

데이터통신과네트워크(주경호)

Lee Jun Hyeok (wnsx0000@gmail.com)

December 17, 2024

목차

I	<u>서론</u>	3
1	서론	3
1.1	Data Communication	3
1.2	Network	4
1.3	Network Types	6
2	Network Protocols	8
2.1	Protocol	8
II	<u>Physical Layer</u>	10
1	서론	11
1.1	서론	11
1.2	Analog Signal	11
1.3	Digital Signal	13
1.4	Impairment	15
1.5	Data rate	17
2	Digital Transmission	18
2.1	Digital to Digital : Line Coding	18
2.2	Digital to Digital : Block Coding	24
2.3	Digital to Digital : Scrambling	26
2.4	Analog(signal) to Digital(data) Conversion	26
2.5	Transmisison Mode	30
3	Analog Transmission	32
3.1	Digital(data) to Analog(signal) Conversion	32
3.2	Analog(signal) to Analog(signal) Conversion	36
4	Bandwidth Utilization	37
4.1	Multiplexing	37
4.2	Multiplexing 기법들	38
4.3	Spread Spectrum	41
4.4	Spread Specturm 기법들	41
5	Switching	43
5.1	Switching	43

III	<i>Data-link Layer</i>	46
1	서론	46
1.1	서론	46
1.2	Addressing	47
2	Error Detection and Correction	49
2.1	서론	49
2.2	Block Coding	50
2.3	Linear Block Code	51
2.4	기타 기법들	54
3	Data Link Control	56
3.1	서론	56
3.2	Protocols	58
3.3	기타 개념들	65
4	Media Access Control	66
4.1	서론	66
4.2	Random-access Protocols	66
4.3	Controlled-access Protocols	73
4.4	Channelization Protocols	74
5	Ethernet&wireless LAN	76
5.1	Ethernet	76
5.2	wireless LAN	80

Part I

서론

1. 서론

1.1. Data Communication

1.1.1. Data Communication

1. Data Communication

통신(Telecommunication)은 전화/정보/텔레비전 등을 포함하는 원거리 의사소통을 의미함.
데이터통신(Data communication)은 두 기기 간에 데이터를 주고받는 통신을 의미함.

data communication은 하드웨어(chipset, NIC chip 등)와 소프트웨어(소켓, 펌웨어 등)로 구현됨.
웹, wifi, bluetooth 등 data communication을 수행하는 다양한 시스템들이 있음.

2. 주요 특성들

data communication은 아래와 같이 4가지 주요 특성들을 가짐.

전달(delivery) : 시스템은 데이터를 올바른 목적지에 전달해야 함.

정확성(accuracy) : 시스템은 데이터를 변동 없이 정확하게 전달해야 함.

적시성(timeliness) : 시스템은 데이터를 고려된 시간 내에 전달해야 함.

파형 난조(jitter) : 시스템은 jitter가 최대한 일어나지 않도록 해야 함.

여기서 jitter는 패킷 도착 시간 지연에 따른 딜레이나 버벅거림을 의미함.

3. 구성 요소

data communication은 아래와 같이 5가지 구성 요소들을 가짐.

메시지(message) : 주고받는 데이터.

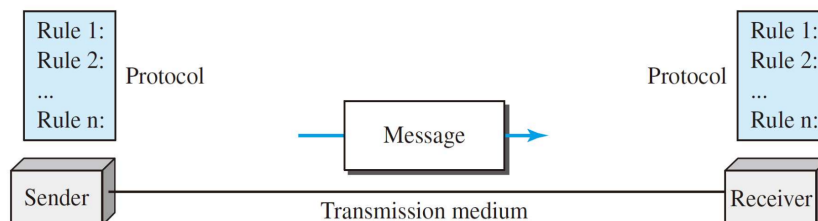
송신자(sender) : message를 보내는 사람.

수신자(receiver) : message를 받는 사람.

전송 매체(transmission medium) : 메시지가 이동하는 물리적인 경로. (cable, 공기)

프로토콜(protocol) : data communication을 제어하는 규약의 집합.

정보를 전달하는 물리적/논리적 경로를 통틀어 채널(channel)이라고도 함.



1.1.2. Data Representation

데이터 표현 방법(Data Representation)에는 아래와 같은 것들이 있음.

1. text

bit pattern의 집합으로 표현함. 이 집합을 code라고 하고, 텍스트를 나타내는 과정을 coding이라고 함.
code에는 Unicode, ASCII 등이 있음.

Unicode는 32bit로 pattern을 구성하고, 세상의 모든 언어를 표현할 수 있음. ASCII는 8bit로 pattern을 구성하고, Unicode의 첫 127개(로마자 등)에 해당하는 문자를 표현할 수 있음.

2. *number*

*number*는 이진수로 단순 변환함.

3. *image, video*

*image*는 픽셀(pixel)에 대한 행렬로 표현함. *image*의 색은 3가지 정보(RGB)로 표현함.

*pixel*은 디지털 *image*를 구성하는 가장 작은 단위로, 각 *pixel*은 하나의 색상을 나타냄. *image*의 해상도(resolution)는 *pixel*의 개수에 의해 결정됨.

*video*는 *image*의 집합으로 표현함.

4. *audio*

이산적으로 표현이 가능한 *text*, *number*, *image*와는 달리, 연속적인 데이터를 가지므로 별개의 표현 방식을 가짐.

1.1.3. Data Flow

데이터가 전송될 수 있는 방향에 따라 데이터 흐름(Data Flow)을 분류할 수 있음. 여기서의 *data flow*는 일대일 통신을 가정함.

1. *simplex*

단방향(*simplex*)는 단방향 통신을 의미함. 연결된 두 장치 중 하나에서는 송신만 가능하고, 다른 하나에서는 수신만 가능함.

예를 들어, 키보드, 단순 모니터 등이 있음.

2. *half-duplex*

반이중(*half-duplex*)는 각 장치에서 송/수신을 동시에 수행하지 못하는 양방향 통신을 의미함. 연결된 두 장치는 송/수신이 모두 가능하지만, 한쪽이 송신하고 있으면 다른 쪽은 수신만 할 수 있음.

예를 들어, 무전기, 개인 라디오(CD radio) 등이 있음.

3. *full-duplex*

전이중(*full-duplex*)는 각 장치에서 송/수신을 동시에 수행할 수 있는 양방향 통신을 의미함. 연결된 두 장치는 데이터를 송신하면서 수신받을 수 있음.

예를 들어, 전화 등이 있음.

1.2. Network

1.2.1. Network

1. *Network*

네트워크(Network)는 소통 가능한 기기들의 상호 연결 집합임. 각 기기들을 *host*가 될 수도 있고, *client*가 될 수도 있음.

2. *connecting device*

*network*에서는 *router*, *switch*, *modem* 등의 연결 장치(connecting device)이 사용됨.

라우터(*router*) : *network*를 다른 *network*와 연결하는 장치. *packet* 전달, 경로 선택, 보안 등을 처리함. 즉, *network layer*에서 *datagram*을 *decapsulation*하여 *ip* 주소를 확인하고 적절한 *network*로 *datagram*을 *encapsulation*해서 전달함. 이를 위해 내부적으로 *table*을 가지고 있고, 여러 작업을 수행함.

스위치(*switch*) : *network* 내부에서 *MAC* 주소를 기반으로 여러 장치를 연결하고 데이터를 전달(*forwarding*)하는 장치. 즉, *data link layer*에서 *frame*의 *MAC* 주소를 확인하여 적절한 위치(장치, *switch*, *router* 등)로 *frame*을 *encapsulation*해서 전달함. 이를 위해 내부적으로 *table*을 가지고 있고, 여러 작업을 수행함.

모뎀(*modem, modulator-demodulator*) : 데이터 형식(디지털-아날로그 등)을 변환하여 인터넷 연결 등을 수행하는 장치. *physical layer*에 해당됨.

3. *Network* 평가 기준

Network의 평가 기준에는 아래와 같은 것들이 있음.

성능(Performance) : 데이터 처리량과 지연 정도를 평가.

신뢰성(Reliability) : 실패 빈도, 실패 시 복구 시간, 재해 등에 대한 견고함 등으로 평가.

보안(Security) : 비인가된 접근, 데이터 보호, 데이터 손실에 대한 복구 정책/과정 등으로 평가.

1.2.2. Physical Connection

물리적인 연결(Physical Connection)에는 두 가지 종류가 있음. 이 연결 방식을 활용해 topology가 구성됨.

1. Point-to-point : 하나의 연결(link)에 두 기기만이 존재하는 경우. 즉, link가 두 기기를 위한 전용 link(dedicated link)인 경우.

2. Multipoint : 하나의 link를 3개 이상의 기기가 공유하는 경우.

1.2.3. Topology

접속 형태(Topology)는 network가 물리적으로 구성되는 방식을 말함. topology는 link들과 장치(노드)들로 구성됨.

아래와 같이 여러 topology가 존재함.

1. Mesh Topology

각 노드가 모든 다른 노드와 point-to-point로 연결되는 방식. $N(N-1)/2$ 개의 link를 가짐.

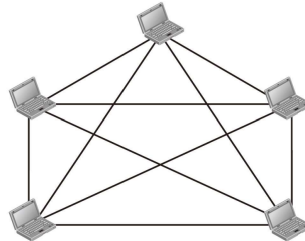
장점.

장치가 많아져도 트래픽에 대한 대처가 용이함. 안전성이 높고, fault 상황에 대한 대처가 쉬움.

단점.

비용이 많이 들고, 하드웨어 복잡도가 높음. 확장성이 안 좋음.

$n = 5$
10 links.



2. Star Topology

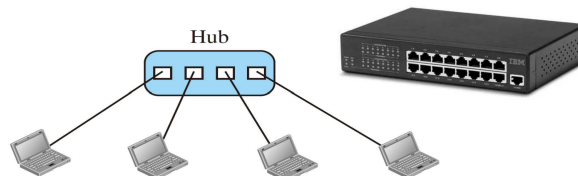
각 노드가 hub 또는 switch로 불리는 중앙 컨트롤 장치에 point-to-point로 연결되는 방식.

장점.

설치와 재설정이 간편함. 하나의 link에서 문제가 생겨도 다른 link에는 영향이 가지 않고, fault 상황에 대한 식별과 격리가 용이함. hub에 연결만 하면 되므로 확장성이 좋음.

단점.

topology 전체가 hub에 의존하고 있기 때문에 hub가 죽으면 전체 시스템이 죽음.



3. Bus Topology

하나의 cable이 모든 노드에 대한 main cable로 사용되는 방식. 각 노드들은 drop line과 tap에 의해 bus에 multipoint로 연결됨.

drop line은 장치와 main cable을 연결하는 link이고, tap은 케이블의 피복을 벗겨 drop line과 연결한 부분임.

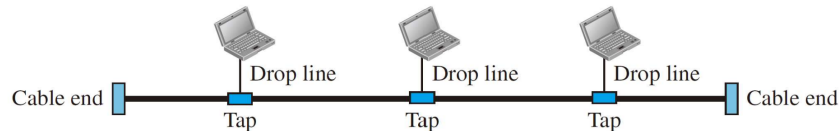
신호가 main cable을 따라 이동할 때 에너지가 손실되므로(열 등으로.), bus가 수용 가능한 tap의 개수와 tap 사이 거리에는 제한이 있음.

장점.

적은 케이블을 사용하여 설치가 간편함.

단점.

재설정과 fault 상황에 대한 격리가 어려움. bus가 끊어지면 전체 시스템이 죽을 수 있음. 케이블 길이 한계에 의해 확장성이 좋지 않음.



4. Ring Topology

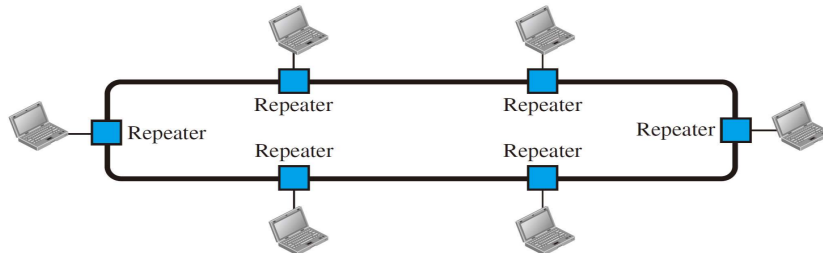
각 노드가 인접 노드와 point-to-point로 연결되는 방식. 신호/데이터가 경로의 노드를 거쳐 목적지에 도달함.

장점.

설치와 재설정이 간편함. ring 형태이므로 적은 데이터 충돌로 지속적인 data flow를 확보할 수 있음.

단점.

failure가 발생하거나 ring이 끊어지면 전체 시스템이 죽을 수 있음. 노드 추가 시 network를 멈춰야 할 수 있음.



케이블을 적게 쓰는 것이 유리한 자동차, 드론 등에서는 bus를 사용하기도 함.

1.3. Network Types

1.3.1. LAN

LAN(Local Area Network)은 가정, 사무실, 건물, 학교 단위의 network를 말함. LAN은 host들을 서로 연결함.

여러 방식으로 구현할 수 있음. 과거에는 bus topology로 구현하기도 했지만, 지금은 주로 star topology로 구현함. 물론 wifi 등 무선 LAN도 존재함.

1.3.2. WAN

1. WAN

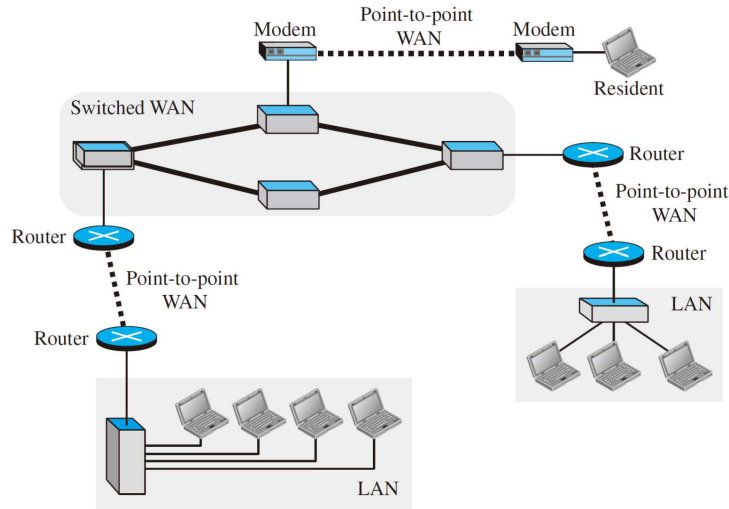
WAN(Wide Area Network)은 도시, 국가 단위의 network를 말함. WAN은 connecting device(router, switch, modem 등)들을 연결함.

WAN은 여러 방식으로 구현됨. connecting device를 활용하여 switched WAN을 구성하기도 하고, point-to-point WAN을 구성하기도 함.

2. internetwork

internetwork(internet) 2개 이상의 network가 연결된 것을 말함.

여기서의 internet은 Internet과는 구분되는 개념임.



3. Internet

인터넷(Internet)은 전세계의 장치들이 서로 연결되어 데이터를 주고받는 internetwork임.

Internet은 backbone, provider network, customer network, peering point 등으로 구성됨.

최상위 layer에 있는 backbone은 주로 거대 통신 회사(AT&T, NTT 등)들에 의해 운영되는 대규모 network임. 그 다음 layer에 있는 provider network는 backbone의 서비스를 하위 network에 제공하는 network임. backbone과 provider network를 통틀어 인터넷 서비스 제공자(ISP, Internet Service Provider)라고 부름.

customer network는 하위 layer으로서 Internet으로부터 제공받는 서비스를 이용하는 network임.

peering point는 두 대상을 연결하여 데이터를 직접적으로 교환할 수 있도록 하는 연결점임.

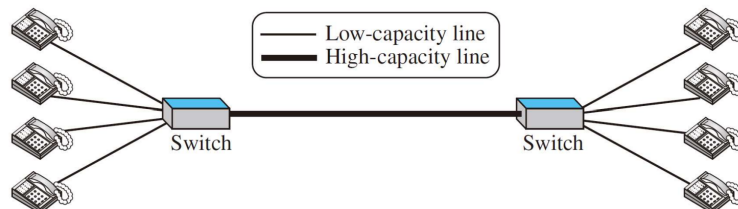
1.3.3. Switching

스위칭(Switching)은 데이터를 적절한 위치로 보내주는(forwarding하는) 작업을 의미함. network에는 switching의 방식에 따라 Circuit-Switched Network와 Packet-Switched Network가 있음.

1. Circuit-Switched Network

switch가 두 장치 사이의 서킷(circuit)이라는 전용 회선(dedicated connection)을 제공하는 방식. switch는 각 회선에 대한 활성화/비활성화를 수행함.

주로 과거 전화선 등에 사용됨. 현재의 전화는 packet-switched로도 처리가 가능함.

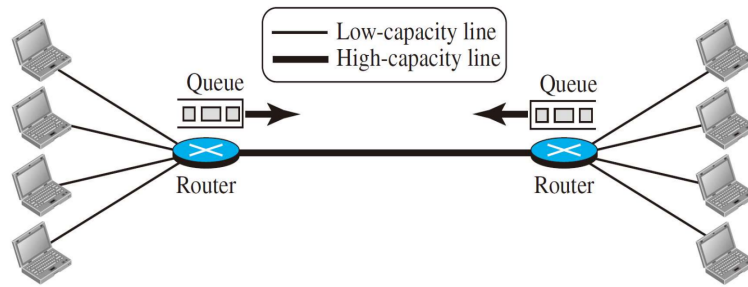


2. Packet-Switched Network

두 장치가 패킷(packet)이라는 단위로 데이터를 주고받는 방식. dedicated circuit을 사용한 연속적인 연결 대신, router를 사용하여 독립적인 packet의 교환으로 통신이 이루어짐.

packet-switched network는 packet을 저장하고 forwarding하기 위한 queue를 사용함.

packet switching은 디지털 데이터를 주고받음. 음성 데이터 등을 교환하기 위해서는 데이터를 인코딩/디코딩해야 함.



2. Network Protocols

2.1. Protocol

2.1.1. Protocol

1. Protocol

프로토콜(Protocol)은 효율적인 통신을 위해 수신/송신/연결 장치들이 지켜야 하는 규칙의 집합임.

2. Protocol Layering

복잡한 통신에 대한 protocol은 각 작업에 대한 계층화(Layering)를 통해 설계/구현/유지보수를 단순화할 수 있음. 이때 각 layer에서는 인접한 layer과의 상호작용만 고려하면 됨.

protocol layering에 의한 network model로는 TCP/IP, OSI Model 등이 있음.

3. principles of protocol layering

protocol layering 시에 각 layer가 지켜야 하는 원칙은 아래와 같음.

1. 각 layer는 작업과 그 반대 작업을 모두 수행할 수 있어야 함.
2. 통신 중인 두 장치에 대해, 동일한 layer에서 서로 대응되는 두 대상은 동일해야 함.

예를 들어, Encrypt/Decrypt layer는 encrypt/decrypt 작업을 모두 수행할 수 있어야 하고, 한쪽에서 encrypt해서 보낸 데이터를 decrypt하면 원래의 데이터와 동일해야 함.

2.1.2. Internet 주소 체계

Internet에서는 특정 장치를 식별하기 위해 아래와 같은 주소들을 사용함.

1. MAC 주소

MAC(Media Access Control) 주소는 네트워크의 각 장치가 가지고 있는 NIC(Network Interface Card)에 할당된 주소로, network 내에서 장치를 식별하기 위해 사용됨. 16진수로 이뤄져 있음.

이더넷(Ethernet)은 LAN에서의 MAC 주소 관련 처리에 대한 프로토콜임. ethernet을 사용하는 network를 ethernet network라고 함.

2. IP 주소

network 내에서 장치를 식별하기 위해 사용되는 주소. 길이에 따라 IPv4, IPv6가 있고, 사설 ip와 공인 ip 등의 분류가 존재함.

2.1.3. TCP/IP

1. TCP/IP

TCP/IP(Transmission Control Protocol/Internet Protocol)는 오늘날 Internet에서 사용되는 프로토콜 집합임.

TCP/IP는 아래와 같이 5개의 layer로 구성되어 있고, 각 layer는 나름의 프로토콜을 포함하고 있음.

Layer5. 응용(Application) layer : 사용자와의 상호작용을 처리하는 layer. (HTTP, HTTPS, DNS 등.)

Layer4. 전송(Transport) layer : port 선택, 데이터 reliability, 흐름 관리 등을 수행하는 layer. (TCP, UDP 등.)

Layer3. 네트워크(Network) layer : 송신자와 수신자의 연결을 생성하는 layer. (IP 등.)

Layer2. 데이터 링크(Data link) layer : 적절한 링크의 탐색을 수행하는 layer. (ethernet 등.)

Layer1. 물리(Physical) layer : 물리적인 연결을 통해 비트를 전송하는 layer.

위에서 표현한 5개의 layer는 OSI 7계층에서 5개의 layer를 가져와 설명한 것이고, 4개의 layer로 설명하기도 함. 이 경우 network layer를 Internet layer로 부르고, data link layer와 physical layer를 묶어서 네트워크 액세스 계층(Network Access layer)으로 부름. 물론 의미는 동일함.

2. 캡슐화/역캡슐화

송신자 측에서 데이터는 application~physical으로 흐르게 되는데, 이때 각 layer에 대한 헤더가 추가되며 캡슐화(encapsulation)됨. 이 헤더는 수신자 측에서 layer를 거치며 역캡슐화(decapsulation)됨. 이때 데이터는 encapsulation된 상태에 따라 다른 명칭으로 부름.

encapsulation에 의한 데이터 교환 단위는 아래와 같이 layer별로 존재함. 각각의 layer에서 캡슐화를 통해 나온 각 결과물을 패킷(packet)이라고도 부름.

메시지(message) : http의 경우 http 메시지. application layer에서 요청/응답에 http 헤더 등을 추가한 것. uri 경로, http 메서드 등을 포함함.

세그먼트(segment, TCP)/데이터그램(datagram, UDP) : transport layer에서 message에 transport layer 헤더를 추가한 것. port 번호, 시퀀스 번호, 흐름 제어 정보 등을 포함함.

데이터그램(datagram) : network layer에서 segment에 ip 헤더를 추가한 것. ip 주소 등을 포함함.

프레임(frame) : data link layer에서 ethernet 헤더를 추가한 것. MAC 주소 등을 포함함.

비트(bit) : physical layer에서는 frame을 비트로 변환하여 전송함.

3. Addressing

bit로 표현되므로 주소를 가질 수 없는 physical layer를 제외한 layer들은 출발 주소와 도착 주소를 가짐.

layer별로 가지는 대표적인 주소의 형식은 아래와 같음.

application layer : name.

transport layer : port 번호.

network layer : ip 주소.

data link layer : MAC 주소.

