-correct apostrophes: ‘ }

16-QAM Example



- Assume we have a collection of **groups of 4 bits** $b_0 b_1 b_2 b_3$ represented by the hex numbers:
- $h_k =$ F, 4, E, A, 2, 6, D, …,
- $k = 1, 2, \ldots, 7, \ldots$

- Then, for 16-QAM the related complex numbers $I + jQ$ are:

$h_k =$ F 4 E A 2 6 D

$d_k =$ [1+j, −1−3j, 1+3j, 3+3j, −3+3j, −1+3j ,1−j]/$\sqrt{10}$

a scaling factor

How to convert the complex numbers into the sinusoidal signals?
The best method is to use the **ifft** (inverse Fast Fourier Transform)

Review of the Spectral Processing

- At this stage it would be beneficial to re-examine examples from tutorial week 1 related to spectral analysis/processing.
- Let us look at sec. 1.5: Synthesizing a two-sinusoid signal
- We have specified the frequency range (aka **sampling frequency**) fs = 8000Hz and the **number of samples** tt =20. The **frequency step** is df = fs/tt = 400Hz
- We defined two frequency points one at 400 Hz and one at 2400Hz so that the **half-spectrum** is:

f	0	400	800	1200	1600	2000	2400	2800	3200	3600
n	0	1	2	3	4	5	6	7	8	9
h[n]	0	2.5	0	0	0	0	7.5	0	0	0

Synthesizing two-sinusoid signal, cont'd

- The **half-spectrum** is:

f	0	400	800	1200	1600	2000	2400	2800	3200	3600
n	0	1	2	3	4	5	6	7	8	9
h[n]	0	2.5	0	0	0	0	7.5	0	0	0

- The full spectrum consists of the half-spectrum and is (conjugated) mirror reflection:

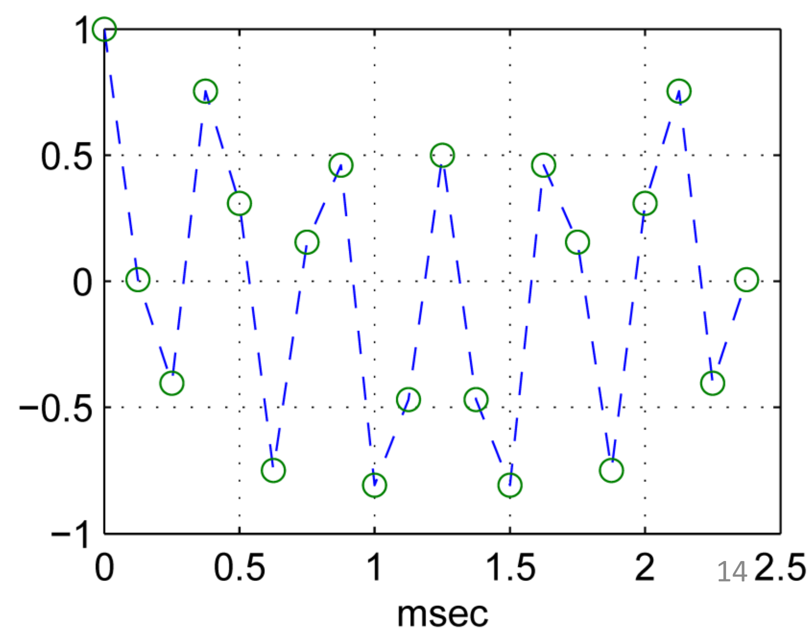
  h = [hn ; hn(end-1:-1:2)]

- The time domain signal is obtained as the inverse Fourier transform:

  s_t = ifft(h)

