

EF012 디지털 통신 PROJECT: WIFI 물리계층 구현

제출일: 5/19(수) 11:55pm

1. Wi-Fi 물리계층

컴퓨터 통신에서 정보는 계층구조를 이용해서 전달되는데, 위에서부터 아래 방향으로 어플리케이션 계층, TCP 계층, IP 계층, MAC 계층, 물리 계층의 다섯 계층이 있습니다. 이 프로젝트에서는 무선랜 또는 WIFI라 부르는 전송 방식의 물리 계층 동작 방식을 공부해보려고 합니다. 즉, 여러분들은 WIFI를 구현하는 칩의 동작 원리를 따라해 볼 것입니다. 이를 위해 메세지로부터 와이파이 패킷을 생성하고 이로부터 WIFI 신호를 생성하는 matlab 프로그램이 여러분에게 제공됩니다. 여러분은 이 프로세스의 반대되는 프로세스, 즉, WIFI 신호를 받아들이어 최종적으로 메시지를 복원하는 작업을 하는 matlab 프로그램을 만들면 됩니다.

2. WIFI 송신단의 code

여러분에게 다음과 같이 정의된 WIFI 신호를 생성하는 "wifitransmitter.m" 파일이 제공됩니다

```
txsignal = wifitransmitter(message, level, snr)
```

wifitransmitter 함수는 다음을 입력으로 받아들입니다.

- message: WIFI 패킷의 내용을 포함하는 문자 메세지. 문자는 ASCII 문자들이어야 하며 문자열의 길이는 10000 바이트를 넘어서는 안 됩니다.
- level: 인코딩의 단계를 나타냅니다. 메시지는 WIFI 신호가 되기 위해 5 단계의 인코딩을 거칩니다. (1) level-1/coding: 비트들에 터보코드(turbo-code)를 적용; (2) level-2/interleaving: 비트들에 자리 바꿈을 수행; (3) level-3/modulation: 비트에 BPSK을 적용; (4) level-4/OFDM: 비트들에 OFDM 적용; (5) level-5/noise: 패킷에 잡음을 추가하고 패킷의 시작과 끝에 zero-padding을 추가. level의 기본값(default value)은 5.
- snr: signal to noise ratio. 신호대잡음비를 데시벨(dB)로 나타냄. snr의 기본값(default value)은 무한대.

위의 함수가 올바른 입력값들을 받아들여 실행되면 txsignal은 복소수의 수열의 인코딩된 패킷이 됩니다.

여러분은 다양한 입력값들을 이용해서 이 함수를 수행해 보고 출력값을 관찰해 보기 바랍니다. 각 레벨의 동작을 이해하기 위해 이 함수를 상세히 공부하기 바랍니다.

3. 해야 할 것: WIFI 수신부의 코드

여러분이 해야 할 일은 wifitransmitter 가 생성한 신호 txsignal 를 입력값으로 읽어 들여 message 를 출력값으로 생성하는 WIFI 수신부 함수를 작성하는 것입니다. 즉, 송신부 함수의 반대되는 프로세스를 만들면 됩니다. 여러분이 제출한 함수 프로그램은 다음과 같이 실행될 것입니다.

```
[message, length, start] = wifireceiver(txsignal, level)
```

여러분들이 제출해야 할 것은 matlab 으로 만들어진 wifireceiver.m 입니다.

여러분들이 제출한 함수는 다음과 같은 테스트를 거칠 것입니다. 예를 들어 'hello world'라는 메시지를 사용한다고 가정합니다. 물론 여러분의 프로그램은 다른 입력 메시지에 대해서도 작동을 해야 합니다.

(1) level 1 test (20 점)

level-1 테스트에서는 송신단에서 turbo coding 이 적용됩니다.

matlab 에서 다음을 수행했을 때에

```
>> txsignal = wifitransmitter('hello world', 1)
>> wifireceiver(txsignal, 1)
```

다음은 출력해야 합니다.

```
hello world
```

여러분의 프로그램은 이 레벨에서 turbo decoding 을 올바르게 수행해야 합니다.

(2) level 2 test (10 점)

level-2 테스트에서는 송신단에서 turbo coding 과 interleaving 이 적용됩니다.

matlab 에서 다음을 수행했을 때에

```
>> txsignal = wifitransmitter('hello world', 2)
>> wifireceiver(txsignal, 2)
```

다음은 출력해야 합니다.

```
hello world
```

여러분의 프로그램은 이 레벨에서 turbo decoding 과 interleave 의 역을 올바르게 수행해야 합니다.