4. 데이터 전달 과정

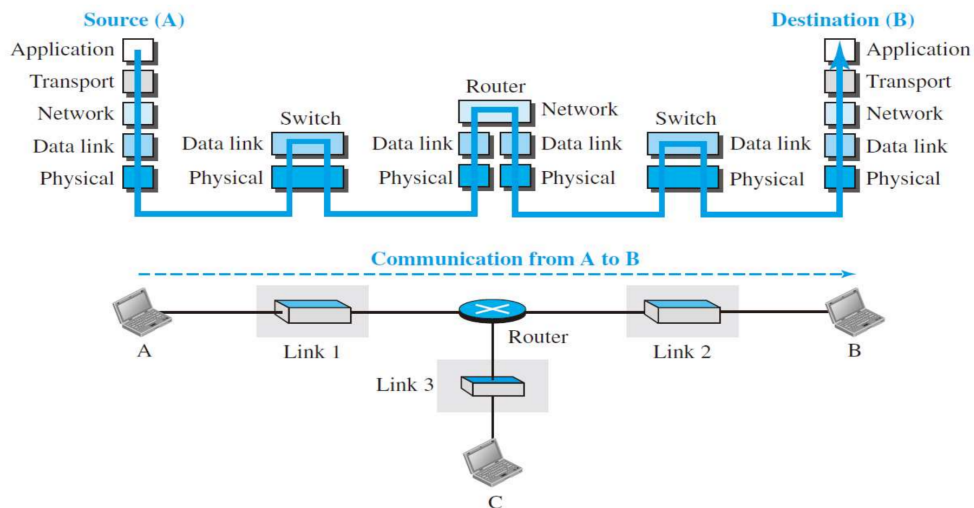
encapsulation된 데이터는 connecting device(switch, router 등)들을 거치며 목적지로 전달됨.

switch는 data link layer를 처리함. 즉, LAN에 대해 동작하고, MAC 주소를 확인하여 적절한 장치(switch, router, 목적지 등)에 데이터를 전송함.

router는 network layer를 처리함. 즉, WAN에 대해 동작하고, ip 주소를 확인하여 적절한 장치(switch, router, 목적지 등)에 데이터를 전송함.

당연히 이런 식으로 packet에 대해 decapsulation과 encapsulation을 처리하려면 MAC address와 ip address가 분리되어 있어야 함.

목적지를 발견했으면 데이터의 헤더가 layer 순서대로 사용/삭제되며 처리됨.



5. end-to-end/hop-to-hop

여기서 3, 4, 5 layer는 end-to-end를 보장해야 함. 즉, 3, 4, 5 layer에 대한 데이터는 중간 장치(hop, host와 router 등)의 개입 없이 송신자와 수신자끼리만 사용해야 함. hop에 의해 패킷이 수정되면 안 됨. 이때 본 layer는 Internet에 대해서 동작하는 것임.

1, 2 layer는 hop-to-hop을 보장해야 함. 즉, 1, 2 layer에 대한 데이터는 hop에서 사용해야 함. 이때 본 layer는 하드웨어적인 link에 대해서 동작하는 것임.

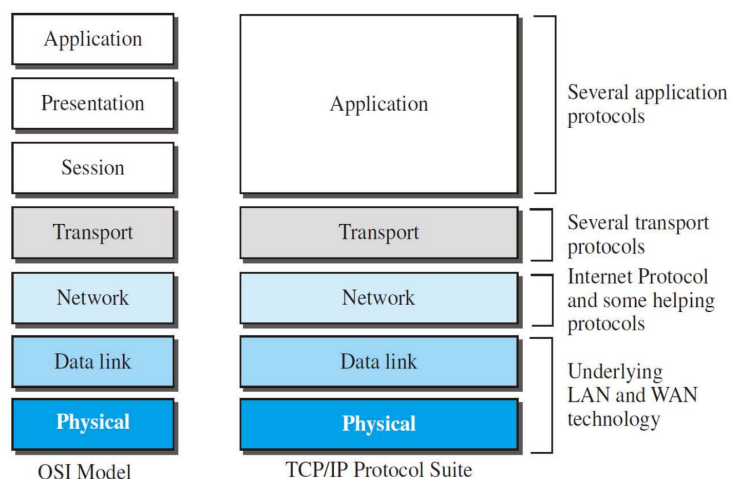
router는 data link layer 데이터(frame)에 대해 decapsulation을 수행하고, switch는 데이터를 수정하지 않음.

2.1.4. OSI Model

OSI(Open System Interconnection) 7계층 model은 ISO(International Organization for Standardization)에서 제정한 프로토콜 집합임.

ISO는 국제 표준을 지정하는 국제 기관임.

모든 통신과 장치에 각 layer를 구현하는 대신, 필요한 layer만 구현하기도 함. 또한 실제로는 TCP/IP 프로토콜을 주로 사용함.



Part II

Physical Layer

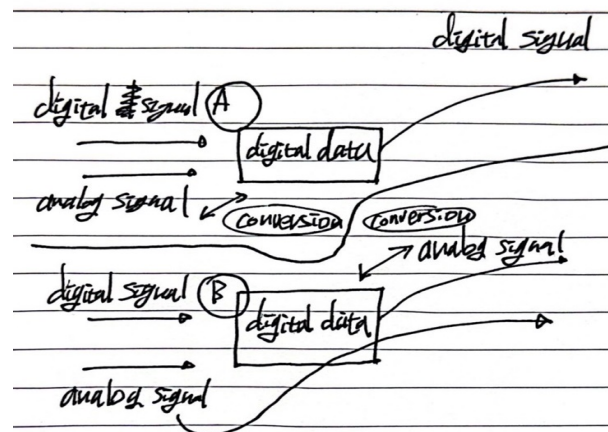
1. 서론

1.1. 서론

1.1.1. Physical Layer

물리 계층(Physical Layer)은 물리적인 연결을 통해 비트를 전송하는 계층임. data를 전자기 signal로 바꾸어 전송함.

physical layer에서의 통신은 signal의 교환임. data가 전달되기 위해서는 우선 signal로 변환되어야 함.



1.1.2. Analog vs. Digital

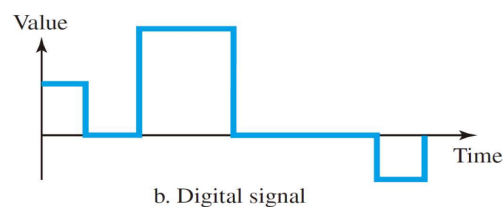
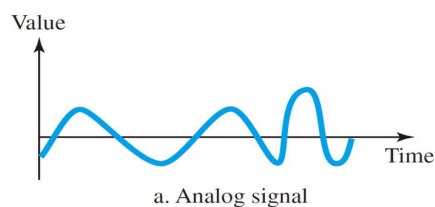
data와 signal 모두 아날로그(Analog)와 디지털(Digital)의 개념을 가짐.

1. analog/digital data

analog data는 연속적인 정보에 대한 data이고, digital data는 이산적인 정보에 대한 data임.

2. analog/digital signal

analog signal은 시간에 따른 signal 값이 연속적으로 무한히 존재하는 signal이고, digital signal은 정의된 몇 가지 이산적인 signal 값만을 가지는 signal임.



1.2. Analog Signal

1.2.1. Sine wave

사인파(sine wave)는 가장 기본적인 형태의 주기적인 아날로그 신호(Periodic Analog Signal)임. 푸리에

변환(Fourier Transform)에 의해 모든 Periodic analog signal은 sine wave의 조합으로 분해될 수 있음.

1. sine wave가 가지는 속성들

sine wave는 아래와 같은 속성들을 가짐.

1. 진폭(Peak Amplitude) : signal 값이 가장 강한 지점에서의 signal 값의 크기. signal이 가진 에너지의 크기와 비례함.
2. 주기(Period, T) : sine wave가 한 cycle을 도는 데에 걸리는 시간(s). Frequency와 역수 관계임.
3. 진동수(Frequency, f) : sine wave가 1초에 진동하는 횟수(Hz). Period와 역수 관계임.

$$T = \frac{1}{f}$$

4. 위상(Phase, Phase shift) : sine wave의 주기적 패턴에서 시간적으로 얼마나 앞 또는 뒤에 있는지를 나타내는 개념. 시간이 0인 지점에서 봤을 때 sine wave가 얼마나 shift되어 있는지를 radian 또는 degree로 나타냄.

5. 파장(Wavelength, λ) : sine wave가 한 번의 period에 이동하는 거리임. 즉, sine wave가 전송 매체를 통해 특정 시간 동안 전송될 때의 이동 거리(빠르기)를 나타내고, 아래와 같이 구할 수 있음.

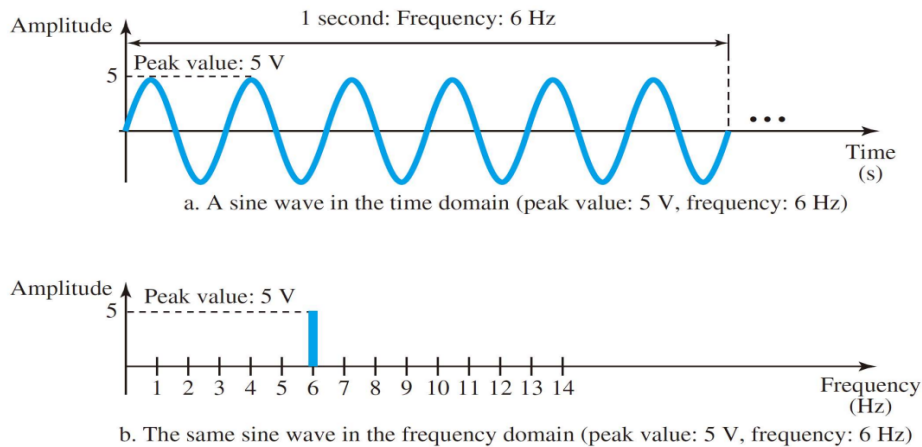
$$\lambda = \frac{c}{f}$$

2. Time/Frequency-domain

sine wave는 time-domain 또는 frequency-domain으로 표현함(plot으로 나타냄).

time-domain plot은 time-amplitude plot으로 시간에 따른 진폭을 확인할 수 있음. frequency-domain plot은 frequency-amplitude plot으로 특정 frequency를 가지는 sine wave에 대한 amplitude를 확인할 수 있음.

frequency-domain은 여러 개의 sine wave를 한꺼번에 나타낼 때 편리함.



si 단위계 외워두자.

1.2.2. Composite Signal

1. Composite Signal

여러 개의 simple sine wave로 구성된 signal을 합성 신호(composite signal)라고 함.

전류를 전달하거나 단순한 통신에서는 simple sine wave만을 전송하기도 하지만, 대부분의 data communication에서는 composite signal을 전송함.

Fourier Series/Transform을 사용하여 composite signal을 분해하여 frequency-domain으로 확인할 수

있음.

2. Periodic Composite Signal의 분해

푸리에 급수(Fourier Series)를 사용하여 periodic composite signal을 이산적인 sine/cosine wave로 분해할 수 있음. 즉, time-domain을 이산적인 frequency-domain으로 간단히 나타낼 수 있음.

frequency가 f 인 composite signal을 fourier series로 분해한 경우, 분해 결과로 나온 sine wave들의 frequency는 f 의 배수임. 이를 고조파(Harmonics)라고 함. 가장 낮은 frequency를 가지는 harmonics의 frequency는 f 이고, 이 harmonics를 기본 주파수(Fundamental frequency) 또는 첫 고조파(First Harmonic)라고 함.

Figure 3.10 A composite periodic signal

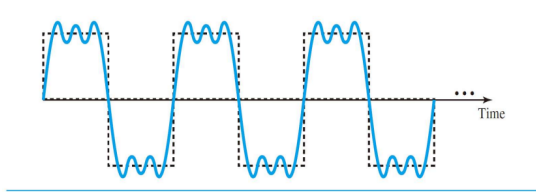
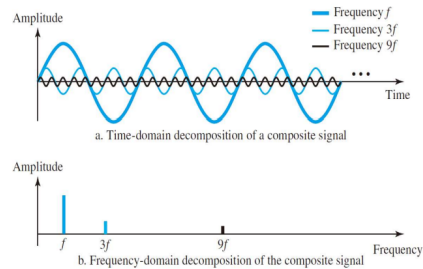


Figure 3.11 Decomposition of a composite periodic signal in the time and frequency domains



3. Non-periodic Composite Signal의 분해

푸리에 변환(Fourier Transform)을 사용하여 non-periodic composite signal을 연속적인 sine/cosine wave로 분해할 수 있음. 즉, time-domain을 연속적인 frequency-domain으로 간단히 나타낼 수 있음.

non-periodic composite signal은 무한히 많은 sine wave의 합성으로 구성됨.

4. Bandwidth

대역폭(Bandwidth)은 composite signal이 가지는 frequency 범위의 길이임. 이는 periodic composite signal과 non-periodic composite signal 모두에 적용됨.

통신에서는 bandwidth가 좁은 것이 유리함. 무선 통신에서는 주파수 영역에 대한 사용이 국가적 단위로 정해져 있으므로, 신호가 적절한 bandwidth를 가져야 송수신이 가능함. 유선에서도 물리적 장치의 사용에서 bandwidth가 영향을 미치는데, 예를 들어 구리선은 32MHz만큼의 bandwidth까지만 원활한 처리가 가능하고, bandwidth가 그보다 넓으면 데이터가 손실됨.

예를 들어, composite signal이 1000~5000Hz의 frequency를 가진다면, bandwidth는 4000Hz임.

1.3. Digital Signal

1.3.1. Digital Signal

1. Digital Signal

digital signal은 해당 data의 특정 비트를 전압으로 표현함. 1을 양의 전압으로, 0을 0에 가까운 전압으로 표현할 수 있음. 여러 개의 전압 level을 사용해 여러 개의 비트를 하나의 전압으로 표현할 수도 있음.

bit rate는 1초에 전송 가능한 비트의 개수임(bpm). bit duration은 한 비트를 전송하는 데에 걸리는 시간임(s). 아래와 같이 bit rate와 bit duration은 역수 관계임.

$$\text{bit rate} = \frac{1}{\text{bit duration}}$$

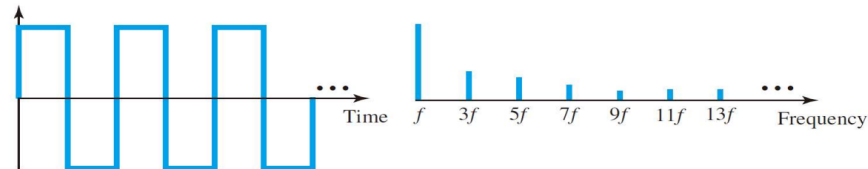
bit length는 transform medium에서 하나의 비트가 차지하는 물리적 거리임. 아래와 같이 전달 속도(propagation speed)와 bit duration을 곱해서 구할 수 있음.

$$\text{bit length} = \text{propagation speed} \times \text{bit duration}$$

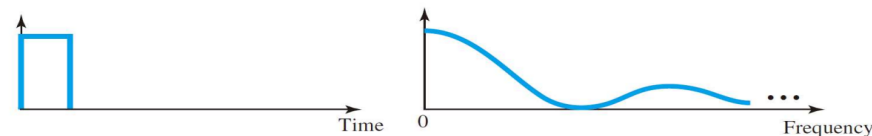
2. Digital Signal의 분해

digital signal도 composite signal로 취급하여 Fourier Series/Transfer를 사용하여 분해할 수 있음.

원본 digital signal은 수직(vertical)/수평(horizontal)인 선(line)으로 구성되어 있는데, vertical line에서는 frequency가 무한대이고, horizontal line에서는 frequency가 0임. frequency가 0과 무한대 사이에서 변화하므로 digital signal은 아래와 같이 무한한 종류의 frequency를 가지게 됨. 물론 periodic의 경우는 이산적인 frequency를 가지고, non-periodic의 경우는 연속적인 frequency를 가짐.

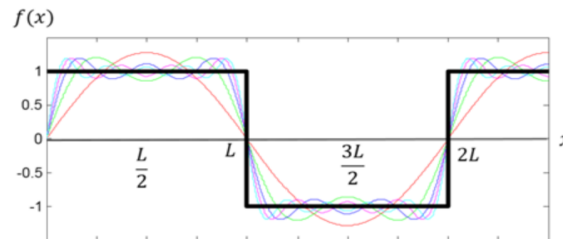


a. Time and frequency domains of periodic digital signal



b. Time and frequency domains of nonperiodic digital signal

digital signal에서의 사각파를 여러 sine wave로 분해하면 사각파와 유사한 형태의 sine wave들로 harmonics가 구성됨.



사각파는 무한개의 사인파를 합한 조합으로 표현할 수 있음.

1.3.2. Transmission of Digital Signal

digital signal의 전송에는 baseband transmission과 broadband transmission이 있음.

1. Baseband Transmission

Baseband Transmission은 digital signal을 analog signal로 변환하지 않고 전송하는 방식임.

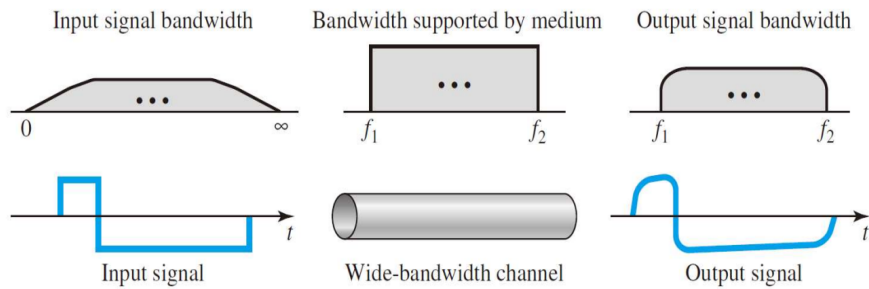
signal을 분해하면 이론적으로 무한 개의 frequency 성분이 존재하므로, 원본 signal을 온전히 전달하려면 무한한 범위의 frequency를 전달해야 하지만 channel은 한정된 bandwidth만을 제공할 수 있으므로 이는 현실적으로 불가능함. channel을 거치면서 원본 signal에 변형이 발생하게 됨.

광섬유 등 넓은 bandwidth를 가지는 medium을 사용하면 높은 정확도로 digital signal을 전송할 수 있음. 물론 이 경우에도 원본과 동일하게는 전송이 불가능하지만 손실된 signal로부터 원본 signal을 유추할 수 있음.

bandwidth가 제한될수록 원본 digital signal과의 차이가 커지므로 정확도가 낮아짐. 이때 적절한 harmonics를 더 추가해서 원본 digital signal과의 차이를 줄일 수 있음.

낮은 frequency만을 통과시키는 channel을 Low-Pass Channel이라고 함. baseband transmission에서는 low-pass channel이 사용됨. 사각파를 low-pass channel에 통과시키면 각진 부분이 둥글게 깎이게 됨.

주로 유선 통신에서 사용하는 방식임.



2. Broadband Transmission

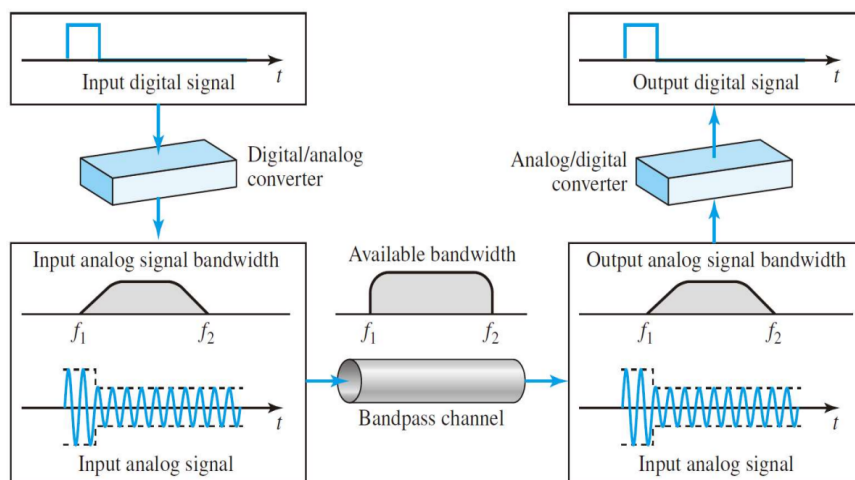
Broadband Transmission은 transmission을 위해 digital signal을 analog signal로 변환(Conversion) 또는 변조(Modulation)하여 전송하는 방식임.

digital signal은 하나의 frequency만을 가지는 analog signal로 변환되는데, 여기에서의 analog signal을 carrier signal이라고 함. carrier signal의 amplitude/frequency/phase가 변화하며 digital signal을 표현함. 자세한 내용은 뒤에서 다룸.

digital signal을 analog signal로 변환한 이후에는, modulation을 통해 특정 frequency 대역을 가지도록 함. 실세계에서는 각 frequency 대역에 대한 활용이 국가 차원에서 관리되고 있기 때문에, 적절한 frequency 대역의 사용이 필수적임. 추가로, modulation의 반대 작업을 복조(demodulation)라고 함.

특정 frequency 대역만을 통과시키는 channel을 Bandpass Channel이라고 함. broadband transmission에서는 bandpass channel이 필수적으로 사용됨. 이때의 signal을 bandpass signal이라고 함.

유선 통신에서도 사용하지만, 주로 무선 통신에서 사용하는 방식임. data를 무선으로 전송하려면 analog signal을 사용해야 함.



1.4. Impairment

1.4.1. Impairment

전송 시에 signal에는 손상(Impairment)이 발생하게 됨. impairment의 원인으로는 attenuation, distortion, noise 등이 있음.

1.4.2. Attenuation

1. Attenuation

감쇠(Attenuation)은 signal의 에너지 감소를 말함. signal이 medium을 지나가면서 받게 되는 저항 때

문에 에너지가 열 등으로 빠져나감.

attenuation에 대한 대책으로 증폭기(amplifier)를 사용하기도 함.

2. Decibel

데시벨(Decibel, dB)은 두 값의 비율에 상용로그를 취하고 10을 곱해 표현하는 물리량 단위임. 큰 수치를 간략하게 표현하기 위해 사용함.

attenuation을 decibel로 나타낼 수 있음. 두 지점에서 signal이 가지는 에너지를 측정하여 아래의 수식으로 계산함. P_1, P_2 는 각 지점에서 측정한 에너지임.

$$dB = 10 \log_{10} \frac{P_2}{P_1}$$

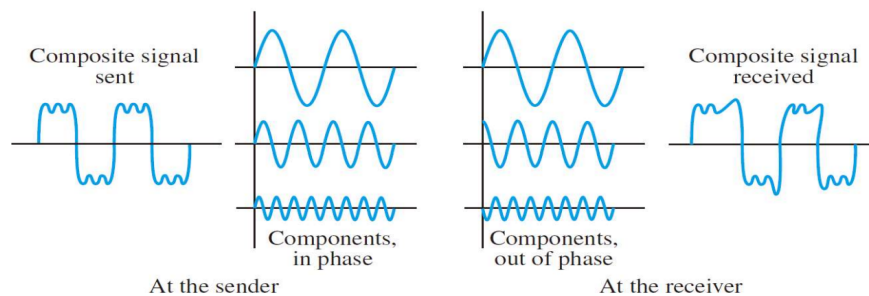
위의 decibel은 구간에 대한 값임. decibel이 음수면 attenuation이 발생한 것이고, 양수면 증폭된 것임. 또한 계산해 보면 당연하게도, 전체 구간의 decibel은 각 부분 구간의 decibel의 합으로 구할 수 있음.

1.4.3. Distortion

왜곡(Distortion)은 회로 시스템의 비선형성에 의해 나타나는 signal의 형태 변화를 말함.

composite signal의 각 harmonics들은 frequency에 따른 medium에서의 전파 속도가 다르고, 이에 따라 성분별로 각각 다른 지연 정도를 가짐. 각 성분이 목적지에 도달했을 때 원본과 다른 위상을 가지게 되므로 signal의 형태가 변화할 수 있음.