- The duration of such a symbol is

  Ts = 1/400 = 2.5ms



Another example on the path to OFMD

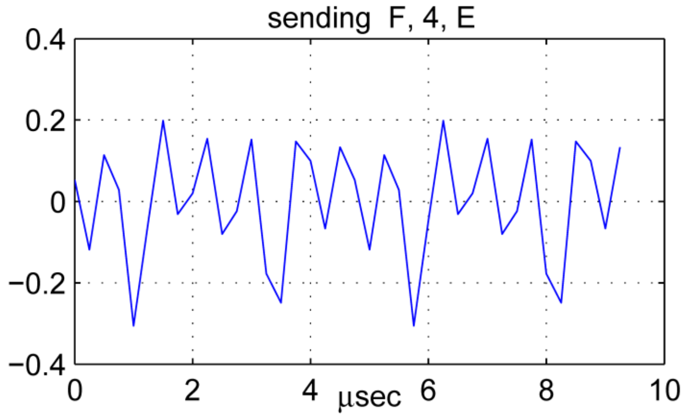
- Following the 16QAM example we would like to get time signals equivalent to the following combination of bits:

$$h_k \;=\; \text{F} \quad 4 \quad \text{E}$$

$$d_k = 1\text{+j, –1–3j, 1+3j}$$

- Let us assume that the **frequency spacing** df = 100kHz and let the groups of bits be located at 400, 800 and 1200kHz, and the **number of frequency steps** be: tt = 40.
- The **sampling frequency** is: fs = df*tt = 4MHz.
- hn = zeros(1, tt/2); hn([5, 9, 13]) = [1+1i, -1-1i, 1+3i] ;
- The full spectrum:

  h = [hn conj(hn(end-1:-1:2))];
- The time signal is s = ifft(h) ;



OFDM principle

- In OFDM (**Orthogonal Frequency Division Multiplexing**) the 20MHz frequency channel is divided into **64 frequency subcarriers**.
- The subcarrier spacing $\Delta_F = 20/64 = 315.5\text{kHz}$
- Conceptually, each **complex number** d_k obtained form QAM (Quadrature Amplitude Modulation) that **codes a group of bits** will represent a sinusoidal signal of the OFDM sub-carrier frequency

$$f_k = k\Delta_F$$

- The equivalent time signal is obtained by the Inverse Fourier Transform and can be written as:

$$r(t) = \sum_k d_k \exp(j2\pi k \Delta_F t) = \mathrm{ifft}(d_k)$$

- Some details have been omitted for clarity.

OFDM Subcarriers

- Each of the OFDM symbols is represented by 64 subcarriers numbered in the following way Figure 18-11 (p.22 out of 48)

$$\begin{array}{lcccccccccccccc} & d_0 \ldots d_4 & p_{-21} & d_5 \ldots d_{17} & p_{-7} & d_{18} \ldots d_{23} & \mathrm{DC} & d_{24} \ldots d_{29} & p_7 & d_{30} \ldots d_{42} & p_{21} & d_{43} \ldots d_{47} \\ \boldsymbol{f} = & -26 & -21 & & -7 & -1 & 0 & 1 & 7 & & 21 & 26 \end{array}$$

- 53 subcarriers are numbered form –26 to 26.
- Subcarrier 0 representing the frequency 0 (DC) is set to zero.
- 48 subcarriers marked d_k are used to carry data
- Data points are complex numbers obtained from QAM mapping
- 4 subcarriers marked p_k carry the pilot signals,
- Pilots are $-1, +1$ values (see 18.3.5.9 for details)
- Pilot signals help to synchronize the transmitter and receiver

Converting OFDM symbol into time domain

$$f = \boxed{\underset{-26}{d_0} \dots d_4 \underset{-21}{p_{-21}} d_5 \dots d_{17} \underset{-7}{p_{-7}} d_{18} \dots \underset{-1}{d_{23}}} \underset{0}{\mathrm{DC}} \boxed{\underset{1}{d_{24}} \dots d_{29} \underset{7}{p_7} d_{30} \dots d_{42} \underset{21}{p_{21}} d_{43} \dots \underset{26}{d_{47}}}$$

