

데이터 통신 강의 정리노트

- 본 수업 교재: fifth-edition-data-communications-and-networking

2020124036

CH.10 Error Detection & Correction

data는 전송했을 때와 동일한 내용으로 보존됐다는, guarantee가 있어야 한다.
link L에선 보통 err 탐지후 프레임을 버리고, 상위 L에서 재전송을 요청하나, link L에서 직접 고쳐 올려보내기도 한다.

10.1 introduction

- 10.1.1 Types of Err.
interference로 인해 bits flow는 전송 중에 바뀌게 된다.
sig.의 모양을 바꾸게 되는 것이다.
이때,

1. single-bit err.:

한 data 단위안에서 only 1 bit가 바뀐다.

2. burst err.:

한 data 단위안에서 2 bits이상이 바뀌게 된다.

burst err.가 single보다 발생할 확률이 더 높은데, 1bit의 duration보다, noise sig.의 duration이 일반적으로 더 길기 때문이다. noise는 대부분 여러 bits에 걸쳐 영향을 준다. 영향을 받는 bit수는 data rate와 duration noise에 달려있다.

- 10.1.2 Redundancy
detecting과 correcting은 redundancy를 이용한다. redundancy는 여분의 비트를 추가로 붙이는 것이다.
- 10.1.3 Detecting v.s. Correction
err. detection은 err.가 발생했는지, 여부만 확인하기 때문에 corr.보단 쉽다.
몇 비트가, 또 어디서 err.가 발생했는지는 신경쓰지 않는다. (=single-bit나 burst err.나 같다.) 반면에, err. corr.은 어디서, 몇개의 bit가 err.가 났는지 알아야 한다.
- 10.1.4 Coding
redundancy는 다양한 coding schemes에 의해 만들어진다. sender가 실제 보낼 내용의 비트와 관계된(함수관계) redundant bits들을 붙여 보낸다.
receiver는 이 함수관계에 맞춰 받은 비트열에 err.가 있는지, 검사한다.

코딩 schemes는 block coding과 convolution coding으로 나눌 수 있다.

10.2 Block Coding

block coding에서는 message를 k개의 bit로 이루어진, block들로 나눈다.

각 block들은 **datawords**라고 부르는 단위이다.

이 k 개 bits에 r 개의 redundancy를 더해 $n=r+k$ bit의 block을 만든다.

redundancy가 추가된 block을 **codewords**라 부른다.

이때, k bit의 datawords였으므로, 2^k 개의 datawords를 만들 수 있다.

codewords는 n bit였으므로, 2^n 개의 codewords를 만들 수 있다.

즉, 실제 2^n 개의 codewords 중, 사용되는 codewords는 (datawords와 매칭, 또는 대응되는) 2^k 개 뿐이다.

$2^n - 2^k$ 개는 버려진다. (= Invalid하다.)

- 10.2.1 Error detection

block coding으로 error detecting하는 방법. 2가지 조건을 만족하면 된다.

1. the receiver has a list of valid codewords.

2. the origin. codeword has changed to an invalid one.

receiver가 valid를 받으면 dataword를 extract하고. corrupted를 받으면(inv.), 버린다.

다만, corrupted의 결과로 valid가 된다면 인지하지 못하고 extract하게 된다.

Hamming Distance

one of the central concepts in coding error control은 해밍거리이다.

해밍거리는 2 words간 대응하는 bit중 서로 다른 갯수이다.

$d(x_{word}, y_{word})$ 로 나타낸다.

해밍거리는 전송중 corrupted된 비트 수이다.

- 2 word 사이에 XOR을 해서 1이 나온 bit수를 세면 된다.

Minimum Hamming Distance for Error Detection

모든 가능한 pairs of codewords 중에서 가장 작은 해밍 거리이다.

err를 최대 s 개 까지 감지하고 싶을 때, sys.에서 min. 해밍거리는 $s+1$ 이 되어야 한다.

그래야 s 개가 바뀌어도 valid한 쌍 끼리는 최소 $s+1$ 개가 달라 err. bits undetected되는 불상사가 없다.

$$d_{min} = s + 1$$

— Minimum Distance for Linear Block Codes

Linear Block Codes block code의 부분집합. non linear은 구조가 분석하고 완성하기 힘들다.

정의는 'abstract algebra'가 필요하다. 따라서 우리는 단순히 linear block code는 2 valid codeword를 xor(=eor)했을 때,

또다른 valid codeword가 되는 것을 말한다고 하자.

linear block code면 min. 해밍거리를 찾는 게 수월하다.

non-zero인 codeword중에서, 가장 1이 적은 codeword를 고르고, 그 1의 갯수가 min.해밍거리이다.

1. Parity-Check Code

가장 익숙한 err.detecting code이다. linear block code이다. k-bit의 dataword => n-bit의 codeword (where $n=k+1$)

parity bit라 불리는 extra bit가 codeword의 1의 갯수를 even(짝수)로 만든다. (odd case도 있다.)

1 bit err.detecting이라 min.해밍거리는 2이다.

- 발측에서는 generator로 $r_0 = \text{parity bit} \oplus \text{dataword}$ 를 연산해 붙여준다.(실제로는 modulo-2)
 - 1갯수가 even이면 parity는 0, odd면 parity는 1.
- 수신측에서는 checker로 $s_0 = \text{syndrome} \oplus \text{dataword}$ 를 xor해서 붙여준다. (1의 갯수가 even이면 $s=0$)

10.3 Cyclic Codes

special한 linear block codes with one extra property.

codeword가 rotated(cyclically shifted)하면, 그 결과도 another codeword.

left-shift(rotating)할때, 첫 비트는 제일 끝으로 간다.(wrapped around)

- 10.3.1 Cyclic Redundancy Check

theoretical background가 이 책을 넘어서므로, 따라서 자세한 건 x. 아무튼 만들 수 있다한다.

Cyclic redundancy check (CRC)라는 걸 배운다. N/W, 특히 LAN이나 WAN에서 쓴다.

- encoder:

- the process of modulo-2 binary division:

십진수 나눗셈과 같다. 다만, 덧셈과 뺄셈을 모두 XOR로 하기 때문에, 두 연산이 모두 같은 결과를 낸다.

십진수 나눗셈처럼, 매 step마다 copy of divisor을 피연산자의 4 bits에 XOR한다.

remainder은 '몫-1' bit이고, 다음 step에서 이 '몫-1' bit에 피연산자로부터 1 bit를 내려붙여 '몫' bit를 만든다.

- LSB가 0이면 0000...(all 0's)로 나눠줘야 한다.
- 더이상 내려붙일 비트가 없을 때, 그만두고 remainder를 $n-k$ bits에 붙여 n bit dataword를 완성한다.
- decoder:
encoder랑 같은 division을 한다. remainder는 syndrome이라 부르며, 모두 0이면 높은 신뢰도로 no err.이다.
no err.면 $n-k$ 의 dataword를 추출해 위 L로 올려보낸다.
- 10.3.2 Polynomials
cyclic code를 이해하고, 분석하는데 좋은 방식은 polynomial 형태로 보는 것이다.
 - pattern of 0s and 1s를 coefficients of 0 & 1의 polynomials로 본다.
 - 제곱수는 bit의 자리를 나타낸다.
 - degree of polynomial은 highest power in the polynomial이다.
adding and subtracting에 있어서는, modulo-2를 쓴다. 즉, 같은 power 항에 대해서 $1+1=0$ 이면 해당 power 항은 지워진다.
($1+1+1=1$ 이면 1이겠죠?)
곱셈/나눗셈은 십진수에서 지수끼리 더하고 빼는 것과 동일하다.
 - shifting:
 m bit를 옮긴다 했을 때, x^m 를 각 항에 곱하거나(left shift), 나눈다(right shift).
- 10.3.3 Cyclic Code Encoder using Polynomials
polynomial을 이용한 계산을 알아보자.
훨씬 간단하다.
똑같이 dividend의 highest degree를 가진 항을 ($n-k$ 개의 0's가 붙은) divisor의 highest degree 항으로 나눈다.
나누기 위해 divisor에 곱해야하는 항이 첫 step에서 결정되는 몇의 첫 항이 된다.
divisor에 곱하면 빼고 나머지는 내려보내 다시 두번째 step을 진행한다.
 - divisor은 cyclic code에서 generator ($= t(x)$)라고도 부른다.
- 10.3.4 Cyclic Code Analysis
Dataword: $d(x)$ Codeword: $c(x)$ Generator: $g(x)$ Syndrome: $s(x)$ Error: $e(x)$
일 때,
 $s(x) \neq 0$ 이면, err.(corrupted)이고, $s(x) = 0$ 이면 err.x거나, 에러탐지실패이다.
이때, $g(x)$ (devisor)를 정하는 기준이 뭘까?

우선,

$$\text{Received codeword} = c(x) + e(x)$$

이고,

$$\frac{\text{Received codeword}}{g(x)} = \frac{c(x)}{g(x)} + \frac{e(x)}{g(x)}$$

이때,

두번째 항은 syndrome이다. 따라서 syndrome이 0이면 err가 없거나, $e(x)$ 가 $g(x)$ 로 divisible해서 err탐지에 실패한 것이다.

- $g(X)$ 로 divisible한 $e(x)$ 를 탐지하기위해 $g(x)$ 를 애초에 잘 설정해야 한다.

the size of the dataword is augmented by adding $n-k$ 0s to the right-hand side fo the word.

그 결과 n 비트는 generator에 들어간 뒤, divisor를 이용해 (size: $n-k+1$ => 나머지는 뭉보다 작아야 한다.) modulo-2 division을 한다.

몫은 버린다. remainder ($r_2r_1r_0$)만 dataword에 붙여 codeword를 만드는 데 쓴다. 여기까지가 sender의 encoder고, receiver의 decoder는 codeword를 받은 후, 전체 n 비트를 checker로 카피해 가져가서, divisor로 나눈다.

여기서 remainder는 syndrome으로, ($s_2s_1s_0$), $n-k$ bit이다. 신드롬은 analyzer가 읽어서 모든 syndrome bits가 0이면 codeword 왼쪽에서부터 k -bit를 통과시킨다.

single bit err. : 1bit err.는 x^i 이고, $x^k + 1$ 에 의해 나눠지지 않는다. 따라서, If the generator has more than one term and the coefficient of x^0 is 1, all single-bit errors can be caught.

Two isolated single bit err. : $e(x) = x^i + x^j$ 이고, $e(x) = x^i(x^{j-i} + 1)$ 이므로,

If a generator cannot divide $x^t + 1$ (t between 0 and $n - 1$), then all isolated double errors can be detected. - $t = 0$ is meaningless and $t = 1$ is needed, as we will see later. This means t should be between 2 and $n - 1$.

Odd number of err. : A generator that contains a factor of $x + 1$ can detect all odd-numbered errors.

다만, $x+1$ 만 있으면, 연속해서 오는 자리들을 가진 (예로, $x^i + x^{i-1}$) codeword의 err.를 detection할 수 없다.

burst err. : $e(x) = (x^j + \dots + x^i)$ 꼴이다.

L 은 burst err.의 길이이고, $g(x)$ 의 highest degree항은 x^r 일 때,

- All burst errors with $L \leq r$ will be detected.
- All burst errors with $L > r$ will be detected with probability $1 - (1/2)^{r-1}$.
- All burst errors with $L > r + 1$ will be detected with probability $1 - (1/2)^r$.

Summary A good polynomial generator needs to have the following characteristics:

1. It should have at least two terms.
2. The coefficient of the term x^0 should be 1.

3. It should not divide $x^t + 1$, for t between 2 and n - 1.

4. It should have the factor x + 1.

- 몇몇 standard polynomials들은 유명한 protocol들에 쓰인다.

CRC-8 : ATM header, CRC-16: HDLC, CRC-32: LANs

- 10.3.5 Advantages of Cyclic Codes:

HW/SW에서 구현하기 쉽고, 특히 HW에서 구현할 경우 빠르다.

- 10.3.6 Other Cyclic Codes:

최근(- 교재가 나온 건 너무 옛날인데 맞나?)에 쓰이는 Reed-Solomon Code라는 게 있다.

암튼 자세한 건 수학내용이 현 학부생이 배우기엔 어려워 생략한다고 한다.

- 10.3.7 HW implementation:

HW를 조금만 쓰고도 Cyclic Code를 싸고 쉽게 구현할 수 있다. 또, Check bit나 Syndrome bit calculation 의 rate를 증가시킨다.

divisor :

1. divisor은 dividend의 일부분에 의해 반복적으로 XOR된다.

2. divisor은 n-k+1 bits를 가진다. 미리 결정되거나, all 0's이다.

보낼 dataword의 LMB(left most bit)에 의해 결정된다고 한다.

=> 해당 step의 a part of dividend의 LMB가 0이면 0000...을 뺐었다!

3. n-k bits만 필요할 것 같은데, n-k+1 bits인 이유는?

교재 원문에서는,

A close look shows that only n-k bits of the divisor are needed because the result of the operation is always 0,

no matter what the value of this bit. The reason is that the inputs to this XOR operation are either both 0s or both 1s.

In our previous example, only 3bits, not 4, are actually used in the XOR operation.

XOR 특성상, 서로 다른(1/0) bit값이 와야 결과가 1이다.

즉, a part of dividend의 LMB가 0이면 차피 divisor도 00..이고, 1이

면 divisor의 LMB도 그대로 1로 같것이고,

해서, 늘 divisor의 LMB가 0인 것과 마찬가지이다.

augmented Dataword :

나누기를 할 때, augmented Dataword의 position을 고정하면, divisor를 1bit씩 right shifting해야하고,

divisor를 고정하면, augmented Dataword를 left shifting해야 한다. **Remainder :**

register(single-bit storage devices)가 $n-k$ 개 필요하다.

최종 remainder를 구하기까지 step by step의 dividing process가 필요하다.

예시에서 divisor은 011 또는 all 0's이다.

각 step마다 밑 줄로 이어진다. (예시의 경우 3열의 register가 8행이나 존재한다.)

1. remainder가 0's라고 가정한다.
2. 매번 time click(arrival of 1bit from an augmented DW)마다 아래 step을 반복한다.
 - A. LMB를 보고 divisor를 결정한다.(0xx...를 할지, 000...를 할지)
 - B. 남은 remainder $n-k-1$ 자리와 augmented dataword로부터 오는 다음 bit(1+2로 총 3bit)를 divisor와 XOR한다.

결과로 next remainder로 한다. > 교재 p.276의 F10.12 설명:

가장 왼쪽 remainder 자리에는 LMB of the part of the dividend를 저장하는 register($=r2$)가 있다. 이 register는 다른 register($r1, r0$)와 일렬로 있는 상태이고, 각각 XOR을 할 때 divisor의 각 자리 (d_1, d_0)와 이어져 있는데, 이때, d_2 와는 끊어져 있으므로 d_2 는 항상 0이다.

매 step(divisor에 1/0을 곱해서, 해당 step에 참여할 dividend의 일부분과 xor할 때마다)에서 이렇게 divisor를 결정한다.

LMB of the part of the dividend가 1이면 011이 나오게 될것이고, 0이면 all 0's 가 될 것이다.

이때, step들 사이에 규칙이 있어 반복되는 부분이 있다.

따라서, simplification을 수행할 수 있다.

F.10.13과 같은 shifting과 memorizing을 하는 shift를 이용해 간단화 한 줄의 devices들로 설계할 수 있다.

10.4 CHECKSUM

any length의 message에도 적용할 수 있는, err. detecting technique이다.

Internet에서는, data LL보다 TP LL나 NW LL에서 더 많이 쓴다.

그래도 주제가 err. detection이라고 이번 CH에서 다룬다.

전통적으로.

1. source(sender쪽)에서, message는 m-bit units로 나뉜다.
2. generator는 추가로 mbit unit(called checksum)을 뒤에다 붙인다. (메시지 가운데 넣는 경우도 있다.)
3. destination에서는, checker가 새로운 checksum을 만드는데, 받은 mbits units들 & 받은 checksum을 가지고 만든다.
4. new checksum이 모두 0이면, message는 accepted된다.(모두 0아니면 버린다.)

mbits units로 이루어진 하나의 message마다 checksum을 붙인다.

- 10.4.1 Concept

예로, m bits unit 각각이 나타내는 수의 합을 redundancy로 보내는 것이다.
(7, 11, 12, 0, 6)이었으면, (7, 11, 12, 0, 6, 36)으로 보낸다.

One's Complement Addition

sum이 m bits를 넘으면 어떡해? => 1's complement를 쓴다.

방법은, 폭력적이게도

오른쪽에서부터 m bits를 짜르고, 남은 왼쪽의 비트들은 2^0 자리부터 맞추어서 서로 더한다.

- (7, 11, 12, 0, 6)이었으면, (7, 11, 12, 0, 6, 6)으로 보낸다.

Checksum

receiver가 편하게, the complement of the sum, the checksum을 보낸다.

앞에서 본 one's complement arithmetic은,

해당 bit열의 모든 bit 값을 뒤집으면 (0->1, 1->0) 만들 수 있다.

이때, all 0's의 positive zero와, all 1's의 negative zero가 있다.

corruption이 없다면, receiver 쪽에서는, 받은 m-bits의 값들의 합 + checksum = 15 일 것이다.

Internet Checksum

전통적으로 Internet은 16bit checksum을 쓴다. sender에서,

1. 메시지를 16bit words로 나눈다.
2. checksum word의 값을 0으로 initai.
3. checksum 포함해서 다 더한다. one's complement를 쓴다.
4. complemented sum은 checksum의 값이 된다.
5. checksum을 data에 넣어 보낸다.

Algorithm

1. sum=0
2. if (more words?):
 - A. sum=sum+New word
 - B. .
3. if (Left(sum) is nonzero?):
 - A. Sum= left(Sum) + Right(Sum)
 - B. .
4. Checksum= Complment(Sum)
5. Checksum=truncate (Checksum)

Performance

Checksum의 단점과, CRC의 장점으로 인해 많은 새로운 프로토콜들에서 checksum => CRC로 하는 중이다.

- 10.4.2 Other Approaches to the Checksum Checksum의 큰 문제:
'the traditional checksum is not weighted'

16bit word를 checksum으로 많이 사용하는데, 1st unit에서는 err.로 -1되고, 2nd unit에서는 err.로 +1되면, 하나의 message 입장에서 봤을 때는 값이 0(=+1-1)이므로, 그대로다. mbits unit들은 순서를 구분하지 않는다.

이를 해결하기 위해 Fletcher, 그리고 Adler라는 방법이 나왔다.(지금은 쓰나..?)

Fletcher 8-bit data item에 대해선 16bit checksum을 붙이고, 16-bit data item에 대해선 32bit checksum을 붙인다.

1. 8-bit data item:

8비트 unit으로 16비트 checksum을 만드는데, 계산은 modulo 256(2^8)로 한다. 중간결과들을 256으로 나누고 remainder는 기억하는 과정이 필요하다.

- 2^8 로 나누면 두개의 accumulators L과 R을 쓴다.
- L은 data items들을 단순히 더한다. (8bits checksum)
- R은 weight를 계산에 더한다. (8 bits checksum)

예를 들어, modulo 연산을 제외한다고 가정하자.