당연하게도 λf 가 sine wave의 속도이므로 λ 가 동일하면 frequency에 비례하는 속도를 가짐.



1.4.4. Noise

1. Noise

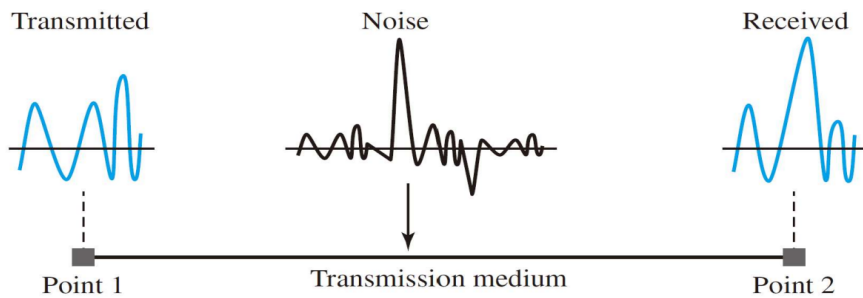
노이즈(Noise)는 의도되지 않은 간섭이나 잡음을 말함.

아래와 같이 noise에는 여러 종류가 있음.

열 잡음(Thermal noise) : wire의 온도가 높아짐에 따라 전자들의 불규칙한 무작위 운동으로 발생하는 noise. 온도가 높아질수록 noise가 심해짐.

크로스토크(Crosstalk) : 인접한 wire에 의한 간섭으로 발생하는 noise.

임펄스 노이즈(Impulse Noise) : 짧은 시간 간격에 큰 에너지를 가진 signal(번개, 전원 선 등)에 의한 간섭으로 발생하는 noise.



2. SNR

SNR(Signal-to-Noise Ratio)는 signal의 세기와 해당 signal에 포함된 noise의 세기의 비율임. SNR이 높을수록 signal이 명확하게 전달된다는 것임.

SNR은 수신 지점에 대한 값임.

SNR은 아래와 같이 signal의 세기와 noise의 세기의 비율로 구할 수 있고, 이를 주로 decibel로 표현함.

$$SNR = \frac{\text{average signal power}}{\text{average noise power}}$$

$$SNR_{dB} = 10 \log_{10} SNR$$

noise가 많은 상황에서는 level을 크게 높일 수 없음.

1.5. Data rate

1.5.1. Data rate

데이터 전송 속도(Data rate)는 통신 시스템에서 가장 중요한 지표 중 하나임.

data rate는 아래의 3가지 요인에 의해 결정됨.

1. bandwidth의 길이. (= 사용되는 frequency = 잦은 signal 변동)
2. signal이 가지는 level.
3. channel의 성능.

bit rate.으로 data rate을 알 수 있음.

1.5.2. Calculating Data rate

data rate를 계산하기 위한 이론적인 공식으로는 Nyquist와 Shannon이 있음.

1. Nyquist bit rate

나이퀴스트(Nyquist) bit rate는 noise가 없는 channel에 대해 data rate(bit rate)의 이론적 최대치를 구하는 공식임.

아래의 공식으로 bit rate를 구할 수 있음. bandwidth는 channel의 bandwidth이고, L은 data를 표현하는 데에 사용된 level의 개수임. 공식을 보면 noise를 고려하지 않은 것을 알 수 있음.

$$\text{BitRate} = 2 \times \text{bandwidth} \times \log_2 L$$

bandwidth와 level을 높이면 data rate를 높일 수 있지만, bandwidth에는 한계가 있음. 또한 level을 높이면 하드웨어 복잡도가 높아지고 reliability가 낮아짐(level이 많아지면 구별 시에 실패 빈도가 높아질 수 있음.).

2. Shannon capacity

샤논(Shannon) capacity는 noise가 있는 channel에 대해 data rate(bit rate)의 이론적 최대치를 구하는 공식임.

아래의 공식으로 $capacity(bit\ rate)$ 를 구할 수 있음. $bandwidth$ 는 $channel$ 의 $bandwidth$ 임.

$$Capacity = bandwidth \times \log_2(1 + SNR)$$

2. Digital Transmission

digital/analog data의 Digital signal transmission에 대해 알아보자.

참고로 여기서의 coding은 부호화를 말함.

2.1. Digital to Digital : Line Coding

2.1.1. Line Coding

1. Line Coding

Line Coding은 digital data, 즉 bit의 배열을 transmission medium(line)에 맞게 digital signal로 변환(coding)하는 방법임. 송신자 측에서 digital data를 digital signal로 인코딩해서 전송하면 수신자는 digital signal을 디코딩해서 digital data를 얻음.

2. data/signal element

data는 전송해야 하는 것이고, signal은 실제로 전송하는 것임. data element는 정보를 표현하는 가장 작은 단위임(하나의 비트와 일치함.). signal element는 digital signal의 가장 작은 시간적 단위임.

data element는 signal element에 의해 운반됨(carried). 이때 각 signal element와 해당 signal element 의해 운반되는 data element의 비율을 r 로 표기함. 즉, 하나의 signal element가 3개의 data element를 운반한다면 r 은 3임.

$$r = \frac{\text{number of data element}}{\text{number of signal element}}$$

3. data/signal rate

Data rate(N , bit rate)는 1초에 전송되는 data element의 개수임. 단위는 bps임.

signal rate(S , pulse rate, modulation rate, baud rate)는 1초에 전송되는 signal element의 개수임. 단위는 baud('보'로 읽음.)임.

data rate와 signal rate 사이에는 아래와 같은 관계가 성립함. 이때 c 는 데이터 패턴에 따른 경우 요인임.

$$S = c \times N \times \frac{1}{r}$$

data communication에서는 data rate는 최대한 늘리고, signal rate(bandwidth, frequency)는 최대한 줄여야 함. data rate를 늘리면 단위 시간 동안 전송하는 비트의 수가 많아지므로 전송 속도가 빨라짐. signal 변화가 많을수록 더 큰 frequency를 사용해야 하므로, signal rate를 줄이면 더 작은 frequency로 signal이 표현되므로 필요한 bandwidth가 줄어듦. bandwidth는 한정되어 있으므로 좁은 것이 유리함. 또한 frequency가 높아지면 이를 위한 하드웨어의 가격이 더 높아짐.

4. Line Coding Scheme

Line Coding Scheme는 line coding에 대한 방법론임. line coding scheme에는 unipolar, polar, bipolar, multilevel, multitransition 등이 있음.

각 line coding scheme에 대해 DC component(Baseline wandering 포함)과 self-synchronization은 단순히 signal만 봐도 확인이 쉬움. 관건은 bandwidth를 구하고 파악하는 것임. 여기서 bandwidth는 N (data rate)로 나타낼 수 있음. digital signal과 유사한 형태의 sine wave(실제로 이렇게 분해됨.)를 그리고, 비트 하나가 전송된 시간이 $\frac{1}{N}$ 이므로 해당 sine wave의 period 및 frequency를 구할 수 있음. 이 과정을 반복해서 여러 frequency를 구하면 frequency-domain에서 bandwidth를 나타낼 수 있음.

signal rate는 데이터 패턴에 따라 달라지고, 이에 따라 주로 평균적인 경우에 대해 signal rate를 이야기함. 이 경우 S를 S_{ave} 로 표기할 수 있음. 의해 왜 S이 데이터 패턴에 따라 달라지는지는 더 알아보자.

2.1.2. Line Coding의 문제 상황들

line coding에서는 아래와 같은 문제 상황들이 발생할 수 있으므로, line coding scheme는 이를 잘 처리할 수 있어야 함.

1. Baseline Wandering

signal의 수신자는 수신된 signal의 세기의 평균치를 계산하여 이후 수신된 signal의 세기에 대한 기준선(Baseline)으로 사용함. 이때 signal이 특정 세기를 연속적으로 가진다면 baseline이 편향될 수 있는데, 이를 Baseline Wandering이라고 함.

좋은 line coding scheme는 baseline wandering을 잘 방지해야 함.

2. DC component

직류(DC)에서는 한 방향으로 일정한 전류가 흐르고, 교류(AC)에서는 크기와 방향이 변화하며 전류가 흐름. 전체 signal에서 교류 성분(AC Component)을 제거하고 남은 부분을 직류 성분(DC component)이라고 함.

AC component는 주기적이므로, 전체 signal의 전압에 대한 총합이 0이거나 0에 가깝지 않은 경우 DC component가 포함된 것임. 이때 전압 총합을 DC value라고 함. digital signal은 사각형 모양의 signal이므로 단순히 +1, -1, 0 등으로 DC value를 계산할 수 있음. 일반적인 signal의 경우 아래와 같이 적분으로 간단히 구할 수 있음.

$$DCvalue = \int_0^T s(t)dt$$

저주파 성분을 통과시킬 수 없거나 변압기를 사용하는 medium이나 시스템에서는 DC component가 문제가 될 수 있으므로, 해당 환경에 대한 line coding scheme는 이를 잘 처리해야 함. data communication에서는 DC component를 제거하는 것이 좋음. DC component가 최소화되거나 제거된 signal을 DC-balanced하다고 함.

참고로, 변압기는 자속의 변화에 따른 전자기 유도(패러데이의 법칙)에 의해 동작하므로, DC 전류(자속 변화 없음.)가 유입되면 처리하지 못하거나 장치에 문제가 발생할 수 있음.

3. Self-synchronization

수신자의 비트 간격(bit interval)과 송신자의 bit interval이 일치하지 않으면 signal을 디코딩할 때 문제가 발생함. 이를 해결하기 위해 data 안에 타이밍 정보를 포함하는 자기 동기화(Self-synchronization) signal을 사용함.

즉, 신호 자체에서 타이밍 정보를 알 수 있어야 하고, 이에 따라 수신자와 송신자가 서로 동기화되어 있어야 함.

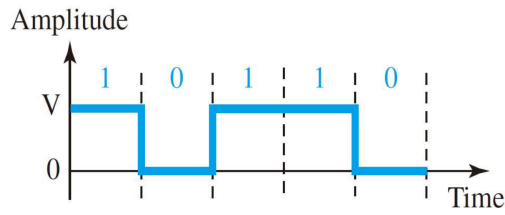
2.1.3. Line Coding Schemes : Unipolar

Unipolar scheme는 signal level이 양과 음 한 방향으로만 변하는 scheme임. 즉, signal level이 한쪽과 0에만 존재하고, 시간 축 반대로 넘어가지 않음.

NRZ가 있음.

1. NRZ

NRZ(Non-Return-to-Zero)는 비트 중간에 전압이 0이 되지 않는 unipolar scheme임. 양의 전압을 1로, 0 전압을 0으로 함.



Baseline Wandering, DC component, Self-synchronization이 처리되지 않음.

uni는 단일, polar는 극을 의미함. 즉, 단일 극만을 가지는 scheme.

2.1.4. Line Coding Schemes : Polar

Polar scheme는 signal level이 양과 음 양쪽 방향으로 변하는 scheme임.

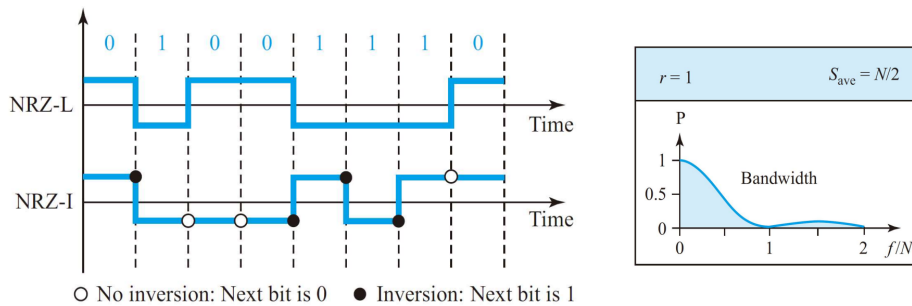
NRZ, RZ, biphase가 있음.

1. NRZ

NRZ(Non-Return-to-Zero)는 비트 중간에 전압이 0이 되지 않는 polar scheme임. NRZ-L과 NRZ-I가 있음.

NRZ-L(NRZ-Level)은 전압 level에 따라 비트가 결정되는 NRZ임. 양의 전압을 0, 음의 전압을 1로 함.

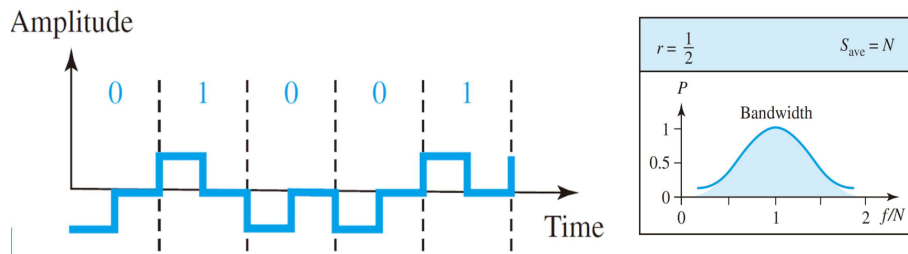
NRZ-I(NRZ-Invert)은 전압의 변화에 따라 비트가 결정되는 NRZ임. 전압 변화가 없으면 0, 변화가 있으면 1로 함.



Baseline Wandering, DC component, Self-synchronization이 처리되지 않음.

2. RZ

RZ(Return-to-Zero)는 비트 중간에 전압이 0이 되는 polar scheme임. 양의 전압, 음의 전압, 0 전압을 모두 사용하여, 특정 비트에 대한 전압을 표현한 후 비트 중간에 0 전압으로 돌아옴. 음의 전압이었다가 0 전압으로 돌아오는 것은 0, 양의 전압이었다가 0 전압으로 돌아오는 것은 1로 함.



Baseline Wandering, DC component, Self-synchronization이 처리됨. 하지만 3개의 전압을 구분할 수 있도록 구현하는 것이 까다롭고, 전압 변화가 많아 높은 frequency가 필요하고 bandwidth가 넓다는 단점이 있음.

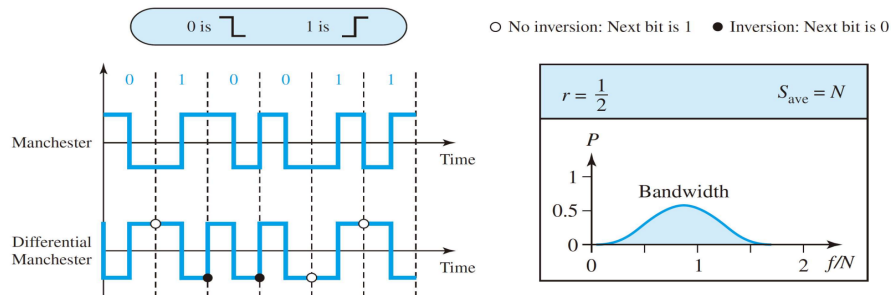
3. biphase(Manchester, Differential Manchester)

biphase는 2개의 phase로 비트를 표현하는 polar scheme임. 양의 전압과 음의 전압을 사용하여, 비

트 하나를 표현할 때 특정 전압을 사용하다가 중간에 다른 전압을 사용함. manchester와 differential manchester가 있음.

Manchester는 NRZ-L과 RZ의 개념을 합친 biphase로, 전압에 의해 비트가 정해지면서 중간에 전압이 바뀜. 양의 전압이었다가 음의 전압인 것을 0, 음의 전압이었다가 양의 전압인 것을 1로 함.

Differential Manchester(차이 Manchester)는 NRZ-I와 RZ의 개념을 합친 biphase로, 전압의 변화에 따라 비트가 정해지면서 중간에 전압이 바뀜. 전압 변화가 없으면 1, 변화가 있으면 0으로 함.



Baseline Wandering, DC component, Self-synchronization이 처리됨. 하지만 NRZ에 비해 signal rate가 2배여서 bandwidth가 넓고 높은 frequency를 사용하게 됨.

biphase가 코스트가 비교적 높기는 해도, 요새는 하드웨어 성능이 워낙 좋아서 요구사항이 엄격한 것이 아니라면 심각한 문제까지는 아니라고 함.

2.1.5. Line Coding Schemes : Bipolar

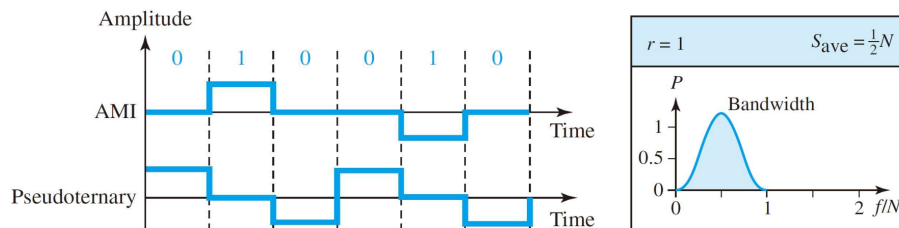
Bipolar는 특정 비트를 표현할 때 양과 음의 전압을 번갈아 사용하는 scheme임.

AMI, pseudoternary가 있음.

1. AMI, pseudoternary

AMI(Alternate Mark Inversion)는 0은 0 전압으로, 1은 양의 전압과 음의 전압을 번갈아 사용하여 표현하는 bipolar scheme임. 장거리 통신에 주로 사용됨.

pseudoternary는 AMI와 동일한데 0과 1의 표현이 반대임. 즉, 1을 0전압으로, 0은 양의 전압과 음의 전압을 번갈아 사용하여 표현하는 bipolar scheme임.



Baseline Wandering, DC component를 처리하면서, NRZ와 같은 수준의 signal rate를 가짐. 하지만 0이 연속적으로 등장하는 경우 synchronization에 문제가 생길 수 있음.

2.1.6. Line Coding Schemes : Multilevel

Multilevel은 여러 개의 signal level을 사용하는 scheme임.

multilevel에 해당되는 일부 scheme들은 $mBnL$ 의 형태로 이름을 붙임. m 은 data element의 개수를, n 은 signal element의 개수를, L 은 사용하는 signal level의 개수를 나타냄.

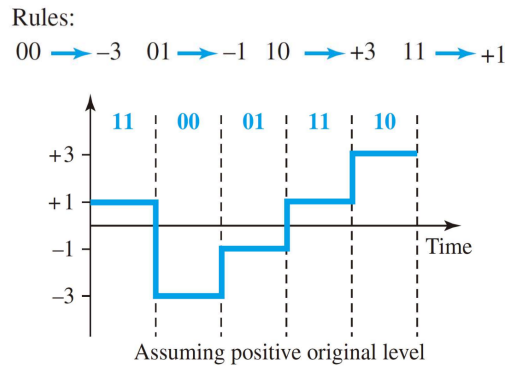
2B1Q, 8B6T, 4D-PAM5가 있음.

1. 2B1Q

2B1Q는 2개의 data element를 4개(quaternary)의 level을 가지는 1개의 signal element로 표현하는

multilevel scheme임.

이런 종류의 scheme에 대해서, 어떤 level(전압)에 어떤 비트 조합을 매핑할 것인지는 hamming distance를 고려함. hamming distance는 두 이진수에 대해 서로 다른 비트 값을 가지는 자리의 개수임(예를 들어, 00과 01은 차이가 1. 01과 10은 차이가 2임.). 전압과 hamming distance가 비례하도록 각 비트에 대한 전압을 설계함. 비트에 차이가 크다면, 전압 차이 또한 크게 해서 잘못 해석될 확률을 줄이는 것. 일례로, 만약 11과 00이 바로 옆 전압을 사용하고, 11에 해당하는 전압이 00으로 해석된다면 오류가 더 커지게 됨.



NRZ에 비해 2배 빠르게 데이터를 전송할 수 있지만, 수신자가 각 level을 구분할 수 있어야 함.

과거에 존재하던 전화선을 사용해 인터넷을 제공하는 DSL(Digital Subscriber Line) 등에서 사용함. (8Mbps의 속도를 제공할 수 있음.)

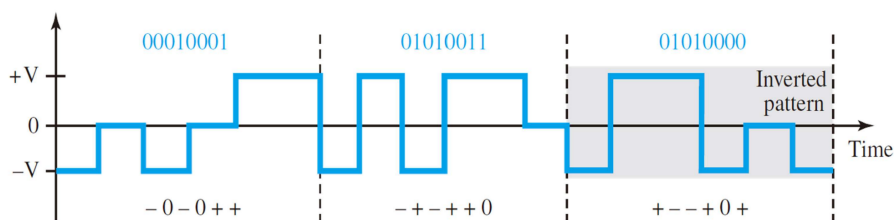
Baseline Wandering, DC component, Self-synchronization이 처리되지 않음.

2. 8B6T

8B6T는 8개의 data element를 3개(ternary)의 level을 가지는 6개의 signal element로 표현하는 multi-level scheme임.

3개의 level을 사용하므로 8개의 비트보다 더 많은 정보를 나타낼 수 있지만, synchronization과 error detection에 남은 signal element들을 사용함.

8B6T에서는 data 하나에 해당되는 signal이 0 또는 +1의 DC value를 가지도록 설계되어 있음. 이때 DC value가 +1인 signal이 등장한 경우 DC balance를 맞추기 위해 그 다음 DC value가 +1인 signal은 상하로 반전되어 DC value를 -1로 가지게 됨. 이는 송신자 측에서 처리하며, 수신자 측에서는 DC value가 -1인 경우 반전된 signal인 것으로 판단할 수 있음.



LAN 구축, 데이터 센터 등에 활용되는 100BASE-4T cable 등에서 사용함. (100Mbps의 속도를 제공할 수 있음.)

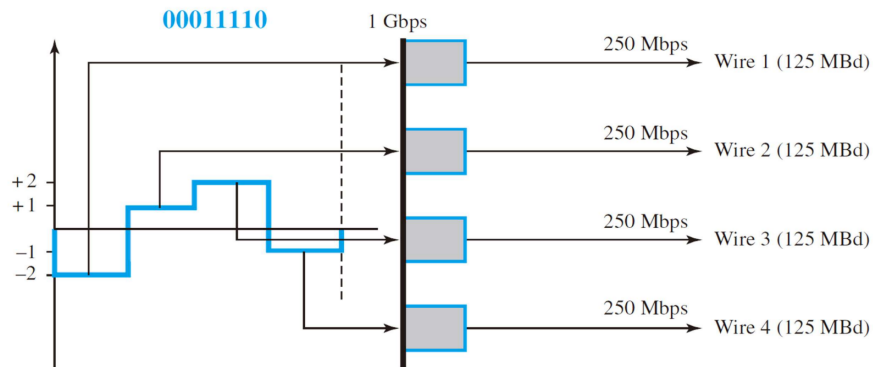
Baseline Wandering, DC component, Self-synchronization이 처리됨.

3. 4D-PAM5

4D-PAM5(4-dimensional 5-level Pulse Amplitude Modulation)는 5개의 level과 4개의 signal element를 사용하고, 각 signal element에 대한 wire(PAM)가 총 4개 존재하는 multilevel scheme임.

만약 1개의 wire만을 사용해서 data rate를 유지하려면 같은 시간 간격 안에 signal 4개가 들어가야 함. 즉, 분해된 sine wave들의 period가 $\frac{1}{4}$ 배, frequency가 4배가 되고 결과적으로 bandwidth가 4배 가까이 됨. 즉, 4D-PAM5를 사용하면 bandwidth가 $\frac{1}{4}$ 배로 줄일 수 있음.

0 전압에 해당하는 level은 error detection에 사용됨.



Baseline Wandering, DC component, Self-synchronization이 처리됨.

error detection 등 자세한 내용은 추후에 다룸.