- In order to convert the OFDM symbol into its time domain representation the 52 subcarriers are reorganised into a 64-point vector as follows 18.3.2.6 (p.11)

$$h = [0 \ \ f(1) \dots f(26) \ \ 0 \ \dots 0 \ \ f(-26) \dots f(-1) \,]$$

- This is equivalent to forming the full spectrum in the previous examples
- The time domain signal is calculated using the inverse Fourier Transform:

$$s = \mathrm{ifft}(h)$$

Duration of an OFDM Symbol

- The duration of one OFDM symbol t_F (symbol interval) is

  the inverse of the subcarrier spacing

$$t_F = \frac{1}{\Delta_F} = \frac{64}{20} = 3.2\mu s$$

  plus **time guard** t_g= 0.8µs:

$$t_s = t_g + t_F = 0.8\mu s + 3.2\mu s = 4\mu s$$

- The time guard is formed by the **cyclic extension** of the **time domain** OFDM symbol s(1) … s(64):

s(49) … s(64)	s(1) … s(64)	s(1)

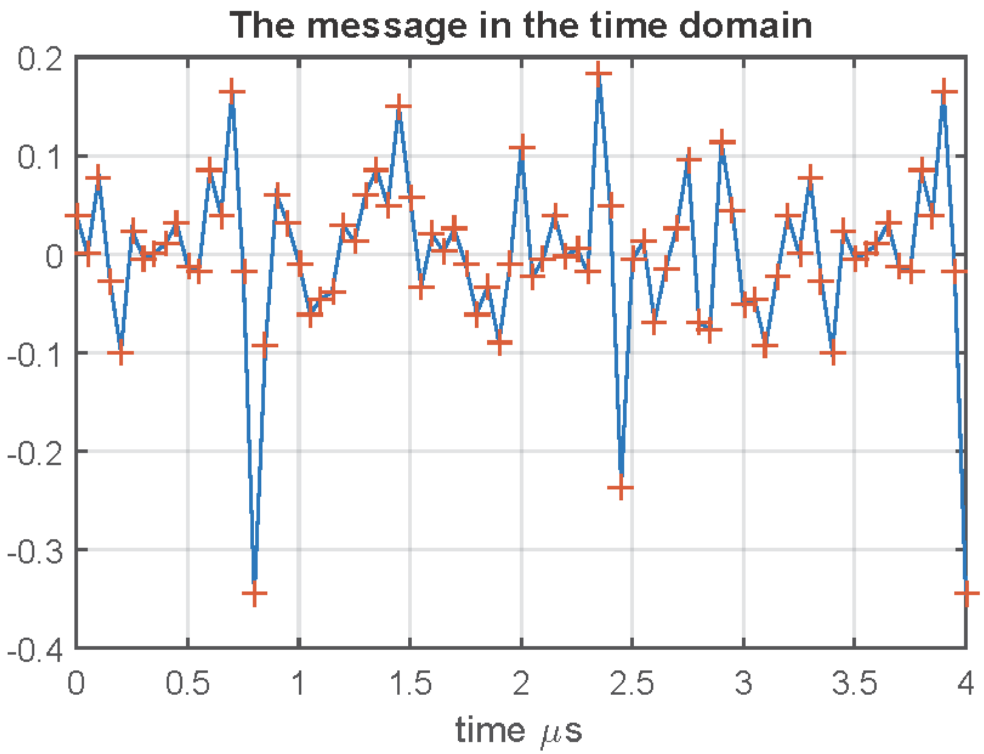
- The resulting time vector has 81 time samples.

OFDM example

- Follow the example that converts the message:

Wireless Networks are great

- into the OFDM time signal:



Modulation dependant parameters

Table 18-4—Modulation-dependent parameters

Modulation	Coding rate (R)	Coded bits per subcarrier (N_{BPSC})	Coded bits per OFDM symbol (N_{CBPS})	Data bits per OFDM symbol (N_{DBPS})	Data rate (Mb/s) (20 MHz channel spacing)
BPSK	1/2	1	48	24	6
BPSK	3/4	1	48	36	9
QPSK	1/2	2	96	48	12
QPSK	3/4	2	96	72	18
16-QAM	1/2	4	192	96	24
16-QAM	3/4	4	192	144	36
64-QAM	2/3	6	288	192	48
64-QAM	3/4	6	288	216	54

- **Coding rate** is related to convolutional coding and puncturing
- Data rate, is related to the **coding rate** and modulation type
- For 36Mb/s **16-QAM with ¾ coding rate** must be used and we have:

- 16-QAM implies $N_{BPSC} = 4$, and $N_{CBPS} = 4 * 48 = 192$ bits where 48 is the number of subcarriers with data
- The number of data bits is smaller by the coding rate

$$N_{BPSC} = \frac{3}{4}192=144 \text{ bits}$$

PHY frame format

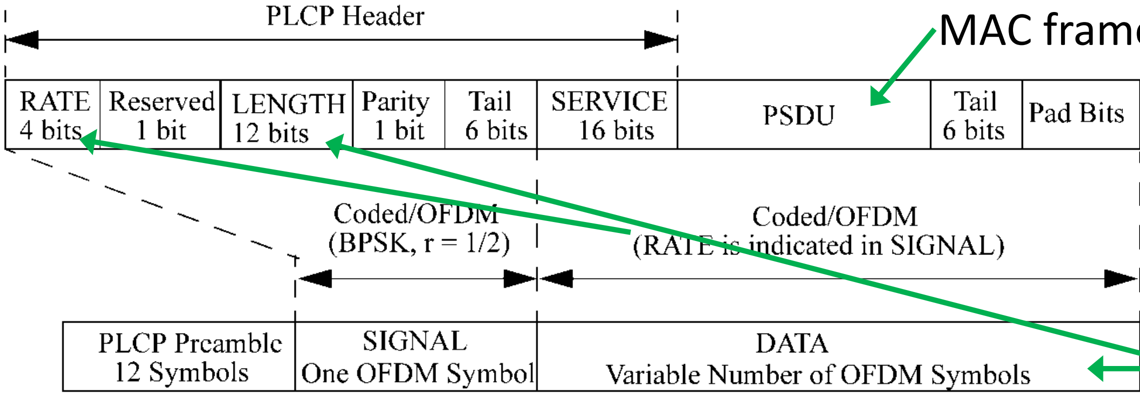


Figure 18-1—PPDU frame format

MAC frame with FCS

PLCP Header = SIGNAL + SERVICE