(3) level 3 test (10 점)

level-3 테스트에서는 송신단에서 turbo coding, interleaving, BPSK modulation 이 적용됩니다.

matlab 에서 다음을 수행했을 때에

```
>> txsignal = wifitransmitter('hello world', 3)
>> wifireceiver(txsignal, 3)
```

다음은 출력해야 합니다.

```
hello world
```

여러분의 프로그램은 이 레벨에서 turbo decoding, interleave 의 역, BPSK demodulation 을 올바르게 수행해야 합니다.

(4)level 4 test (10 점)

level-4 테스트에서는 송신단에서 turbo coding, interleaving, modulation, OFDM 신호 생성이 적용됩니다.

matlab 에서 다음을 수행했을 때에

```
>> txsignal = wifitransmitter('hello world', 4)
>> wifireceiver(txsignal, 4)
```

다음은 출력해야 합니다.

```
hello world
```

여러분의 프로그램은 이 레벨에서 turbo decoding, interleave 의 역, BPSK demodulation, OFDM demodulation 을 올바르게 수행해야 합니다.

(5)level 5 test (30 점)

level-5 테스트에서는 송신단에서 노이즈 추가를 포함해서 모든 단계가 적용됩니다.

matlab 에서 다음을 수행했을 때에

```
>> txsignal = wifitransmitter('hello world', 5, 30)
>> [message, length, start] = wifireceiver(txsignal, 5)
```

다음은 출력해야 합니다.

```
message = 'hello world'
length = 11
start = <number of padded zeros before the packet>
(= length(noise_pad_begin) in wifitransmitter.m)
```

여러분의 프로그램은 이 레벨에서 turbo decoding, interleave 의 역, BPSK demodulation, OFDM demodulation, 패킷의 시작 찾기를 올바르게 수행해야 합니다. length 는 보내는 메시지의 길이를 의미합니다. start 는 preamble 이전에 있는 padded zero 들의 길이를 의미합니다. 위의 세 개의 출력에 대해 각각 10 점이 부여됩니다. 이때 snr 값은 30 dB 이상이어야 합니다.

4. 추가 조언

반드시 일찍 시작하기 바랍니다. 여러분이 필요로 하는 대부분의 함수가 matlab 안에 이미 구현되어 있습니다. 그걸 이용하면 수월하게 프로그램을 만들 수 있습니다. level-1 부터 만들어 가기 바랍니다. level-1 이 성공하면 그 level-2,3,4,5 로 이동하면서 진행하기 바랍니다. 질문하기 전에 우선 많은 노력을 해보기 바랍니다. (matlab 사용법 및 명령어 공부 등) 그래도 모르겠다면 내게 이메일로 연락하기 바랍니다. (msoh@kyonggi.ac.kr)

5. 제출해야 할 것들

LMS 를 통해서 다음 것들을 제출하기 바랍니다.

- wifireceiver.m
- 다음에 대한 간략한 설명을 보여주는 파워포인트 design.pptx 문서
 - o 각 단계에서 사용한 함수에 대한 설명
 - o preamble 을 어떻게 찾았는지

6. 배점

- 제출 여부 10 점
- design.pptx 10 점
- level-1 20 점
- level-2 10 점
- level-3 10 점
- level-4 10 점
- level-5 30 점

7. matlab 사용

공대 PC 실(제2공학관 101 호)에 비교적 최신판(2019b) matlab 이 설치되어 있습니다. 그리고 본 과제를 수행하는데 필요한 패키지도 설치되어 있습니다. 사용 가능 시간은 주중 12:00 - 14:00 그리고 월목금 17:00 시 이후 공대교학탕에 요청하고 사용 가능합니다. 전자공학과 PC 실(402 호)의 PC 에는 "communications system toolbox"가 없어 프로그램이 실행되지 않습니다.

본 프로젝트를 실행하는데 필요한 matlab toolbox 는 다음과 같습니다.

- communications toolbox
- deep learning toolbox (필요할 수도 또는 필요 없을 수도 있음)
- DSP system toolbox
- signal processing toolbox

convolutional codes (level 1)

- message bit 들이 block 단위로 들어오는 것이 아니라 연달아 들어오는 경우가 있다
- 이런 경우에 convolutional encoder 는 연달아 들어오는 입력에 대해 연달아 출력을 생성한다.
- binary convolutional encoder 은 다음으로 이루어져 있다
 - M -stage shift register
 - n 개의 modulo-2 adder
 - adder 의 출력을 serial 하게 만드는 multiplexer