2.1.7. Line Coding Schemes : Multitransition

Multitransition은 여러 개의 전압 변경(transition) 방식을 사용하는 scheme임.

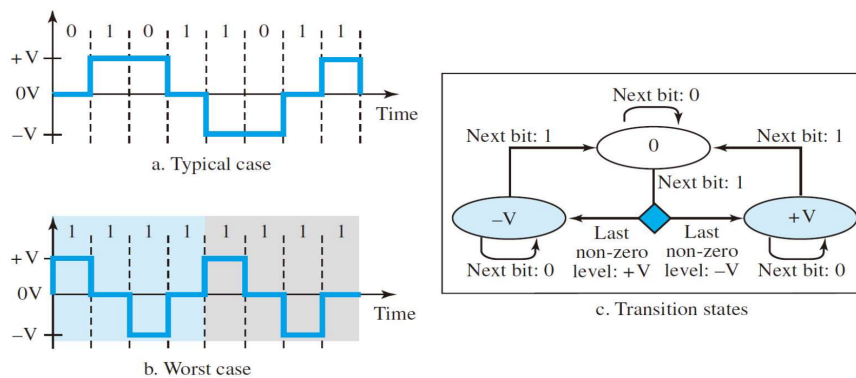
NRZ-I와 differential manchester에서는 비트 변화의 유무에 대해 전압이 변경되었지만, multitransition에서는 transition에 대한 규칙이 더 많음.

MLT-3이 있음.

1. MLT-3

MLT-3는 3개의 level(양의 전압, 음의 전압, 0 전압)을 사용하고, 아래와 같이 level 변화에 대한 3가지 transition 규칙을 사용함.

1. 다음 비트가 0이면, 변환하지 않음.
2. 다음 비트가 1이고 현재 level이 0이 아니면, 다음 level을 0 전압으로 지정.
3. 다음 비트가 1이고 현재 level이 0이면, 다음 level을 마지막으로 등장한 0이 아닌 level의 반대 level로 지정함.



MLT-3은 더 복잡한 시스템을 사용하는데에 반해 NRZ-I와 동일한 signal rate를 가짐. 하지만 실제로 MLT-3는 NRZ-I에 비해 전압 변화가 적고, 1인 반복되는 등의 상황에서 주기적인 형태를 보이므로 적은 종류의 frequency로 구성되며 더 좁은 bandwidth를 지원함. 그래서 구리선(구리선은 frequency가 32MHz보다 높아지면 전자기 방사가 발생함.) 등을 통해 빠른 속도(100Mbps 가량)로 데이터를 전송해야 할 때 MLT-3을 사용함.

Baseline Wandering, DC component이 처리되지만, 0이 연속적으로 등장하는 경우 Self-synchronization가 처리되지 않음.

전자기 방사가 발생한다는 것은, signal이 전자기파의 형태로 손실된다는 것임.

2.2. Digital to Digital : Block Coding

2.2.1. Block Coding

1. Block Coding

Block Coding은 m 개의 비트를 n 개의 비트로 변환하는 기술로, 더 많은 비트를 사용해 scheme에 따라 기존의 데이터가 가질 수 있는 문제점(Baseline wandering, DC component, Self-synchronization 등)을 보완함.

block coding은 m 과 n 에 따라 mB/nB 의 형태로 이름을 붙임.

block coding은 주로 line coding scheme에 의해 인코딩되기 전과 디코딩된 이후에 사용됨.

2. Block Coding Step

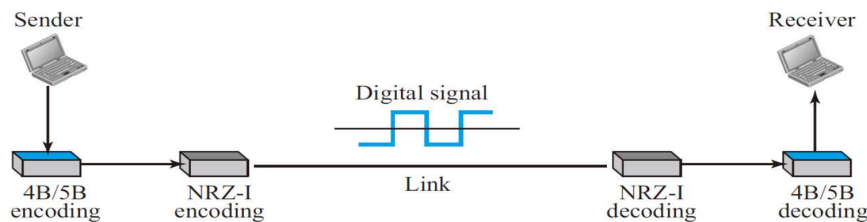
block coding은 아래의 3가지 단계를 가짐.

1. Division : 비트들을 m 개씩 나눠 묶음.
2. Substitution : m 개의 비트를 n 개의 비트로 변환함.
3. Combination : 변환한 결과 비트들을 모음.

block coding은 n 이 m 보다 더 큰 경우임. m 이 n 보다 더 큰 경우는 data의 압축으로, png, jpg, mp3 등에서 수행함.

2.2.2. 4B/5B

4B/5B는 4개의 비트를 연속된 0이 없는 5개의 비트로 변환하는 block coding으로, NRZ-I에서 사용하기 위해 디자인되었음.



NRZ-I는 0이 연속적으로 나오는 경우 때문에 Self-synchronization을 처리하지 못함. 4B/5B를 사용하면 변환된 5개의 비트는 1개보다 많은 0으로 시작하지 않고, 중간에 0이 2개를 넘어서서 연속되지 않음. 결과적으로 3개 이상의 0이 연속되는 경우가 존재하지 않음.

제어를 위한 부분(Control Sequence)에는 0이 연속적으로 들어가기로 함. start/end delimiter는 packet의 시작과 끝에 사용되는 control sequence임.

Data Sequence	Encoded Sequence	Control Sequence	Encoded Sequence
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (Start delimiter)	11000
0100	01010	K (Start delimiter)	10001
0101	01011	T (End delimiter)	01101
0110	01110	S (Set)	11001
0111	01111	R (Reset)	00111
1000	10010	Enables Error Detection	
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

4비트를 5비트로 표현했으므로 잉여 비트(Redundant Bit)가 발생하는데, 이는 error detection에 사용됨. 4B/5B에 미리 정의되어 있지 않은 비트 집합이 전송된 경우 error가 발생했음을 알 수 있음.

NRZ-I에 4B/5B를 사용하면 signal rate가 20%(25%?) 증가하게 되고, signal 변동이 많아지므로 더 높은 frequency와 넓은 bandwidth를 사용하게 됨. 하지만 그래도 biphase보다는 여전히 bandwidth가 좁음. 4B/5B는 DC component를 해결하지는 못하므로 DC에 대한 제한이 있는 환경에서는 biphase를 사용해야 함.

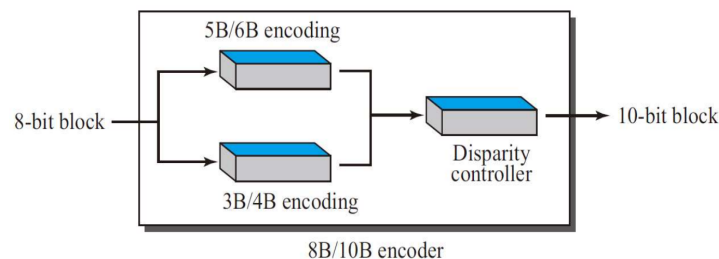
2.2.3. 8B/10B

1. 8B/10B

8B/10B는 8개의 비트를 연속된 1 또는 0이 없고 DC balance를 유지하도록 10개의 비트로 변환하는 block coding임.

8개의 비트를 5개와 3개로 나누어 각각 5B/6B, 3B/4B 인코딩을 수행한 뒤(매핑 테이블의 단순화를 위해 나누어 수행함), 결과물을 합해 Disparity Controller라는 요소에 전달함.

8개의 비트를 10개의 비트로 표현했으므로 4B/5B보다 redundant bit의 수가 많고, 이에 따라 더 향상된 error detection을 수행할 수 있음.



2. Disparity Controller

Disparity(=DC weight) Controller에서는 1과 0이 불균형한 개수로 존재하거나, 인접한 다른 비트들과의 불균형이 존재하는 경우를 비트 반전 등을 통해 처리함. 즉, DC balance를 맞추는 부분임.

8B/10B의 변환 table을 보면 하나의 8bit 데이터가 두 개의 10bit 데이터에 매핑되어 있음. 이때 10bit 데이터 각각이 모두 DC weight가 0인 것도 있고, -1과 +1인 것도 있음. disparity controller는 이전에 등장한 값의 DC weight를 고려해 적절한 DC weight를 가지는 10bit 데이터를 전송함.

2.3. Digital to Digital : Scrambling

2.3.1. Scrambling

1. Scrambling

Scrambling은 line coding scheme(주로 AMI)에 적용하여 synchronization을 방해하는 부분(0이 연속적으로 등장하는 부분)을 특수한 bit 배열로 대체하는 기법임.

기존 line coding scheme 중 biphas(polar)는 bandwidth가 너무 넓었고, NRZ-I에 block coding을 사용한 경우는 DC component 문제가 있었고, AMI(bipolar)는 bandwidth도 좁고 DC component도 발생하지 않는데 synchronization 문제가 있었음. 이때 AMI는 synchronization 문제만 해결하면 됐는데, scrambling을 AMI에 적용하여 이 문제를 해결할 수 있음.

scrambling에는 B8ZS와 HDB3이 있음.

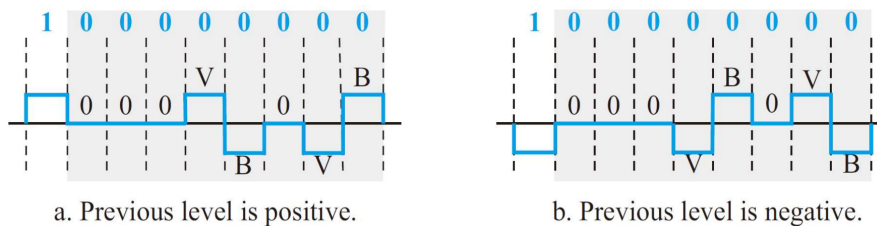
2. B8ZS

B8ZS는 0 bit 8개가 연속해서 등장하면 scrambling이 적용되어 0들을 00VB0VB으로 대체하는 방식임.

V(Violation)는 AMI 규칙을 위반하는 0이 아닌 전압(직전 0이 아닌 전압(B 포함)과 동일한 전압.)이고, B(Bipolar)는 AMI 규칙에 부합하는 0이 아닌 전압(직전 0이 아닌 전압(V 포함)과 반대인 전압.)임. 즉, 마지막에 등장한 0인 아닌 전압의 부호에 따라 00VB0VB가 뒤집혀 나타나게 됨.

bit rate, DC valance와 bandwidth에 악영향을 끼치지 않으면서 synchronization 문제를 해결할 수 있음.

북아메리카에서 주로 사용되는 방식임.



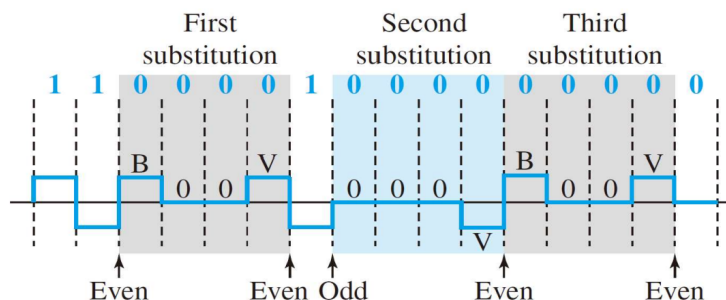
3. HDB3

HDB3은 0 bit 4개가 연속해서 등장하면 scrambling이 적용되어 0들을 000V 또는 B00V로 대체하여 전송하는 방식임.

마지막 scrambling(substitution)의 종료 시점부터 새로운 scrambling의 시작 지점까지 0이 아닌 전압의 개수를 확인하여, 짝수(even)개(또는 0개)이면 B00V로 치환하고, 홀수(odd)개이면 000V로 치환함.

bit rate, DC valance와 bandwidth에 악영향을 끼치지 않으면서 synchronization 문제를 해결할 수 있음.

유럽, 일본 등에서 주로 사용되는 방식임.



참고로 일본은 통신과 관련된 역사가 깊음. 태평양 전쟁에 참여하느라 무선통신 기술이 많이 발달했기 때문.

2.4. Analog(signal) to Digital(data) Conversion

2.4.1. Analog(signal) to Digital(data) Conversion

실세계의 데이터는 *analog*이고, 이를 그대로 컴퓨터 시스템에서 사용하거나 전송해야 하는 경우가 존재함(음성 등). 또는 송신자가 전송한 *analog signal*을 수신자 측에서 수집해 사용하는 경우가 있음. 이를 위해서 *analog signal*을 *digital data*로 변환할 수 있어야 하고, 여기서는 그 방법에 대해 알아봄.

*analog signal*을 *digital data*로 변환하는 기법에는 PCM과 DM이 있음.

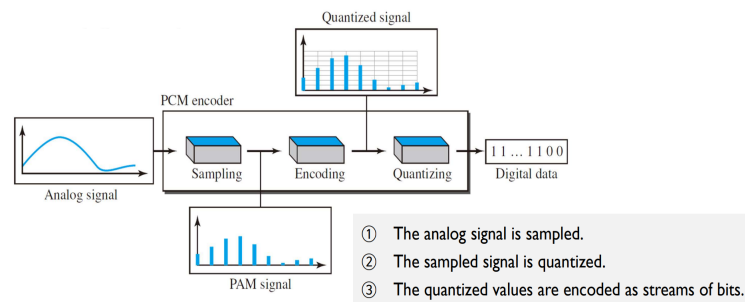
이렇게 변환한 *digital data*을 적절한 *line coding scheme*와 기법들을 사용해 *digital signal*로 전송하거나, *digital(data) to analog(signal) conversion*을 통해 *analog signal*로 전송할 수 있음.

2.4.2. PCM

1. PCM

PCM(Pulse Code Modulation)은 *analog signal(pulse)*를 *digital data(code)*로 변환(modulate)하는 기법임. 가장 흔하게 사용되는 digitization 기법 중 하나임.

PCM은 *sampling*, *encoding*, *quantizing*의 단계로 구성되고, 연속적인 값을 이산적인 값으로 변환하는 과정을 거침.

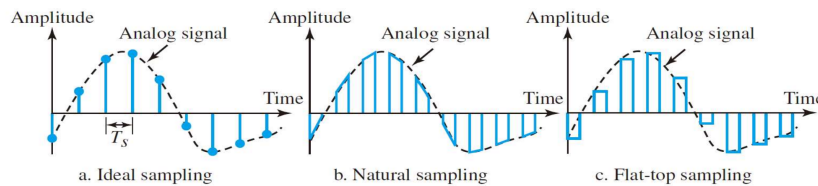


(위 그림에서는 *encoding*과 *quantizing*의 순서가 잘못 나와 있음.)

2. Sampling

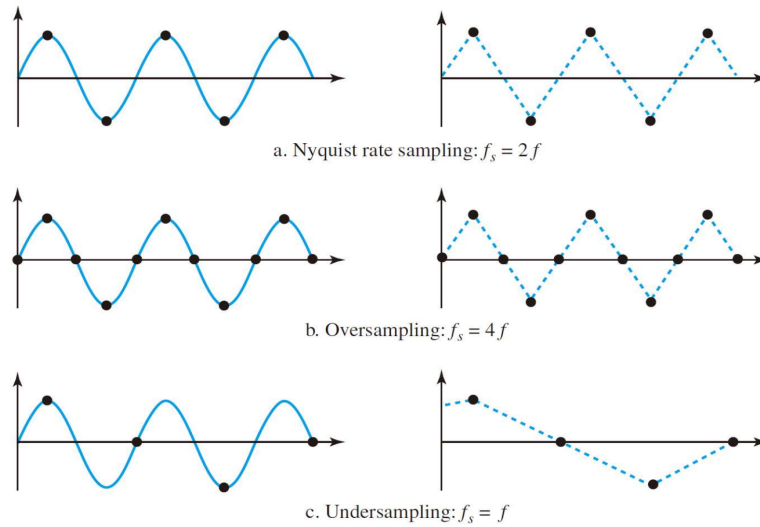
Sampling 또는 PAM(Pulse Amplitude Modulation)은 시간(x 축)에 대해 *analog signal*을 이산적으로 쪼개는 단계임. *sampling*에 의해 도출된 결과물을 *PAM signal*이라고 함.

*sampling*에서는 *analog signal*을 특정 시간 간격으로 쪼개는데, 이 간격을 *sampling interval* 또는 *sampling period(s)*라고 하고, T_s 로 표기함. *sampling interval*의 역수를 *sampling rate* 또는 *sampling frequency(Hz)*라고 하고, $f_s = \frac{1}{T_s}$ 로 표기함.



*Nyquist (sampling) theorem*에 의하면, *sampling rate*가 최소한 원본 *analog signal*의 2배는 되어야 변환한 *digital data*로 원본 *analog signal*을 충분히 복원할 수 있음. 즉, 한 번의 주기에 대해 최소 2번은 값을 확인할 수 있어야 한다는 것인데, 파동에 일정한 간격으로 점을 찍어 보면 당연함.

물론 *sampling frequency*가 크면 *analog signal*을 더 정확히 복원할 수 있지만, 당연하게도 그만큼 많은 *cost*가 발생함.



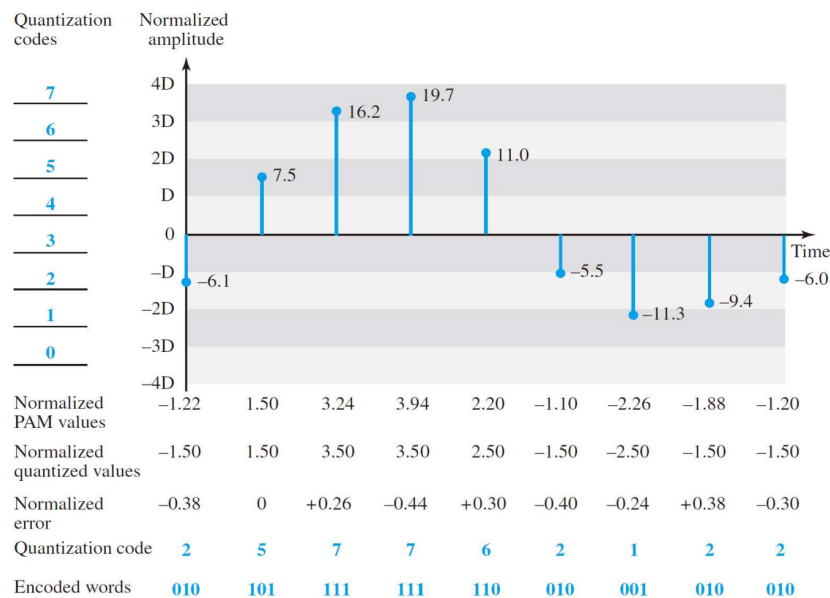
3. Quantizing

Quantizing(양자화)은 진폭(전압)(y축)에 대해 PAM signal을 이산적으로 쪼개는 단계임. quantizing에 의해 도출된 결과물을 quantized signal이라고 함.

quantizing에서는 PAM signal을 특정 전압 간격으로 쪼개는데, 이 간격을 Δ 라 하고, 아래와 같이 당연하게 구할 수 있음. 여기서 L 은 quantization level임.

$$\Delta = \frac{V_{max} + V_{min}}{L}$$

PAM signal의 각 값을 정규화하는 등의 처리를 한 뒤, 해당 값이 속하는 구간의 값으로 근사함. 이후 해당 구간에 매핑된 code로 변환함. 이후 Encoding 단계에서 해당 code를 사용해 특정 비트 집합으로 인코딩함.



Quantization Level(L)은 quantizing에서 사용하는 구간(level)의 개수임. L 이 클수록 analog signal을 더 정확히 복원할 수 있음. L 이 작아지면 정확도가 떨어지고 quantization error가 커짐.

Quantization Error는 quantizing에서의 근사에 의해 발생하는 오차 값임. 즉, 실제 값과의 차이를 나타내는 값임.

4. Encoding

Encoding은 *quantized signal*을 *bit*의 집합으로 인코딩하는 단계임.

당연하게도 하나의 *quantized*된 값이 가지는 비트의 개수(n_b)는 아래와 같이 L 로 구할 수 있음.

$$n_b = \log_2 L$$

*bit rate*는 1초에 전송되는 비트의 개수이므로, 아래와 같이 *sampling rate*와 n_b 를 곱해 얻을 수 있음.

$$\text{bit rate} = f_s \times n_b$$

참고로, 사람 목소리와 음악은 20kHz까지의 주파수를 가짐. 이에 따라 녹음, 음악 재생 관련 기기는 *sampling rate*를 최소한 44.1kHz까지 지원하도록 설계함. 또한 L 은 주로 256으로 지정함.

2.4.3. DM

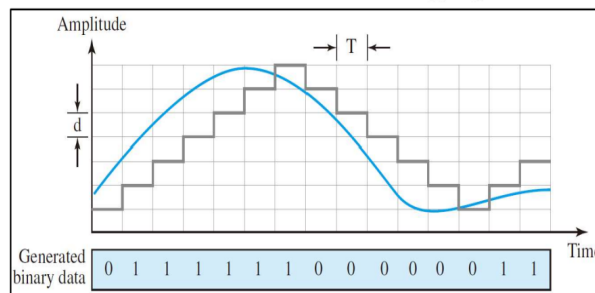
1. DM

DM(Delta Modulation)은 현재 *analog signal*의 전압 수준과 이전 시간 간격에서 변조에 의해 생성된 전압 수준의 차이(delta, δ)가 가지는 부호에 따라 바로 다음 시간 간격의 비트가 결정되는 기법임. 현재의 *signal*이 크면 1, 작으면 0이 생성됨.

*digital data*로 변환 시에는 DM modulator를 사용하고, 이를 다시 *analog signal*로 변환할 때에는 DM demodulator를 사용함.

맨 앞 비트는 시작 비트로, 변화량을 확인할 수 없으므로 0으로 지정함.

PCM보다 낮은 하드웨어 복잡도를 가짐.



2. DM Modulator

DM Modulator는 DM 기법에서 *analog signal*을 *digital data*로 변조하는 부분임.

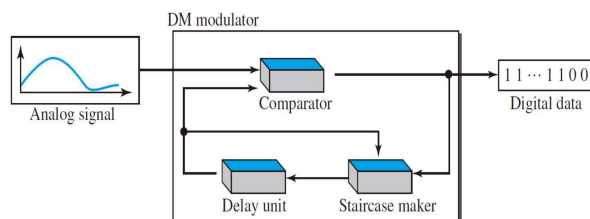
DM modulator는 *comparator*, *staircase maker*, *delay unit*으로 구성됨.

Comparator : 전압 수준을 비교하여 비트를 생성함.

Staircase Maker : *analog signal*을 변조하여 전압 수준을 지정함.

Delay Unit : 정의된 시간 간격 동안 지정된 전압을 유지함.

즉, *comparator*를 거친 이전 시간 간격에서의 *analog signal*을 *staircase maker*에서 변조하여 특정 전압 수준을 가지게 하고, *delay unit*에서 해당 전압을 비교에 사용할 때까지 유지하고, *comparator*에서 그 전압을 가져다가 현재의 *analog signal*과 비교하여 비트를 생성하는 것.



3. DM Demodulator

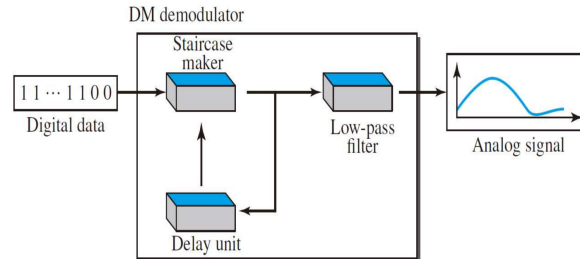
DM Demodulator는 DM 기법에서 digital data를 analog signal로 복원하는 부분임. 우선 비트 집합으로부터 각진 형태의 signal로 복원하고, 이후 각진 부분을 깎음.