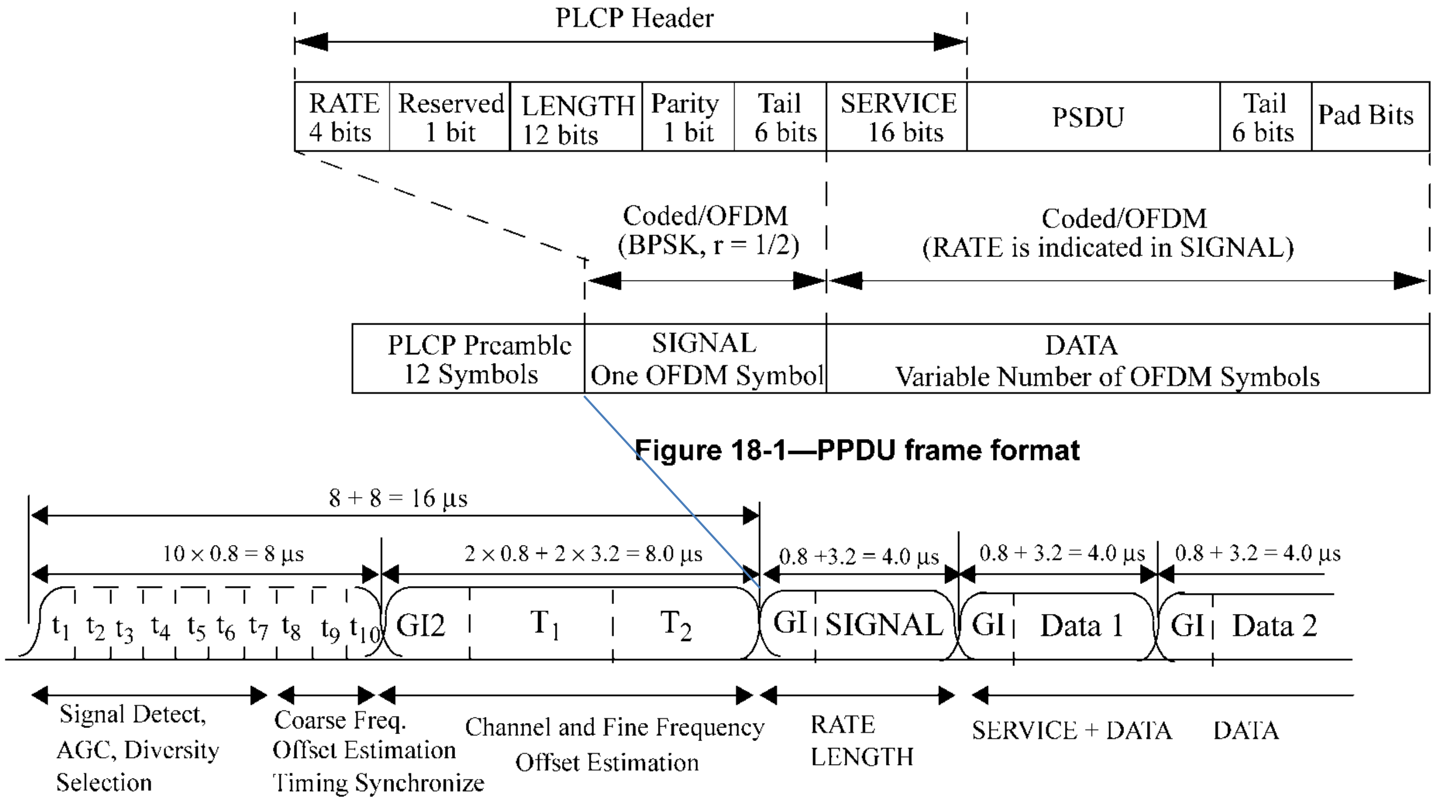
Number of data bits = $m * N_{CBPS}$

Pad bits must be added

Data includes:

- the PSDU=MPDU (MAC header, data, FCS),
- 16 SERVICE bits (see 18.3.5.2),
- 6 tail bits
- a number of pad bits so that the total number of bits is a multiple of N_{CBPS} , that is, all OFDM symbols are fully filled in with bits.