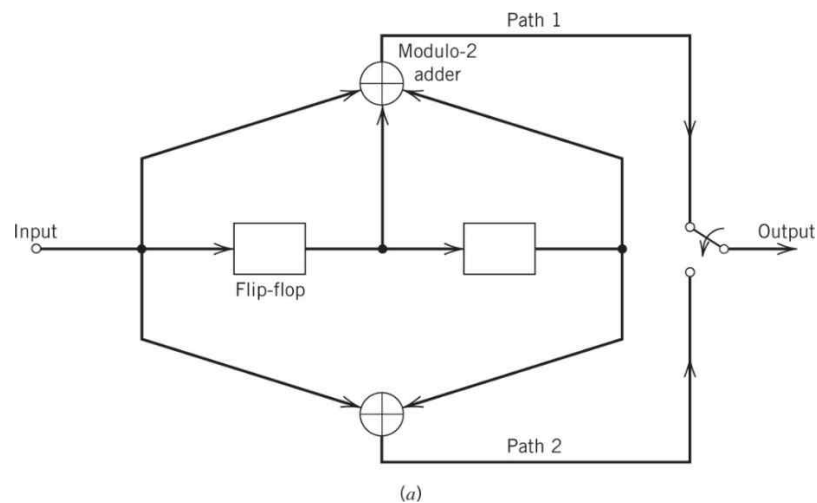


Figure 26 (a) $K = 3, n = 2, \text{rate-}1/2$ convolutional encoder

- L bit 의 message 는 길이 $n(L + M)$ bit 의 출력 sequence 를 생성한다
 - message 의 첫 bit 가 들어갈 때에 M -stage 의 shift register 에 있는 값 0으로 가정하고 출력
 - 입력 bit 이 이동할 때에 하나의 stage 만큼 이동
- 그러므로 code rate 은

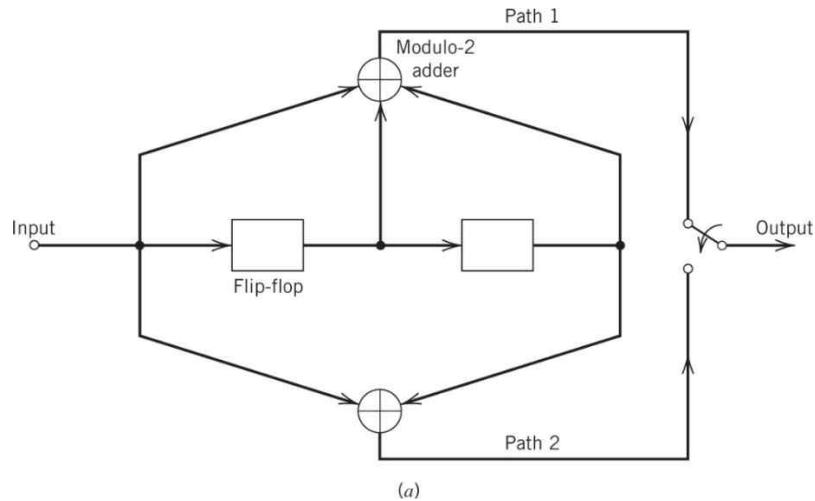
$$r = \frac{\text{input bit length}}{\text{output bit length}} = \frac{L}{n(L + M)}$$

- 이식에서 입력 message 가 연속적으로 계속해서 들어오므로 $L \gg M$ 라고 놓을 수 있다

$$r = \frac{L}{n(L + M)} \approx \frac{L}{nL} = \frac{1}{n}$$

- constraint length K 는 하나의 입력 bit 가 encoder 의 입력에 들어가서 모든 stage 를 거쳐 빠져나오기 직전까지의 shift 횟수를 의미

- M stage의 shift register가 있는 경우에 $K = M + 1$ 이 된다
- 그림 10.26a에서 $M = 2$ 개의 shift register가 있으며, constraint length $K = 3$ 이 된다. modulo-2 adder의 개수는 2개이어서 code rate은 $r = \frac{1}{n} = \frac{1}{2}$ 이 된다.
- convolutional encoder의 다항식 표현



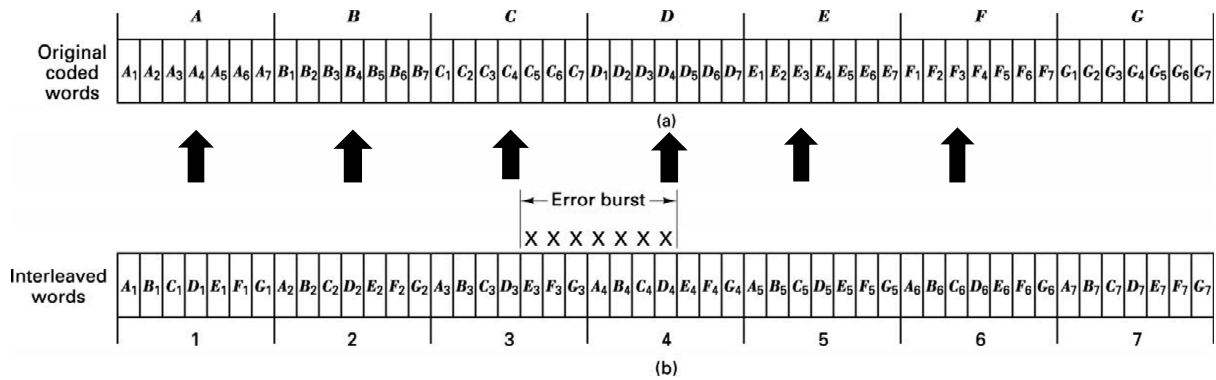
- 출력단자까지의 경로를 다항식으로 표현할 수 있다.
- 그 다항식을 generator polynomial이라 부른다
- convolutional encoder는 modulo-2 adder 개수만큼의 다항식으로 표시된다.
- i 번째 modulo-2 adder에 대한 다항식은 다음과 같이 쓸 수 있다