1. $R=L=0$
2. data가 있으므로(들어오므로)
3. $R=0+D_1$
4. $L=0+D_1$
5. data가 있으므로(들어오므로)
6. $R=D_1+D_2$
7. $L=D_1+D_1+D_2$
 $=2*D_1+D_2$
8. data가 있으므로(들어오므로)
9. $R=D_1+D_2+D_3$
10. $L=D_1+D_1+D_2+D_1+D_2+D_3$
 $=3*D_1+2*D_2+D_3$

즉, 블록들(혹은 item)들의 메시지 내 위치에 따라 가중치(곱해지는 상수, weight)의 크기가 달라진다.

checksum은 단순히 합뿐만 아니라, 가중치가 부가된 정보도 포함되게 된다.($L \times 256 + R$)

교수님께 질문도 드려봤다.

기본 checksum은 메세지의 어떤 두 부분이 교환되는 형태로 에러가 발생한다면 이를 감지하지 못합니다. 전체 sum은 동일할테니까요, fletcher checksum은 메세지를 구성하는 블록들의 순서를 checksum을 계산하는데에 고려하겠다는 것입니다. 교재 그림 10.18을 보면 R 계산에서 데이터블록 D_i 가 포함됩니다. 따라서 두 블록 D_i 와 D_j 가 R계산에 이용되는 순서가 달라진다면, 최종 checksum도 달라질 것입니다. 예를 들어서 mod 256을 제외하

고 생각하면, $R1=D0$, $L1=R1=D0$, $R2=R0+D1=D0+D1$, $L2 = L1+R2=D0+D0+D1 = 2*D0 + D1$ 이런 식으로 $D0$ 가 $D1$ 보다 더 많이 더해집니다. 더해지는 횟수를 가중치 weight라고 하면 블록의 전체 메시지 내의 위치에 따라서 weight가 달라진다고 볼 수 있습니다.

Adler Checksum

32 bit checksum이다.

어렵다(?). 생략하겠다....

Forward Error Correction

real time multimedia transmission에서는 reproducing으로 delay를 만들어내기 위해, retransmission은 적합하지 않다.

즉각적으로 reproduce 해야한다.

=> 'forward err. correction (FEC)' technique 들을 사용한다.

- 10.5.1 Using Hamming Distance err. detection에서는 $d_{min} = t + 1$ 이었지만, err. correction에서는 $d_{min} = 2t + 1$ 이어야 한다.
valid code를 중심으로해서, t 반지름안에 있는 invalid들은 중심의 valid로 correction한다.

- 10.5.2 Using XOR

Data를 N개의 chunk로 나눈다.

N개의 chunk를 모두 XOR해서 R을 만든다. 보낼때는 N개의 chunk에 R까지 같이 보낸다.

만일 P_i 가 손실되면,

XOR의 특성에 의해서,

$P_i = P_1 + P_2 + \dots + R + \dots + P_N$ 로 구할 수 있다.

자기 자신과의 XOR은 사라진다. 따라서 P_i 만 남게 되는 것이다.

- 10.5.3 Chunk Interleaving

하나의 chunk는 XOR로 복구가 가능했다.

한 패킷에서 2개 이상의 에러는?(burst err.)

- 한 패킷에 최대 하나의 에러만 발생하도록 하자.

- 보내야될 패킷을 순서대로 보내지 않고,

original packet들이 있으면, 보내는 패킷은 이 ori. packet의 일부분들을 모아 보낸다.

두번째 보낼 패킷은 ori. packet들의 다음 일부분들을 모아서 보낸다.

- 다 받고나면, 결국 각 패킷마다 하나씩의 chunk가 발생했다.

다만, 패킷을 바로바로 처리하지 못해 delay가 발생하는 단점이 있다.

- 10.5.4 Combining

hamming distance를 이용해 보낼 수도 있다.

원리는 위와 같다.

각 m 비트의 패킷이 t비트의 err를 복구할 수 있는 상황.

m row(m개의 패킷)을 column방향으로 interleaving을 통해 전송한다.

mxt의 에러가 발생한다해도, 결국 m개의 패킷에 나눠지는 것이기 때문에, ori packet(row)에 대해서는 1개만 발생. 따라서 복구가 가능하다.

- 10.5.5 Compounding

redundancy로 duplicate of each packet을 전송한다.

너무 redundancy가 크므로, 해상도(resolution)을 낮춰 보낸다.

- 아예 안보내는 것보다는 낫지 않겠는가, 하는 것이다.

CH.11 Data Link Control (DLC)

data LL은 2개의 sublayers로 나뉜다.

1. Data Link Control(DLC) - upper sub L
2. Multiple Access Control(MAC) - Lower sub L

여기서 DLC를 배울 것이다.

11.1 DLC SERVICES

DLC는 인접한 두 노드간의 communication 절차를 다룬다.

- dedicated link는, broadcast link는 상관 X

DLC funtions들은 framing, flow and err. control을 포함한다.

- 11.1.1 Framing

physic. L에서 Data transmission이란 bit를 소스에서 des.로 sig의 형태로 전송하는 것을 말한다.

sender와 receiver가 동일한 bit duration과 timing을 가지도록 bit sync.를 제공한다.

D.L에서는, 비트를 frame으로 pack해서 서로 구분한다.

일종의 포대에다가 단순히 글자를 넣음으로써 정보 한 단위를 다른 단위와 구분한다.

포대는 속도측정 기능에 사용된다.

추가로, 각 포대는 sender와 receiver의 addr.를 정의한다.

D.L에서는 source에서 des.로 보낼 때 sender의 addr와 des의 addr를 덧붙인다.

des.의 addr는 패킷을 어디로 보낼지 정의한다. src.의 addr는 recipient가 수신확인을 위해 사용한다.

한번에 모든 메시지를 하나의 패킷(프레임)으로 만들지 않는 이유는, 크면 클 수록 single bit err.가 치명적이기 때문이다.

Frame Size

frame의 boundaries가 필요 없다. size 자체가 delimiter로 사용된다. 예) ATM WAN -> cell이라 부르는 fixed size frame 사용.

- fixed-size framing:

- variable-size framing:

prevalent(널리 사용된다) in local-area networks.

- end of frame과 beginning of the next의 정의가 필요하다.

역사적으로 두개의 접근방식이 사용됐다.

- a character-oriented approach

- a bit-oriented approach

1. Character-Oriented Framing(or byte-oriented):

data(from upper level)는 8bit의 character, 예를 들어 ascii 같은 coding sys.를 사용 한다.

header는 src.와 des.의 addr.와, other control info.를 담는다. 그리고 trailer에는 error detection redundant bits (= 역시 8bits이다.)

이때, frame끼리 구분을 위해, 1byte의 flag를 시작과 끝에 붙인다. 프로토콜에 기반한 (따르는) flag는 시작과 끝을 나타낸다.

DL에서 text만 exchange될 때 주로 사용된다. flag는 text communication상에서 사용되지 않는 어떤 문자 하나를 사용하게 된다.

그러나, 요즘에는 info.나 graph, 혹은 오디오나 비디오를 보내기 때문에, 사용되지 않는 문자가 없다. 계속 옛 방식대로 하면

frame 중간에 해당 문자를 보고 frame이 끝났다고 판단하는 사고가 일어날 수 있다. 이 때문에, 'a byte stuffing strategy'를 도입했다.

'byte stuffing(or character stuffing)'에서는, flag로 사용되는 문자와 중복된 (pattern) 문자가 data 중간에 있다면, 특수한 byte(한 문자)를 중간에 덧붙여준다.

the escape character(ESC)라고 불리는데, 미리 bit pattern을 정의해 놓는다.(알고 있다.)

receiver가 ESC 문자를 맞닥뜨리면, 제거하면서 다음에 오는 flag나 ESC를 flag/ESC가 아닌, data의 character로 인지하게 된다.

- 다만, ESC가 text의 일부라면(제거할 extra character가 아니라면,) 또한 ESC를 extra로 붙여줘야 한다. 그렇지 않으면, ESC의 패턴이 데이터 안에 있어서 ESC 패턴의 문자를 제거한 뒤, 뒤를 온전히 사용하는 등, 문제가 발생하기 때문이다.

2. Bit-Oriented Framing:

frame의 bit들은 upper L에서 txt, graphic, video 등등으로 해석될 것이다.

그러나 delimiter를 위해 header가 붙는 (trailer도) frame끼리는 구분할 필요가 있고, 보통 대부분 8-bit의 특정 패턴을 가진 flag를 사용한다.

하지만 역시 character oriented protocol과 같은 문제가 있는데, flag pattern이 data에 있다면, frame의 끝이 아니라는 추가 정보를 제공해줘야 한다는 것이다.

'bit stuffing'은, 1 single bit를 flag처럼 보이는 pattern 뒤에 붙여주는 것이다. 예를

들어, flag가 01111110이었다면, 011111라는 패턴 뒤에는 항상 0을 붙여준 뒤, receiver쪽에서 제거하게 하는 것이다.

- 11.1.2 Flow and Error Control

CH9에서 flow and err. control에 대해 언급했었다. 여기서 배우게 된다.

DLC sublayer는 flow and err. control에 대해 맡고 있다.

Flow Control

production과 consumption사이의 비에는 balance가 있어야 한다.

production이 비교적 빠르거나, 느린 문제가 있는데, flow control은 비교적 빠른 문제를 다룬다.

DL의 communication에서는 4개의 entities가 있다고 한다.

NW의 sender/ receiver node, DL의 sender/ receiver node.

CH23에서 배우겠지만, 실제론 NW와 DL의 복잡한 관계가 있지만, 여기서는 DL의 두 노드 간의 관계에 대해서만 집중한다.

교재의 Figure 11.5에서는, sending node가 DL을 통해 receiving node로 frame을 push하고 있다.

receiving node가 받는 frame을 process할 수 없고, 도착한 rate에 맞게 상위 L(NW)로 보낼 수 없다면, overwhelmed.

Flow Control은 이때 feedback을 sending node로 보내서, stop or slow down pushing frames를 요청한다.

Buffers

flow control의 일반적인 방법. 두개의 버퍼를 이용한다.

sending DLL(node)과 receiving DLL(node)에 각각 한개씩 준다.

버퍼는 holding packet을 한다.

flow control은 consumer가 producer에게 sig.을 보냄으로써 시작한다.

- receiving DLL의 버퍼가 다 차면, sending DLL에게 그만 보내라고 신호를 보낸다.

Error Control

physic. L는 완전히 신뢰할 수 없기에, DLL에서 err. control을 해줘야 한다. (NW로 넘기기 전에)

보통 아래 두가지 중에 한 가지를 쓴다. 모두 CRC라는 걸 header에다가 더해주는 방법이다.(receiver에서는 check한다.)

1. 프레임이 망가지면, 그대로 버린다. 안 망가졌으면 그대로 올려보낸다.
wired LANs, such as Ethernet에서 쓴다.
2. 프레임이 망가지면, 그대로 버린다. 안 망가졌으면, ack를 sender에게 보낸다. (flow& err. control을 위함이다.)

Combination of Flow and Error Control

flow control을 위한 ack이 간단한 상황에선 err. control, 그러니까 frame이 봉고없이 잘 도착했다고 알려주는 용도로도 사용될 수 있다.
ack이 안 오면, frame에 문제가 생겼다고 해석하면 된다. (acknowledgment를 실은 frame은 ack라 부르며, data frame과 구분.)

- 11.1.3 Connectionless and Connection-Oriented
NW랑 Transport L를 배울 때 다시한번 자세히 배울 내용이다.

Connectionless protocol

frame은 노드 사이를 오갈 때, 어떠한 frame과의 관계가 없다. each frame은 independent하다.
connectionless는 medium의 부재(physic. connection)이 아닌, 프레임끼리의 관계이다.
프레임끼리는 not numbered고, 따라서 ordering을 sensing하지도 않는다.
대부분의 LANs의 DLL은 connectionless protocol이다.

Connection-Oriented Protocol

논리적 연결이 두 노드간 설정돼야 한다.(setup phase)
서로 연관된 (순서가 있는) frame들이 모두 전송된 뒤에는 (transfer phase)
논리적 연결이 제거된다.(teardown phase)
즉, frame끼리 넘버링되고, 순서대로 보내진다.
순서대로 안오면, 기다린다. 순서대로 한 세트를 모아다 NW로 올려보낸다.
rare in wired LANs이고, point-to-point protocols이나 some wireless LANs, and some WANs에서 쓴다.

11.2 Data-Link Layer Protocols

전통적으로, 4개의 프로토콜이 DLL에서 flow&err. control을 위해 존재했다.

1. simple
2. stop-and-wait
3. go-back-n
4. selective-repeat

1, 2는 DLL에서 아직 쓰지만, 뒤의 3, 4는 사라졌다(?)

상위 L에서 나온다는 것 같다. 자세한 건 wired/ wireless LANs를 알 필요가 있어, CH23에서 배울 것이다.

DLL의 동작을 a Finite State Machine (FSM)을 이용해 표현하면 편리하다.
유한개의 동작을 하는 (number of state) 기계로 생각한다는 뜻이다.
항상 event가 일어나기 전까진 하나의 상태를 유지하고 있다.
event 마다 두가지 반응을 정의한다.

1. actions to be performed
2. determining the next state

항상 처음 시작할 때의 state는 정의돼야 한다.

- state마다 rectangle(or circle)로 표현되고, movement를 화살표, 그리고 화살표마다 (event)/(actions) 형식으로 기입한다.
- colored arrow shows the starting state.

- 11.2.1 Simple Protocol

flow, err control 모두 안 한다.

receiver가 receive하자마자 frame을 handling한다고 가정한다.

sender는 NW L에서 Message(패킷)가 내려와야지 Frame을 만들어서 보낸다.

receiver는 그대로 받는다.

FSM에서는, request가 올때까지 sending node는 ready상태이고, request가 NW L로부터 내려오면 그때 보낸다.

- 11.2.2 Stop-and-Wait Protocol

err. & flow control 모두 한다.

primitive ver.을 배우고, 더 자세한 건 CH23에서 배운다.

(sliding window라는 걸 배우고 배우게 된다.)

sender는 한 번에 1frame을 보내고 wait한다.

CRC를 data frame마다 붙여 보내고, receiver가 checking한다.

붕괴되면 frame을 버리고, 아무런 응답을 보내지 않는다.

sender는 보낼 때마다 타이머를 작동하고, ack(응답)을 기다린다.

ack가 타이머가 expire하기 전에 오면, 다음 frame을 보내고,

반대로 ack도착 전에 타이머가 expire하면, lost or corrupted로 간주한다. 이때, sender는 frame의 copy를 ack가 올 때까지 가지고 있었다.

즉, ack가 오면 copy를 버리고, 다음 frame을 보내는 것이다.

- sender에게 corrupted ack가 올 수도 있다. 이 경우, 버린다.
- 문제점: 만약 receiver가 잘 받아서 ack를 보냈는데, 이 ack를 lost하면, sender는 한번 더 보내게 된다. receiver의 NW L에서는 2개의 패킷을 받게 될 것이다.
- 해결을 위해서는 패킷들의 시퀀스 넘버링, 그리고 ack또한 넘버링 돼야 할 것이다.

Sequence and Acknowledgment Numbers

sequence numbers are 0, 1, 0, 1, 0, 1, ...;

ack또한 마찬가지로 numbering이 1, 0, 1, 0, 1, 이 된다.

다만, ack의 넘버링은 항상 다음 받아야 할 frame의 넘버이므로, 1부터 시작한다.

- 11.2.3 Piggybacking

앞선 두개의 디자인은 unidirectional communication을 위해 만들어졌다.
communication의 효율성을 위해, 데이터를 보내면서, ack도 같이 보낸다. (piggybacked with the ack)
자세한 건 CH23에서 배울 것이다.

11.3 HDLC

High-level Data Link Control (HDLC)은 point-to-point 그리고 multipoint links를 위한, bit-oriented protocol이다. stop-and-wait을 보완한다.
이론적으로 (구현말구) issue가 있으나, PPP나, Ethernet protocol에 사용되는 기초적인 concept를 제공하므로 배운다.

- 11.3.1 Configurations and Transfer Modes

point-to-point, multipoint links 모두에서 사용된다.

1. normal response mode (NRM)

HDLC는 두가지 전송 모드가 있는데, primary station이라는 하나의 station은 명령을 보내고,
secondary station이라는 여러개의 station은 오직 응답만 할 수 있는, 비대칭적인 구조이다.
2. asynchronous balanced mode (ABM)

각 station들은 primary이자, secondary가 될 수 있으며 (acting as peers), link들은 point-to-point이다. 오늘날 많이 쓴다.

- 11.3.2 Framing

여러 옵션들을 지원하기 위해, HDLC는 3가지 frame의 type을 지정한다.

각 type의 frame들은 최대 6개의 필드를 가진다.

control field가 frame의 type과 기능을 정의한다.

- flag
- address
- control
- info.
- frame check sequence(FCS)

HDLC의 err. detection field이다. 2 or 4 byte CRC를 가진다.

user's data from the NW L 혹은, management info.를 가진다. nw 를 지날때마다 길이가 바뀐다.

1 or 2 bytes size로 flow and err. control에 사용된다.

secondary station(response하는 station)의 주소를 담는다.
secondary 쪽에서 frame을 만들때는, (response용 이므로) from의 주

소를 담게된다.

NW의 필요에 의해 해당 필드의 사이즈가 결정된다.

sync. pattern인 01111110으로 처음 flag와 끝 flag는 동일한 값을 가진다.

- ending flag

1. information frames(I-frames)

2. supervisory frames(S-frames)

3. unnumbered frames(U-frames)

sys. management를 지원한다.link 자체를 다루는 관리하는 경우가 많다.

연결을 establishment하고, 해제한다. 두 연결된 디바이스 사이에서,

exchange session(연결 설정 세션) management & control info.

S-frame과는 달리, info. field가 있지만, user data가 아닌, sys.

management info.를 위한 field이다.

그러나, S-frames과 같이, 대부분의 정보가 U-frames control field에 있는, code section에 들어간다.

- 다시 code는 2가지로 나뉘는데, P/F앞의 2비트의 prefix(접두) code부분과, P/F 뒤의 3비트 suffix(접미) 부분이다. > 합쳐서 5비트 => 32개의 U-frame 종류를 만든다.

1. node A는 a set asynchronous balanced mode (SABM) frame을 B에게 보내 연결을 요청한다.

2. B가 positive response를 unnumbered acknowledgment (UA) frame을 통해 준다.

3. 이 두가지 exchange가 일어나고, data transfer가 이루어질 수 있게 된다.

4. 다 보내고, A는 DISC (disconnect) frame을 보낸다.

5. B에서 받아들여졌다는 의미로 UA (unnumbered acknowledgment)를 보낸다.

only transport control info. piggybacking이 불가능하거나(딱히 보낼 data가 없을 때), 부적절한 상황에서 flow & err. control에 쓰인다. info. field가 없다.

control field의 첫 2비트가 10이면, S-frame이라는 얘기다.