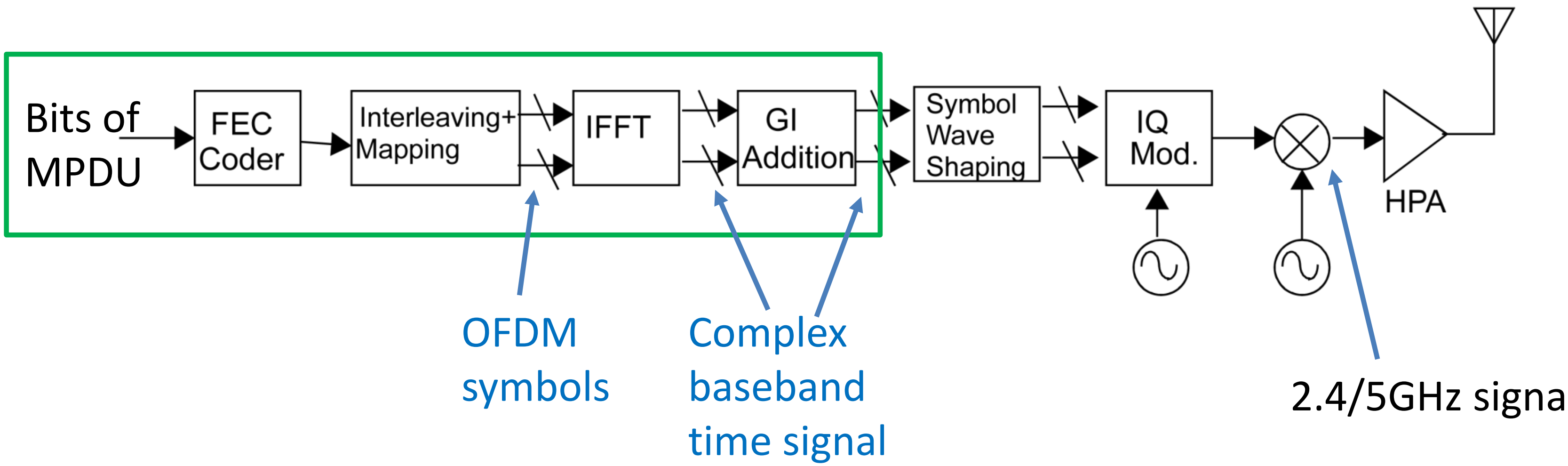
Timing relationships



OFDM signals

- The PLCP Preamble field is used for synchronization (sec. 18.3.3)
- It consists of 10 short symbols and two long symbols.
- GI represents a guard interval.

Transmitter block diagram (p.23)



- FEC – Forward Error Correction
- IFFT – Inverse Fast Fourier Transform
- IQ Mod. – modulation – frequency shift from the baseband 20 MHz channel into 2.4/5GHz band, e.g. 2407 to 2412 MHz (channel #1)
- HPA – Antenna amplifiers

Overview of the PPDU encoding process (18.3.2.2) p.7

PLCP Preamble:

a) Produce the **PLCP Preamble** field, composed of
   - 10 repetitions of a “short training sequence” and
   - two repetitions of a “long training sequence”
b) Produce the **PLCP header** by filling in the RATE, LENGTH, and SERVICE fields.
c) Calculate from RATE field
   - the **number of data bits per OFDM symbol** (N_{DBPS}),
   - the coding rate (R),
   - the number of bits in each **OFDM subcarrier** (N_{BPSC}), and
   - the **number of coded bits per OFDM symbol** (N_{CBPS}).
d) Append the PSDU to the SERVICE field of the TXVECTOR (data).

Overview of the PPDU encoding process -- bits

e) Initiate the **scrambler** with a pseudo-random nonzero seed, generate a scrambling sequence, and

   XOR is with the extended string of data bits.

f) Replace the six scrambled zero tail bits following the data with six nonscrambled zero bits to return the convolutional encoder to the zero state.

g) Encode the extended, scrambled data string with a **convolutional encoder** (R = 1/2).

   Omit (**puncture**) some of the encoder output string (chosen according to “puncturing pattern”) to reach the desired “coding rate.”

Overview of the PPDU encoding process: OFDM 1

h) Divide the encoded bit string into groups of N_{CBPS} bits.
Within each group, perform an "**interleaving**" (reordering) of the bits according to a rule corresponding to the desired RATE.

i) Divide the resulting coded and interleaved data string into groups of N_{CBPS} bits.
   – For each of the bit groups, **convert the bit group into a complex number according to the modulation** encoding tables.

j) Divide the complex number string into groups of 48 complex numbers.
   – Each such group will be associated with one OFDM symbol.

k) Four subcarriers are inserted as pilots .
   – The total number of the subcarriers is 52 (48 + 4).

Overview of the PPDU encoding process: OFDM 2

l) For each group of subcarriers –26 to 26, convert the subcarriers to time domain using inverse Fourier transform.
   – Prepend to the Fourier-transformed waveform a circular extension of itself thus forming a GI.

m) Append the OFDM symbols one after another, starting after the SIGNAL symbol describing the RATE and LENGTH fields.

n) Up-convert the resulting "complex baseband" waveform to an RF according to the center frequency of the desired channel and transmit.

Example of BCC Frame Encoding

- We will follow the steps of encoding a BCC (Block Convolutional Code) frame as presented in IEEE Std 802.11 Annex L (Moodle).
- You have practiced most of the bit processing in the week 6 tute.
- In today's tute you can re-use the code from tute 6.
- Most of the code is available from Moodle

Summary

Reflect on:

- Explain how digital modulation techniques (QPSK, QAMs, etc) are used to convert from bits to signals
- Explain the principle used in OFDM and how multiple access is achieved using OFDM and SC-FDMA.
- Understand PPDU encoding process

Tutorial 7: The implementation of BCC. Reminder: submit together tutorial 6 by 23:55hr on Thursday 12 April 2018

Lecture 08: Mobile Network/LTE