DM demodulator는 staircase maker, delay unit, low-pass filter로 구성됨.

Staircase Maker : digital data로부터 전압 수준을 지정함.

Delay Unit : 정의된 시간 간격 동안 지정된 전압을 유지함.

Low-pass filter : 주파수가 높은 성분을 제거함. 즉, 각진 부분을 원만하게 깎아서 원본 analog signal과 유사한 부드러운 신호로 변환함.



2.5. Transmisison Mode

2.5.1. Transmission Mode

Transmission Mode는 구성된 digital data(비트 집합)를 transmission medium을 통해 전송할 때, 어떻게 효율적으로 전송할지에 대한 분류임.

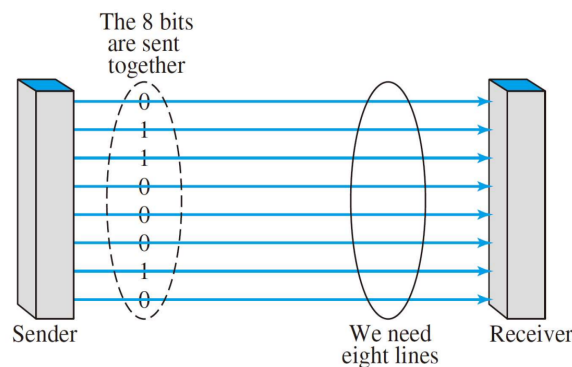
transmission mode에는 parallel mode와 serial mode가 있음.

지금까지 line coding scheme와 여러 기법들, analog to digital을 통해 전송할 비트 집합을 구성하는 방법을 다뤘고, 여기서는 그렇게 구성한 비트 집합을 어떤 형태로 전송할 것인지를 다룸.

2.5.2. Parallel Mode

병렬 모드(Parallel Mode)는 비트 집합의 비트들을 특정 개수로 그룹화하고, 비트 수 만큼의 wire/channel을 사용하여 비트들을 동시에 병렬적으로 전송하는 transmission 방식임.

빠르지만, cost가 많이 듦.

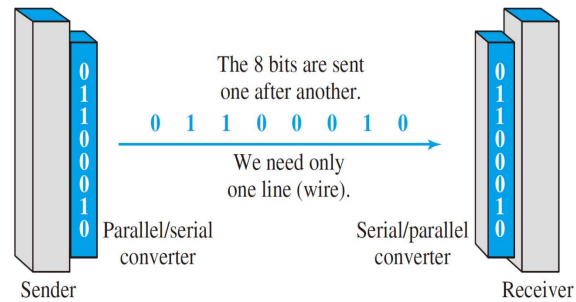


2.5.3. Serial Mode

순차 모드(Serial Mode)는 하나의 wire/channel만을 사용하여 각 비트를 순차적으로 전송하는 transmission 방식임.

이때 송신자와 line 사이에는 parallel/serial converter가, line과 수신자 사이에는 serial/parallel converter가 필요함.

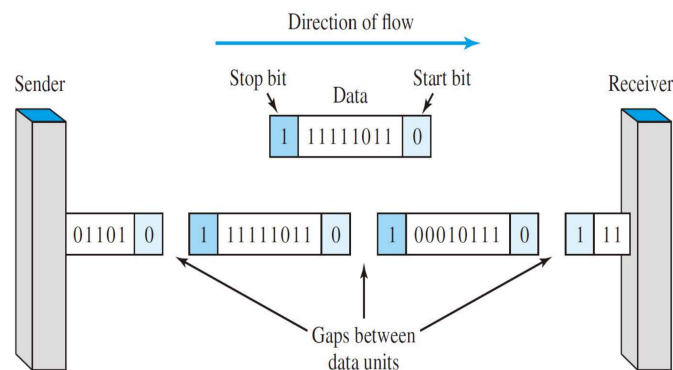
주요 방식으로는 *Asynchronous*와 *synchronous*가 있음.



1. *Asynchronous*

비동기 전송(*Asynchronous transmission*)은 *signal*을 비동기적으로 전송하는 방식임.

기본 전송 단위는 1바이트임. 해당 바이트 앞에 *start bit(0)*를, 뒤에 *end bit(1)*를 붙여 해당 데이터의 시작과 끝을 나타냄.



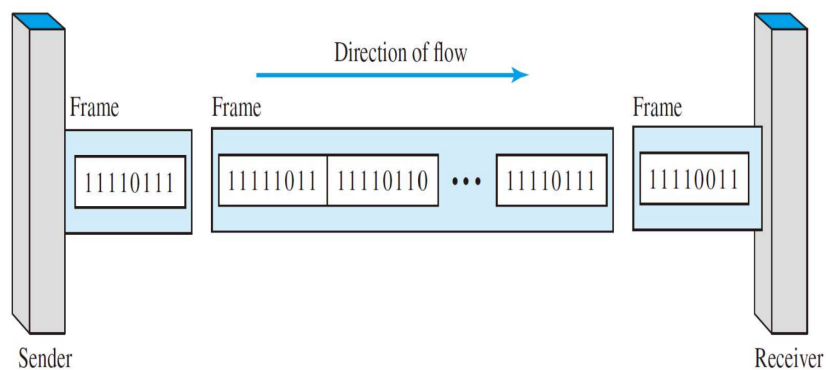
당연하게도 각 바이트는 비동기적으로 전송되지만, 해당 바이트 안의 각 비트는 동기적으로 처리됨.

*asynchronous*은 간단한 *transmission*에서 사용하는 방식임. 예를 들어, 컴퓨터에서 모니터로 화면 정보를 전달하는 경우에는 단순히 바이트 단위로 정보를 전송하면 됨. 하지만 한 번에 1바이트보다 많은 데이터를 전송해야 하는 복잡한 비교적 통신에서는 사용이 불가능함.

2. *Synchronous*

동기 전송(*Synchronous transmission*)은 *signal*을 동기적으로 전송하는 방식임.

기본 전송 단위는 *frame*임. *frame*은 *link layer*에서의 데이터 전송 단위로, *physical layer*에서 *frame*을 생성하고 *link layer*에서 이를 전송하는 것. *frame*은 *MAC* 주소, *ip* 주소, 포트 번호 등 계층별 처리를 위한 다양한 정보를 가지고 있음. 수신자는 미리 정의된 프로토콜에 의해 *frame*으로부터 적절한 데이터를 얻을 수 있음.



이더넷, wLAN(wifi) 프로토콜 등을 사용하는 네트워크에서는 *synchronous* 방식을 사용함.

3. Isochronous

등시 전송(*Isochronous Transmission*)은 *synchronous*와 유사하게 *signal*을 동기적으로 전송하지만, 각 *frame* 사이의 시간 간격을 일정하게 한 방식임.

오디오/비디오 등 데이터 중간의 시간 간격에 예민한 *real-time* 데이터의 전송에 특화된 방법임. 동기에서는 각 *frame* 사이의 시간 간격이 일정하다는 것을 보장하지 않음.

사실 중간에 무슨 일이 생길지 모르므로 받는 쪽에서 항상 정해진 시간 간격으로 패킷을 받을 수는 없음. 하지만 *physical layer*에서의 처리에서는 이게 최선임. 그래서 *real-time* 데이터에 대한 주된 처리는 대체로 *application*에서 구현함. 중간에 패킷 전송이 끊기더라도, 소프트웨어가 신호를 적절히 생성하거나 늘리는 식으로 그 사이를 적절하게 메꿈.

3. Analog Transmission

3.1. Digital(data) to Analog(signal) Conversion

3.1.1. Digital(data) to Analog(signal) Conversion

1. Digital(data) to Analog(signal) Conversion

*Digital(data) to Analog(signal) Conversion*은 시스템에 존재하는 *digital data*를 *analog signal*로 변환해 전송하는 것임. 즉, 송신자는 *digital data*를 *analog data*로 변환해 전송하고, 수신자는 *analog data*를 *digital data*로 변환해 확인함.

*digital data*를 *analog signal*로 변환하는 것은 해당 *signal*을 *broadband transmission*(무선)으로 전송하기 위한 것임. 지금까지는 *baseband transmission*이었음.

digital to analog conversion 방법에는 ASK, FSK, PSK가 있음. *sine wave*는 3가지 변수(*amplitude*, *frequency*, *phase*)를 가지는데, ASK, FSK, PSK는 각각 대응되는 변수에 변화를 줌으로써 *digital data*를 *analog signal*로 표현함. QAM도 있는데, 이는 ASK와 PSK의 개념을 합친 것임.

2. Carrier Signal

*Carrier signal*은 *digital to analog conversion*에서 전송하는 *analog signal*의 *base signal*임.

ASK, FSK, PSK에서 전송하는 *analog signal*은 *carrier signal*이거나 *carrier signal*을 변형한 것임. 또한 *carrier signal*에 의해 *signal*이 특정 주파수 영역을 가지게 됨. 즉, *carrier signal*은 *baseband signal*을 *bandpass signal*로 변환하는 중간 다리임.

*carrier signal*의 주파수를 *carrier frequency*라고 하고, f_c 로 표기함. *carrier signal*은 $\sin(2\pi f_c t)$ 또는 $\cos(2\pi f_c t)$ 임. 여기서 2π 는 수학적 편의를 위해 사용하는 상수임(*period*를 계산해 보면 당연함.).

블루투스에는 주로 FSK를, WAN, wifi, LTE/5G에는 QAM를 사용함.

3.1.2. ASK

1. ASK

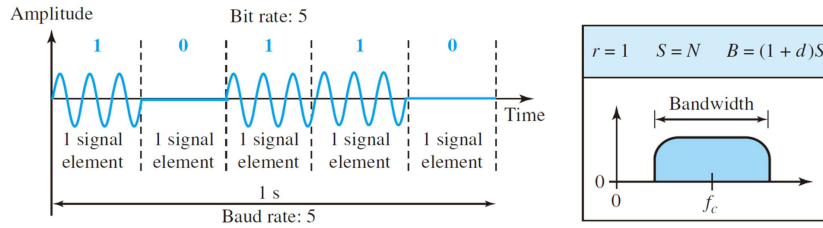
ASK(*Amplitude Shift Keying*)은 *carrier signal*의 *amplitude*만을 변형해 *digital data*를 *analog signal*로 나타내는 방식임.

2. BASK

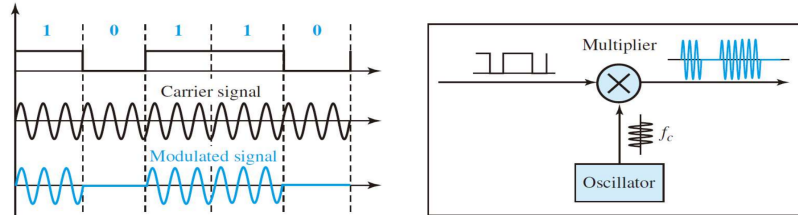
BASK(*Binary ASK*) 또는 OOK(*On-Off Keying*)은 2개의 *level*만을 사용하는 ASK임.

하나의 *signal level*은 *carrier signal*과 동일하고, 다른 하나는 0임.

이때 *bandwidth*는 f_c 를 중심으로 형성됨(왜 그런지는 아직 고려할 필요 없음.). 즉, *carrier signal*을 잘 지정해야 적절한 주파수 대역을 사용할 수 있음.



1을 carrier signal로, 0을 0으로 한다면 단순히 아래와 같이 unipolar NRZ를 사용하여 구현할 수 있음. 발진기(Oscillator)는 계속해서 carrier signal을 생성하는 부분이고, Multiplier는 단순 곱을 수행하는 부분임. 곱 연산의 결과에 따라 analog signal이 구성됨.



3.1.3. FSK

1. FSK

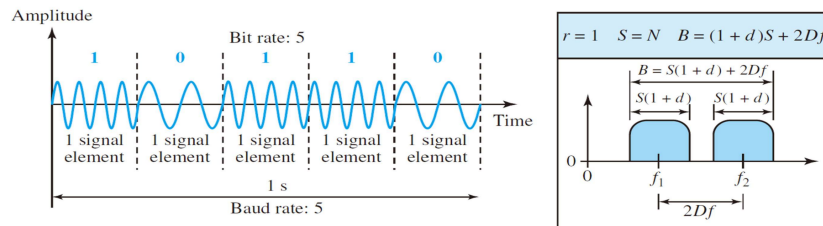
FSK(Frequency Shift Keying)은 frequency를 변형해(여러 frequency를 사용해) digital data를 analog signal로 나타내는 방식임.

여러 f_s 를 사용하므로 당연히 ASK에 비해 bandwidth가 넓음. 또한 수신자 측에서의 frequency 감지에 따른 하드웨어 복잡도가 높음. 또한 서로 다른 frequency를 많이 사용할수록 데이터 전송 속도가 빨라짐. 즉, 돈이 많고 bandwidth 제약에 여유롭다면 FSK로 빠른 통신을 구현할 수 있음.

2. BFSK

BFSK(Binary FSK)는 서로 다른 frequency를 가지는 2개의 carrier signal을 사용하는 FSK임.

이때 bandwidth는 두 carrier signal의 f_c 각각을 중심으로 형성됨. 즉, carrier signal을 잘 지정해야 적절한 주파수 대역을 사용할 수 있음.



3. Multi-level FSK

Multi-level FSK는 서로 다른 frequency를 가지는 3개 이상의 carrier signal을 사용하는 FSK임.

n 개의 carrier signal(f_s)을 사용한다면 하나의 analog signal로 $\log_2 n$ 만큼의 비트를 표현할 수 있음. 예를 들어, 한 번에 4비트를 표현하려면 16개의 frequency를 사용해야 함.

블루투스에서는 70여개의 frequency를 사용하고, 블루투스와 wifi는 서로 주파수 대역을 공유하고 있음. 이에 따라 블루투스 기기가 많거나 강한 wifi 신호가 있는 곳에서는 통신 충돌이 발생할 수 있음.

3.1.4. PSK

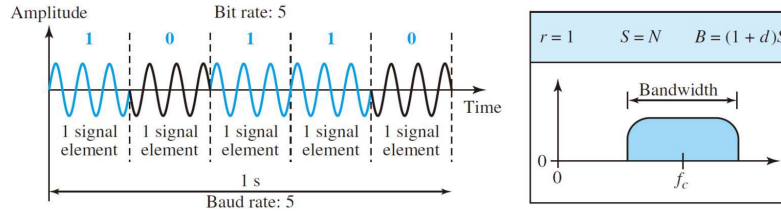
1. PSK

PSK(Phase Shift Keying)는 carrier signal의 phase만을 변형해 digital data를 analog signal로 나타내는 방식임.

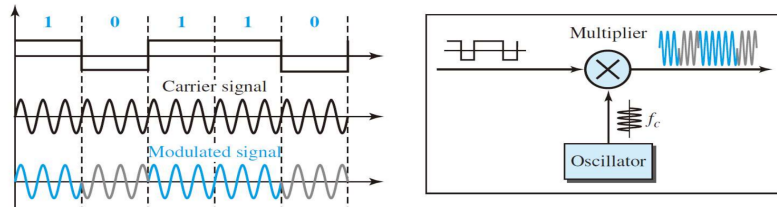
2. BPSK

BPSK(Binary PSK)는 phase가 0인 carrier signal과 phase가 π 인 carrier signal만을 사용하는 PSK임.

이때 bandwidth는 f_c 를 중심으로 형성됨. 즉, carrier signal을 잘 지정해야 적절한 주파수 대역을 사용할 수 있음.



1을 phase 0으로, 0을 phase π 로 한다면 단순히 아래와 같이 polar NRZ를 사용하여 구현할 수 있음. 발진기(Oscillator)는 계속해서 carrier signal을 생성하는 부분이고, Multiplier는 단순 곱을 수행하는 부분임. 곱 연산의 결과에 따라 analog signal이 구성됨. -1이 곱해지면 phase를 π 로 한 것이 됨.



3. Quadrature PSK

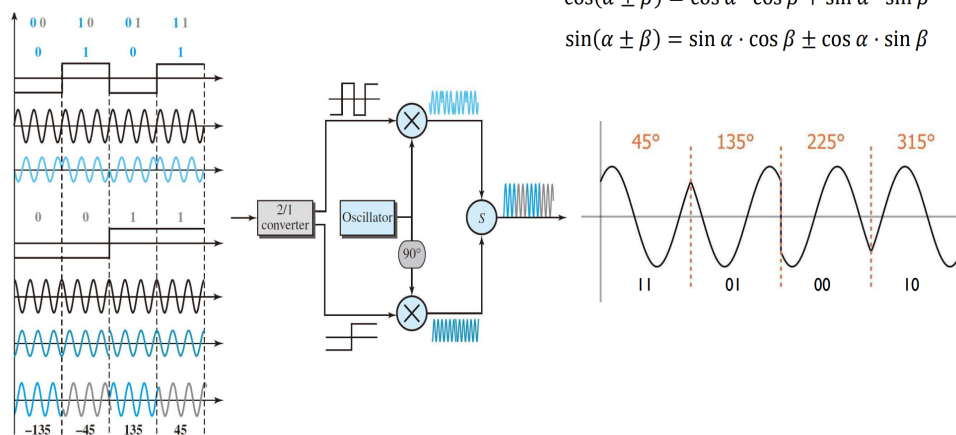
Quadrature PSK는 2개의 BPSK를 동시에 사용하는 PSK로, in-phase signal과 quadrature signal을 carrier signal로 사용하는 PSK임.

각 signal element는 $\frac{1}{4}\pi$, $\frac{3}{4}\pi$, $\frac{5}{4}\pi$, $\frac{7}{4}\pi$ 의 4가지 phase 중 하나를 가지게 됨. 즉, 하나의 signal element로 2개의 비트를 전송할 수 있음.

quadrature PSK에서 비트에 따라 적절한 analog signal을 생성하는 장치의 구현은 아래와 같음. 비트 값이 00/01/10/11 중 무엇인지에 따라 두 가지 digital signal이 생성됨. Oscillator에서는 펄스를 생성하여 sine wave와 cosine wave(phase가 $\frac{\pi}{2}$)로 사용됨. 곱셈 연산에 의해 생성된 결과를 합쳐서 전송함. $\sin(2\pi f_c t + \frac{1}{4}\pi)$ 등을 삼각함수의 덧셈정리로 정리하면, sine wave 2개(sine 하나와 cosine 하나)의 합성인 것을 알 수 있으므로, 각각 생성된 신호를 합쳐 적절한 신호를 만들 수 있음.

$$\cos(\alpha \pm \beta) = \cos \alpha \cdot \cos \beta \mp \sin \alpha \cdot \sin \beta$$

$$\sin(\alpha \pm \beta) = \sin \alpha \cdot \cos \beta \pm \cos \alpha \cdot \sin \beta$$



이때 oscillator에서 생성된 기본 signal을 I component(in-phase signal), $\frac{\pi}{2}$ 만큼 이동된 signal을 Q component(quadrature signal)이라고 함. 여기서 in-phase는 기준 신호와 위상이 같음을 의미함. quadrature는 직교(또는 $\frac{\pi}{2}$ 만큼의 위상 차이)를 의미하고, I component와 Q component는 서로 직교하는 signal임.

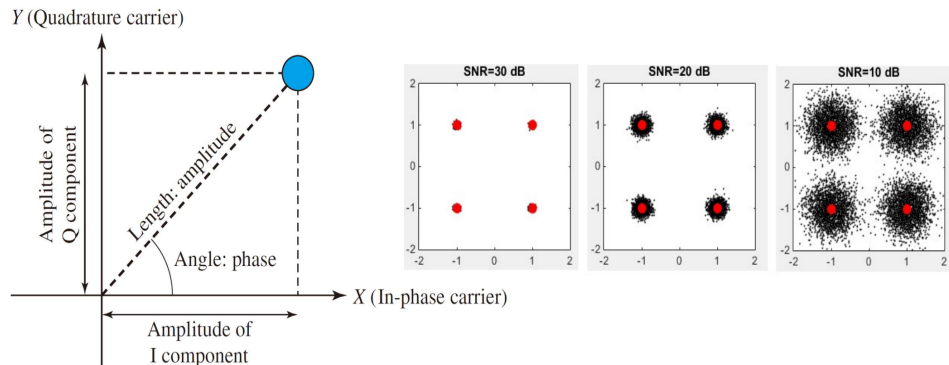
3.1.5. Constellation Diagram

1. Constellation Diagram

Constellation Diagram은 각 signal element를 amplitude와 phase에 따라 그래프에 하나의 점으로 표현한 diagram임. 이때의 한 점을 symbol이라고 함.

원점과 점 사이의 거리는 signal element(I와 Q의 합성 signal)의 amplitude이고, x축과의 각도는 phase 임. 가로축은 I component의 amplitude가 되고, 세로축은 Q component의 amplitude가 됨.

아래와 같이 통신에 따른 각 signal element의 분포를 확인할 수 있음. 각 점은 어느 영역에 속하는지에 따라 특정 signal로 해석됨. SNR이 높을수록(noise가 많이 걸수록) 다른 signal로 해석될 위험성이 커짐.

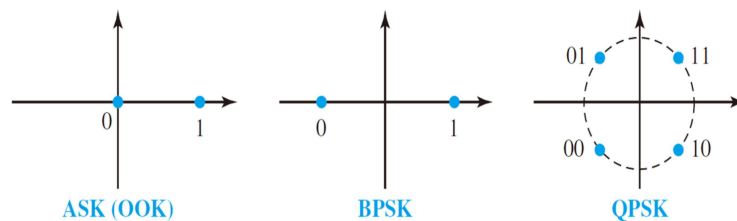


2. ASK, FSK, PSK로의 적용

ASK(BSK)에서는 in-phase signal과 0만을 사용하므로 constellation diagram에서 x축 위에 한 점과 원점 위에 한 점을 가짐.

BPSK에서는 in-phase signal만을 사용하지만, NRZ 신호와의 곱셈에 의해 signal이 x축을 기준으로 뒤집어지므로(phase가 π) constellation diagram에서 x축 위에 두 점을 가짐.

QPSK에서는 in-phase signal과 quadrature signal를 사용하고, 두 signal은 동일한 amplitude를 유지하면서 위상에 π 씩 변화를 줘서 총 4개의 점을 가지게 됨.

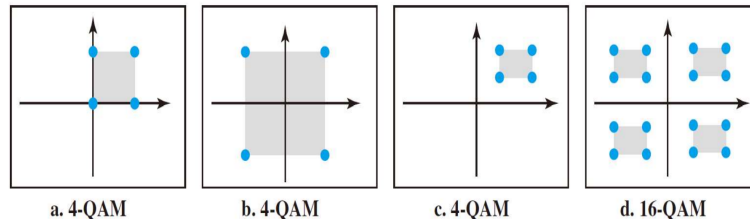


constellation diagram은 amplitude와 phase에 대한 diagram이므로, frequency에만 변화를 주는 FSK에는 잘 사용하지 않는 것 같다.

3.1.6. QAM

QAM(Quadrature Amplitude Modulation)은 in-phase signal과 quadrature signal을 carrier signal로 가지고, 각 carrier signal의 amplitude에 변화를 주는 방식임. 즉, signal ASK와 quadrature PSK의 개념을 합친 것.

QPSK에 amplitude 변화까지 적용한 것이므로, constellation diagram으로 표현해 보면 아래와 같은 상황들이 존재할 수 있음.