마지막 3비트 (N(R))이 ack(acknowledgment number)과 같거나, nak(negative acknowledgment number)과 같다.

S-frame의 종류를 정의하기 위해 2비트의 code라고 불리는 부분이 있다.

(N(S)가 없다!) - 2비트 = 4개 종류의 S-frame이 존재한다. > 1. Receive ready(RR):

subfield의 값이 00이면, RR S-frame이다.

안전하게 받았는지 알려주는데, 이 경우, N(R)이 ack or nack와 같다. > 2.

Receive not ready(RNR):

subfield가 10이면, RNR S-frame이다.

RR 뿐만 아니라, 추가적인 기능을 수행하는 역할이다.

frame or a group of frame을 받았음을 알리고, receiver가 바빠서 더 이상 받을 수 없음을 알린다.

congestion-control mechanism; sender에게 slow down하라고 요청하는 기능을 한다.

N(R)의 값은 ack 가 된다.(acknowledgment number)

> 3. Reject(REJ):

subfield의 값이 01이면 REJ이다.

NAK 프레임이고, Selective Repeat ARQ에서 사용되는 NAK과는 다르다.

Go-Back-N ARQ에서 사용되는 NAK으로, sender의 timer가 expire되기 전, 알림으로써, 효율성을 높이는 데 사용된다.

N(R)은 NAK(negative acknowledgment number)이다. > 4. Selective reject(SREJ):

subfield의 값이 11이면, SREJ이다.

Selective Repeat ARQ에서 사용되는 NAK으로, HDLC protocol에서는 'Selective reject'라고 쓴다.

N(R)은 NAK이다.

DL user data와 이에 기반한 control info를 담는다.(piggy-backing)

piggybacking으로 flow & err. control info.도 포함하는데, 이는 control field 내 subfields에서 정의하는 기능이다.

- control field:

I frame은 NW L로부터 온 user data를 실어나르기 위해 만들어졌다.

첫 비트는 타입을 결정한다. control field의 첫 비트가 0이면, I frame이라는 뜻이다.

다음 3비트는 N(S)라 불리며, 해당 프레임의 시퀀스 넘버를 정의한다.

그 다음 P/F bit를 건너뛰고 3비트는 N(R)이라고 불리며, piggybacking을 쓸 때, ack의 넘버를 담고 있다.

P/F는 두 가지 목적으로 쓰는 1bit이다. 값이 1일 때는 poll(= primary가 secondary한테 보낼 때) or final(= secondary가 primary에게 보낼 때) - addr. field가 receiver의 addr인지(poll), sender의 addr인지(final)로 구분 한다.

piggy backing을 사용하는 예시 Figure 11.18 => piggy backing을 사용.

11.4 Point-To-Point Protocol (PPP)

point-to-point access로 가장 흔한 프로토콜 중 하나.

Internet service provider의 server와 개인 home computer 사이의 Internet이 PPP를 쓴다.(이것도 옛날얘기라고 교수님이 말하셨다.) 대다수가 traditional modem에서 왔다.

telephone line(= Physical L)을 통해 Internet으로 연결됐으나, control과 data 전송의 manage를 위해서, DLL에서의 protocol이 필요하다.
그 결과, PPP가 널리 사용됐다. => 요즘엔 조금 쓴다.

- 11.4.1 Services

PPP는 point-to-point에 맞춰 설계됐지만, 단순화를 위해 전통적인 서비스들을 제외하기도 했다.

Services Provided by PPP

PPP는 디바이스간 교환되는 프레임의 형식을 정의한다.
또, 어떻게 두 디바이스는 link를 확립하고, data를 교환할지도 정의한다.
PPP는 IP뿐만 아니라 NW L로부터 payloads를 받도록 설계됐다.

인증(=패스워드나, 로그인 시스템 등)까지는 옵션이라고 한다....

새로운 PPP는 Multilink PPP라고 불리며, multiple links를 지원한다.
놀랍게도, PPP가 NW addr. 구성(IP addr.)을 제공한다

home user가 Internet에 연결하기 위해 일시적으로 NW addr.가 필요할 때 유용하다.

<참고>

- ack number는 받은 시퀀스 넘버 그 자체일 수도 있지만, 일반적으로 그 다음 받을 것으로 예상되는 ack number를 쓴다.
 - go-back-n: 중간에 생략되면, 거기부터 다시.
 - selective reject/ repeat: 빠진 것만 다시.
-

Services Not Provided by PPP

flow control을 지원하지 않는다.

하나씩 보낸다면 receiver의 포화를 신경쓰지 x.

err. control에는 아주 간단한 mechanism을 가진다.

CRC field가 err. detection에 이용된다. err.가 있으면 프레임을 바로 버린다.

upper L에서 주의가 요구된다.

err. control과 sequence numbering의 부재는 패킷을 뒤죽박죽으로 만들 수 있다.

- 11.4.2 Framing

PPP는 character-oriented (or byte-oriented) frame을 사용한다.

- flag:
1byte의 01111110 pattern으로 frame 앞 뒤에 flag가 붙는다.
- addr.:
addr field는 상수로 1byte; all 1's (broadcast addr.)를 가진다.
- control:
HDLC의 unnumbered frames를 따라한 00000011이라는 상수 값을

가진다.

err. control은 detection에만 그치고, flow control을 제공 x.

- protocol:

data field에 무엇을 실을 건지 정의한다. (user data or other info.)

2 byte long이 default지만 1byte일 수도 있다.

- payload:

user data or other info, 어떤 것이든 실을 수 있다.

가변적인 사이즈를 가진다.(padding을 위한 용도의 field이다.) 보통 프레임은 1500bytes까지 고정된 기본 크기를 가진다.

하지만, 프레임이 전송되는 도중에 사이즈가 바뀌기도 하고, byte stuffed될 수 있기 때문에(flag pattern이 이 필드에 나타나면) '까지'로 maximum을 고정해놓는데,

그래도 기본 크기를 채우지 못하면, 이때 payload field에서 padding을 해준다.

- FCS:

2-byte or 4-byte standard CRC.

Byte stuffing

01111101이 escape byte로, 앞에서 배웠던 것과 마찬가지로 같은 비트패턴이 data field에 나오면, 또 그 앞에 이 esc byte를 붙여줘야 한다.

- 11.4.3 Transition Phases

transition phase에서, PPP는 dead state(no active carrier(physical L sig.), the line is quiet in Physic. L)부터 시작한다.

두 개의 노드가 통신을 시작하면, establish state가 되고, 두 노드간 옵션에 관한 협상이 진행된다(?).

그 후에, 만약 두 노드가 인증('password 입력'등의 작업)을 필요로 한다면(서로를 모른다면), 시스템에서 인증(확인)해준 뒤,

NW에서 NW L configuration(IP addr. 같은 필요한 구성)을 할당받는다.

link-control protocol 패킷들은 간단한 communication에 쓰인다. 여러 패킷들은 여기서 교환된다.

Data transfer은 open state에서 동작한다. connection이 이 state에 접어들면, data packet들의 교환이 시작된다.

하나의 endpoint에서 끊을 때까지 connection은 유지된다.

한쪽 endpoint에서 연결을 끊고자 할 때, sys.는 terminate state에 들어선다. 모든 carrier가 dropped될 때까지 terminate state를 유지하다가, 종래에 dead state로 돌아온다.

- 11.4.4 Multiplexing

PPP의 역할은 link를 확립하고, 파티들끼리 (통신을 하는 노드끼리) 인증(password)하고, NW data를 나르는 다른 프로토콜들을 사용하는 것이다. PPP packet은 data field 내에서 data를 아래 3개의 프로토콜(PPP의 기능을 위한 프로토콜들) 중 한개를 넣는다.

1. the Link Control Protocol (LCP)

2. two Authentication Protocols (APs)

user id가 필요한 dial-up links(전화선 이용한 인터넷)에서 사용되도록 PPP 가 만들어졌기 때문에, 중요한 부분이다.

Authentication이란, resource에 접근하기 위해 유저의 id를 확인한다는 뜻이다.

이를 위해 두가지 프로토콜을 만들었다. protocol value 0xC223로 시작하고, challenge, response, success, failure라는 4개 종류의 CHAP packet 종류가 있다.

1. PAP>Password Authentication Protocol):

- sys.에 접근하길 원하는 유저는 authentication identification(usually the user name)과 password를 sys에 전송한다.
- sys.는 id와 password를 검증하고, 승낙/ 거절한다. 0xC023라는 PPP frame안 protocol field값을 가진다. 그 다음 payload부분에 내용이 들어갈 것이다. 종류로는 유저가 요청하는 packet인 authen. req., 시스템이 보내는 ack & nak이 있다.

2. CHAP(Challenge handshake Authentication Protocol):

- sys.가 user에게 challenge packet을 보낸다. challenge value가 담겨있다.
- user는 그 값을 받아 미리 정해진 함수에 본인 password와 함께 대입한다. 그 결과를 response packet에 담아 sys.에 보낸다.
- sys.는 user의 password와 함께 아까 정해진 함수에 똑같이 ch. val.와 함께 대입해, 결과를 유저의 결과와 비교한다.

PAP보다 더 보안이 좋다. password를 온라인으로 보내지 않는다.

link를 설립하고, 유지하고, 규정하고(configuration), terminating하는 역할을 맡는다.

두 endpoint간의 옵션을 설정할 negotiation mechanism도 지원한다.

link가 설립되기 전에 두 endpoint는 이런 옵션들에 합의한다.

모든 LCP 패킷들은 PPP frame의 payload field안에 들어간다. (protocol field의 값이 0xC021일 때 이렇게 해석한다.) 다시 LCP내용에서도, Code, ID, Length, Info.가 들어가고, COde에 따라 packet type이 나뉜다.

패킷 타입은 req., ack., nak, reject, echo(살아 있는지 확인용) 등등이다.

이때, data들은 많은 다양한 NW L에서(프로토콜에서) 내려온다. 즉, Muxing 이다.

참고: 모든 frame에 대해선 ack를 보내는 건 아니지만, authentication에 대해선 항상 ack를 보내야 한다.

3. several Network Control Protocols (NCPs)

PPP는 multiple-network-protocol이다. 즉, IP 외에도 어떤 NW protocol을 사용할 수 있다.(각 프로토콜에 맞는 파라미터를 넣겠지.) 일반적으로 IP addr.를 쓰는 IPCP이고, 게다가 IP를 고정적으로 못써서 할당받아야 한다.

- IPCP(Internet Protocol Control Protocol):
IP packets in the Internet을 실어나르는 link에 대해 틀(configuration)을 잡는 프로토콜이다.
protocol field에서 0x8021값을 가지며, 마찬가지로 payload에 IPCP info.와 code, id, length정보를 가진다.
code는 configure- req., ack, nak, reject와 terminate- req., ack, 그리고 code-reject의 종류를 가진다.
물론, 다양한 NW L을 (OSI, IP, so on...) 사용하므로, 이때마다 각각 protocol field의 값은 다르다.

물론, 여기서 logical frame을 여러개의 실제 frame으로 나눴기 때문에,

Multilink PPP

PPP는 원래 single-channel point-to-point physic. L을 위해 개발됐다. 그러나 multiple channel을 이용한 방법이 발전했고, logical PPP frame은 여러 실제의 PPP frame으로 나뉜다. 그 후 여러 channel들로 전송한다.
multilink PPP임을 표기하기 위해, 실제 프레임들의 porotocol field는 0x003d로 설정한다.
frame마다 complexity가 필요하다.(특별한 표시를 해줘야 한다.) 예를 들어, sequence numbering이 필요하다.

CH.12 Media Access Control (MAC)

node or station들이 공통 link로 이어졌을 때, multipoint or broadcast link라 부른다. common link를 쓰므로, 서로 통신이 조율되도록 multiple-access protocol이 필요하다. 독점하거나, 서로 방해하거나, 동시에 말하지 못하게 하는 등, 다양한 규칙을 정의한다.

이런 기능을 수행하는 layer가, DLL의 sublayer 중 하나인 *media access control (MAC)*이다.

Multiple-access protocols

Random-access protocol	Controlled-access protocols	Channelization protocols
ALOHA	Reservation	FDMA
CSMA	Polling	TDMA

Multiple-access protocols

CSMA/CD	Token passing	CDMA
CSMA/CA		

12.1 Random Access

random access, or contention이라 부르는 방법에서는, 어떤 station도 다른 station보다 superior하지 않다.

(+control 할 수 없다.)

보낼 게 있는 station은 보낼지, 말지를 결정하는 절차를 실행하는데, medium의 idle/busy 상태에 따른다.

전송이 station에서 랜덤하게 이루어지고, 다음번에 어떤 station이 보낼지도 정해지지 않는다. (randomaccess 라 불리는 이유)

station끼리 medium을 사용하기 위해 경쟁해야 한다.(contention이라 불리는 이유)

경쟁하다보면, access conflict(collision)이 발생하기도 한다.frame은 사라지거나 수정된다.

conflict를 피하기 위해 각 station들은 다음 질문들에 답하는 절차를 따른다.

- 언제 station이 medium에 접근할 수 있는지
 - medium이 busy상태면 station이 뭘 할 수 있는지
 - 전송 성공/실패를 어떻게 station이 확인할 수 있는지
 - access conflict가 발생하면 station이 무엇을 해야할지
- 12.1.1 ALOHA

초기의 random access method. 하와이대에서 만들어졌다.

radio (wireless) LAN을 위해 만들어졌지만 shared medium에서 쓰인다.

Pure ALOHA

각 satation이 보낼 data가 있을 때마다 frame을 보낸다.

그러나, only 1 channel이기 때문에, (shared) 붕괴위험이 있다.

ALOHA protocol은 receiver로부터오는 ack에 달렸다.

station에서 frame을 보내면, receiver로부터 ack를 기대하는데,

ack가 오지 않으면(timeout period가 지났는데도), station은 frame (or ack)이 손실됐다 가정하고 재전송한다.

이때, 충돌한 station 모두가 동시에 재전송하면, 결국 또다시 collide될 것이다.

그래서 Pure ALOHA는 만일 timeout period가 지나면, 각 station은 'random한 시간' 동안 대기 후 각자 frame을 재전송하게 한다.

이때의 '시간'을 *backoff time* T_B 라고 한다.

또한, 만일 K_{max} , 즉 '최대 재전송 횟수'만큼 시도한 뒤에는 station은 포기하고 나중에 다시 시도해야 한다.

time-out period는 *round-trip propagation delay*의 최댓값과 같다.
*round-trip propagation delay*는 두 station간 frame을 전송하는 데 걸리는 시간의 두배이다.

$2 \times T_p T_B$ 는 K(the number of attempted unsuccessful transmissions)에 의존하는 랜덤한 값이다.
 T_B 수식의 한 예로, *binary exponential backoff*가 있다.
 매 재전송 마다, multiplier로 $R = 0 \text{ to } 2^k - 1$ 을 랜덤하게 고르고, T_p (maximum propagation time)에 곱한다.
 혹은, T_{fr} (the average time required to send out a frame)에 곱한다.
 최종적으로 앞의 두 방법 중 하나로 T_B 를 구한다.
 이 방법에서는, 계속해서 collision이 일어날수록, random numbers의 range가 증가한다.

K_{max} 는 주로 15로 선택된다고 한다.

pure ALOHA의 성능들

1. Vulnerable time

collision이 일어날 것으로 예상되는 시간 간격이다. 만일 station이 fixed-length frame을 쓴다면, 각 frame을 보내는 데에 T_{fr} (avg. trans. time)초가 걸릴 것이다. 이때, 만일 패킷 b를 어느 한 시점 t에서 보냈다고 하면, $t - T_{fr} < \tau < t$ 인 τ 에서 다른 station도 보냈거나, $t < \tau < t + T_{fr}$ 인 τ 에 보냈다면 충돌할 가능성이 있을 것이다. 즉, *pure ALOHA*에서,

$$\text{Pure ALOHA vulnerable imte} = 2 \times T_{fr}$$

2. Throughput

G를 한 frame의 전송 time 동안 생성되는 frame의 avg 갯수라고 하자.

pure ALOHA에서는

$$S = G \times e^{-2G}$$

예를 들어, $S_{max} = 0.184$ when $G = 1/2$ 인데, 이 말은, 1/2개의 frame이 생성되는 동안을 한 프레임 transmission time으로 하고,
 이때, 성공적으로 frame이 목적지에 도착할 확률은 18.4%란 뜻이다.

추가로 G=1/2일 때 maximum throughput 성능을 낼 것으로 기대되는데, 왜냐하면 vulnerable time은 $2 \times \text{transmission time}$ 이고,
 이 vulnerable time 동안 한 frame을 생성해 충돌이 없을 것으로 기대되기 때문이다.

Slotted ALOHA

pure ALOHA는 vulnerable time이 $2 \times T_{fr}$ 이다.

왜냐하면 station이 언제 보낼 수 있는지에 대한 정의가 없기 때문인데,
station이 막 보내기 직전이나 막 보낸 직후에 보내게되는 경우가 발생할 수 있다.

slotted ALOHA에서는 시간을 T_{fr} 초의 slot들로 나눈다.

그리고 station들은 각 slot 시작점에서만 frame을 보낼 수 있다.

결과적으로, vulnerable time은 절반인 T_{fr} 로 줄었다.

동시에 같은 slot 시작점에서 보내려다 collision이 일어날 수 있기 때문이다.

slotted ALOHA 성능들

1. Throughput

avg. number of successful transmissions는

$$S = G \times e^{-G}$$

이다.

maximum throughput S_{max} 는 0.368(when G=1)이다.

즉, 한 frame이 전송되는 시간동안 한 프레임이 생성되고, 이때 frame이 목적지에 성공적으로 도착할 확률이 36.8%라는 것이다.

(2배!)

G=1일 때 throughput maximum일 것으로 기대하는 이유는 vulnerable time이 곧 frame transmission time이기 때문이다.

- 12.1.2 CSMA

더욱 collision을 줄이고 performance를 향상시킬 목적으로 등장한 방법이다.

station이 medium을 사용하기 전에 검사한다면 collision을 줄일 수 있을 것이다.

Carrier sense multiple access (CSMA)는 각 station이 보내기전에 medium을 확인하게 한다.

(Listen before talk)

완전히 collision을 막을 수 없는 이유는, CSMA의 a space and time model(F.12.7)에 보듯이,

propagation delay 때문이다.

station이 frame을 보낼 때, 첫 비트가 도착하기까지 짧은 시간이 필요하고,
각 station이 이 비트를 감지하기까지도
마찬가지로 시간이 걸린다.

- 쓰고 있다면 다른 station이 medium을 통해 흘려보낸 data의 첫 비트 부터 다른 station도 감지할 텐데, 이 다른 station까지 전파되기까지 시간이 걸린다.

즉, 각 station이 medium이 현재 idle인지 알 수 있을 때까지 frame을 통해 알다보니, 이 frame을 주고받는데 시간이 걸리는 것이다.

CSMA 성능들

1. Vulnerable Time:

CSMA의 vulnerable time은 propagation time T_p 와 같다.