$$g^{(i)}(D) = g_0^{(i)} + g_1^{(i)}D + g_2^{(i)}D^2 + \dots + g_M^{(i)}D^M$$

- 계수 $g_0^{(i)}$ 은 i 번째 modulo-2 adder가 첫 번째 shift register의 왼쪽에 연결되어 있으면 1, 아니면 0.
- 계수 $g_1^{(i)}$ 은 i 번째 modulo-2 adder에 첫 번째 shift register의 오른쪽에 연결되어 있으면 1, 아니면 0.
- 계수 $g_2^{(i)}$ 은 i 번째 modulo-2 adder에 두 번째 shift register의 오른쪽에 연결되어 있으면 1, 아니면 0
- 계수 $g_M^{(i)}$ 은 i 번째 modulo-2 adder에 M 번째 shift register의 오른쪽에 연결되어 있으면 1, 아니면 0.
- 여기서 D 는 unit-delay 변수를 의미하며 $+$ 부호는 modulo-2 addition을 의미.
- D^j 은 j 개의 shift register를 지나왔다는 의미.
- 여기서 $(g_0^{(i)}, g_1^{(i)}, g_2^{(i)}D^2, \dots, g_M^{(i)})$ 를 generator sequence라 한다

interleave (level 2)

- bit sequence 가 전송될 때에 bit 에러는 연속된 bit 들에 걸쳐 일어나는 경우가 많다. 이렇게 뭉쳐 있는 bit error 는 수신단에서 복원을 어렵게 만든다. 이를 해결하기 위해 전송하기 전에 bit sequence 에서 bit 의 순서를 바꾸어 이 새로운 sequence 에 설사 연속된 bit error 가 일어날 지라도 sequence 를 원상태로 바꾸어 놓았을 경우 bit error 가 연이어 일어나지 않게 만드는 기법
- 아래의 그림에서 (a)의 codeword 를 interleave 하여 (b)를 만들었다. interleaved words 에 burst error 가 발생하였다면 실제로 이는 (a)의 codeword 에서 $A_4, B_4, C_4, D_4, E_3, F_3$ 의 거리가 떨어진 bit error 들이 발생한 것으로 보여 error correction 이 쉽게 된다.



BPSK (binary phase-shift keying) (level 3)

- 송신단에서 보내고자 하는 0 과 1 의 bit 를 정현파의 위상값에 대응시켜 전송하는 방식

$$s_0(t) = A \cos(2\pi f t + \phi)$$

- 고정된 진폭과 고정된 주파수의 정현파가 서로 다른 위상을 가짐으로 1 과 0 을 나타냄
- symbol 0 의 경우 $A \cos(2\pi f_0 t + \pi)$, symbol 1 의 경우 $A \cos(2\pi f_0 t)$ 을 사용