사용하는 점의 수에 따라 4-QAM, 16-QAM, 32-QAM 등으로 표기함. 당연히도 n 개의 점이 있다면 하나의 signal element는 $\log_2 n$ 개의 비트를 나타낼 수 있음. 숫자가 커질수록 phase와 amplitude 각각에 대한 구분도 어렵고 복잡도가 높음. 또한 amplitude가 커지므로 전력 소모량도 많아짐.

digital to analog conversion에서 가장 지배적으로 사용되는 방식임.

많은 점(구분)을 사용하는 것은 높은 성능을 보장하지만 cost가 많이 듦. 그래서 주로 하나의 시스템에서 각 방법끼리 전환하여 사용하도록 구현되어 있음. 특히 무선 통신의 경우 통신 환경이 계속 변화하는데, 통신 상태가 좋은 경우 64-QAM를 쓰고 그렇지 않은 경우 4-QAM를 쓰는 등으로 동작함. 최근 wifi에서는 1024-QAM까지도 사용한다고 함.

3.2. Analog(signal) to Analog(signal) Conversion

3.2.1. Analog(signal) to Analog(signal) Conversion

Analog(signal) to Analog(signal) Conversion은 digital data로의 변환 없이, 감지한 analog signal을 바로 analog signal로 전송하는 것임.

analog to analog conversion에서의 input signal을 modulating signal이라고 함. analog to analog conversion에서도 carrier signal이 사용되는데, 해당 conversion은 carrier signal을 통해 modulating signal을 표현하는 기법임.

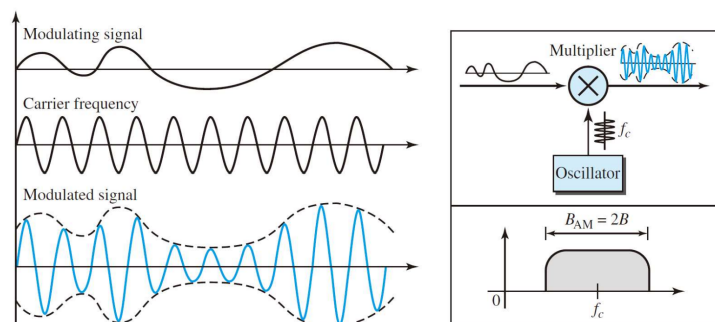
3.2.2. AM/FM/PM

analog to analog conversion의 방법으로는 AM, FM, PM이 있음.

1. AM

AM(Amplitude Modulation)은 carrier signal의 amplitude만을 변형해 analog signal을 또 다른 analog signal로서 전송하는 방식임.

modulating signal과 carrier signal을 곱해서, carrier signal이 modulating signal의 모양과 유사한 형태의 amplitude를 가지도록 함. 이때 bandwidth는 f_s 를 중심으로 형성됨. AM은 아래와 같이 구현됨.



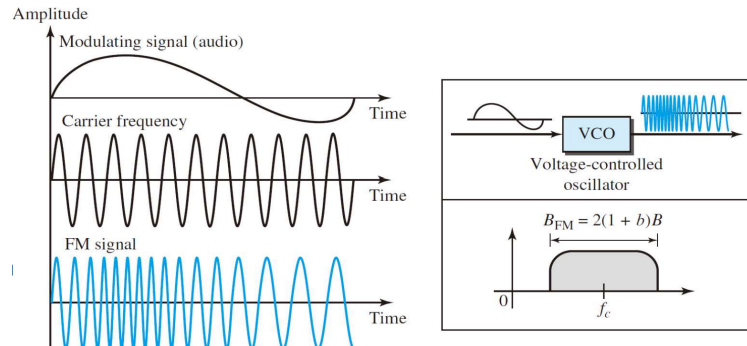
각 부분의 amplitude 극댓값들을 모아서 원본 signal을 복원할 수 있음. 문제는, 거리가 멀어질수록 attenuation이 발생하는데, attenuation의 정도가 심해 위와 아래의 신호가 서로 겹치게 되면 더 왜곡됨. 즉, 장거리 통신에 불리한 방법임.

2. FM

FM(Frequency Modulation)은 carrier signal의 frequency만을 변형해 analog signal을 또 다른 analog

signal로서 전송하는 방식임.

VCO는 입력 전압 변화에 따라 frequency에 변화를 주는 장치로, modulating signal의 전압 수준에 따라 carrier signal이 다른 frequency를 가지도록 함. 이때 bandwidth는 f_s 를 중심으로 형성됨. FM은 아래와 같이 구현됨.

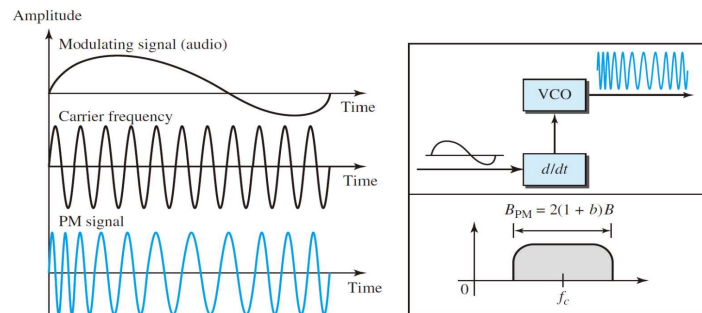


거리가 멀어짐에 따라 attenuation이 발생해도, frequency에 따라 신호를 구분하므로 영향이 적음. 그래서 장거리 전송에 주로 사용됨.

3. PM

PM(Phase Modulation)은 carrier signal의 phase만을 변형해 analog signal을 또 다른 analog signal로서 전송하는 방식임.

modulating signal의 전압 수준을 미분하고 VCO에 넣는 방식으로 구현되는데, 자세한 내용은 설명하지 않음. 이때 bandwidth는 f_s 를 중심으로 형성됨. FM은 아래와 같이 구현됨. PM은 아래와 같이 구현됨.



위와 같이 analog signal을 바로 analog signal로 전송할 수도 있지만, digital data로 변환해야 저장, 가공, 암호화 등을 수행할 수 있음. 실제로 digital data를 전송할 때는 압축하고 암호화하는 과정이 포함되어 있음.

4. Bandwidth Utilization

실세계에서의 link는 제한된 bandwidth를 가지고 있음. data communication에서 bandwidth는 가장 중요한 자원 중 하나이므로 이를 잘 활용하는 것, 즉 bandwidth utilization이 중요함. 대표적인 bandwidth utilization 방법으로는 multiplexing과 spread spectrum이 있음. multiplexing은 효율 개선의 측면에서 bandwidth를 활용하고, spread spectrum은 보안과 안정성의 측면에서 bandwidth를 활용함.

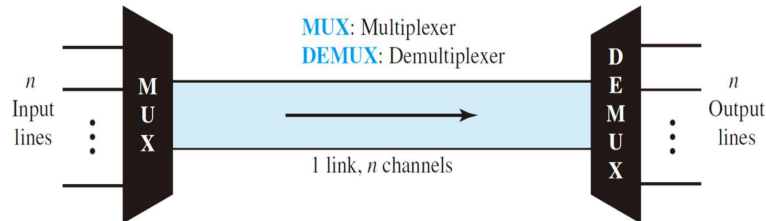
4.1. Multiplexing

4.1.1. Multiplexing

Multiplexing은 여러 channel을 하나로 합쳐 하나의 data link에서 여러 signal에 대한 동시 전송을 가능하게 하는 기법으로, link에서의 전송 효율을 개선함.

여기서 *link*는 물리적인 경로를 의미하고, *channel*은 *line* 한 쌍(input/output)의 송수신을 담당하는 *link*의 한 부분임. *multiplexing*이 적용된 시스템에서는 여러 개의 *line*이 한 *link*의 *bandwidth*를 공유하여, 각 *channel*을 통해 전송됨.

이때 각 *line*을 합쳐 하나의 *link*로 전송하는 부분을 *MUX*(Multiplexer), *link*를 각 *line*으로 분리하는 부분을 *DEMUX*(Demultiplexer)라고 함.



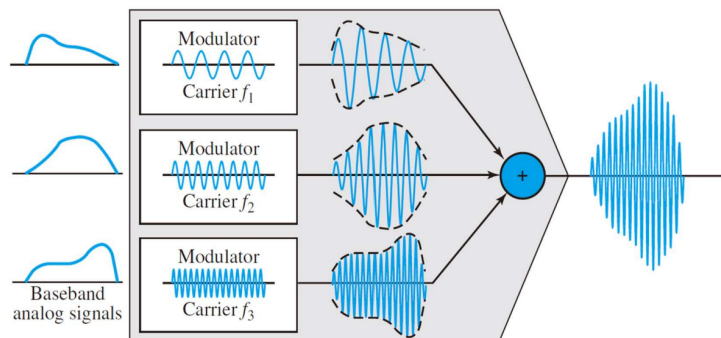
4.2. Multiplexing 기법들

4.2.1. FDM

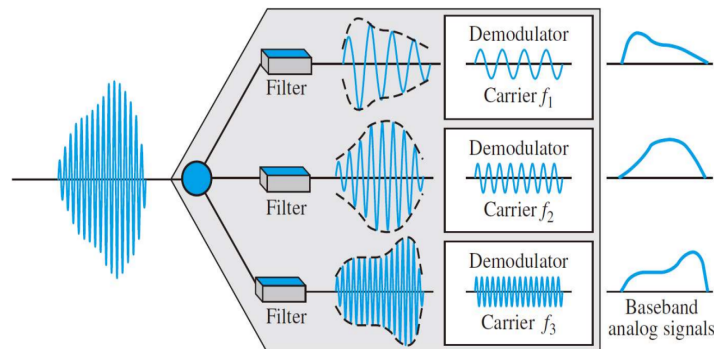
Frequency-Division Multiplexing(FDM)은 *analog signal*을 조합하는 *analog multiplexing* 기법임. 즉, *signal*을 서로 겹치지 않는 *frequency* 대역으로 옮겨 한꺼번에 전송함.

당연하게도 *link*의 *bandwidth*가 각 *signal*들의 *bandwidth*를 합친 것보다 클 때 사용이 가능함.

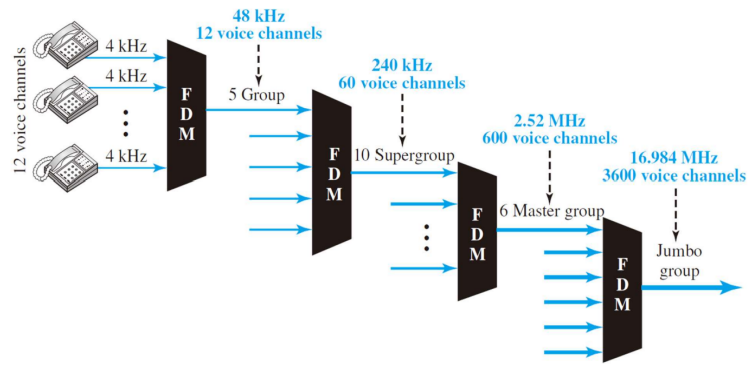
각 *signal*은 *MUX*에서 서로 다른 *carrier signal*을 통해 변조되고, 이후 합쳐서 *link*를 통해 전송됨. 이때 변조된 각 *signal*이 구성하는 *bandwidth*는 서로 겹치지 않도록 특정 간격만큼 떨어지도록 구현되는데, 이때의 사용되지 않는 *bandwidth* 간격을 *보호대역*(Guard Band)이라고 함.



*link*의 끝에 도달한 *signal*은 *filter*에 의해 분해되고, *DEMUX*에 의해 원본 신호로 복구됨.



FDM은 과거 아날로그 전화망에서 주로 사용되었음. 아래와 같이 점점 더 큰 *bandwidth*로 grouping하여 효율을 최대화했음.

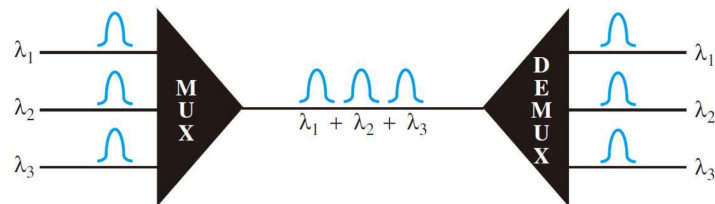


4.2.2. WDM

*Wavelength-Division Multiplexing(WDM)*은 광섬유 *signal*을 조합하기 위한 *analog multiplexing* 기법임. 즉, 각 *signal*을 서로 다른 빛 신호를 사용하여 한꺼번에 전송함. $f = \frac{c}{\lambda}$ 이므로 *frequency*에 차이를 준 것과 다르지 않음.

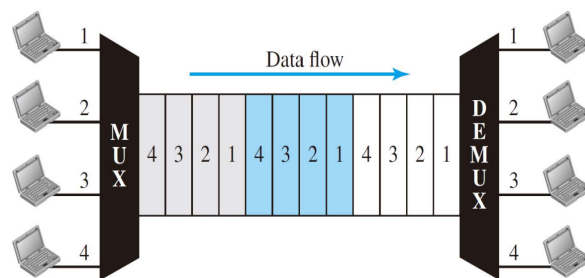
이는 광섬유의 고속 전송률을 활용하기 위한 방식으로, 빛 신호를 사용하고 *frequency*가 매우 높다는 점 외에는 *FDM*와 동일함.

이때 빛 신호의 결합과 분리에 대한 기술적인 구현은 복잡하지만, 그 원리는 단순히 프리즘을 응용한 것임.



4.2.3. TDM : Synchronous TDM

*Time-Division Multiplexing(TDM)*은 여러 개의 저속 *channel*을 하나의 고속 *channel*로 조합하는 *digital multiplexing* 기법임. 즉, 각 *line*의 *data*가 *link*를 특정 시간 간격으로 나누어 전송됨. *link*의 높은 *bandwidth*를 활용하는 것.



1. Synchronous TDM

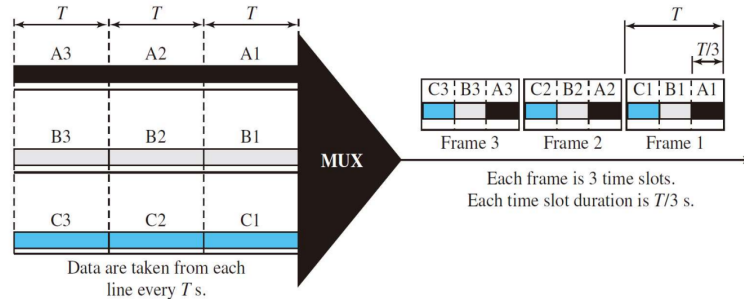
*Synchronous TDM*은 각 *line*에 대한 *unit*으로 고정된 *frame*을 구성하여 전송하는 방식임.

*unit*은 *input line data*에 대한 전송 단위임. *link*는 각 *line*별로 하나의 *unit*을 가져와서 하나의 묶음으로 처리하는데, 이를 *frame*이라고 함. 이때 *unit*이 차지하는 시간 간격을 *time slot*이라고 하는데, *MUX*와 연결된 *line*에서의 *time slot*을 *input time slot*, *frame*에서의 *time slot*을 *output time slot*이라고 함.

*input time slot*이 T 이고, 총 n 개의 *line*이 *MUX*에 연결되어 있다면, *frame*은 T 시간마다 n 개의 *unit*을 전송해야 함. 즉, *output time slot*은 최소 $\frac{T}{n}$ 여야 하고, 다시 말해 *link*는 *line*보다 *data rate*이 최소 n 배

빨라야 함.

이때 각 time slot에 대응되는 line은 고정되어 있음. 이에 따라 해당 line이 어떤 시점에 전송할 data가 없으면 empty slot이 발생하게 됨. empty slot이 많이 발생할수록 synchronous TDM의 성능은 떨어지게 됨. empty slot 문제는 statistical TDM으로 해결이 가능함.

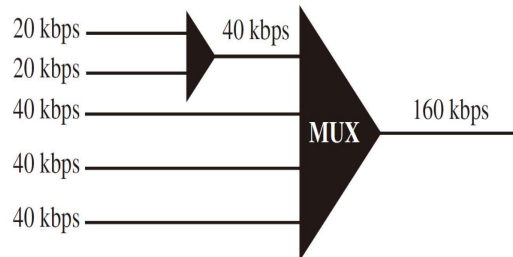


2. Synchronous TDM : data rate이 다른 경우

앞에서 설명한 synchronous TDM은 각 line의 data rate이 같다고 가정한 것임. 만약 그렇지 않은 상황에는 아래와 같은 방식으로 구현할 수 있음.

1) Multilevel Multiplexing

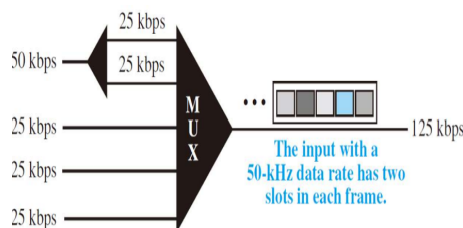
Multilevel Multiplexing은 각 line의 data rate이 서로 정수배일 때 MUX를 사용해 line을 합쳐 data rate을 맞추는 기법임.



2) Multiple-Slot Allocation

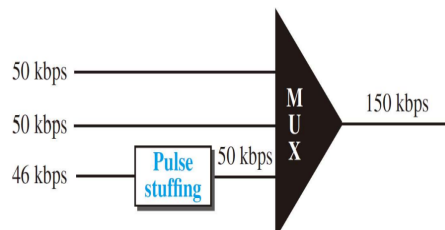
Multiple-Slot Allocation은 각 line의 data rate이 서로 정수배일 때 DEMUX를 사용해 line을 분리하여 data rate을 맞추는 기법임.

이 경우 data rate이 빠른 line은 하나의 frame에 여러 unit을 넣게 됨.



3) Pulse Stuffing

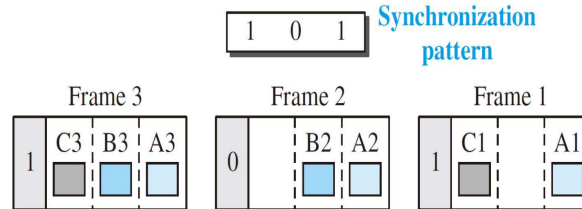
Pulse Stuffing은 각 line의 data rate이 서로 정수배가 아닐 때, data에 dummy bit를 추가하여 느린 data rate을 가지는 line을 가장 빠른 data rate 값으로 맞추는 기법임.



3. Frame Synchronizing

Frame Synchronizing은 MUX와 DEMUX 사이의 synchronization 기법임.

주로 각 frame 앞에 framing bit라고 하는 하나의 비트를 붙이는 식으로 구현함. framing bit에는 synchronization pattern이 정의되어 있어, 해당 pattern에 맞춰 bit 값이 지정됨.

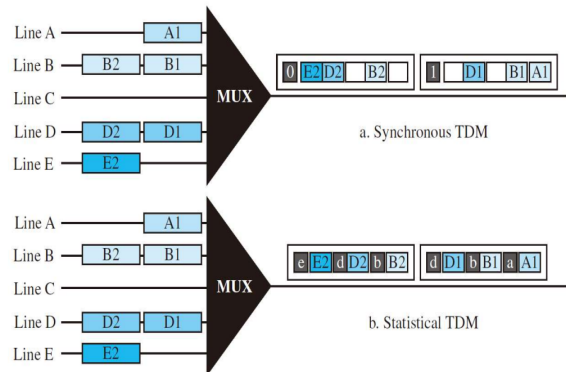


MUX와 DEMUX가 잘 맞아야 각 time slot을 적절하게 읽고 쓸 수 있으므로, frame synchronizing은 TDM에서 가장 중요한 과제 중 하나임.

4.2.4. TDM : Statistical TDM

Statistical TDM은 각 line에 대한 unit으로 가변적인 길이의 frame을 구성하여 전송하는 방식임. 즉, 미리 정의된 slot이 존재하지 않고, 전송 시점에 존재하는 unit들만으로 frame을 구성해 전송함.

이를 위해 각 slot에는 해당 unit이 DEMUX에서 어느 line으로 전달되는지에 대한 addressing 처리가 되어야 함. n 개의 line이 존재한다면, $\log_2 n$ 개의 bit를 frame 앞에 붙이는 식으로 구현할 수 있음.



4.3. Spread Spectrum

4.3.1. Spread Spectrum

Spread Spectrum은 redundancy의 추가로 bandwidth를 확장하여($B_{SS} \gg B$) link에서의 보안과 안정성을 개선하는 기법임. 즉, 기존의 data를 redundancy로 포장하여 안전하게 전송하는 것.

bandwidth를 확장하므로 기존의 필요 bandwidth보다 더 넓은 bandwidth를 사용해야 하고, bandwidth의 확장은 원래의 signal에 독립적으로 수행되어야 함.

무선 통신(LAN, WAN)에서 활용하는 기법임. 무선 통신에서는 공기를 transmission medium으로 사용하므로, 효율보다는 보안과 안정성을 우선 확보할 필요성이 있음. 또한 무선 통신이므로 analog signal에 대한 것임.

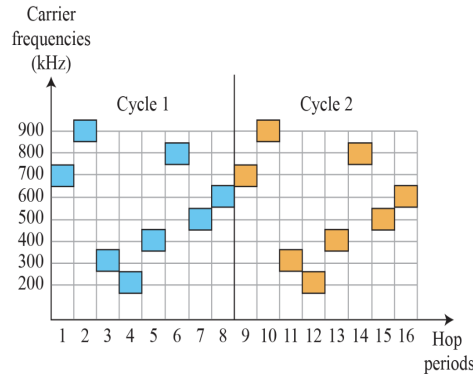
4.4. Spread Spectrum 기법들

4.4.1. FHSS

1. FHSS

Frequency Hopping Spread Spectrum(FHSS)은 M 개의 서로 다른 frequency들을 가지고 원본(Source) signal을 변조하여 전송하는 방식임.

FHSS는 source signal을 Hopping Period(T_h)마다 다른 frequency를 사용하여 변조하여 전송함. 즉, 어떤 시점에 한 frequency를 사용하여 변조되었다가, 특정 시간 간격이 지나면 다른 frequency를 사용하여 변조됨. 이때의 시간 간격을 hopping period라고 함. 여러 개의 frequency로 변조하여 전송하므로 bandwidth가 확장됨($B_{FHSS} \gg B$).



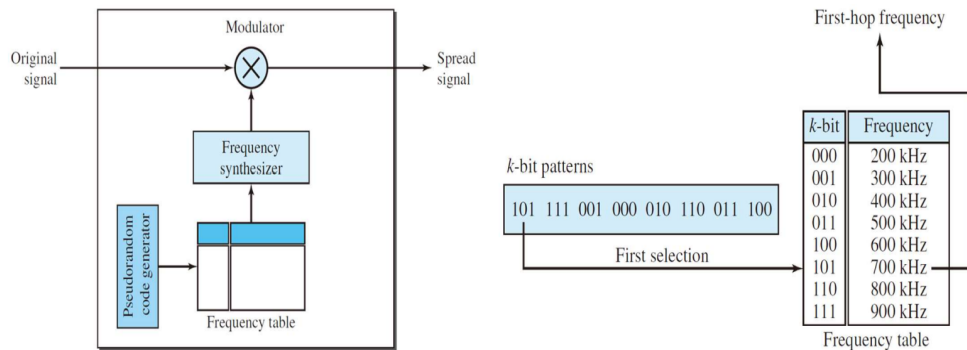
2. 구현

충돌 방지와 보안을 고려하면, 변조에 어떤 frequency를 사용할 지가 무작위로 결정되어야 함. 이를 위해 PN과 frequency table, frequency synthesizer를 사용함.