이 시간은 sig.가 medium 끝에서 끝까지 가는 전파시간과 같다.

2. Persistence Methods:

channel이 busy / idle일 때 station이 할 행동을 정의한 3가지 methods가 있다.

A. 1-persistent method:

단순히 idle한 지 sensing한 뒤, 즉시 보낸다. (with probability 1)

B. nonpersistent method:

sensing해서 idle이면 즉시 보내고, busy라면 random한 시간만큼 기다린 뒤, 다시 sensing한다.

앞 방식과 달리, 두개 이상의 station이 동시에 idle로 sensing하고 보낼 확률이 적다.

반대로 idle상태인 medium에 어떤 station도 전송하지 못하고 있는, nw의 efficiency가 떨어지는 상황이 올 수도 있다.

C. p-persistent method:

channel이 일정한 size의 time slot을 가지고, 이때 slot의 duration은 maximum propagation time과 같거나 더 크다.

이 경우 앞선 두 방식과 달리, collision을 줄이면서 efficiency도 쟁긴다.
다음의 step을 따른다.

A. p의 확률로, station은 frame을 보낸다.

B. $q = 1-p$ 의 확률로, station은 다음 slot의 시작 시간까지 기다리고, line을 checking(sensing)한다.

a. line이 idle하다면, step1으로 간다.

b. line이 busy하다면, collision이 일어난 것처럼 backoff process를 사용한다.

• 12.1.3 CSMA/CD

CSMA method는 collision 발생 시 따라야 할 process를 정의하지 않았다.

Carreir sense multiple access with collision detection (CSMA/CD)는 collision을 다루는 부분에서 알고리즘을 더 다듬었다.

여기서는 station이 frame을 보내고 성공적으로 도착하는지 monitoring한다.
성공적이면 monitoring을 끝내고, 실패하면, 재전송한다.

- A는 persistence procedure를 실행하고, frame의 bit들을 보내기 시작 한다.
- A의 첫 비트가 t_1 에 출발한다.

- t_2 에서 C는 아직 A의 첫 비트를 받지 못했다. (공유 medium을 타고 전파되지 않고 있다.
- 따라서 C도 persistence procedure를 실행하고 frame의 bit들을 보내기 시작한다.
- C의 비트와 A의 비트가 t_2 시점 이후 어느 시점에서 collision 한다.
- C는 t_3 , 즉, A의 첫 비트가 도착한 시점에서야 collision을 감지한다.
- C는 즉시 전송을 중단한다.
- A는 t_4 , 즉, C의 첫 비트가 도착한 시점에서야 collision을 감지한다.
- A도 즉시 전송을 중단한다.

Minimum Frame Size

For CSMA/CD, frame 사이즈를 제한할 필요가 있다.

마지막 비트를 보낼 때까지 collision은 detection해야 한다.

- 프레임을 다 보내고 나면 frame의 copy도 지우고, 더 이상 medium을 monitoring하지 않기 때문이다.
- 따라서, frame transmission time T_{fr} 은, the maximum propagation time T_p 의 2배여야 한다.
 - worst case의 경우, 서로 max propagation distance 만큼 떨어져 있어서 끝에서 끝까지 갔다가 collision을 인지, 다시 원래 끝으로 돌아올 때 $2 \times T_p$ 이기 때문이다.

Procedure Time

F12.13의 diagram.

- ALOHA와 닮았지만 차이점이 존재한다.
1. the persistence process를 추가했다. 앞에서 배운 3가지 중 한 개를 택한다.
 2. 알로하는 frame 하나 보낸 뒤 ack를 기다린다.

frame을 전부 보내지 않고 collision을 살핀다. CSMA/CD는 transmission과 collision detection이 연속되고, 동시에 진행되는 process이다.

- (두 개의 다른 포트/ bidirectional)

monitoring 과정 중에는 transmission이 잘 끝났는지/ collision이 발생했는지 확인한다.

둘 중 하나의 상태이면, 즉시 전송을 중단한다.

루프를 나올 때까지 collision이 일어나지 않으면, transmission이 잘 끝났다는 뜻이다. 전체 frame이 잘 전송됐다. 이분법적으로, 반대의 상황에선 collision이다.

3. 짧은 jamming sig. 을 다른 station들이 collision에 대해 인지하도록 보낸다.

- idle.

1. zero

- station이 channel을 잘 감지하고, frame을 보내고 있는 상태.
- 별거 없다. 그냥 원래 기능을 잘 할 수 있는, 정상적인 상태이다.

Energy Level

channel상에서 sig. energy는 3개의 값을 갖는다고 할 수 있다. 1. zero 2. normal 3. abnormal

- collision이 있어서 level of energy가 normal보다 2배일 것이다.

이렇게해서 채널이 idle / busy / collision mode 중 어느 mode인지 구분짓는다.

throughput

CSMA/CD의 throughput은 pure or slotted ALOHA보다 크다.

- Maximum throughput은 G값에 차에 의해 발생한다.
 - 1-persistence method에서,
G=1일 때, *maximum throughput = around 50%*
 - non persistence method에서,
G=3~8일 때, *maximum throughput = can go up to 90%*

Traditional Ethernet

LAN protocol 중 하나는 CSMA/CD를 사용하는 전통적인 Ethernet - Data rate of 10Mbps이다.

- traditional Ethernet은 broadcast LAN이었고, 1-persistence method to control access to the common media.
- 12.1.4 CSMA/CA

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

- wireless NW를 위해 개발됐다.

collision은 처음에는 channel의 idle상태를 확인할때까지 미루는 과정을 통해 피해간다.

만일 idle이라면, 그 즉시 보내지 '않고', IFS라 부르는 시간동안 대기하게 된다.

- 왜냐하면, 아직 멀리 떨어진 station에서 medium으로 frame을 전파하더라도, 거리가 멀기에 도착하지 않았을 수도(not sensed) 있기 때문이다.

1. the interframe space(IFS)

IFS시간을 기다리고, 여전히 idle이라도 주의해야할 부분이 있다.

Contention window는 여러 슬롯들로 나누어진 일정 시간을 말한다.

보낼 준비가 된(보낼 data/ frame이 있는) station은 wait time의 길이 (= slot 수)를 임의로 정한다.

- binary exponential backoff strategy에 따라 window의 slot을 변경 한다.
 - 이말인 즉슨, IFS time뒤에도 idle이 아니면, station○느 계속해서 slot의 수를 2배로 늘려나간다.
- contention window는 매 time slot마다 channel을 sensing해야 한다.
- 만일 채널이 busy한 걸 찾았으면, process(idle하면, IFS기다리고, contention window만큼 기다리기)를 restart하지 않는다. 다만, timer 를 멈추고, 채널이 idle일 때 다시 재개한다.

3가지 전략을 통해 collision avoidance를 한다.

2. the contention window 3. acknowledgments

여전히 collision resulting in destroyed data가 있을 수 있다. 혹은, 전송 중에 corrupted 일수도 있다.

- receiver에게 frame을 잘 받았다고 보증해주는 무언가가 필요하다.
 - the positive acknowledgment
 - time-out timer

Frame Exchange Time Line

F.12.17은 exchange of data와 control frames in time을 보여주고 있다.

1. station은 channel의 energy level을, frame을 보내기 전에 check한다. (carrier freq.가 기준 레벨이다.)

- A. channel이 idle 상태가 될때까지 persistence strategy with backoff를 쓴다.
- B. idle상태인 걸 감지하면, station은 DCF interframe space (DIFS)라는 시간만큼 기다린다. 그리고 *request to send(RTS)*라 부르는 frame을 보낸다.

2. des.station에서는, RTS를 받고, short interframe space(SIFS)라 불리는 시간만큼 기다린다. 그리고 des.는 *clear to send(CTS)*라 부르는 control fram을 src. station에게 보낸다.

- 이 CTS는 src에게 des.는 받을 준비됐다는 걸 알려준다.

3. src. station은 data를 SIFS만큼 기다린 뒤 보내기 시작한다.

4. des. station은 SIFS만큼 기다린 뒤, frame을 잘 받았다고 알리기 위해 ack를 보내게 된다.

- 반면에 CSMA/CD에서는, collision이 발생하지 않으면, 그 자체가 data가 잘 도착 했다고 src.에게 알리는 수단이다.

Network Allocation Vector

한 station이 access를 얻어버린다면, 나머지 station들은 data 전송을 어떻게 미루게 될까?

그러니까, 이런 경우에 collision avoidance를 어떻게 해결하는가?

- 해결책은 NAV(Network Allocation Vector)이다.

src. station은 RTS frame을 보내고, channel을 타고 frame은 전파될 것이다 (occupying).

station들은 frame의 비트들을 인지하게 되고, timer를 만들게 된다.

- called, a *network allocation vector* (NAV)
- that shows channel을 다시 체크하기까지 이 station이 얼마나 기다려야 하는지.

station이 매번 sys.에 접근하고, RTS 프레임을 보낼 때마다, 다른 station들은 NAV를 시작한다.

Collision During Handshaking

RTS나 CTS control frames가 보내지고 있을 때(*Handshaking period*), collision이 일어난다면 어떨까?

- RTS는 des에게 이제 보내도 되냐고 src가 DIFS를 기다린 끝에, 보내는 frame.
- CTS는 src에게 이제 보내도 된다고 des가 SIFS를 기다린 끝에, 보내는 frame(ack).

=> 2개 이상의 station들은 RTS frame을 동시에 보내려 할 것이다. => 그리고 이 frame들은 충돌한다.

하지만, 이때는 collision detection을 위한 mechanism이 따로 없기 때문에, sender는 CTS frame(des.에서 오는 ok sig.)이 receiver로부터 안 왔을 때, 비로소 collision이 일어났다고 가정하게 된다.

- 그 후, backoff strategy(T_B 만큼 기다린 뒤, $k=0$ 인, idle인지 확인하는 단계로 초기화)가 발생하고, sender는 다시 시도하게 된다.

Hidden-Station Problem

use of the handshake frames(RTS & CTS)이 hidden station problem을 해결할 것이다.

- F.12.17을 보면,
 1. src A가 receiver B에게 보낸다.
 2. B는 반대편 방향으로 C가 있다.
 3. C가 frame을 전송중이라면, B에게도 frame이 간다.

4. B는 A와 C에게 동시에 받는다.
5. A는 C존재를 모른다. (= hidden terminal / station)
6. A는 어떻게 C를 알아?
7. RTS를 B에게 보내고, CTS를 B로부터 받지 못한다면 (C에게 받고있으니), A는 frame을 보내지 못한다.

CSMA/CA and wireless NW

CSMA/CA는 wireless NW에 많이 쓰인다. CH15에 더 자세히 배우겠다.

12.2 Controlled Access

Controlled Access에서, station들은, 서로 합의해서 어느 station이 보낼 권리가 있는지 정한다.

- 3가지 controlled access methods를 알아볼 것이다.
- 12.2.1 Reservation

reservation method는, station이 data를 보내기 전에는, 예약을 해야 한다는 뜻이다.

시간을 여러 구간으로 나눈 뒤, 각 구간마다, reservation frame과 data frame을 순차적으로 보낸다.

N station이 통신에 참여하게 된다면, 정확히 N개의 reservation minislots이 reservation frame에 있게 된다.

각 station은 각자의 own minislot을 가지게 되고, reservation frame의 own minislot 값을 1 비트로 바꾼 뒤,
다음에 이어서 data frame을 실을 수 있게 된다.

- 12.2.2 Polling

primary station과 secondary station을 정의한 topology를 갖고 있다.

- primary device는 link를 control한다.
 - 어떤 device(secondry)가 given time에 channel을 쓸지 정해준다.
 - 즉, session의 initiator 역할을 한다.
- secondary device는 지시를 따를 뿐이다.

이 method는 *poll*과 *select function*들을 써서 collision을 막는다.

- primary station이 control하기 때문에, 애가 fail이면, sys. 전체가 영향을 받는다.

Poll

poll function은 primary가 secondary에게 transmission을 요청할 때 쓴다.

1. primary가 data를 받을 준비가 되면, 각 차례마다 station들에게 ask (poll) 한다.
2. secondary는 보낼 게 있으면, data frame을 보내고, 없으면 NAK frame을 보내게 된다.
 - A. NAK이 왔으면, 다음차례의 secondary에게 같은 방식으로 물어본다.
 - B. data frame이 왔으면, primary는 frame을 읽고 잘 받았다고 ACK frame을 보낸다.

select

select function은 primary가 보낼 게 있을 때 쓴다.

link가 available할 때,

- target device가 prepared한 건 어떻게 알 수 있나?
- 때문에 primary는 secondary에게 전송을 미리 알고(SEL(=select) frame; 이때, 한 field는 secondary의 addr.), ready라는 ack를 기다리게 된다.
- 12.2.3 Token Passing

토큰 패싱 method에서는, nw안의 station들이 하나의 logical ring으로 구성돼있다.

각 satation의 앞, 뒤에 위치한 (ring상에서) station을 *predecessor*(전임자)와 *successor*(계승자)라 부른다.

*predecessor*은 논리적으로 링 상에서 앞에 있는 station을 가리킨다.
*successor*은 논리적으로 링 상에서 뒤에 있는 station을 가리킨다.

channel을 쓸 수 있는 권리는 predecessor => current => successor 순으로 넘어간다.

- 채널을 쓸 수 있는 권리는 어떻게 주어질까?

special한 패킷인 *token*이 ring을 따라 순환하면서, token을 소유하고 있는 station이 이 권리를 갖게 된다.

- 소유한 station이 보낼 게 있으면, data를 보내고, 없으면 다음 station에 이 패킷을 넘기는 것이다.

token management function들도 필요하다.

- token이 nw상에서 사라지지 않도록 보장.
- station과 전송되는 data type (?)

교수님이 생략하신 내용인 걸 보니 영... 안 중요한가보다.

Logical Ring

굳이 physical한 ring -> 일 필요는 없다. logical하게 ring을 구성하면 된다.

- 교수님 추가. Scheduling

time duration을 time slot으로 나눈다.

특정 time slot을 어떤 node/ station이 사용할 것인가를 scheduler라는 central controller가 정한다.

1. 각 station들은 할당 요청을 scheduler에게 보낸다.

- multiple access protocol(앞에서 배웠던 방법들 중 하나)을 통해 scheduling req.를 보낸다.

2. scheduler가 빈 time slot을 station에게 알려준다.

12.3 Channelization

Channelization (or channel partition)은 channel의 bandwidth를 매 시간단위/ freq./ code 동안 나눠쓰는, multiple- access method이다.

- channel을 sub channel로 나누겠다.

FDMA, TDMA, CDMA

- CH16에서 적용하는 것까지 살펴볼 예정이라고 한다.

- 12.3.1 FDMA

Freq.-division multiple access(FDMA)

available bandwidth is divided into freq. bands.

freq. 구간에 따라 나눠진 sub channel들이 각 station에 all time(=entire period of commu.) 할당된다.

- 각 station들은 bandpass filter를 둬서, 해당 BW에서만 보내도록 전송 freq.를 고정 한다.
- 서로 간섭이 없도록 (station/ sub channel끼리), small guard bands를 sub channel 사이사이에 둔다.

FDMA와 FDM(freq. division muxing)과 닮았으나 차이점이 있다.

1. FDM

CH6에서 봤듯이, physical L technique이다.

FDM은 저대역폭을 고대역폭의 채널에 실는다.

- combined channel들은 low-pass이다.
- multiplexer가 sig.들을 modulate하고, 합친 결과로 bandpass sig.가 나온다.
- multiplexer가 BW를 shifted 한 것이다.

2. FDMA

DL L의 access를 위한 방법이다.

- 각 station의 DLL에서는 physical L에게 data를 보낼 bandpass sig.를 만들라고 한다.
 - sig.는 할당된 band에서 만들어져야 한다.
 - 이때, physical L에서는 multiplexer가 없다.
- sig.들은 자동적으로 bandpass-filter에 의해 bandpass sig.로 만들어 진다.
- 그 후, 공통 채널에서 mix돼서 한꺼번에 보내진다.

전자기학적인 이론에 의거해 여러 carrier에 원본 sig.를 실어보내는 것/ 필터로 자동적으로 나오는 sig.들을 보내는 것

에 차이가 있는 것 같다.

- 12.3.2 TDMA

Time-division multiple access (TDMA)은, station들이 BW를 시간 간격으로 나눠쓰는 방법이다.

각 station은 time slot을 할당받게 된다.

큰 문제점은 서로 다른 station들이 sync.를 어떻게 맞추느냐이다.

- 각자 할당받은 slot의 시작점을 알아야 한다. 타이밍을 맞춰야 치고 들어가지(?)

문제는 광범위한 지역에 station들이 퍼져있으면, propagation delay가 있기 때문에 무척 sync.를 맞추기가 어려워진다. 이런 delay를 고려해서, guard time이란 것을 넣는다.

- 각 slot 시작점에 (slot들 사이사이가 되겠다.) synchronization bits(or preamble bits)를 넣는다.

CH6에서 봤듯이, physc. L technique로,
더 느린 data rate를 가진 channel들에서 오는 data들을 섞어, 더 빠른 data rate를 가진 channel에 보내는 것이다.

- 해당 process는 물리적인 multiplexer를 쓴다.

1. TDM(Time-division muxing)

TDMA와 TDM도 헷갈릴 수 있겠다. 2. TDMA

access를 위한 방법으로, DataLink L에서 쓴다. DLL의 각 station들은 physic. L에게 할당된 time slot을 쓰라고 한다. 역시 물리적인 multiplexer 를 쓰지 않는다.

- 12.3.3 CDMA

Code-division multiple access (CDMA)는, 수십년 전부터 구상됐다.

- FDMA와 달리, 1 channel이 전체 BW를 쓴다.
- TDMA와 달리, 모든 station들이 동시에 data를 보낸다.

Analogy(대강 이렇다~)

CDMA는 다른 코드들을 이용한 통신이다.

예를 들어, 큰 방안에서 원하는 몇몇끼리씩 대화를 하고, 이때 대화를 참여하는 모임들은 각자들만 할 수 있는 언어(code)를 쓴다.

Idea

4개의 station이 있다고 가정하자.

- 각 station들은 같은 channel에 연결됐다.
 - 각 station으로부터 나온 data는 d_i 이다. 예를 들어, station 1로부터 나온 data는 d_1 .
 - 각 station에게 할당된 코드는 c_i 이다. 예를 들어, station 2에게 할당된 code는 c_2 /

그리고 추가로 두 가지 가정을 한다.

1. code들은 서로 다른 code와 곱해질 경우, 0이된다.
2. code들은 같은 code와 곱해질 경우, 4 (= the num. of stations)가 된다.

F 12.23을 참고하자.

1. station 1은 $d_1 * c_1$ 을 한다. station 2, station 3 ... so on.
2. 채널 상에서 data는 이 곱해진 항들의 합이다.
3. 다른 3개의 station 중 어느 곳으로부터 data를 받길 원하는 station은 sender의 code와 함께 곱한다.
4. 앞에서 정한 성질에 의해, sum인 data 중에서는 sender station에서 온 항만 살아남는다.
5. 4로 나눠주면 원래 data를 얻을 수 있다.