$$s_1(t) = A \cos(2\pi f_0 t) \quad \text{for 1}$$

$$s_0(t) = A \cos(2\pi f_0 t + \pi) = -A \cos(2\pi f_0 t) \quad \text{for 0}$$

- BPSK 에서 $s_0(t) = -s_1(t)$ 이 성립

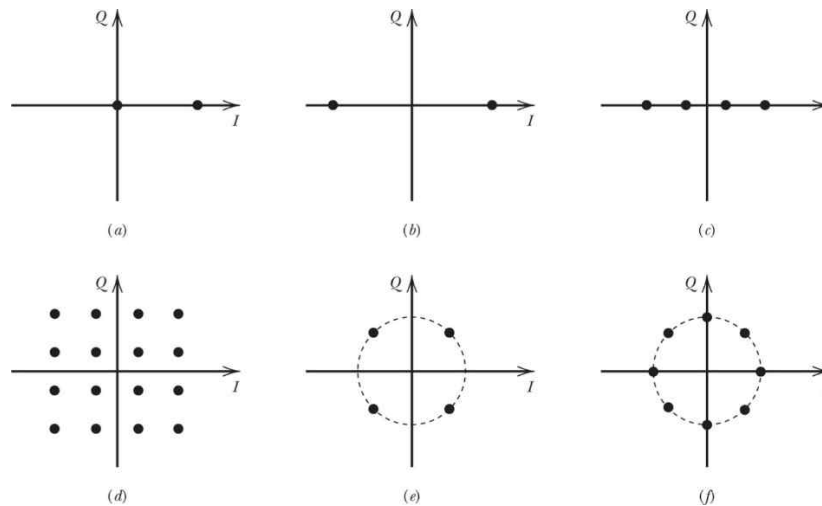


Figure 9.15 Representation of different band-pass modulations in signal space: (a) BASK; (b) BPSK; (c) M-ASK ($M = 4$); (d) M-QAM ($M = 16$); (e) 4-QAM and QPSK; and (f) M-PSK ($M = 8$)

PREAMBLE (level 3)

보내는 패킷의 맨 앞에 추가되며 수신단에게 패킷의 시작을 알려준다

OFDM (Orthogonal Frequency Division Multiplexing) (level 4)

- 지금까지 대역통과 채널은 선형이고 신호를 왜곡시키지 않는다고 가정
 - 선형 채널: 신호와 잡음이 단순한 덧셈으로 합쳐지는 채널
 - 무왜곡(undistorted) 채널: 채널의 진폭응답(amplitude response)이 모든 주파수에 대해 편평(flat)하다. 즉, 모든 주파수에 대해 배율이 똑 같다.
- 그림 9.21 은 무선랜 채널에서의 진폭 응답을 보여준다
 - 채널의 중심주파수를 중심으로 보여줌
- 무선랜은 2.4 GHz 대역에서 최소 20 MHz 의 대역을 사용. 그림 9.21 에서 20 MHz 대역으로 잘라 놓고 보았을 때에 편평하지 않음.
- 하지만 300 kHz 의 대역으로 잘라 놓고 보았을 때에 편평하다고 볼 수 있음

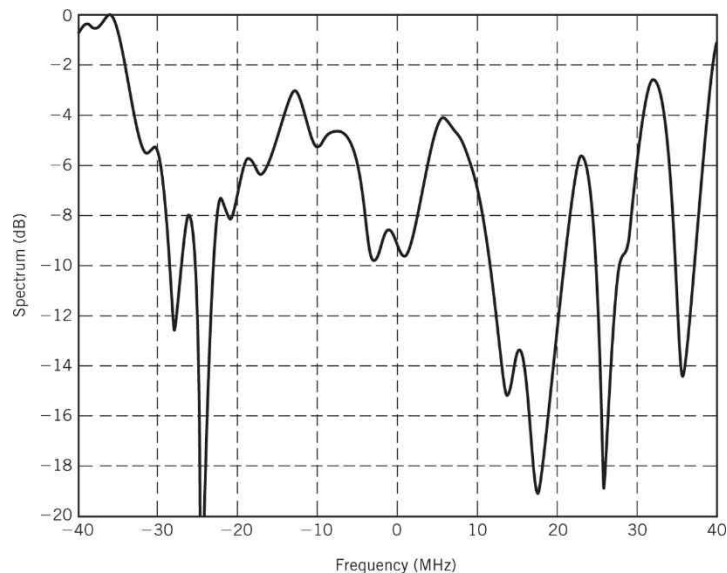


Figure 9.21 An example of amplitude response spectrum of wireless LAN channel

- 다중반송파 변조(multicarrier modulation) 기법은 동시에 여러 개의 반송파 신호를 사용하는 방식을 의미
- 그림 9.22a 는 다중반송파를, 9.22b 는 그 다중반송파에 보내고자 하는 신호를 실은 스펙트럼을 보여준다
- 9.22a 에서 개별 반송파를 f_i 에 위치하는 부반송파(subcarrier)라 부른다
- 다중반송파 전체의 중심주파수를 f_c 라 놓자. ($f_c = \frac{f_0 + f_{47}}{2}$)

- 변조된 부반송파의 대역폭이 300 kHz 이하인 경우에 채널의 왜곡이 없다고 간주할 수 있다. 왜냐하면 채널의 진폭응답이 편평하다고 볼 수 있기 때문이다.
- 그림 9.22의 예에서는 48개의 부반송파가 존재. f_0, \dots, f_{47}