Pseudorandom Code Generator(Pseudorandom Noise, PN)는 무작위 bit열을 생성하는 하드웨어임. 물론 때의 무작위는 수학적으로 구현된 것으로, 당연하게도 완전한 무작위(truerandom)는 아님. PN에서 생성한 특정 bit열은 주기적으로 반복되어 사용됨.

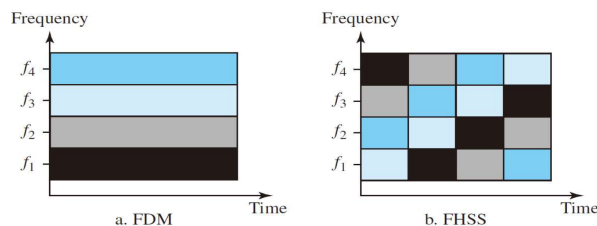
Frequency Table은 bit열과 frequency에 대한 매핑을 저장하는 table임. PN이 bit열을 생성하면 매핑된 frequency 값을 내보냄.

Frequency Synthesizer는 frequency table로부터 frequency 값을 받아 해당 frequency의 signal을 전송함.



3. Bandwidth Sharing

FHSS는 hopping period마다 매번 다른 frequency를 사용하여 변조되므로, 총 M 개의 서로 다른 frequency를 사용한다면 해당 link에서 총 M 개의 channel을 동시에 전송할 수 있음.



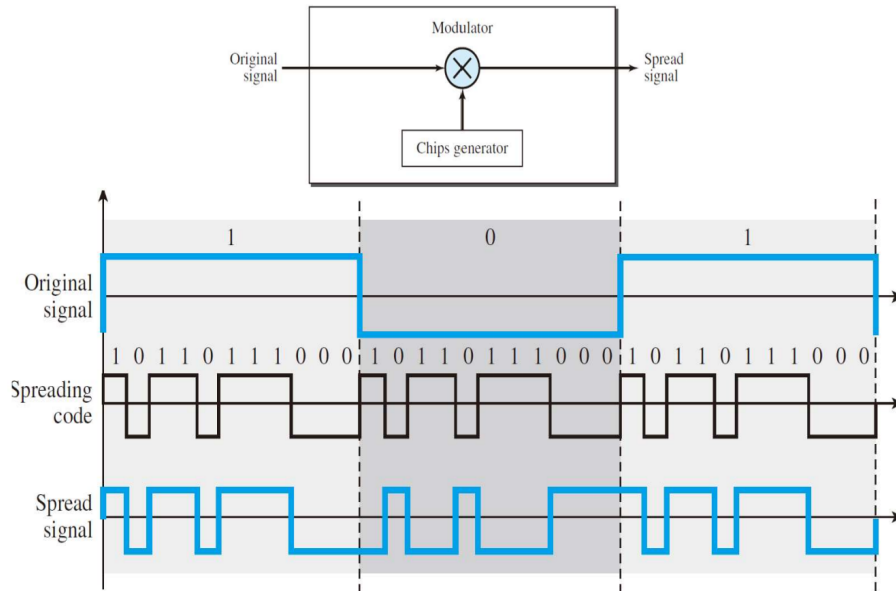
bluetooth가 이 방식을 사용하는데, 70개 가량의 frequency들을 활용함.

4.4.2. DSSS

*Direct Sequence Spread Spectrum(DSSS)*는 각 *data bit*를 *chip*을 사용해 변조하여 전송하는 방식임.

*chip*은 길이가 n 인 *bit*열로, *chip*에 대한 *bit rate*는 기존 *data bit*의 *bit rate*의 n 배임.

*chip*을 알고 있어야 원래의 *data bit*을 알아낼 수 있으므로 보안 상 이점이 있음. 참고로, 만약 서로 다른 *chip*을 사용하여 변조한 *signal*을 합쳤다가 다시 분해할 수 있는 환경이라면 *DSSS*에서도 *bandwidth sharing*을 적용할 수 있음.



3G 등에서 사용되던 방식임.

5. Switching

5.1. Switching

5.1.1. Switching

1. Switching

*Switching*은 *switch*를 활용하여 *device*들끼리의 연결을 관리하는 기법임.

스위치(*Switch*)는 여러 개의 *device*와 연결되어 여러 *link*들 간의 일시적인 연결을 생성할 수 있는 장치임.

복수의 *device*로 구성된 *network*에서, 특정 *device*끼리는 1대1 *communication*을 할 수 있어야 함. 이는 단순히 모든 *device*에 대해 *point-to-point*인 *mesh topology*로 구현할 수도 있지만 *network*의 규모가 조금만 커져도 자원 낭비가 너무 심해지므로, 대신 *switching*으로 구현할 수 있음.

switching 방식으로는 *circuit-switching*, *packet-switching*, *message-switching*이 대표적이고, *packet-switching*에는 *Datagram approach*, *Virtual-circuit approach*가 있음. 오늘날에는 *circuit-switching*, *packet-switching*이 주로 사용되므로, *message-switching*은 정리하지 않음.

2. TCP/IP 계층에 따른 Switching 적용

*switching*은 *TCP/IP*의 각 계층에서 구현됨. 계층 별 구현 대상은 아래와 같음.

physical layer : *circuit-switching*. (*packet* 처리 불가능.)

data-link layer : *virtual-circuit approach*. (*packet(frame)* 처리 가능.)

Network layer : *virtual-circuit/datagram approach*. (*packet* 처리 가능.)

5.1.2. Circuit-switching

1. Circuit-switching

Circuit-switching은 여러 link와 switch로 두 지국(Station) 사이에 물리적인 전용 경로(dedicated path)를 제공하는 방식임.

circuit-switching은 dedicated path를 제공하므로, 만약 6개의 input과 6개의 output이 연결되어 있다면 해당 network는 총 36개의 path를 제공할 수 있어야 함.

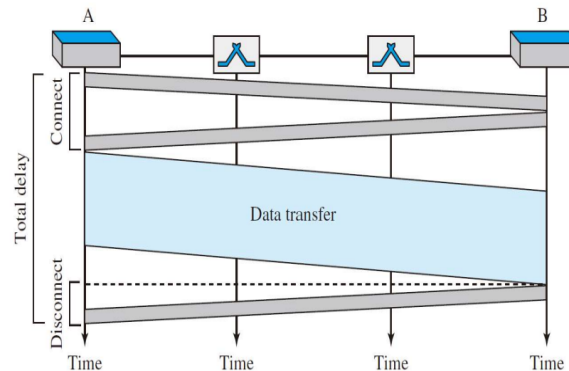
circuit-switched network에서는 아래의 과정을 통해 두 station이 통신하게 됨.

A가 B에게 통신하려고 하는 상황을 가정하자.

1. Setup Phase : A는 B와 각 switch들에게 연결 요청을 보내고 승인받음. dedicated path가 구성됨.

2. Data-transfer Phase : A가 B에게 통신함.

3. Teardown Phase : dedicated path가 해제됨.



2. 장단점

dedicated path를 사용하므로, connection이 생성된 이후에는 통신에 있어서의 delay가 최소화됨.

dedicated path를 구성하고 통신 중에는 이를 해제할 수 없기 때문에, 한 connection이 사용 중이면 다른 connection이 해당 자원을 활용할 수 없음. 전화 network 등에서는 전화를 끊으면 path를 해제하면 되지만, 컴퓨터 간 통신 등에서는 장시간 connection을 유지해야 하는 경우가 존재하므로 비효율적일 수 있음.

각 line에서의 실질적인 signal 전송에는 앞에서 배운 line coding 등이 적용됨.

하나의 link가 충분한 bandwidth를 가지고 있다면, FDM/TDM 등을 적용하여 여러 channel로 사용됨.

5.1.3. Packet-switching

Packet-switching은 두 device가 packet 단위로 데이터를 주고받는 방식임. 이때 packet에는 미리 정해진 자원 할당이나 우선순위가 존재하지 않고, 필요에 따라 자원이 할당되며 선착순으로 전송됨.

packet-switching에는 datagram approach와 virtual-circuit approach가 있음.

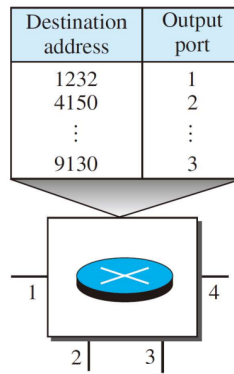
5.1.4. Datagram approach

1. Datagram Approach

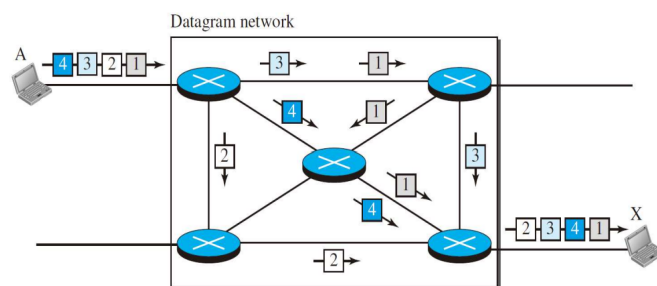
Datagram approach는 각 packet(datagram)을 독립적으로 취급하는 비연결성(Connectionless) 방식임.

connection을 유지하지 않으므로 비연결성이라고 함.

datagram network에서의 switch를 라우터(Router)라고 함. 각 router는 Routing Table을 가짐. 이는 destination address와 그에 대응되는 output port에 대한 table로, 주기적으로 갱신됨. packet의 header에는 destination address(최종 목적지의 주소. 주로 ip 주소)가 있어서, router에서 어떤 port로 전송될지 결정될 수 있음.



각 packet에 대한 path는 전송되는 과정에서 동적으로 결정됨. 이에 따라 각 packet은 순서가 뒤바뀐 채로 도착하거나, 일정하지 않은 시간 간격을 두고 도착하거나, 목적지에 도착하지 못할 수도 있음. 이에 대한 처리는 상위 layer에게 책임이 있음.

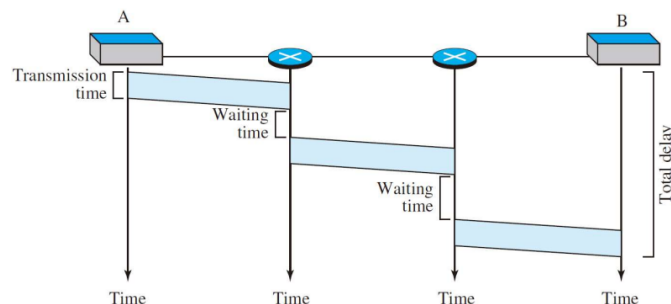


원래 datagram switching은 network layer에서 처리되는데, 여기에서는 다른 switching 방식과의 비교를 위해 정리함.

2. 장단점

packet이 전송되는 시점에 자원이 필요에 따라 동적으로 할당되므로, circuit-switching 방식에 비해 자원 활용에 대한 효율이 좋음.

각 packet은 우선순위를 가지지 않고 router에 도착한 순서대로 처리되므로, delay/waiting time이 발생할 수 있음.



5.1.5. Virtual-circuit approach

1. Virtual-circuit approach

Virtual-circuit approach는 circuit-switching과 datagram approach의 특성을 모두 가지고 있는 방식임.

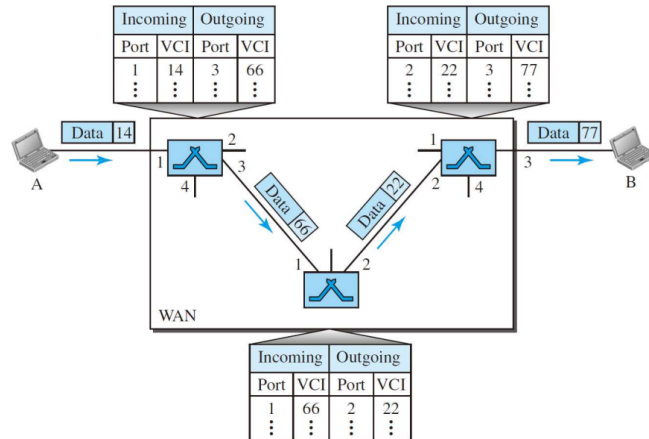
circuit-switching처럼 setup/data-transfer/teardown phase의 단계를 거쳐 처리되고, datagram switching처럼 header에 도착지 주소를 담고 있는 packet(frame)을 전송함. 실제로 물리적인 dedicated path를 구성하지는 않지만, setup phase에서 path가 결정되면 해당 path만을 사용함.

virtual-circuit approach를 사용하면 전송 전에 얼마만큼의 자원이 할당될지를 파악할 수 있음. 이때 setup/teardown phase에 따른 처리 시간이 소요되지만, data-transfer phase에서 delay는 발생하지 않음.

2. VCI

virtual-circuit approach에서 도착지 주소는 Virtual-circuit Identifier(VCI)라고 하는 식별자로 구분함. VCI는 local jurisdiction을 가짐. 즉, 이는 최종 목적지의 주소(end-to-end jurisdiction)가 아니라 바로 다음 switch로의 주소임.

각 switch는 VCI와 output port에 대한 매핑이 저장된 table을 가지고 있음. 이는 setup phase에서 그 내용이 구성됨. data-transfer phase에는 packet에 포함된 VCI 값과 해당 switch가 가진 table의 내용으로 각 packet을 어디로 보낼지가 결정됨. 이때 packet이 가진 VCI는 switch를 이동하며 계속 갱신됨.



Part III

Data-link Layer

1. 서론

지금까지 physical layer에서 살펴본 것은 하나의 link에서 물리적으로 전송을 어떻게 할 것인가였음. 이제부터는 그 윗 계층인 data-link layer에 대해 알아보자.

1.1. 서론

1.1.1. Data-Link Layer

1. 역할

data-link layer에서는 두 node 사이에 존재하는 하나의 link에서의 data communication만을 고려함. 즉, data-link layer에서 frame에 포함시키는 정보는 node와 node 사이 link에서의 통신에만 필요한 데이터임.

Node는 data communication에서의 device임. router, switch, source/destination 등을 포함함. 이때 첫 번째 노드를 source host, 마지막 노드를 destination host라고도 함.

link는 하나의 node에서 다른 node로의 연결임. 즉, 전체 path가 아닌, node-to-node임.

이때 개별 link에는 서로 다른 protocol이나 기법이 사용될 수 있음. 예를 들어, 스마트폰-기지국에는 LTE/5G를, 기지국에서 다른 곳으로는 ethernet을 사용하는 식으로 구현됨.

2. Service

data-link layer는 physical layer의 service를 활용하여 network layer에 service를 제공함.

data-link layer에서 제공하는 service로는 아래와 같은 것들이 있음.

1) Framing

data-link layer에서는 datagram을 frame으로 encapsulation하고, frame을 datagram으로 decapsulation함.

2) Flow Control

data 전송의 흐름을 조정함.

3) Error Control

전자기 신호를 사용하여 통신하기 때문에 error가 발생할 수 있는데, 이에 대한 detection과 correction을 수행함.

4) Congestion Control

다중 접속 등에 의한 충돌 관리 등으로 혼잡도(congestion)을 조정함.

1.1.2. Sublayers

data-link layer는 DLC와 MAC을 sublayer로 가짐.

1. point-to-point link vs. broadcast link

point-to-point link는 두 node를 1대1로 연결하는 link로, medium의 모든 자원을 활용하여 통신함. 반면 broadcast(multipoint) link는 여러 node와 연결된 link로, medium의 일부 자원을 활용하여 통신함.

2. DLC/MAC

Data-link Control(DLC)은 point-to-point link와 broadcast link 모두에서 공통적으로 처리되어야 하는 부분을 포함하는 sublayer임.

Media Access Control(MAC)은 broadcast link에서 추가적으로 처리되어야 하는 부분을 포함하는 sub-layer로, 충돌에 대한 처리 등을 함.

bus topology 등을 생각하면 broadcast link에서 충돌에 대한 별도의 처리는 당연히 필수적임. 물론 point-to-point에서도 서로 전송하려고 하는 경우 문제가 될 수 있지만, 그럴 때는 대체로 2개의 선을 사용하여 입출력을 분리하여 구현함. 그렇게 구현되어 있지 않다면 MAC이 필요함.

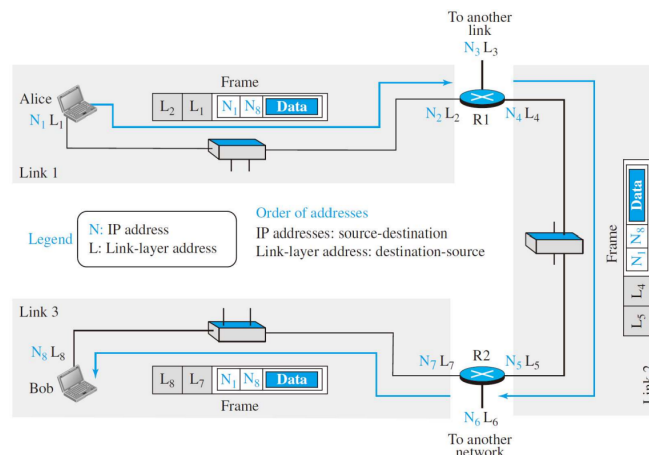
1.2. Addressing

1.2.1. MAC address

1. MAC address

MAC address는 data-link layer에서 사용하는 address로, physical address, link address라고도 함.

data-link layer에서는 하나의 link에서의 통신만을 고려함. MAC address는 하나의 link에서 출발 node와 도착 node에 대한 address이고, 하나의 link에서 다음 link로 넘어갈 때 그 값이 새롭게 지정됨.



2. 종류

MAC address에는 3가지 종류가 있음.

1) Unicast Address : link에 존재하는 하나의 node에 대한 address.

ethernet에서 사용되는 unicast address는 48bit 크기이며, 이는 주로 12개의 16진수를 :으로 구분하여 표기함. 예를 들어, A3:34:45:11:92:F1 등으로 표기함.

2) Multicast Address : link에 존재하는 여러 node에 대한 address.

3) Broadcast Address : link에 존재하는 모든 node에 대한 address.

ethernet에서 사용되는 broadcast address는 48bit 크기이며, 모든 값이 1임. 즉, FF:FF:FF:FF:FF:FF 임.

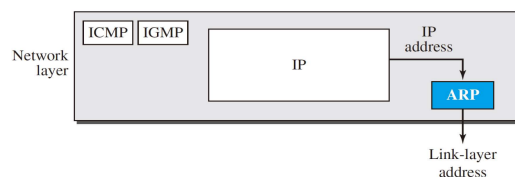
ethernet은 현재 physical/data-link layer에서 가장 일반적으로 사용되는 protocol임.

1.2.2. ARP

1. ARP

Address Resolution Protocol(ARP)는 IP address에 대해, 매핑된 MAC address(data-link address)을 찾는 network layer의 보조 프로토콜(Auxiliary Protocol)임.

즉, 어떤 node가 network 내에 존재하는 다른 node의 ip address만을 알고 있을 때, 각 link에 대한 MAC address를 찾을 때 사용됨.



2. Packet Format

ARP packet의 format은 아래와 같음.

0		8		16		31	
Hardware Type				Protocol Type			
Hardware length		Protocol length		Operation Request:1, Reply:2			
Source hardware address							
Source protocol address							
Destination hardware address (Empty in request)							
Destination protocol address							

Hardware: LAN or WAN protocol

Protocol: Network-layer protocol

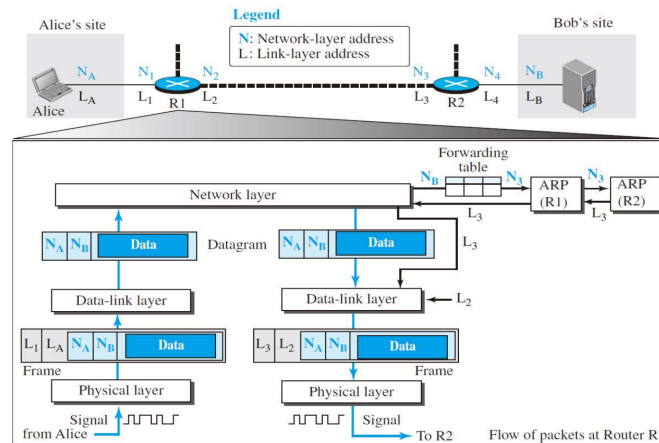
여기에서 hardware는 data-link layer를 의미하고, protocol은 network layer를 의미함. 또한 type에는 각 계층에서 사용된 protocol을 나타내는 정보를(ex. IPv4는 0x0800임.), address는 해당 address를 저장함.

3. 동작 과정

어떤 packet을 destination으로 전달하려고 할 때, IP address만 알고 path를 모르면 ARP가 동작함. ARP에서는 ARP request packet을 link에 존재하는 모든 node에(broadcast) 전송함. 해당 packet에는 송신자의 IP/MAC address와, 수신자(목적지)의 IP address가 포함되어 있음.

ARP request packet을 전달받은 node 중 path와 관련 없는 node에서는 해당 packet이 무시되고, path와 관련이 있는 node에서는 자신의 ARP를 통해 ARP response packet을 전송함. ARP request packet에는 해당 node의 MAC address가 포함되어 있음.

ARP response packet을 전달받았으면 포함된 MAC address로 원래 전달하려던 packet을 전달함. 해당 packet을 받은 node에서 다시 ARP를 사용하여 다음 node로 packet을 전달할 수 있음.



물론 각 node에서의 table/cache 등의 사용, proxy ARP, destination의 위치 등에 따른 구체적인 동작과 구현 방법이 존재하지만, 이를 모두 정리하지는 않음.

2. Error Detection and Correction

2.1. 서론

2.1.1. Error

오류(Error)는 data communication에서 전송 중 간섭(Interference) 등에 의해 발생하는 data의 변경임. error로는 single bit error와 burst error가 있음.

- 1) Single Bit Error는 data 단위 중 하나의 bit에만 변경이 발생한 경우를 말함. 일반적으로 하나의 bit에 해당하는 시간 간격은 매우 짧기 때문에, single bit error는 잘 일어나지 않음.
- 2) Burst Error는 data 단위 중 2개 이상의 연속적인 bit들이 변경되는 경우를 말함.

2.1.2. Modular Arithmetic

모듈러 연산(Modular Arithmetic)은 나머지 연산을 기반으로 한 산술 체계임. a 와 b 에 대한 modular arithmetic은 $a \bmod b$ 로 표기함.

$\bmod-N$ arithmetic에서는 0부터 $N-1$ 까지의 제한된 정수만을 사용함. 해당 범위를 벗어난 값은 해당 값에 N 으로 나머지 연산을 한 것을 값으로 함.

독특한 점은, $\bmod-2$ 에서는 덧셈과 뺄셈이 같은 결과를 도출하고, 이는 XOR 연산과 동일하다는 것임. 당연히 $\bmod-N$ 에서도 적용되는 성질은 아님.

참고로 $\bmod-N$ 에서 연산 결과가 음수로 나왔을 때 나머지 연산이 혼동되면, N 의 배수를 더한 뒤 나머지 연산을 적용할 수 있음.

modular arithmetic에서 두 수의 덧셈/뺄셈/곱셈에 대한 modular arithmetic은 연산을 하고 \bmod 를 취하나, \bmod 를 취하고 연산을 하나 결과가 동일함. 나머지 연산의 분배법칙 성립 여부를 생각해 보면 아주 당연함.