Chips

CDMA는 Coding theory에 기초한다. (* 대강 복호화, 암호화, 압축 등에 대한 통신관련 이론이다.)

- 각 station은 code를 할당받는데, 이때 code란, *chips*라 부르는 seq. = 순열(순서가 있는 숫자의 열)이다.

대략 아래의 계산(?) 규칙이 생긴다.

1. 각 seq.는 N개의 station이 있을 때, N elements를 가진다.
2. seq.를 숫자와 곱하면, 스칼라배가 된다. (각 seq.의 ele.들이 숫자배가 될 것이다.)
3. 2개의 seq. 끼리 곱한다면, inner product가 되고, 서로 같은 seq. 즉, 같은 code면 그 값이 N이 될 것이다.

이런 성질을 갖게되는 seq.를 orthogonal seq.라고 부른다.

Data Representation

F 12.25에서 보듯이,

- data bit 0은 -1로 보내고,
- data bit 1은 +1로 보내고,
- silence는 0으로 보낸다.

아무래도 spreading BW를 쓸 수밖에 없다.

sig. level

역시 spreading BW..... 0을 표현하는데 -1이 되고, 다시 code가 곱해져 -N 혹은 N이 될 수 있다.

Sequence Generation

chip sequence를 생성해내기 위해, *Walsh table*라는 2차원 table을 이용한다.

별로 안 중요하다. 교수님이 언급조차 하지 않으셨다....

CH.13 Wired LANs: Ethernet

CH 9~12에서 DLLL에 관한 general한 issue들을 알아봤다.

이번 장에서부터는 wired LANs 프로토콜을 배울 것이다.

- 많은 wired LANs 프로토콜 중에서도 오늘날까지 Ethernet 기술이 살아남았다. 책에서 오늘날이라고 해봤자, 90년대가 아닐까....

- 교재에서는 기술(technology)과 protocol을 자주 혼용해서 쓴다.

13.1 Ethernet Protocol

CH 1에서, TCP/IP protocol suite는 DLL이나 Physic. L 프로토콜 어느것도 정의하지 않았다고 배웠다.

- 그 말은 즉슨, NW L에게 서비스하는 어떤 프로토콜도 수용하겠다는 뜻이다.

DLL과 Physic. L는 실제로 local과 wide area NW인데,

- local area NW인 LAN을 CH1에서 배웠었고,

지리적으로 같은 빌딩이나, 캠퍼스 안과 같은 지역 안에서 쓰도록 만들어진 프로토콜이다.

- 하지만 오늘날, 안에서만 쓰지 않고 바깥과 - LAN은 WAN이나 Internet과 이어져있다.
- 80's, 90's, 여러 타입의 LANs가 쓰였는데, 대부분 sharing media 문제를 해결하기 위해 media-access method를 썼다.

Ethernet 또한 CSMA/CD 방법을 썼다.

- 이외에도 token ring, token bus, and FDDI(Fiber Distribution Data Interface) 같은 프로토콜들은 token-passing 방법을 썼다고 한다.
- 교수님 대학원시절 선배들은 ATM LAN 논문을 쓰고 졸업했다는 전설이 있다.

결국 대부분의 기술들을 Ethernet이 흡수, Ethernet이 살아남았다

- 13.1.1 IEEE Project 802

Ethernet에 관해 얘기하기 전에, IEEE standard를 잠시 짚고 넘어간다.

85'에, IEEE는 project 802를 시작했는데, 다양한 장비들간 내부통신을 위한 기준을 세우기 위해서였다.

- OSI나 TCP/IP protocol suite를 대체하는 모델이 아닌, Physic. L와 DLL에서 주로 쓰이는 LAN protocol의 기능을 정의하기 위해서였다.

TCP/IP suite와의 관계를 보자면, F 13.1에서 보듯이, DLL을 두 개의 subLayer로 나눴다.

1. Logical link control (LLC)

- 앞에서 우리는 DLC(data link control)를 배웠었는데, framing, flow control, err. control을 다루는 subL였다.

IEEE에서는 이런 기능들을 하나의 subL로 묶었는데, 이걸 LLC라고 한다. single link L protocol을 제공한다.

- single link L protocol은 MAC의 투명성을 제공한다는데, 아무래도 모든 LAN에서 LLC는 기능이 같다보니, 나머지 기능을 MAC의 기능으로 쉽게 간주할 수 있게돼서 그런가보다(?)

2. Media access control (MAC)

- Ethernet LANs는 CSMA/CD를, token rings나 FDDI 등은 token-passing을 사용하게끔 MAC을 프로토콜에 따라 정해놨다.

물론, Physic. L에 대해서도 여러 표준을 정했다.

- 13.1.2 Etherent Evolution

점차 bps를 늘려가는 방향으로 발전해왔다.

standard Ethernet	Fast Ethernet	Gigabit Ethernet	10 Gigabit Ethernet
10Mbps	100Mbps	1Gbps	10Gbps

최근에는 TeraBps 개발중....

13.2 Standard Ethernet

standra에서 evolution하는동안 바뀌지 않았던 특징 몇개를 알아볼 것이다.

- 13.2.1 Characteristics

Connectionless and Unreliable Service

conenctionless sv.를 제공한다.

- 전송하는 각 frame은 independent하다.
- connec. est. / connec termin. phase가 없다.
 - receiver의 ready상태를 확인하지도 않고 보낼 거 있으면 바로 보내버린다.
 - receiver가 처리하기 벅차 dropping frame이 일어날 수도 있다.
 - sender는 frame이 drop됐는지도 모른다. 못된것.

때문에, Ethernet의 sv.를 받는 IP도 역시 connectionless고, 갔는지 못 갔는지 알지 못한다.

심지어 이 IP sv.를 받는 transport의 L(계층)이 UDP라면, 책임(frame이 사라진 걸 확인할)은 온전히 APP L의 몫이된다. 못된것.

- 물론, TCP를 쓴다면 connection oriented기 때문에 packet(segment)를 다시 보낼 것이다.

만일, Ethernet에서 receiver가 corruption을 CRC-32와 같은 방식으로 감지한다면, 바로 버려버리고 그 외에는 신경쓰지 않는다.

- 패킷을 lost했는지 여부는 상위 L의 일이기 때문이다. 못된것.

Frame Format

Ethernet frame의 구조는 다음과 같다.

Preamble	SFD	Des. addr.	Src. addr.	Type	Data and Padding	CRC
7bytes	1byte	6bytes	6bytes	2bytes	46~1500bytes	4bytes
<--Physic. L-->						

1. preamble
2. Start frame delimiter(SFD)
3. Des. addr. (DA)
4. Src. addr. (SA)
5. Type.
6. Data
7. CRC

data가 data지. 상위 L에서 encapsulated된 부분이다.

- 1500 bytes까지 되는데, 위에서 내려온 애가 1500B넘으면 더 쪼개서 frame으로 encapsulate한다.
- 아래에서 올라온 애가 46B를 못 넘으면 (최소 바이트도 못되면), padding으로 0을 채워서 넣고, 나머지는 상위 L가 이 padding을 제거할 의무가 있으므로, 알아서 하게 냅둔다.

이 frame이 누구의 패킷이 encapsulate 됐는지, upper-Layer protocol을 정의한다.

- 여기서 upper L protocol은 IP, ARP, OSPF 등등이 될 수 있다.
- datagram의 protocol field나, segment의 port number와 같은 기능이다.
- muxing과 demuxing에 쓰인다. -> 여러 프로토콜에서 내려왔지만 (<=> 여러 프로토콜로 나눠 올라가지만) DLL에선 독립적으로, 하나의 처리과정을 거칠 뿐이다.

6B의 크기로, src. station의 LL addr를 갖고 있다.

6B의 크기로, Des. station의 LL addr를 갖고 있다.

- 본인의 DLL addr거나, multicast 그룹에게 보내는 multicast addr. 대상 멤버중 하나거나, broadcast면, decapsulates해서 upper L로 올려보낸다.
- 특정 비트열로 beginning of the frame을 알림
- sync.의 마지막 기호이기도 해서 경고의 의미를 가지기도 한다.
- 다음 field는 des. address이라고 알린다.
- Ethernet frame은 variable length frame이라서 시작을 알리는 게 중요하다.

얘까진 physic. L이다.

7bytes로 0/1이 왔다갔다하면서 sys에게 coming frame이 있음을 알리고, clock에 sync를 할 수 있게 돋는 등, physical L에서 synchronization을 위한 부분이다.

CRC-32 만일 receiver가 addr - type, data field에 걸쳐 CRC를 계산해서 0이 아닌 field를 발견하면, frame을 버린다.

frame Length

Ethernet은 길이 제한이 있다.

- min: 64bytes (=512bits)
- max: 1518bytes

Header와 trailer 길이도 포함이라 Header길이와 trailer길이(src.addr, des.addr, length or type, CRC)를 뺀 나머지 부분이 64B 넘거나 1518B보다 작아야 한다.

대강 역사적으로 메모리(버퍼)가 비싸기도 했고, (과거에는...그랬지)
shared medium 특점을 막는다고 한다. <= 별로 안중요한게 교수님이 그냥 넘어가셨다!

- 13.2.2 Addressing

Ethernet NW에서는 각 station(예로, PC, workstation(전문 분야 작업용 PC), printer 등이 있다.)에서는 각자의 *Network interface card* (NIC)가 있다.

- NIC는 station안에 있고, station의 DLL addr를 제공한다.

Ethernet addr.는 6B (48bits)이다.

- 16진수로 써있고, 1bytes마다 colon(:)으로 구분된다.
- 예시는 다음과 같다. 4A:30:10:21:10:1A

Transmission of Address Bits

ex.13.1을 보면, 47:20:1B:2E:08:EE라는 주소가 보내지고 있다.

- addr.는 left to right로 보내지고, byte by byte로 보내진다. 각 byte에 대해서, right to left로 보내지고, bit by bit로 보내진다.

Hexadecimal	47	20	1B	2E	08	EE
Binary	01000111	00100000	00011011	00101110	00001000	11101110
Transmitted <-	11100010	00000100	11011000	01110100	00010000	01110111

Unicast, Multicast, and Broadcast Address

1. least significant bit of the first byte in a des. addr가 0이면 (짝수면) => 'unicast'
2. least significant bit of the first byte in a des. addr가 1이면 (홀수면) => 'multicast'
3. 모든 byte가 F면 => 'broadcast'

Distinguish Between Unicast, Multicast, and Braodcast Transmission

stdard Ethernet은 coaxial cable의 bus topology라거나, twisted-pair cable의 hub를 중앙에 두는 star topology를 썼다.

- 모두 shared medium이고, 그러다보니 physical L 상에서는 broadcast이다.
- DLL에서 frame을 받아보고 자신한테 온 게 아니면 버린다.

- 13.2.3 Access Method

std. Ethernet protocol이 broadcast nw이었기 때문에, sharing medium에서 access control method가 필요하다.

- std. Ethernet은 CH12에서 얘기했던 CSMA/CD with 1-persistence를 썼다.

예시를 들어보자.

1. station A는 D에 보낼 frame이 있다.
2. A는 carrier sensing을 한다.
 - A는 medium의 energy level을 체크하고, sig. energy가 medium상에서 없으면, medium이 idle하다고 판단한다. frame을 보낸다.
 - 만일 sig. energy가 not zero라면, idle이 될때까지 관찰한다. 그 후, idle하면 frame을 보낸다.
3. A는 frame을 보내고, 확실히 전달됐다고 알게될 때까지 자신의 buffer에 frame의 copy본을 가지고 있게 된다.

A는 sensing을 계속하게 되는데, 언제까지 sensing을 할지, 두 가지 케이스가 있다.

1. A는 512bits를 보내고, collision을 감지(energy LV이 regular 보다 훨씬 높다)하지 못했다.

sensing medium을 중단한다. 이때, 잘 생각해보면, 1. std. Ethernet일때, 10Mbps로, 512bits는 $512/(10Mbps) = 51.2\mu s$ 의 시간동안 station에서 선로상으로 나간다. (send out)

2. cable의 speed of propagation = $2 \times 10^8 \text{ meters}$ 이므로, 51.2us동안 frame의 첫 비트는 10,240m(쭉 갔을 때), 혹은 5120m(round trip)일 것이다.

- RTT(round trip time):

<https://www.geeksforgeeks.org/what-is-rttround-trip-time/>
src station에서 frame을 보내고, des. station으로부터 ack를 받은 시간이라고 봐도 된다.

비트가 끝까지 갔다가 튕겨 돌아오는 건 아니다. <= 생각해보면 bus topology에서 terminator가 있었다. 3. cable상의 마지막 station에서 collision이 일어나고, 돌아올 것이다.

즉, collision이 감지되려면, 최악의 경우를 상정하더라도 512bits를 sender가 모두 보내기 전에,

그리고 첫 비트가 5120m의 round trip을 마치고 돌아오기 전에 collision이 발생해야 한다.

또 주의할 점이 있다. cable이 5120m라는 가정을 위에서 했었는데, 실제로 디자이너들은 2500m정도로 한다. 여러 delay들을 고려하기 때문이다.

결국 이 모든 걸 고려해서 station A가 512bits를 다 보내기 전에 collision을 감지하도록 해야 한다.

=> 이렇게 잘 고려해서 만들었으니까, station A는 512bits를 다 보낼때까지 collision이 감지되지 않았으면 안심하고 copy of frame을 버린다.

2. station A가 512bits를 다 보내기 전에 collision을 감지했다.

어느쪽에서든 다른 station에서도 frame을 보내다가 같이 충돌했을 것이다. 따라서 A와 다른 statoin 모두, line이 available할 때까지, buffer에 copy본을 keep해야한다.

- 이때, collision을 nw상의 다른 station에게도 알리기 위해 48bits의 jam signal을 보낸다.

- 13.2.4 Efficiency of Standard Ethernet

교수님이 pass.

- 13.2.5 Implementation (구현방식)

10BaseX로 이름을 붙임.

- 10 => 10Mbps의 속도
- Base => baseband digital sig.를 이용해 전송한다는 뜻.
- X => 100m단위 표시 max. cable 길이일수도 있고, type일수도 있다.

- 13.2.6 Changes in the Standard

다음 단계의 Ehthernet으로 가기 위한 여러가지 기술 발전이 있었다. (shared Ethernet -> Bridged Ethernet -> Switched Etherent->Full-Duplex Ethernet)

1. Bridged Ethernet(Bridged LAN):

2. Switched Ethernet (Switched LAN):

3. Full-Duplex Ethernet:

좀 더 발전됐다.

- multiport bridge = bridge 수가 늘면 늘수록 bridge port수만큼 나눠서 쓸 수 있는데, 더 나눠서 최종적으로 N-port 스위치까지.
- N-port 스위치는 N개의 station과 이어져있으며, N개의 collision domain을 가지게 된다.

layer-2 switch 는 N-port bridge에다가 패킷을 빠르게 다룰 수 있는 기능들이 추가된 장비이다.

bridge => LAN topology를 여러 sub domain(=브릿지의 port수)로 나누는 기능을 한다.

- bus topology라면 shared medium에 여러개 station이 연결돼 있으니 문제가 있을수도....

- 이걸 다시 여러개의 sub domain로 나눠서 수월하게.
- collision domain을 분리할 수도 있다. collision이 일어나 영향을 미치는 지역을 줄인다.

link 2개를 쓴다.

- CSMA/CD를 쓸 필요가 없는데, 각 station들은 switch와 2개의 link로 이어져있기 때문이다. station끼리 독립적으로 data를 주고 받는다.
- point-to-point
- MAC sublayer의 collision sensing 기능은 꺼버릴 수 있다.
- 대신에, flow/err. control을 위해 LLC와 MAC subLayer 사이에 MAC control sublayer를 넣었다.

13.3 Fast Ethernet (100 MBPS)

90's, LAN 기술이 발전해 전송속도가 10 Mbps보다 빨라졌다.

- FDDI나 Fiber Channel 등이 시장에 등장했다.

FDDI: http://www.ktword.co.kr/test/view/view.php?m_temp1=272

우리 Ethernet LANs도 살아남고 싶어서 열심히 노력한 끝에 Fast Ethernet이 나오게 됐다.

- 100MBPS라고 한다.
- std. Ethernet과 비교했을 때, MAC sublayer를 변경하지 않았다. 즉, frame의 format이나 max. min. size를 바꾸지 않았다.

대강 목표가 다음과 같았다.

1. data rate를 100Mbps로.
2. Std. Ethernet에 비교해 우월한 성능
3. 48-bit addr.를 유지.
4. frame format을 유지.

- 13.3.1 Access Method

전 소단원(?)에서 CSMA/CD는

- transmission rate
- min. size of the frame
- max. network length(물리적 거리)

에 달려있다고 했다. (영향을 받는다?)

min.size of the frame을 keep한다면,
max. length of the NW를 바꾸어야 한다.

즉, 512bits의 min. frame size를 가지고, 전송속도를 10배 빠르게 한다면 (10Mbps -> 100Mbps),

1. collision detection이 10배 더 빨라져야 하고
2. max. length of the NW는 10배 더 짧아져야 한다.

두가지 solution을 교재는 제시한다.

1. CSMA/CD를 유지하고, Bus topology -> passive hub와 star topology를 쓰겠다. 대신, NW길이를 250m로 (1/10) 줄이겠다.
2. CSMA/CD를 사용하지 않고, full duplex link-layer switch를 사용하겠다. (point-to-point라서 collision이 발생하지 않겠다.)

Autonegotiation

동작 모드와 전송 rate를 두 device가 자동으로 협상(negotiate)하도록 하는 기능을 추가했다.

- 13.3.2 Physical Layer

star topology에서는 center로 hub(CSMA/CD 사용)나 switch(CSMA/CD를 사용하지 x)를 사용할 수 있다.

- 전송 rate가 올라가면서 새로운 encoding 방식을 사용하게 된다.

Encoding

Ethernet에서는 manchester encoding 방식을 사용했었는데,

fast Ethernet에서는 구체적인 implementation에 따라 다양한 방법을 사용한다.

- 전송 rate가 올라가면서 BW requirement, noise 등을 고려한다. 자세히 알 필요는 없다.

13.4 GIGAbit Ethernet

1000Mbps까지. IEEE에서는 standard 802.3z라고 불렸다.

1. Data rate를 1 Gbps로.
2. std.나 fast Ethernet보다 낫게.
3. 48-bit addr. 계속 사용.
4. 같은 frame format 사용.
5. min. and max. frame lengths 사용.
6. fast Ethernet에서 정의했던 autonegotiation 지원.

- 13.4.1 MAC Sublayer

두 가지 모드로 medium access를 지원한다.

1. Full duplex mode:

central switch를 사용. 대신에 collision이 없다.

- maximum cable length는 collision과는 무관. attenuation에 의해 결정된다.

2. Half duplex mode: CSMA/CD (rare)

- 13.4.2 Physical Layer

topology : star with a hub or a switch(를 이용하는 경우가 더 많다.) at the center

implementation이나 encoding 방식은 굳이 더 자세히 알 필요는 없다고 교수님이 말씀하셨다.