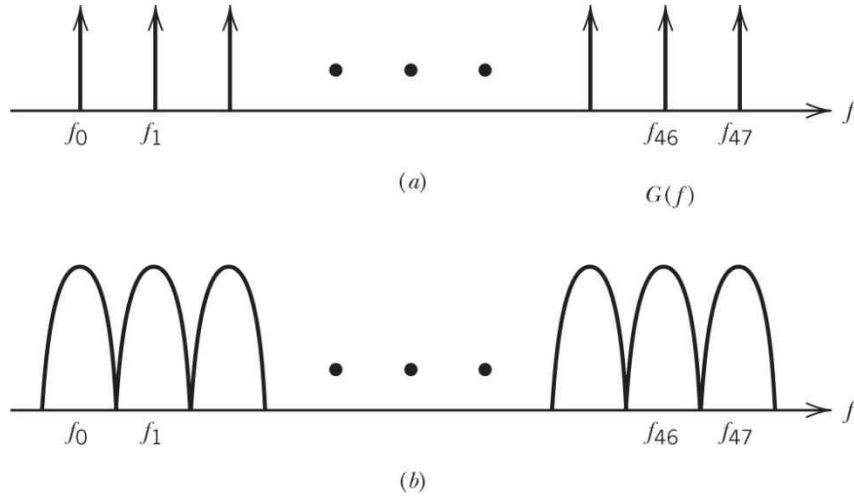


Figure 9.22 Conceptual illustration of complex baseband subcarriers; (a) unmodulated, and (b) modulated

- 부반송파를 수식으로 표현하면

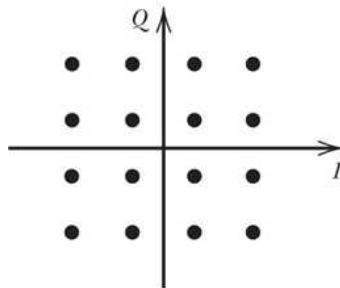
$$\tilde{c}_n(t) = e^{j2\pi f_n t}, \quad n = 0, 1, \dots, 47$$

- n 번째 부반송파의 복소수 envelope는 (이것이 메시지를 포함하는 신호임)

$$\tilde{g}_n(t) = b_{k,n} p(t - kT), \quad (k-1)T \leq t < kT$$

- $p(t)$ 는 구형파, T 는 심볼 간의 거리 (심볼 구간)
- $b_{k,n}$ 는 constellation에서 선택된 위치에 의해 정해지는 복소수. 예를 들어 16-QAM(quadrature amplitude modulation)의 경우 다음 16개의 값 중에서 선택된다.

$$\{\pm 1 \pm j, \pm 3 \pm j, \pm 1 \pm 3j, \pm 3 \pm 3j\}$$



- OFDM 에서 incoming 데이터의 속도가 48 Mbps 이고, 48 개의 부반송파와 $b_{k,n}$ 를 위해서 16-QAM 을 사용한다고 가정하자.
- 전송해야 할 0 과 1 의 데이터 스트림은 16-QAM 을 통해서 4 개의 bit 가 16 개의 심볼 $\{\pm 1 \pm j, \pm 3 \pm j, \pm 1 \pm 3j, \pm 3 \pm 3j\}$ 중에 하나로 바뀐다
 - 심볼 rate 은 $\frac{48 \text{ Mbps}}{\log_2 16} = 12 \text{ M symbols/s}$
- 이 심볼은 demux 를 통해서 48 개의 가지로 나뉘게 된다. 그 나뉘어진 심볼을 $\{b_{k,n}\}, n = 0, \dots, 47$ 이라 하자
 - demux 다음의 가지에서의 심볼 rate 은 $\frac{12 \times 10^6 \text{ sym bols/s}}{48} = 250 \text{ M symbols/s}$

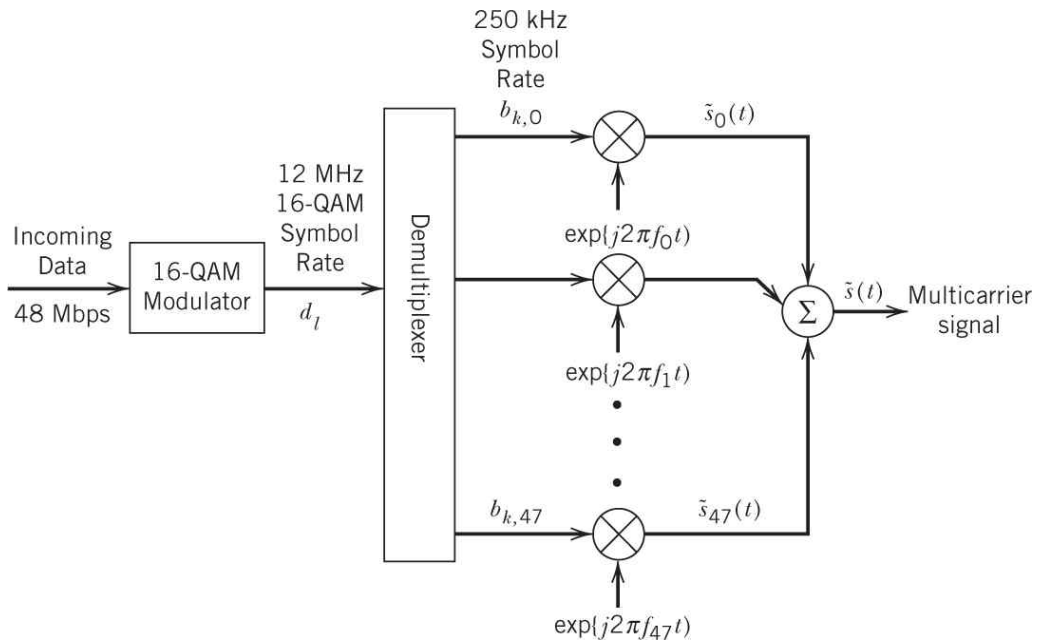


Figure 9.23 Conceptual OFDM modulation process

- 심볼 $b_{k,n}$ 에 구형파 $p(t - kT)$ 를 곱하고 주파수 변조를 위해 $e^{j2\pi f_n t}$ 를 곱하면 결과는 다음과 같다

$$\tilde{s}_n(t) = b_{k,n} p(t - kT) e^{j2\pi f_n t}, \quad (k-1)T \leq t < kT, n = 0, 1, \dots, 47 \quad (9.54)$$

- 매 심볼 구간마다 48 개의 $\tilde{s}_n(t)$ 신호를 모두 더한다. 즉

$$\tilde{s}(t) = \sum_{n=0}^{47} \tilde{s}_n(t) \quad (9.55)$$

- 마지막으로 $\tilde{s}(t)$ 의 중심 주파수를 f_c 로 옮겨 전송. 즉,

$$s(t) = \tilde{s}(t)e^{j2\pi f_c t} (= \sum_{n=0}^{47} \tilde{s}_n(t) e^{j2\pi f_c t} = \sum_{n=0}^{47} b_{k,n} p(t - kT) e^{j2\pi f_n t} e^{j2\pi f_c t})$$

- (9.54)로부터 심볼 $b_{k,n}$ 은 주파수 f_n 를 가지는 신호의 크기임을 알 수 있다. 즉, 심볼 $b_{k,n}$ 은 frequency-domain 에서의 주파수 성분의 크기이다.
- 이것을 수신단으로 전송하기 위해서 (9.54)을 time-domain 으로 표현하여야 한다.
- inverse discrete Fourier transform(IDFT)을 이용해서 (9.54)을 time-domain 으로 표현 가능

$$\begin{aligned} \text{DFT: } b_n &= \sum_{m=0}^{M-1} B_m e^{-\frac{j2\pi m n}{M}}, \quad n = 0, 1, \dots, M-1 \\ \text{IDFT: } B_m &= \frac{1}{M} \sum_{n=0}^{M-1} b_n e^{\frac{j2\pi m n}{M}}, \quad m = 0, 1, \dots, M-1 \end{aligned} \quad (9.57)$$

- $\{b_n\}, \{B_m\}$ 은 각각 frequency-domain sample, time-domain sample 들이다
- IDFT 를 적용하기 위해 다음의 가정이 필요
 - $p(t)$ 는 구형파. 즉,

$$p(t) = \begin{cases} 1, & 0 \leq t < T \\ 0, & \text{elsewhere} \end{cases}$$

- 부반송파의 주파수는 아래와 같도록 선택 (부반송파의 개수가 48 인 경우)

$$f_n = \frac{n}{T}, \quad n = 0, 1, \dots, 47$$

- 각 부반송파의 출력을 심볼 구간마다 M 번씩 샘플링한다. 이는 다음을 의미

$$t = \frac{m}{M} T, \quad m = 0, 1, \dots, M-1$$

- 위의 세가지 가정을 (9.54)와 (9.55)에 적용하면

$$\begin{aligned} \tilde{s}(t) &= \sum_{n=0}^{47} \tilde{s}_n(t) = \sum_{n=0}^{47} b_{k,n} p(t - kT) e^{j2\pi f_n t} \\ \Rightarrow \tilde{s}\left(\frac{m}{M} T\right) &= \sum_{n=0}^{47} b_{k,n} e^{j2\pi \frac{n}{T} \frac{m}{M} T} = \sum_{n=0}^{47} b_{k,n} e^{j2\pi \frac{nm}{M}} \quad (9.58) \end{aligned}$$

- 여기서 m 은 시간 상에서 심볼 당 샘플링 개수, n 은 부반송파 개수
- $p(t) = 1$ 는 $0 \leq t < T$ 구간에서 1 이라 볼 수 있다