- 대강, fiber 를 사용하면 attenuation이 작아서 길이가 더 길어진다.

13.5 10-Gigabit Ethernet

Ethernet을 LAN으로만 썼었는데, MAN(metropolitan area NW)로 쓰게 됐다.

- NW 사이즈가 크다보니 전송 rate도 커져야 했다.
- backbone NW에 해당하는 부분도 존재하기 때문에, 오고가는 트래픽 양도 늘어났기 때문이다.

책에는 10Gigabit까진데, 2010년 100Gbit, 2017년에는 Terabit Ethernet 얘기가 나오고 있다.

Ethernet 기술은 전송 rate를 올려가면서 MAN이나 backbone망까지도 사용하고 있다.

- 13.5.1 Implementation

CSMA/CD를 사용하지 않고, Full Duplex mode만 사용하고 있다. fiber의 경우, km단위까지 쓴다.

CH.17 Connecting Devices And Virtual LANs

Host, 또는 LANs는 독립적으로 작동하지 않는다. 하나 이상의 다른 Host/ LANs, 또는, Internet과 연결돼 있다.

- host들, 또는 LANs과 연결하기 위해, connecting device가 필요한데, connecting device들은 Internet model의 여러 계층(Layer)안에서 동작한다.

몇개의 device들을 살펴본 뒤, 어떻게 사용되고, 또 VLANs를 만드는 지 알아볼 것이다.

Internet과 internet(work)의 차이를 다시한번 재고하자.

- <https://ko.wikipedia.org/wiki/%EC%9D%B8%ED%84%B0%EB%84%B7>

대략, Internet은 TCP/IP를 쓰는 거대한 망; 고유명사고, internet(work)는 내부 망; 일반명사이다.

17.1 Connecting Devices

Connecting devices: hubs, link-layer switches, and routers

host들을 하나의 network로, 또는 network들을 하나의 internet으로 이어 준다.

- 17.1.1 Hubs

Physical L에서 동작하는 일종의 repeater - physci. L의 신호의 세기가 감소하면 세기를 키워 다시한번 재전송.

- Ethernet LAN이 bus topology을 썼을 때는, 동축 케이블의 길이의 한 계를 극복하기 위해 두 개의 LAN을 이어주는 용도로 썼다.
- 오늘날(90's?)에는 star topology를 쓰고, 여기서 multipoint device, 즉 hub로 쓴다.
 - packet을 broadcast(받은 구멍빼고 모든 port로)한다.
 - physically로는 모든 station이 받지만, LL에서 LL addr.를 확인
=> 본인 addr아니면 폐기.

- 17.1.2 Link-Layer Switches

physic. L와 LL에서 동작한다.

1. Physic. L: 들어온 sig. 를 regenerates.
2. LL: frame의 header를 읽어(참조)- MAC addr.를 읽어서, 동작을 결정한다. = Filtering 기능.

- Hub 입장에서는 참조하지 못하는 부분이다.
- mac addr인 des. addr를 체크해서, 어느 outgoing port로 frame을 내 보낼지 결정한다.
 - Switching table(= forwarding table)을 보고 결정한다.

Transparent Switches

station들은 switch의 존재를 알지 못한다.

Learning 초기에는 static하게 des.와 port를 관리자가 지정. 하지만 일일히 수정해줬어야했다.

후에는 dynamic table을 사용,

1. des. addr를 가진 frame이 특정 port로 들어온다.
2. 다음의 두 가지 경우가 발생한다.
 - A. sw table에 해당 des. addr & port 항목이 없다. flooding을 해서 frame을 모든 port(incoming port제외) port로 broadcast. 그러면서 src addr와 그때의 특정 port는 그대로 sw table에 des. addr와 port로 기록된다.
 - B. sw table에 해당 des. addr & port 항목이 있다. 그대로 보낸다.

1. frame들은 한 곳에서 다른 곳으로 가야한다.
2. forwardingg table은 sw.가 NW상의 frame 움직임을 보고 자동적으로 만든다.

IEEE 802.1d에 따르면, transparent sw.를 포함한 sys는 다음을 만족해야한다. 3. 루프는 방지돼야 한다.

F 17.2의 예시 LAN1은 허브와 host A,B,C,
LAN2는 허브와 host E,D,F가 있다.
그리고 LAN1과 LAN2 사이를 두 개의 스위치가 이어주고 있는 상황이다.

1. A에서 D로 frame을 보낸다.
2. 스위치들은 A와 port를 sw table에 기록한다. flooding한다.
3. LAN2의 HUB를 통해 E,D,F로 간다. 다만, HUB는 서로 반대 스위치와도 연결돼있으므로, 서로 반대 sw로 보낸다. 또, 양쪽 스위치로 온 copy본이 2개이다.
4. 서로 반대 sw는 여전히 src가 A인, 새로운 port로 온 frame을 받았고, 또 D에 대한 정보도 없으므로, 또 flooding한다.
5. LAN1의 HUB를 통해 A,B,C로 간다. 다만, HUB는 서로 반대 스위치와도 연결돼있으므로, 서로 반대 sw로 보낸다. 또, 양쪽 스위치로 온 copy본이 2개이다.
6. LOOP

루프 해결을 위해 Spanning Tree Algorithm이 쓰인다.

- loop가 없다.
- 최적화된 spanning tree를 만드는 법은 잘 알려져있다.
 - 자료구조에서 배웠으나 나는 사실 기억이 안난다.

Advantage of SW

- Collision Elimination:
CH13에서 배웠듯, collision이 발생하지 않아서 Carrier sensing이 필요 없다.

- Connecting Heterogenous Devices:
LLC sw는 서로 다른 physical L protocol(data rate)를 쓰는 devices, 또는 다른 전송매체를 쓰는 디바이스들을 이어준다.
 - frame의 format은 변하지 않기 때문이다.

- 17.1.3 Routers

router는 NW L를 배울 때 보다 더 자세히 배울 것이다.
이번 CH에서는 router를 2L sw, 그리고 hub와 비교를 할 것이다.

라우터는 3-Layer device이다.physic. L에서 NW L까지.

1. physic. L:

2. LL:

3. NW L:

NW L addr를 체크한다.

router는 physical addr.를 체크한다.

받은 sig.를 regenerates.

라우터는 networks를 잇는다.

independent 한 NW들을이어 internetwork를 형성한다.

- 즉, 두 개의 NW가 라우터로 이어지면, 하나의 internetwork or internet이된다.

연결된 link? port를 I/F라 하는 것 같다(?)

1. 라우터는 각각의 interface에 대한 physical & logical(IP) addr.를 갖고 있다.

2L sw, hub와 비교하자면,

2. 라우터는 패킷이 도착한 interface의 주소와 LL des. addr.가 일치해야 작동한다. 3. 라우터는 packet의 LL addr(des & src)을 forwarding하면서 바꾼다.

arp로 다음 라우터의 DLL addr/ MAC addr로 바꾼다는 뜻이다.

17.2 Virtual LANs

한 station이 physical하게 다른 station들과 LAN을 이루고 있다면, 그 station은 LAN의 일부라고 할 수 있겠다.

LAN의 일부, 즉 membership의 기준은 geographic 하겠다.

책에서 정의하기를, Virtual local area NW (VLAN)은, physical하게 wiring된 여부와 관계 없이, S/W로 정의된 LAN이라고 한다.

- 17.2.1 Membership

*vendor들은 grouping하는 characteristic으로

What Is a Vendor?

A vendor is a party in the supply chain that makes goods and services available to companies or consumers. The term "vendor" is typically used to describe the entity that is paid for goods provided rather than the manufacturer of the goods itself. However, a vendor can operate as both a supplier (or seller) of goods and a manufacturer.

1. Interface Numbers

2. MAC addresses

3. IP addr.

4. Multicast IP Addr.

5. combination

SW로 이런 모든 특성들을 섞어쓴다.

multicasting at the IP addr.를 특성으로 쓴다.

또 어디서는, 48-bit의 MAC addr.를 쓰기도 한다.

Some VLAN vendor들은 switch interface numbers(=port number)를 membership의 특징으로 삼는다. 예를 들어, ports 1,2,3, and 70이 VLAN 1이고, 하는 식이다.

- 17.2.2 Configuration

station들이 서로 다른 VLANs로 묶이는지.

3가지 방식이 있다.

1. Manual Configuration:

2. Autoamtic Configuration:

3. Semiautomatic Configuration:

station은 VLAN으로 관리자가 정한 기준에 의해 connected or disconnected된다.

NW 관리자는 VLAN S/W를 사용해서 setup때 서로다른 VLANs로 할당한다.

initializing은 manually 처리되고, migration은 자동적으로 끝난다.

- 17.2.3 Communication between Switches

multi-switched backbone에서,

각 switch는 각 station이 어느 VLAN에 속했는지 뿐만 아니라, 다른 switch와 연결된 station들이 어느 VLAN에 속했는지도 알아야 한다.

다음과 같은 방법들이 고안됐다.

Table Maintenance

station이 같은 group member들에게 broadcast하면, 스위치는 table에 station membership을 기록한다. 스위치들끼리는 table을 주고받아 주기적으로 updating한다.

Frame Tagging

frame이 switch들 사이를 오가면서, VLAN의 des.를 정의하기 위해 MAC frame에 추가 header를 붙인다.

frame tag는 받은 switch가 VLAN상에서 broadcast된 메시지를 받을지, 맡지를 결정한다.

Time-Division Multiplexing (TDM)

switch들 사이의 connection이 time-shared channels로 나눠진다.

IEEE Standard

96's, IEEE 802.1는 802.1Q라는 std.를 내놓았다.

format과 tagging에 관한 것이다.

- multi-switched backbones에서 쓰일 format을 정의하고,
- VLANs상에서 multivendor equipment를 쓸 수 있게 했다.

802.1Q

VLAN상에서 쓰기 위해, Ethernet frame에서, src. MAC addr부분과 type/size부분 사이에 802.1Q header를 넣었다.(tagging)

- Ethernet frame에 대해, bridge에서 절차와 switch에서 어떻게 다를지에 관해 정한다.
- QoS 우선순위 (숫자가 높을수록 높다) = PCP (priority value)
- Multiple Registration Protocol (MRP)

PCP value 따라 다른 처리를 해야하고, 어떤 처리에 관한 parameter나 특징들을 정한 프로토콜이다.

tag format은 다음과 같다.

16 bits	3b	1b	12 bits
TPID	TCI	TCI	TCI
TPID	PCP	DEI	VID

- Tag protocol ID(TPID):
0x8100. 고정 값이다. 802.1Q tag frame이라는 걸 알린다.
- Tag control identifier(TCI):
 - priority code point(PCP)가 있어서 우선순위 레벨을 정함. 0~7
 - Drop Eligible Indicator (DEI): congestion이 발생했을 때, 해당 frame을 drop 할 수 있는지
 - VLAN identifier (VID): 특정 VLAN에 해당하는 id. 0 to 4094 (0이면 VLAN기능을 사용하지 않고, priority 값만 사용하겠다는 뜻이다.

이때, PCP val.를 꼭 VLAN에서 사용해야 하느냐? ㄴ ㄴ

이때, PCP val. 관련해서, TSN(Time sensitive networking)이라는 게 등장한다.

TSN: 802.1Q에서 usec단위까지 very low transmission latency를 보장할 수 있어야 한다. 아주아주 작은 전송 latency(지연)를 보장할 수 있어야 한다.

- switch간에 엄밀한 time sync.가 필요함.
- TSN에 대해서, sv하는 traffic class는 Scheduled Traffic (ST)(link를 time slot으로 나눠서, 각 slot을 베타적으로 flow-frame을 전송하는 일종의 conenction-에 할당한다.) , Audio-Video Bridging (AVB), Best effort(BE)(=Qos 보장이 없다. 가기만 하면 된다. 앞의 두개는 빠른 도달을 보장.) 이런 것들이 있다.
 - 이전까지는 Ethernet에서 BE로 가면 간거지~ 했는데, Qos를 보장하는 (sheduling 기능이 있는)ST와 같은 방식이 생겨났다.

요즘 TSN은 차량용 NW (전기차), plane(비행기 NW), 교통망, industry 4.0(I IoT - Industrial internet of things); 방대한 센서들로부터 수집한 data를 잘 가공, 분석해서 현장에서 적절한 생산계획을 위해 사용., 비슷한 개념의 cyber-physical sys.(CPS) 등등... 사용.

- 전송지연을 안정적으로 보장해준다.

CH.15 Wreless LANs

wired LANs과 wired WANs를 배웠었는데,
이번에는 wireless LANs를 배울 것이다.

두가지 타입을 배우는데,

1. IEEE 802.11 project = wireless Ethernet
2. a personal wireless LAN, Bluetooth (= personal area nw or PAN)

15.1 Introduction

무선으로 장비들끼리 connecting 해달라는 수요가 생김

- 15.1.1 Architectural Comparison

Medium

shared medium을 사용하고, sig.는 broadcasting.

무선이다보니까 모든 방향으로 근처 station는 받는다.

- station은 본인한테 온 거 아니면 버릴 것이다.

Hosts mobility

이동하면서 data받x

어느 지역에서 data를 받다가, 다른 곳에서 다시 받을 수 있다.

한 AP(access point)에서 받다가, 그 AP를 벗어나면 data를 못 받게 된다.
다른 AP로 가서 마저 data를 받아야 한다. LL addr.는 유지되는데, NW L
addr.가 AP따라 바뀌기 때문이다. break before make이다. ; 다른 곳에 연
결하기 전에 현재 연결을 끊는다. 반대는 make before break; 끊김없이 이
동하면서 데이터를 받을 수 있다.(=celluar 이동통신)

Isolated LANs

무선랜 노드들끼리 하나의 작은 NW를 형성한 것. 외부로 Internet과 연결되지 X.

ad hoc = 고정되지 않고, 잠시 만들어졌다 사라지는. 이란 뜻.

Connection to other networks

다른 network나 internetwork, 예를 들어 Internet에 연결하기 위해 access point(AP)를 사용할 수 있겠다.

AP란, 한쪽으론 무선 LAN, 다른 한 쪽으론 유선 LAN nw와 연결되는 device
를 말한다. 일반적으로 이 AP를 통해 인터넷과 연결.

wireless LAN은 스위치가 필요 없다. wireless LAN은 Infrastructure network라고 한다.

wired와 달리,
스위치 - 라우터- wired internet(wired LAN - router - wired internet)이
아닌,
Ap - 라우터 - Infrastructure.Infrastructure network - router -
Infrastructure)이다.

Infrastructure이란 단어를 쓰는 이유를 chatGPT에게:

Wired LANs와 wireless LANs는 네트워크 연결 방식에 있어서 차이가 있습니다.

Wired LAN은 유선으로 연결된 LAN으로, 컴퓨터나 장치들이 이더넷 케이블 등을 통해 직접적으로 라우터 또는 스위치에 연결됩니다. 이 때, 라우터는 인터넷에 연결하기 위한 게이트웨이 역할을 수행합니다. 따라서 wired LAN에서는 라우터를 통해 다른 wired 인터넷과 연결된다고 표현합니다.

반면에 wireless LAN은 무선으로 연결된 LAN으로, 컴퓨터나 장치들이 라우터를 통해 무선 신호를 주고받습니다. 이를 위해 라우터는 무선 인터페이스를 가지고 있어야 하며, 이 인터페이스를 통해 무선 클라이언트와 통신합니다. 이때 라우터는 인프라스트럭처 모드로 설정되어 있어야 합니다. 즉, 라우터가 인프라스트럭처 모드로 동작하면 무선 클라이언트들이 해당 라우터를 통해 인터넷에 접속할 수 있습니다. 따라서 wireless LAN에서는 라우터를 통해 인프라스트럭처와 연결된다고 표현하는 것입니다.

무선 네트워크에서 "인프라스트럭처"라는 용어를 사용하는 이유는, 무선 클라이언트들이 라우터와 같은 기반 구조를 통해 통신하고 인터넷에 접속하기 때문입니다. 이는 무선 네트워크가 별도의 중심 장비를 필요로 한다는 것을 의미하며, 이 중심 장비가 인프라스트럭처 역할을 수행하기 때문에 해당 용어가 사용됩니다.

Infrastructure: 사회[공공] 기반 시설.

무선 네트워크에서 "인프라스트럭처(Infrastructure)"라는 용어는 무선 통신에 사용되는 기반 구조를 의미합니다. 무선 네트워크는 기지국, 라우터, 스위치 등과 같은 중앙 장비를 필요로 합니다. 이러한 장비들은 무선 클라이언트들과 통신을 담당하고, 무선 신호를 전달하며, 네트워크 연결을 관리합니다.

인프라스트럭처 모드로 동작하는 라우터는 무선 클라이언트들과 통신하기 위해 무선 신호를 발생시키고 수신합니다. 이를 통해 클라이언트들은 라우터와의 통신을 통해 인터넷에 접속하거나 내부 네트워크의 다른 장치들과 통신할 수 있습니다.

무선 네트워크에서 "인프라스트럭처"라는 용어를 사용하는 이유는 다음과 같습니다:

1. 중앙 제어: 인프라스트럭처 모드에서는 중앙에 위치한 라우터나 기지국이 무선 클라이언트들을 관리하고 통제합니다. 이러한 중앙 제어 구조는 효율적인 네트워크 관리와 보안을 가능하게 합니다.
2. 신호 전달: 인프라스트럭처 모드에서는 라우터나 기지국이 무선 신호를 발생시켜 클라이언트들에게 전달합니다. 이는 무선 클라이언트들이 특정 기지국이나 라우터와의 통신을 수행하며, 신호를 받아들여 데이터를 전송하고 수신하는 방식을 가능하게 합니다.

3. 네트워크 연결: 인프라스트럭처 모드에서는 무선 클라이언트들이 라우터나 기지국과의 통신을 통해 네트워크에 연결됩니다. 이를 통해 클라이언트들은 네트워크 내의 다른 장치들과 통신하고 인터넷에 접속할 수 있습니다.

이러한 이유로, 무선 네트워크에서는 "인프라스트럭처"라는 용어를 사용하여 무선 클라이언트들이 중앙의 장비와 통신하며 네트워크에 연결되는 구조를 나타냅니다.

- 15.1.2 Characteristics

wireless link의 특징들:

1. attenuation
2. Interference
3. Multipath propagation results in sig. received at different phases and less recognizable.
4. Err. due to low sig.-to-noise ratio(SNR)

SNR이 낮은 이유 중 하나. sender가 보낸 sig.가 line of sight(LOS)처럼 직선으로 갈수도 있지만, non LOS (NLOS)로 전파될 수도 있다.
=> path마다 도달 시간이 다르다. 하나의 sig.가 여러 길을 거쳐오다보니, 서로 delay가 다르므로 delay spread가 발생. 지형마다, 거리마다 도달하는 time에 영향을 다르게 미친다..결국 sig. 왜곡이 발생한다.

다른 노드로부터 IF가 있다. 유선보다 심하다.

유선보다 심하다.

낮다. => err가 많이 발생할 환경.

<참고> delay spread:

https://www.sharetechnote.com/html/Handbook_Communication_DelaySpread.html

요런 문제를 해결하기 위해

- AP를 촘촘히 박아서 sender- receiver 사이 거리 줄이기
- 높은BW을 가지는 고주파 대역을 사용한다던지
- OFDM 방식을 사용

sig. duration이 길어서 인접 pulse와의 interference가 신호왜곡으로 이루 어질 가능성이 낮다.

- 15.1.3 Access Control

host가 어떻게 shared medium에 대해 접근권을 가지나, 에 대한 문제가 있다.

- CSMA/CD는 wireless에서 3가지 문제점이 있을 것이다.
 1. 초창기 무선 host의 power문제로 send/ receive가 동시에 할 수 없었다. (half-duplex mode에 그쳤다.)
 2. hidden station problem:
A가 B에게 보낼 때, C라는 노드도 동시에 B에게 보내면 collision이 발생. C가 A에게 도착했어야 collision을 감지할 텐데, 거리가 멀다면 C의 sig.가 A에게 도착했을 때의 energy가 매우 작아져 감지하기 어렵다. 유선에 비해 무선은 attenuation(감쇠)가 심하기 때문이다.
 3. statino 사이의 거리 문제(거리가 멀다면 power가 약해져요.)

15.2 IEEE 802.11 Project

802.11에서는 wireless LAN의 physical과 datalink L에 관한 표준을 만들었다. wireless Ethernet이라고도 부른다.

WiFi(= wireless fidelity)라고도 부른다.

- 15.2.1 Architecture

1. Basic service set(BSS):

2. Extended service set(ESS):

AP를 가진 BSS를 두 개 이상 연결한 것.

- 이때, BSS들의 연결을 distribution sys.라고 한다.
 - wired or wireless로 이루어진 nw이다.
- 더 나아가서, server or gateway를 통해 wireless가 아닌, 다른 nw와 이어진다. mobile node지만 움직이는 동안 통신이 끊기지 않는 걸 보장하지 않는다.
- 보장하면 cellular nw sys.
 - cellular는 기지국이라 함. 기지국들이 coverage가 중첩되게 설치. 음영지역 없애려고 촘촘히 설치. 비쌈.
- wireless LAN은 hot spot = data 접속이 가능한 어떤 spot에서만 data를 주고받을 수 있음. 벗어나면 또다른 spot으로 가야함 (음영지역이 존재. 혹은 AP coverage가 중첩돼있어도 다른 AP coverage로 가는 순간 결국 끊김).

wireless station들과 an AP(optional)로 이루어져 있다.

- without Ap => ad hoc architecture (지들끼리. => 어떤 노드를 거쳐 갈지 라우팅 프로토콜이 있어야 한다. 자세히는 배우지 않는다.)
- with AP => infrastructure BSS (AP와 통신.)

- 15.2.2 MAC Sublayer

IEEE 802.11에서 MAC Sublayer 두 개를 정의했다.

1. Distributed coordination function (DCF):

2. Point Coordination Function (PCF):

시간민감 데이터전송을 위한 프로토콜. contention-free polling access method

: AP가 어떤 station에 대해서 특정 시점에 전송을 하라고 polling을 해주는 방식. (contention에 밀리면 영원히 전송을 못할 수도 있으니까.)

- CSMA/CA 사용 프로토콜.

F 15.6: MAC Layers in IEEE 802.11 std.

physic. L(FHSS, DSSS, Infrared, DSSS, OFDM...)가 있고 LL가 있는데,

LL은 다시 LLC sublayer(IEEE 802.1)와 MAC sublayer가 있다.

MAC sublayer는 다시 PCF(contention-free)와 DCF(contention) 두 가지가 있다.

F 15.7:

DCF에서는 CSMA/CD를 사용.

1. src가 medium을 sensing 한다. 없으면? DIFS만큼 기다렸다가,
2. src가 des.에게 RTS를 보내고,
3. des는 SIFS만큼 기다렸다가 CTS를 src에게 보낸다.
4. src가 CTS를 받으면, data를 보낼 자격을 얻게된다.

hidden station들은 RTS를 발진 못했지만, des의 CTS를 받을 수 있기 때문에, NAV filter(timer)를 설정하게 된다.

그 후, des.는 ack를 보낼 것이다.

- 유선LAN에서는 station이 collision의 유무에 따라서 데이터전송 성공/실패 여부를 파악. 그 후 재전송.
- 무선LAN에서는 collision detection에 기반한 전송 성공여부를 파악할 수 없으므로, des에서 ack를 SIFS만큼 기다린 다음, (패킷 처리 시간 보장용) 보낸다.
=> 이 동안 다른 station이 SIFS동안 carrier sensing하면? RTS를 보내면?
=> 어떤 노드들은 DIFS만큼 기다렸다 보내게 한다. 즉, DIFS > SIFS이다.

PCF는 time-sensitive transmission에 사용된다.

PCF는 분산형 x, 중앙에 controller = AP가 있어서 AP가 어떤 station이 전송할지 알려준다. (=centralized)

- 기본적으로 PCF가 DCF보다 우선순위가 높다.

PIFS(PCF의 IFS)를 사용한다. DIFS(DCF의 IFS)보다는 짧다.

- DCF만을 사용하는 노드들은 전송할 기회를 얻지 못할 수도 있겠다.

때문에, 반복되는 한 주기마다 PCF구간(=contention-free구간), DCF구간을 따로 둔다.

- PCF 시작을 알리기 위해 beacon frame을 둔다.

- beacon frame에는 PCF의 길이가 들어있어서, PCF구간의 NAV를 만들 수 있게 된다. (끝나는 지점 = DCF구간시작을 알 수 있게 된다.)

한 주기마다 PC (point controller) = AP에서 poll frame을 보내게 되고, 받은 node가 medium을 쓰게 된다. 물론 PC는 그밖에 data를 받고, ACK를 주고 받는 역할도 한다.

PCF끝내고 DCF의 contention based가 시작된다.

Fragmentation :

packet이 너무 크면 작게 쪼개서 전송. => 무선은 err.이 높다. 작게 쪼개 보내면 err.가 발생해서 재전송시 좀 더 효율적이다. 이런 field가 포함된다.

Frame Format

F 15.9: Frame format of wireless LANs

2B	2B	6B	6B	6B	2B	6B	0 to 2312B	4B
FC	D	addr 1	addr 2	addr 3	SC	addr 4	Frame body	FCS

FC(Frame control)는 다시... F15.1참조.

Protocol version	Type	Subtype	To DS	From DS	More frag	Retry	Pwrmgmt	Moredata	WEP	...
2bits	2bits	4bits	1bit	1bit	1bit	1bit	1bit	1bit	1bit	...

1. To DS/ From DS는 FC안에서, Addr 1~4의 성격을 알려주는 flag.
2. SC는 fragment number = 어떤 frame을 쪼갰을 때, 각각의 쪼갠 fragment의 순서.
3. subtype은 RTS/CTS/ACK를 나타낸다.(ACK가 있다는 건, 재전송 가능이 있다는 뜻.)
1011:RTS, 1100:CTS, 1101:ACK.

이런 frame중에서도...

F 15.10: Control frames

RTS:

2B	2B	6B	6B	4B
FC	Duration	addr 1	addr 2	FCS

이 duration으로 NAV 설정.

CTS or ACK:

2B	2B	6B	4B
FC	Duration	addr 1	FCS

이런것보다 hidden station이 RTS는 못 받지만 CTS로 medium을 파악하는 과정이 더 중요하다! (앞에 배웠었다.)

- 15.2.3 Addressing Mechanism

바로 위에서 잠깐 언급했었는데, addr 1~4의 필드가 의미하는 것이 무엇인지.

FC field의 To DS와 From DS가 결정한다.(각 DS field는 0/1 값을 가짐.)

- Addr1: 실제 frame 받는 receiver
- Addr2: 실제 frame 전송하는 transmitter

문제는 src가 transmitter가 될 수도 있지만, frame이 AP를 거쳐서 des.로 전달이 된다면, 받을 때 receiver도 AP가 될 수도.

10 case는, src가 frame을 전송하는 경우, des.가 src를 관리하는 서비스하는 AP가 receiver가 될 것이다.

receiver는 receiving AP이고, transmitter는 src. 그러면서도 최종 des.의 주소는 포함시키기.(addr 3에다가 표기)

이 프레임이 nw를 죽 거쳐서 최종 des.에게 갈 때, 01 case는, des.를 서비스하는 AP가 frame을 des.에게 전달할 때는,

transmitter는 AP, receiver는 Des.. 그러면서도 src는 원래 src (addr 3에다가 표기)

case 00은 AP를 가치지 않는 경우. case 11은 두 AP(각 src와 des.를 sv하는 AP)가 wireless distribution sys.일 때.

다 무선 LAN이라 addr 필드를 우리가 적절히 채워줘야 한다.

F 15.11:

src가 보낼때에는, case 10으로 frame을 전송하게 된다.
AP가 받고 보니, des.관장 AP를 찾고, case 11로 전송.
마지막 순간에 des.에 전달되려면, case 01로 frame전송.

<참고> src가 자신을 관장하는 AP에게 frame을 전송할 때:
어떤 AP가 이 frame을 받아처리해야하는지 알려줘야함.

주변에 AP여러개있을수도. 스위치라면, 스위치와 station은 physical link로
이어져있어서
흔동할 리가 없지만, AP는 wireless로 이어진 여러개가 있을 수 있기 때문이다.

Exposed Station Problem

1. A가 B에게 전송하고픔.
2. A는 B에게 RTS를 보냄.
3. C는 RTS받음.
4. B는 CTS를 보냄. C는 NAV filter set.
5. A가 data를 B로 보내는 동안, C도 D로보낼 data가 있음.
6. C가 D로 보내는 건 A가 B로 보내는 것과 동시에 가능함.
7. But A가 보냈던 RTS로 인해 C는 보내지 못함.

이런 문제가 있다.

- 15.2.4 Physical Layer

기본적으로 industrial, scientific, and medical한 (ISM)한 band (대역폭)을 사용해서 외부로부터 interference가 있을 수도 있다.
크게 3가지 대역폭을 가진다.

15.3 Bluthooth

wireless LAN이고, 짧은 거리에 전송을 하는 경우에 쓴다.

ad hoc network를 구성한다. 그래서 다른 유선 인터넷 망으로 연결하는 용도로 사용하지 않는다.

gadget이라부르는 device들이 모여서, piconet이라 불리는 nw를 이룬다.

IEEE 802.15에서는 wireless personal area network (WPAN)이라는 관련 표준을 만들었다.

- 15.3.1 Architecture

Bluetooth는 2가지 타입의 nw가 있다.

1. piconet:

2. scatternet:

최대 8개의 station. 하나를 primary, 나머지를 secondaries.
항상 primary를 통해 통신이 이루어진다.

piconet들을 합쳐서 형성한다.

- 한 piconet의 secondary가 다른 piconet의 primary로.
이런 식으로 확장돼서 scatternet을 이룰 수 있다.

- 15.3.2 Bluetooth Layers

기존의 Internet model과는 다르다.

Physical L = radio layer,

Link L= baseband layer(=MAC sublayer) & L2CAP layer(logical link control and adaptation protocol = LLC)

그런데 계층 구조(F 15.19)를 보면 APP layer가 바로 존재한다. 바로 app layer에서 데이터를 내려 받으면, NW나 transport L을 거치지 않고, L2CAP Layer부터 내려온다.

F 15.21: single-secondary communication

bluetooth의 MAC 구조는 TDD(time-division duplex)-TDMA ; half-duplex 구조를 사용한다.

- TDMA: 625us slot으로 나눠 사용한다.
- TDD: primary->second. second-> prim.

FDD는 HW가 복잡해져... bluetooth는 데이터도 그렇게 안 많고, HW도 간단 해야 해서 TDD사용.

- f_0 는 이 time slot동안 사용하는 freq.를 말한다.
- 다음 slot에는 다른 freq.를 사용한다.

secondary가 여러개면 multiple-secondary communication. secondary 마다 다른 slot을 할당한다. 실제는 slot전체를 안쓰고 data control등을 위해 남겨둔다.

F 15.23: Frame format types

72bits 54bits 0toNbits

Access code Header payload

최대 5slot을 모아서 한 frame을 만들수도 있다. => slot이 많아질수록 payload가 늘어난다. (1 slot-> 240bits, 3-> 1490bits, 5-> 2740bits)

256us of 625us는 hopping & control을 위해 사용하지 않는다. 이걸 이용해서 N bits 계산 가능. 3slot이면 3(625 - 256)

Header에 adr 3bits, type 4bits외에.

1. F: flow control
 2. A: stop-and wait
 3. S: seq. number (1/0)
 4. HEC: header err. correction
-

CH.16 Other Wireless Networks

교수님께서 16.2 cellular telephony 위주로 강의한다고 하셨다.

16.2 Cellular Telephony

이동하는 노드들(단말들)에 대해서, 이동하는 도중에도 끊김없이 음성통화 서비스를 제공하는 nw sv.로부터 시작했다.

이동하는 단말들을 mobile station (MS)라고 부른다.

이동성관리라는 것이 중요한데, 전화를 건 사람이 현재 어디에 있는가?

여기서 어디라는 건, cellular는 기지국(Base station)으로 이루어져 있는데, 이 기지국(BS)이 관리하는 영역을 cell이라고 부름.

- 이 cell의 boundary안에 위치해야 (전파가 미치는 지역) 모바일 스테이션이 sv.를 제공받을 수 있다.
- 기지국을 촘촘히, 전파세기를, 기지국 위치를 잘 조절해 음영지역을 없애야 한다.
- MS(단말)이 이동하면, 특정 기지국 cell에서 다른 기지국 cell로 이동할 것이다. 이걸 실시간으로 파악 (= locate and track a caller)해야 할 것이다.
- 음성 sig.를 한 기지국에다 보내다가, 다른 기지국 cell로 이동하게 되면, 현재 단말이 다시 이 다른 기지국으로 sig.를 보내야 한다.

AP는 실시간으로 끊김없이 (wireless LAN은 data 접속 sv를 위해 만들어진 sys라 음성용은 아니었음.) 불가능하다. 음영지역 없이 sv해준다는 개념도 아니었다.

F 16.6: cellular sys.를 보면,

같은 cell안의 MS끼리 통화하는 경우도 있겠지만, 외부의 유선전화망(Public switched telephone nw- PSTN)에 있는 stationary phone(유선전화망의 단말)과 통화하는 경우도 있겠다.

이때, 다른 nw와 연결하는 것이므로, gateway가 필요할 것이다.

- 여기선 Mobile Switching Center(MSC); connecting calls, recording call info. billing을 이용해서 PSTN과 이어줬다.
 - cellular nw의 음성신호 format이 있을 거고, PSTN의 format이 있을 건데, format을 중간에서 변환, 그리고 PSTN으로 보내는 것도 처리할 것이다.

- 16.2.1 Operation

Freq.- reuse principle :

freq. 갯수를 reuse factor라고 한다. $f_1 \sim f_3$ 쓰면 reuse factor of 3.

1. cellular nw sv. 사업자가 사용할 수 있는 대역들(carrier freq.)이 f_1, f_2, f_3 , 이렇게 있다.
2. 어떤 cell에서 f_1 을 사용. 바로 인접한 cell에서 동일 freq.
3. interference(간섭) 발생! MS1과 MS2에게 동일하게 f_1 을 통해 보낸 것. SNR이 낮아져 err.발생

cellular nw의 physical 성능은, interference를 얼마나 줄일 수 있느냐가 관건이다.

- 따라서, carrier를 재사용하되, 멀리 떨어진 cell에서 f_1 을 재사용하자! BW가 한정돼 있으니, carrier(곧, 대역폭을 말한다.)를 무작정 늘릴 수는 없기 때문이다.

Transmitting :

단말은 자기와 가장 가까운 (=신호 세기가 큰) BS를 찾고, connecting해야 할 것이다.

Receiving :

paging: 단말이 있으면, 다른 쪽에 있는 단말에서 송신하는 목적지가 될 수도 있다.

목적지 위치를 알기전에 받을 단말이 있는 cell을 관장하는 기지국이 어디있는지를 알아야 한다.

- 기지국이 본인 cell에서 des.인 단말을 찾아 신호를 보내는 걸 paging이라고 한다. 항상 sys는 단말이 어느 기지국 영역에 있는지 추적을 해야하고, 그 위치를 기록하는 DB를 가지고 있어야 한다. (= register server)

무선LAN과의 차이점이다. 무선 LAN에서는 단말이 다른 AP영역에 있는 단말에 전송을 하고자 할 때와 달리, 단말로 data를 전송하고자 할 때! 이 단말이 어느 AP영역에 있는지 알아야 하는데, 무선 LAN에서는 그런 sv를 제공해주지 않는다.

계속 같은 말인데,

단말이 data 전송을 시작하는 경우와, 반대로 단말에게 data전송이 되는 경우(단말을 찾아서 보내줘야함)는 sv의/ sys 구축의 난이도가 있다.

단말의 고정된 id가 필요하다. 음성통화에선 고정된 id(전화번호?)가 있지만, data 통신에선 IP addr.가 있어야 한다. 하지만, mobile 단말에는 유동 IP (IP 고정은 갯수에 한계가 있어서....)를 할당하고, 이때부터 단말들은 인터넷 접속이 가능해진다. 반대로 이 단말에 접속하려고 하면, 상대편은 알 수가 없다. (고정 IP가 아니기 때문.) cellular와 무선 LAN의 중요한 차이점 중 하나가 이런, 단말을 목표로 하는 paging이 가능하느냐? cellular는 매번 단말을 추적 해주는 기능이 필요하다.

Handoff (Handover) :

동일 cellular nw에 여러 기지국들이 존재할 때, 이 기지국들 간에서, 다른 cell로 넘어갈 때 끊기지 않게 서비스 제공해 주는 것.

한번에 여러 기지국과 통신가능토록 해야 하겠다.

1. hard handoff

기존 기지국과의 연결을 끊고 바로 다음 기지국과의 연결 (break-before-make)

2. soft handoff 겹치는 domain에서 동시에 받고 있다가 좀 더 이동해서 이전 기지국과의 연결을 끊음 (make-before-break)

Roaming :

서비스 provider는 여러개 (한국, 외국1)

한국 kt망으로 통신을 하다가 미국으로 갔다. 미국에 있는 서비스 provider(사업자)에 연결해서 써야함.

- 즉, 다른 사업자에서 오는 단말들도 sv를 제공해주는 것.

• 16.2.2 First Generation (1G)

cellular nw는 1 generation부터 현재는 6g까지..?

진화과정을 간략히 알아보자.

voice commu.를 위해 개발됐고, analog sig.를 이용한다. (carrier에 실어서 보내는 sys.)

대표적인 sys이 북미에서 쓴 AMPS (Advance Mobile Phone Sys.)라는 시스템이었다.