- (9.58)을 (9.57)의 IDFT를 이용해 구현하고자 한다. 이를 위해 $M = 48$ 이어야 한다. (sum의 upper limit 으로부터)
- M 이 2의 거듭제곱인 경우, DFT는 FFT로 쉽게 구현된다
- 그래서 실제 구현에서는 IFFT를 적용할 때 M 이 2의 거듭제곱이 아닌 경우 0의 주파수 성분을 추가하여 인위적으로 거듭제곱으로 만든다.
- 그림 9.24a는 OFDM 송신기와 수신기의 블록 다이어그램을 보여준다
- 송신기에서 IFFT를 적용할 때에 16개의 빈 주파수 성분을 추가한 것 유의
 - 여기에 수신부와 동기화를 위한 부반송파 또는 보호대역을 위해 zero carrier를 포함시킨다

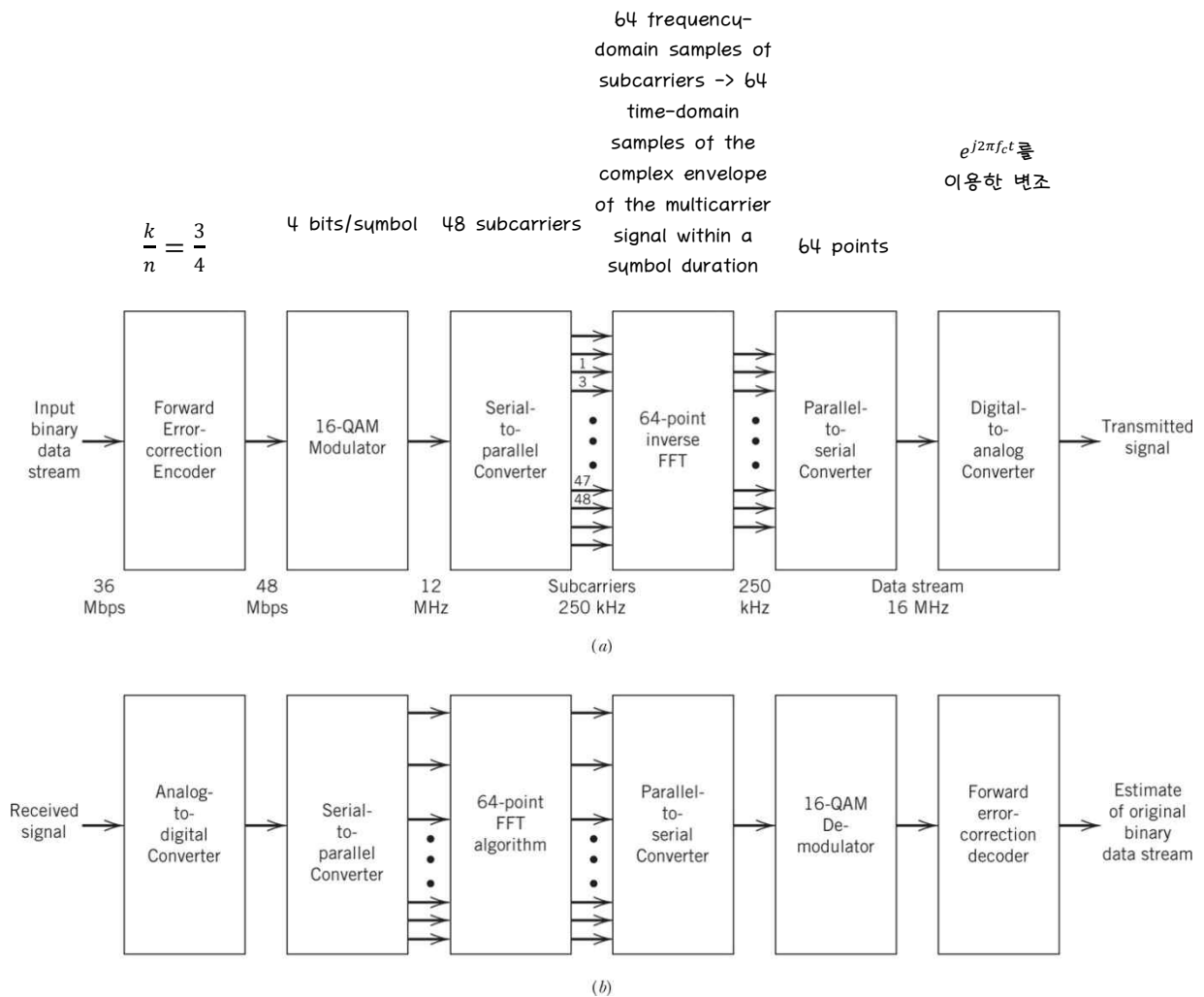


Figure 9.24 Block diagram of (a) OFDM transmitter, and (b) OFDM receiver