- 유선망과도 통화가 가능했다.
- 800MHz band를 사용하고, 2개의 분리된 채널을 사용해서 forward(BS->MS), reverse(MS->BS)를 동시 통화가 가능하게. (FDD; full duplex.)
- FDMA(freq.를 여러 sub carrier 대역으로 나눔. 아까 봤던 f_1, f_2, f_3 , FM(아날로그 전송을 했다.) for voice, FSK(비트를 다루니까.) for control

F 16.8. F 16.9 설명: FM을 쓰다보니까, 3kHz를 30kHz의 대역폭으로까지 늘려서 보냈다.

- 16.2.3 Second Generation (2G)

좀 더 좋은 음질의 voice commu.를 제공한다.

anal -> digitized voice.

digital 전송을 위해 TDMA를 사용.

- PCM을 이용, 음성 -> 디지털. 그 후 compression(압축)
- err. correction도 가능하다. 추가 비트(redendancy)를 붙여서! 1G와의 차이: anal -> digit.으로 용량이 증가했다!

1. AMPS와의 호환이 되는 D-AMPS

TDMA 사용.

3가지 major sys.가 있다. 2. 유럽에서 썼다는 GSM 3. IS-95; CDMA를 이용, 우리나라 쪽에서 퀄컴과 같이 개발.

- GPS 이용한 sync
- CDMA 방식에서는 soft handoff가 가능.
 - 넓은 대역폭을 동시에 사용하기 때문에, freq. reuse가 없다. 대신, 다른 단말마다 다른 orthogonal code를 할당해준다. 이걸 곱하면 여러 신호가 오는 하나의 대역폭으로부터 자기한테 오는 신호만 filtering할 수 있다.
 - 인접한 두 개의 기지국은 같은 freq. 대역을 단말로 보낼 수 있다. 단 말은 하나의 carrier 신호만 받음. carrier freq.에 맞춰 동작해야 하기 때문에, 동시에 다른 freq.를 받게 하다보면 복잡하고, 비용도 커진다.
- 같은 단말에 대한 동일한 freq.를 전송.
- TDMA는 time 축에서 time slot을 나눠 단말마다 할당. 두 개의 기지국에서 전송할 때는 같이 할당하는 time slot을 갖게 한다. 전송하

는 시간을 맞춰주기 때문에 어려워서 hard handoff를 쓴다.

F 16.14 : IS-95 forward transmission에서 sync. using pilot sig.

1. digitized
2. err. correcitng이나 repeating, interleaving을 위해 size가 커지고,
3. 채널 구분을 위한 orthogonal code를 곱해주게 됨.
4. 곱해준 애들을 모은다. BW가 더욱 넓어나게 된다.
5. 하나의 채널 안에 여러 voice들이 포함된 구조이다.
6. 수신 측에서는 orthogonal code를 곱해서 자기 채널의 음성신호만 끄집어낼 수 있다.

이 과정들을 baseband 처리(합하기까지)와 radio 전송에 대한 부분으로 나눠 볼 수 있다.

- 16.2.4 Third Generation (3G)

기존의 voice 통신외에, digital data부분이 강화됐다.

최대 2Mbps까지 data 전송이 가능해졌다.

- 무선 주파수 중에서 좀 더 넓은 대역폭 2GHz 대역을 사용하게 됐다.

표준 규정을 ITU(International Telecommunication Union)에서 정의했다.

IMT-2000 이라는 규약.

(IS-95; us, korea)에서 쓰던 CDMA를 다른 국가들에서도 쓰게됨. GSM같은 경우도, physical L에서는 CDMA를 사용하는 방향으로 발전.(WCDMA; 5MHz BW)

CDMA2000 v.s. WCDMA

결과적으로 WCDMA가 우세하게 됐다. CDMA2000은 사장....

data BW를 키웠으니 전송 data rate가 올라가게 되겠죠?

voice전송은 실시간성.(재전송이란 게 없다.)

--> data는 delay있어도 됨 = 재전송이 있다. = HARQ (hybrid automatic repeat request)

여기서 중요한 점!

일반적으로 LL에서는 CRC로 err. check후 동일 패킷을 또 보낸다. 또 CRC 체크하고... err없으면 받아들인다. 재전송하는 패킷들은 서로 누적이 안된다. 독립적이라, 앞에 몇번을 받았던지 간에, last frame의 err 체크에는 아무런 영향을 끼치지 못한다.

하지만, HARQ는 Physical Layer에서 일어나는 err check가 있다. receive는 이전에 받은 frame에 대한 sig. 정보를 가지고 있다. err가 나면 다시 한번 재전송 할 텐데, 두번째 보낸 frame에 대한 sig. 정보가 이전에 보낸 frame의 정보와 누적된다.

- receiver의 demodulation에 성공할 가능성이 높아진다. = receive에 성공할. err 검사 통과할 확률이 높아진다. 성능이 좋아진다.
 - delay와 digital 전송이 가능한 새로운 전송이 새롭게 도입되면서 HARQ라는 새로운 기술이 들어오게 됐다.

UMTS nw를 대략 살펴보면,

크게 User Equipment - Access NW - Core NW이고,

User Equipment

U-sim - mobile phone

Access NW

node라는 BS와 애네를 컨트롤 하는 RNC(Radio NW control).
RNC를 통해 데이터가 각각 node들로 넘어간다.

Core NW

외부 NW와 이어준다. PSTN같은 유선망이나 Internet과 이어져있다.

- SGSN: 데이터 스위칭 역할(노드로 보내줌)
- GGSN : 외부 NW와의 게이트웨이 역할.
- GMSC: PSTN과의 접속을 위한 gateway.
- HLR: HOMe location register; 현재 이 단말이 어디 위치에 있는지 기록하는 DB.
- VLR: visitor locationregister;
- AuC: 과금과 인증 기능을 하는 server.

중요한 건, Cellular NW는 access NW와 Core NW로 분리가 된다. SSGN과 GGSN: 단말이 이동하면 서비스하는 노드(기지국)들의 위치가 달라진다.

--> 서로 다른 기지국으로 데이터를 내려보내주는 역할 & 데이터를 기지국쪽으로 스위칭해 주는 역할

- (SGSN)은 데이터를.
- (MSC)는 voice sig.를.

음성신호만 서비스할때는 (2G까지는), MSC , GMSC, PSTN부분만 있었을 것이다.
그러다 data전송, Internet 접속까지 하면서 SGSN ,GGSN 부분도 추가됐을 것이다.

- 16.2.5 Fourth Generation (4G) (LTE-A)

ITU-R에서 최대 1Gbps의 전송 Rate가 나와야 된다는 요구사항을 제정.

CDMA를 사용한다 치면, CDMA의 대역폭은 전송 rate보다 커야 한다. (spreading을 시켜야해서.) 2G에서 1.25MHz였으면 3G에서는 5MHz까지. CDMA를 사용하려면 너무 커진다.
Chipduration이 굉장히 짧다. (pulse 사이의 간격) 무선구간에선 multipass로 전송이 될 것이다. delay가 달라지기 때문에 여러 sig.로 도착하게 된다. 이 delay spread가 전송에서, transmitter에서 보낸 chip duration이 짧으면 짧을수록 둉개지는 게 심해질 것이다.

- 따라서 wireless LAN처럼 전송속도가 높아지다보니 OFDM을 사용한다.
- 대략 각 subcarrier별로 15kHz의 BW를 둔다. 하나의 bit의 duration이 길어진다고 볼 수 있겠다. multipass에 의한 sig. 품질 저하를 줄인다.
- 64-QAM사용
- smart antenna arrays for MIMO(multiple-input multiple-output): 기지국에서 전송하는데 사용하는 안테나가 여러개라는 뜻이다. receiver측에서도 여러개의 안테나로 수신한다.
 - 송신과 수신 안테나 사이마다 무선 channel이 존재할 것이다. 이 무선 channel들은 서로 sig. 전파 특성이 다를 것이다. (=correlation이 작다)
즉, diversity(원래뜻은 다양성)라는 걸 이용하는 것이다. (=무선전파의 특성이 다른 여러 무선 채널을 이용해 동일 시그널을 전파하면, 어떤 채널들은 환경에 영향을 받고있는 채널일 수 있지만, 또 다른 어떤 채널들은 sig.특성이 좋아서 잘 보낼 수 있을 것이다.) 피지컬 전송 성능을 높였다.
- Coordinated multipoint (CoMP) based on network MIMO: 아까 MIMO는 기지국에서 송신안테나 여러개 사용. nw mimo는 송신 안테나는 여러갠데, 여러 기지국에서 한 노드에게 데이터 sig.를 동시에 보내겠다. 즉, 기지국 A의 안테나 여러개, 기지국 B의 안테나 여러개에서 한 노드에게 sig.를. diversity가 좋은 성능을 보이려면 각 안테나에서 오는 무선 채널의 특징이 서로 독립적일수록 좋다. (유전적 다양성이 떠오른다....)
즉, 멀리 떨어진 다른 station에서도 전송해주면 이 diversity효과가 좋지 않을까? 하는 것이다. (space diversity); 기지국 간의 coordination이 필요해서 어렵다! <--> time diversity: 한번은 이 채널이 안 좋더라도, 시간이 흐른 뒤에 보내면 그때는 이 무선 채널의 품질이 좋을 가능성 이 크다. (HARQ의 기본 원리 - 여러번 보낼 수 있는 근거가 된다.)
- Inter-cell interference coordination(ICIC)는 어떤 단말간의 통신에 어떤 sub carrier을 쓰고 있다면, 이 subcarrier freq.는 다른 단말간 통신에 쓰지 말자. 역시, 기지국 간의 subcarrier 할당 (scheduling)이 필요하다.
- All-IP: 3G에서는 음성신호를 위한 circuit sw. nw와 data를 위한 packet switched nw.로 구분했었지만, (기존 2G망에 패킷 스위치 망도

합쳤던 것.) --> packet switched로 다 통합시켰다. packet switched는 IP기반으로 돌아간다. 따라서 보이스도 IP기반으로 돌아간다.

사실 voice는 QoS요구사항이 더욱 타이트할 것이다. 워낙 voice의 data 전송 rate가 data보다 낮고, data 전송 rate가 많이 올라갔기 때문에, 미미한 트래픽이라 이 QoS기준을 만족시키기 어렵지 않다. 따라서 사업자 입장에서 굳이 나눌 필요가 없다.

<참고> chip: [https://en.wikipedia.org/wiki/Chip_\(CDMA\)](https://en.wikipedia.org/wiki/Chip_(CDMA))

교수님 자꾸 예고에도 없는 용어 불쑥불쑥 꺼내시면 화가납니다. 15주동안 늘....그래왔어요 하핳..

F 16*: 4G nw

UE <-> eNB <-> SGW <-> PGW <-> PDN
<-> MME <-> HSS

SGW: serving gateway, 여러개의 기지국 간의 data를 스위칭. PGW: PDN gateway, 인터넷과 이어준다. PDN: packet data network = 인터넷
Internet MME: mobility management entity = 단말기가 이동하니까 어느 영역에 존재하는지 저장해놓는다. HSS: homesubscriber server eNB: 기지국

따로 음성/ 데이터 nw를 분리하지 않고, 하나의 core nw (SGW, PGW, MME, HSS 부분)가 존재한다.

F 16*: CoMP

SISO: single input, single output; 기지국 안테나 하나, 기지국 안테나 하나 MIMO: 단말 기도, 기지국도 안테나 여러개

전송 용량 늘리는 방법

- carrier Aggregation:

여러 기지국으로부터 subcarrier로 수신하는데, 기지국이 보낸 freq.가 달라서 단말기가 서로 다른 freq.로 오는 sig.를 수신(aggregation하면서- 합치면서 받음)

- CoMP:

여러 기지국에서 전송하는데, 동일한 carrier freq.로 전송. 일종의 MIMO이다. interference는 감소, sig. 수신파워 증가, throughput증가.

CoMP를 위해 기지국 간의 coordination이 중요하다.

기지국 하나하나의 기능이 많으면 coordination이 어렵다. 앞서 배웠을 때, baseband(physical) 처리와 radio(전송)에 대한 부분을 분리하는 걸 배웠다.

이렇게 분리해서, 공통된 부분은 한 곳에 보내고, 전송에 관한 부분은 기지국으로 내려 보내는(맡기는) 방식으로 nw가 진화하고 있다.

F 16*: ICIC

cell안 쪽에, 겹치는 부분도 아니기 때문에 low power로 전송해도 상관없고, 전송 파워가 적다보니 다른 cell로 갈리도 없어지고. 같은 freq. 써도 된다.
<--> edge쪽에 존재하면(겹), high power로 보내야 한다.
이쪽 구역에선 서로 high power로 보낼테니까 interference가 심하다. 그래서 같은 freq. 를 쓰면 안된다.

- 16.2.6* Fifth Generation (5G)

방금 깨달았는데, 별표(*)가 붙은 건 교수님이 교재외에 따로 추가하신 내용이었다.

IMT-2020 requirements

- max. 20Gbps
- latency 1ms

IMT-2020 use cases

- eMBB(enganced Mobile Broadband) : virtual augmented reality; 같은 방식인데 전송 rate를 높여줬다.
- URLLC (ultra-reliable and low latency commu.): 차량(vehicle-to-X) 제어 관련 통신, 스마트 공장, 드론 등등에서는 실시간제어를 위해 latency가 낮아야.
- mMTC (massive Machine Type Comm.) : environmental monitoring, logistics(유통, 물류), smart grid(실시간 검침 - 전력 발전 변화 등), e-health같이 방대한 센서에서 오는 정보를 제어.

eMBB는 기존에 있던거 고도 발달. URLLC나 mMTC는 상당히 새로운 요구사항이다.

5G에서는 NR(new radio) 특성을 새롭게 규정했다.

1. 사용하는 전송 대역폭을 sub 6GHz --> 25GHz 이상의 mmWave spectrum을 사용 하자.
 2. massive MIMO - 안테나 갯수를 훨씬 키우자. 8개보다 훨씬 많이. beamforming: 안테나 전송파를 단말 방향으로 모아서 보낸다. 즉 interference를 줄인다.
 3. flexible subcarrier spacing: 이전에 4G에서는 15kHz, 1ms slot을 고정적으로 가졌다. 5G로 와서, 하나의 subcarrier와 인접한 다른 subcarrier사이의 freq.차이를 15kHz --> 240kHz까지 늘릴 수 있었다. 더 넓은 subcarrier의 BW를 가지게 됐다. slot의 길이도 1ms --> 62.5us로 줄었다. slot단위로 scheduling이 일어나니까, 그만큼 더 짧은 latency가 가능해졌다.
 4. scalable transmission time interval (TTI): TTI는 1slot을 미니 slot들로 더 나눠서 전송할 수 있다. URLLC를 위한 전송 방식이다.
- URLLC처럼 delay 요구사항이 타이트해지면, HARQ를 사용할 수 없게 되므로, flexible subcarrier spacing을 이용한 짧은 slot 전송을 사용하

게 된다.

- 전송의 err.율을 줄이려면 comp같은 것들 이용.

F 16*: 5G frame structure

carrier BW를 여러개의 subcarrier로 나눠 사용한다.

subcarrier별, time slot별 resource block이 (RB) 존재한다.

- slot당 14개 symbol을 포함
- 12개 subcarrier를 모아서 1개의 RB만듦.

애네를 각 단말별로 allocation, scheduling을 한다.

필요한 전송rate따라서 할당되는 RB도 달라진다. 가변적 scheduling이다.

5G에서 throughput(품질의 일종)을 올리기 위해 여러 개념을 사용한다.

UDN(Ultra-Dense NW) & HetNet(Heterogeneous NW)

여기서 말하는 nw는 기지국. 기지국을 촘촘히 박아서 한 기지국 관할 cell을 줄여서 (coverage 줄이기) throughput올림.

- 그런데, 너무 작은 coverage 기지국만 사용하면, handover 등의 문제점 발생. 큰 coverage도 사용한다.(= heterogeneous NW; 크기가 크고작고)
 - 핸드오버(hand-over)는 단말기가 연결된 기지국의 서비스 공간에서 다른 기지국의 서비스 공간으로 이동할 때, 단말기가 다른 기지국의 서비스 공간에 할당한 통화 채널에 동조하여 서비스가 연결되는 기능을 일컫는다.

cell-free

기지국 하나가 관장하는 cell coverage.

coverage안의 단말은 association돼서, 전송 sv를 받는다.

여러 기지국으로부터 여러 sv를 받는다면 도대체 관장하는 cell은 뭐냐?

- 여러 AP(기지국)으로부터 physical 접속을 해서 데이터를 받는다는 개념이다.

C-RAN(centralized Radio Access NW)

앞에서 physical 전송처리를 할 때, baseband영역에서의 처리부분이 있고, carrier를 따라 antenna에서 전송하는 부분이 있다.

이렇게 나눈 것이다.

- BBU (baseband unit)
- RRU (remote radio unit)

일종의 repeater로 생각할 수 있다. 음영지역만 있으면 거기다 rru만 (= 안테나로 전송하는 부분만) 꽂으면 된다.

BBU하나에 대해서 (Centralized) RRU여러개를 연결해 사용할 수 있다.

구조화에서 오는, 제어하기 편해지는 이점을 얘기하고 싶은 것 같다.

- fronthaul(BBT-RRU)
bbu와 실제 전송하는 안테나 사이의 연결을 fronthaul이라고 한다.
- midhaul
- backhaul(BBU-CN)
BBU와 core nw사이의 연결을 backhaul이라 한다.

bbu가 있으면, 뒤로 core NW가 있을 것이다. 이 CN로 외부망과 연결될 것이다.

- Machine-learnign technologies for nw design, operation & management
점차 이런 디자인, 작동과 관리에 머신러닝을 사용하고 있다고 한다.
- TSN core network/backhaul for URLLC
URLLC 얘기하면서, 무선 구간에서의 latency를 줄이자는 얘기를 했었다. (subcarrier spacing을 늘림으로써 전송단위를 줄이는 등...)
사실 latency는 end-to-end latency이다.

무선 부분과 유선부분이 있으므로 두 부분을 고려하지 않을 수 없다.

- 유선부분 Qos제공은 TSN이다. --> core nw와 backhaul부분은 TSN사용.
- 무선구간에는 URLLC를 위한, flexible subcarrier spacing과 같은 방법을 사용한다.

CoMP vs HARQ reduction for URLLC

URLLC를 위해선 HARQ는 latency증가 문제로 사용하기 어렵다. CoMP를 사용한다.

F 16*: backhaul/ fronthaul

smallcell과 Macrocell Site가 있는데, small cell이 혼재돼있는 nw를 HetNet/(또는) UDN이라고 부른다. nw controller가 외부 인터넷망과 연결되는 core nw부분이다. 이것과 DU=BBU를 연결하는 부분이 backhaul, RU=RRU와 DU를 연결하는 부분을 fronthaul이라고 한다.

16* Sixth Generation(6G)

확실히 define은 안됐지만, 얘기가 나오고 있다.

- teraHz까지 높이자.
- blockchain과 연결하자.
- 모든 것들이 AI 사용을 높이자.
- satellite NW

끝!

In []:

This notebook was converted with convert.ploomber.io