2.1.3. Error Detection/Correction

1. Error Detection/Correction

Error Detection은 error를 검출하는 것을 말함. 즉, error 발생의 유무만을 확인하고, 발생 규모와 위치는 고려하지 않음.

Error Correction은 error를 검출하고, error가 발생한 bit의 수와 정확한 위치를 찾아 정정하는 것을 말함. 이때 redundant bit를 활용해 원본 data를 추측하여 복원하거나, data를 재전송받는 식으로 error를 정정함.

detection보다 correction이 더 어렵고 많은 redundancy를 필요로 하는 것이 일반적임.

2. Redundancy

중복(Redundancy)은 error detection/correction을 위해 data에 redundant bit들을 추가하여 전송하는 것을 말함. 이는 detection/correction의 핵심 개념임.

redundancy는 부호화(coding)를 통해 구현됨. coding에는 block coding과 convolution coding이 있는데, 여기에서는 block coding만을 다룸.

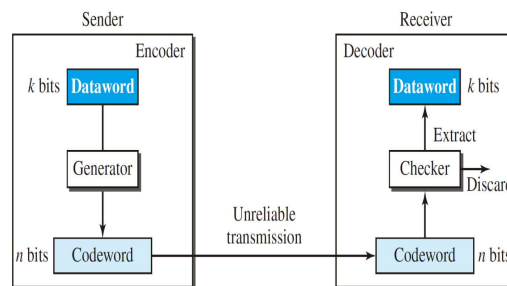
2.2. Block Coding

2.2.1. Block Coding

1. Block Coding

Block Coding은 datawords에 redundant bit를 추가한 codewords를 생성하여 전송하는 기법임.

redundant bit는 송신자에 의해 추가되고, 수신자에 의해 삭제됨.



2. Dataword/Codeword

block coding에서는 dataword와 codeword라는 개념을 사용함.

Dataword는 message(data)를 k bit 길이로 쪼개서 만든 block을 말함. block coding에서는 r개의 redundant bit를 각 dataword에 추가하여 $n = k + r$ bit 길이의 block을 생성하는데, 이를 Codeword라고 함.

block coding에서 하나의 dataword는 하나의 codeword에 일대일로 매핑됨. dataword는 k bit이므로 총 2^k 가지가 존재할 수 있고, codeword는 n bit이므로 총 2^n 가지가 존재할 수 있음. dataword와 codeword는 일대일 대응이므로 총 $2^n - 2^k$ 가지의 사용되지 않는 codeword가 존재함. bit 수에 따라 가능한 codeword 중 어떤 dataword와 매핑된 것을 유효한 codeword라고 하고, 그렇지 않은 것은 유효하지 않은 codeword라고 함.

3. Detection/Correction

수신자가 유효한 codeword의 집합을 알고 있고, 해당 집합에 속하지 않는 codeword를 전달받았다면 error가 발생했음을 detection할 수 있음. 이때 당연하게도 error로 인해 codeword가 바뀌었는데 유효한 codeword 값과 겹친다면 error를 detection하지 못함.

충분한 redundant bit를 사용하여 error가 발생한 codeword가 어떤 유효한 codeword였음을 자명하게 유추할 수 있다면, correction도 가능함. 예를 들어, 원래 01011인데 01010으로 전달되었고, 유사한 형태의 유효한 codeword가 존재하지 않으면 원래의 codeword는 01011인 것으로 correction할 수 있음.

hamming distance를 활용하여 detection과 correction의 가능성에 대한 명확한 기준을 세울 수 있음.

2.2.2. Hamming Distance

1. Hamming Distance

Hamming Distance는 동일한 길이의 두 word(bit열)에 대해, 동일한 위치에서 서로 다른 bit 값을 가지는 자리의 개수임.

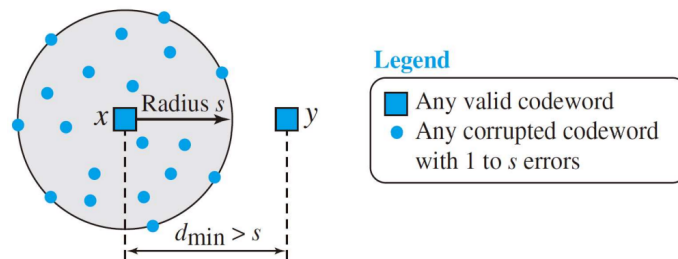
hamming distance는 두 bit열에 대해 XOR 연산을 하고 1의 개수를 세서 구할 수 있음. 참고로 XOR 연산은 각 자리에 대해 bit가 같으면 0, 다르면 1을 값으로 하는 연산으로, a와 b의 XOR 연산은 $a \oplus b$ 로 표기함.

error가 발생했을 때, 원본 codeword와 도착한 codeword의 hamming distance는 error가 발생한 bit의 개수임.

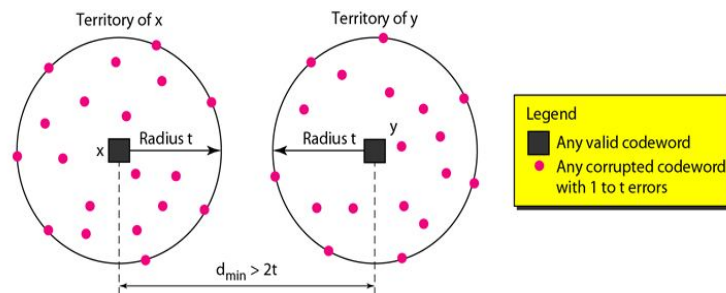
2. Minimum Hamming Distance

Minimum Hamming Distance(d_{min})는 유효한 codeword의 집합에서 존재할 수 있는 hamming distance 중 최솟값임. 이때 error가 아니라 codeword 집합에 대한 값인 것 유의하자.

통신에서 s 개 이하의 bit에서 error가 발생한다고 했을 때, 모든 경우에 대한 error detection을 위해서는 minimum hamming distance가 최소 $s + 1$ 이어야 함. 즉, s 보다 커야 함. 그래야 error가 발생한 codeword가 유효한 어떤 codeword와 겹치지 않음.



통신에서 t 개 이하의 bit에서 error가 발생했다고 했을 때, 모든 경우에 대한 error correction을 위해서는 minimum hamming distance가 최소 $2t + 1$ 이어야 함. 즉, $2s$ 보다 커야 함. 위의 그림과 같이 2개의 codeword를 생각해 본다면, 그래야 error가 발생한 codeword가 속하는 원이 서로 겹치지 않아 구분이 가능함. 이때 어떤 두 유효한 codeword가 가지는 범위에 모두 속하지 않는 error가 발생할 수 있는 게 아닌가 생각할 수 있는데, 가정에서 error의 개수가 제한되었기 때문에 이런 상황은 발생할 수 없음.



정리하면, detection/correction의 d_{min} 조건을 만족시키기 위해서는 기대되는 error bit 발생 수를 감당할 수 있는 codeword 집합을 사용해야 함.

2.3. Linear Block Code

2.3.1. Linear Block Code

Linear Block Code는 두 유효한 codeword에 XOR 연산(mod-2의 덧셈 연산)으로 또 다른 유효한 codeword가 생성되는 block coding 방식임.

여기에서는 이렇게 XOR 연산으로 codeword가 생성되는 것을 linear하다고 함.

오늘날 사용하는 대부분의 block coding은 linear block code임.

2.3.2. Parity Check Code

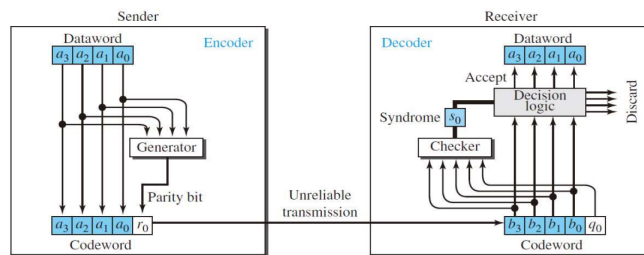
1. Parity Check Code

Parity Check Code는 1개의 bit를 redundant bit로 사용하여 codeword를 구성하는, $d_{min} = 2$ 인 linear block code 방식임.

이때 최하위 bit(맨 오른쪽)에 추가하는 1개의 bit를 parity bit라고 하고, 이는 codeword가 가진 1의 개수가 짝수가 되도록 하는 값을 가짐.

parity check code에서는 dataword에서 한 bit 차이가 parity bit에 의해 codeword에서는 두 bit 차이가 되므로 $d_{min} = 2$ 임. parity check code가 모든 경우에 대해 detection을 하려면 error가 발생한 codeword와 유효한 codeword 사이의 최대 hamming distance가 1임이 보장되어야 함. 물론 일반적인 경우 이를 보장할 수는 없음.

Dataword	Codeword	Dataword	Codeword
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110



2. Detection

bit열의 각 bit 값을 mod-2에서 더했을 때(즉, 각 자리에 대해 XOR 연산을 했을 때) 결과가 0이면 1이 짝수개, 1이면 1이 홀수개인 것임.

송신자 측에서는 dataword의 각 bit 값을 mod-2에서 더한 결과를 parity bit로 사용함($r_0 = a_3 + a_2 + a_1 + a_0 \pmod{2}$).

수신자 측에서는 codeword의 각 bit 값을 mod-2에서 더한 결과를 Syndrome Bit로 함($s_0 = b_3 + b_2 + b_1 + b_0 + q_0$). syndrome bit가 0이면 error가 없는 것으로, 1이면 error가 발생한 것으로 판단함. parity check code에서는 홀수 개의 bit에 error가 발생한 상황을 detection할 수 있음.

당연하게도 발생과 발생하지 않음은 해당 coding에서의 판단일 뿐, 실제 발생 여부는 다를 수 있음. 즉, 모든 경우에 대한 hamming distance가 명확히 정의된 상황이 아니라면, block coding은 완전한 보장을 할 수 없음.

2.3.3. Cyclic Code

1. Cyclic Code

Cyclic Code는 codeword를 순환(rotate, cyclically shift)시키면 또 다른 codeword가 되는 linear block code임.

여기에서는 주로 사용되는 cyclic code인 CRC를 다룸.

2. CRC

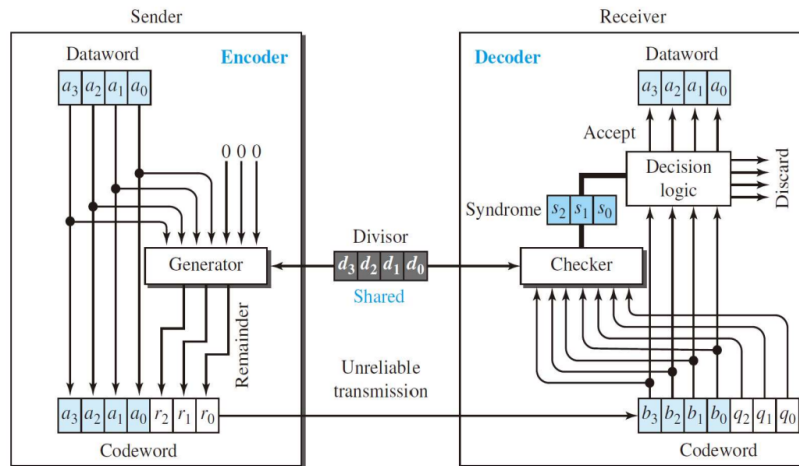
Cyclic Redundancy Check(CRC)는 encoder와 decoder가 공유하는 divisor로 mod-2에서의 나누기 연산을 하여 그 나머지를 활용하는 linear block code임.

dataword의 크기를 k , codeword의 크기를 n 이라고 하자. redundant bit의 길이는 $n - k$ 임. Divisor는 길이가 $n - k + 1$ 인 bit열로, encoder와 decoder가 나누기 연산에 사용하는 값임. divisor는 redundant bit보다 길이가 1 길기 때문에 나머지는 redundant bit와 길이가 맞음.

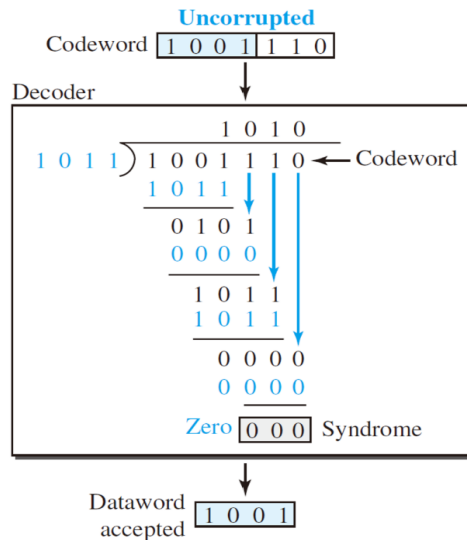
3. CRC의 동작 과정

송신자 측의 encoder에서는 dataword의 복사본을 generator에 전달함. generator에서는 dataword를 codeword 길이에 맞춰서 하위 bit에 0들을 추가한 bit열을 divisor와 나눔. 해당 연산의 나머지가 redundant bit로 사용되어 dataword의 하위 bit에 추가됨. 참고로 나눗셈 시에 뺀 최상위 bit 값을 맞춰주면 됨.

수신자 측의 decoder에서는 전달받은 codeword의 복사본을 checker로 보냄. checker에서는 codeword를 divisor와 나눔. 해당 연산의 나머지가 syndrome bit로 사용됨. syndrome bit는 redundant bit와 길이가 같음. syndrome bit의 각 bit가 모두 0이면 error가 발생하지 않은 것으로 판단되어 dataword 부분이 사용되고, 어느 하나라도 1이면 error가 발생한 것으로 판단되어 값을 버림.

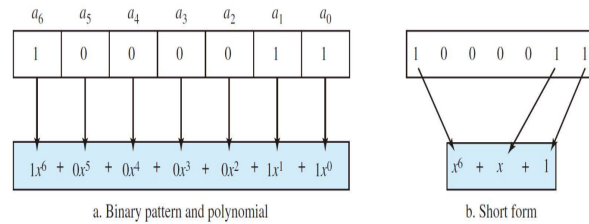


이때 나누기 연산은 단순한 10진수 나눗셈과 동일한 방식으로 수행할 수 있음. 최상위 bit의 값에 따라 뺄이 결정되는데, 중요한 점은 mod-2에서의 연산이므로 덧셈과 뺄셈을 XOR 연산으로 처리해야 한다는 것임.



3. Polynomial

CRC에서는 편의를 위해 bit열을 다항식(Polynomial)로 나타내기도 함. 하위 bit부터 0번째로 시작하여 n번째 bit는 x^n 으로 나타내고, 각 bit 값을 계수(Coefficient)로 함.



CRC에서의 연산은 mod-2 에서 수행됨. *polynomial*의 덧셈/뺄셈은 동일하게 수행되는데(XOR 연산), 단순히 두 *polynomial* 사이에 동일한 차수의 항이 있다면 제거함. 곱셈은 두 항의 차수를 더하고, 나눗셈은 두 항의 차수를 뺌.

이때 곱셈에 대한 분배법칙이 성립하고, 이에 따라 다항식끼리의 곱셈을 할 수 있음. 다항식끼리의 나눗셈은 위에서 *bit*열끼리 나눈 것처럼 할 수 있음(당연하게도 덧셈과 뺄셈은 mod-2 에서의 덧셈과 뺄셈임.).

*polynomial*에 대한 *shift* 연산은 단순히 각 항의 차수를 *shift*하는 만큼 올리거나 내리면 됨. 물론 *bit*에 할당된 공간을 벗어나는 항은 제거함.

앞에서 수행한 나누기(나머지) 연산도 다항식으로 수행할 수 있음.

CRC는 LAN/WAN에서 주로 사용됨.

cyclic code의 수학적인 원리와 나눗셈기의 구현 등은 복잡하므로 다루지 않음.

protocol에 따라 사용하는 CRC와 divisor로 사용하는 표준 *polynomial*이 다름. 예를 들어, LAN에서 사용하는 CRC-32는 최고차항의 차수가 32인 *polynomial*을 divisor로 사용함.

2.4. 기타 기법들

2.4.1. Checksum

1. Checksum

*Checksum*은 각 *unit*으로 구한 *checksum*을 추가로 *message*에 포함시켜 전송하는 *error-detecting* 기법임. 이는 임의의 길이를 가지는 *message*에 대해 적용이 가능함.

*Unit*은 *message*를 *m*개의 *bit*씩 묶은 것을 말함. *Checksum*은 각 *unit*을 더하는 등의 연산으로 계산하는 길이가 *m*인 *bit*열임. *checksum*을 구하는 구체적인 방식은 아래에 따로 정리함.

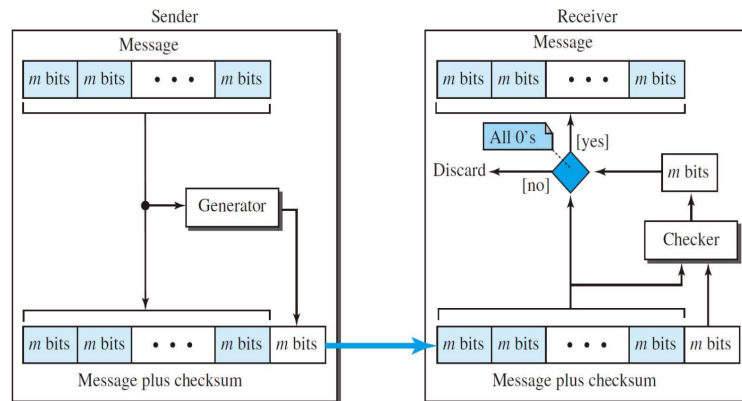
*Internet*에서는 *unit*의 길이를 16bit로 하는 *checksum*을 사용함.

*checksum*은 *data-link layer*에서도 사용되지만, *network/transport layer*에서 주로 사용됨.

2. 동작 과정

송신자의 *generator*는 *message*를 *unit* 단위로 쪼개 *checksum*을 구하고, 이를 *message*에 포함시켜 전송함. 구현 상의 관점에서 더 정확히 하면, *checksum*을 0으로 하고, *checksum*을 포함하는 전체 *message*에 대해 *checksum*을 계산하여 그것을 *checksum*으로 하는 것임.

수신자의 *checker*는 수신받은 *message*(*checksum* 포함)를 *unit* 단위로 쪼개 *checksum*을 다시 계산함. 해당 값이 0이면 *error*가 발생하지 않은 것으로, 1이면 *error*가 발생한 것으로 판단함.

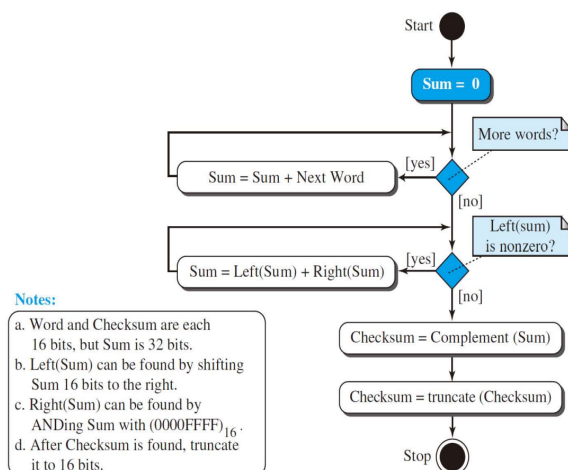


3. Checksum 계산

checksum은 대상을 모두 더한 뒤 1의 보수(One's Complement)를 취한 것을 값으로 가짐. 이때 checksum은 길이가 m 이므로, 덧셈의 결과값은 $0 \sim 2^m - 1$ 의 값만을 가질 수 있음. 덧셈 시에 길이가 m 을 넘어가면 넘어간 왼쪽 bit열을 잘라 오른쪽 bit열에 다시 더함.

더하는 방식이 조금 독특하기는 한데, 어떤 방식으로 더하든 checksum 값은 그것의 1의 보수이므로, checker에서 checksum을 다시 구할 때 error가 발생하지 않았다면 0이 도출됨.

이때 계산을 편리하게 하려면, bit열을 십진수로 바꾼 뒤 다 더하고, 다시 이진수로 바꿔서 넘어가는 값을 잘라서 더하면 됨. 이후 1의 보수를 취함.



4. 성능

checksum은 CRC만큼 정확한 error detection을 하지는 못함. error로 인해 어떤 unit을 값이 커지고 어떤 unit은 값이 작아져 checksum 값이 동일하거나, unit의 값이 커졌음에도 제한된 bit 길이에 따라 checksum 값이 동일한 경우를 detection하지 못함.

그래서 checksum은 CRC로 대체되는 추세임.

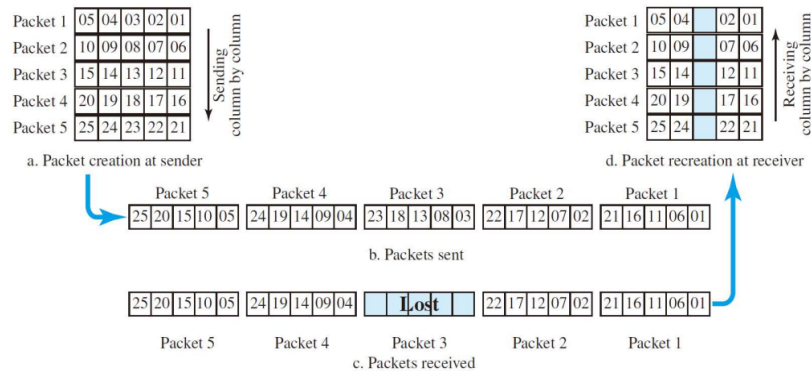
참고로, 어차피 detection만 하면 되므로 checksum을 반드시 message의 최하위 bit 쪽에 붙일 필요는 없음. 실제 구현 시에는 message의 중간에 넣도록 하기도 함.

2.4.2. Chunk Interleaving

Chunk Interleaving은 원본 packet을 각각 전송하는 대신, 각 packet의 bit 값을 하나씩 가져와 구성한 packet을 전송함.

이 경우 전송한 어떤 packet에 error가 발생하더라도, 원본 packet의 관점에서 보면 일부만 손실된 것일 수 있음. 원본 packet의 다른 값들을 활용해 correction하기가 용이함. 특히 burst error에 대한

detection/correction이 가능하다는 장점이 있음.



2.4.3. XOR를 활용한 Correcton

XOR 연산에 대한 역연산은 단순히 원래의 연산을 한 번 더 수행하면 됨. 즉, 아래와 같이 단순히 자리만 바꾼다고 생각할 수 있음.

$$R = P_1 \oplus P_2 \oplus \dots \oplus P_i \oplus \dots \oplus P_N \longleftrightarrow P_i = P_1 \oplus P_2 \oplus \dots \oplus R \oplus \dots \oplus P_N$$

어떤 message를 n 개의 unit으로 쪼개 전송하는 상황 등에서, 모든 unit에 대한 XOR 연산을 통해 계산한 결과값을 함께 전송할 수 있음. 그러면 하나의 unit에만 error가 발생한 경우, 나머지 unit들과 XOR 연산의 결과로 역연산을 수행하면 error에 대한 correction이 가능함.

3. Data Link Control

3.1. 서론

3.1.1. Data Link Control

Data Link Control(DLC)은 point-to-point link와 broadcast link 모두에서 공통적으로 처리되어야 하는 부분을 포함하는 sublayer임. 즉, link의 종류와는 상관 없이 두 인접한 node 사이의 node-to-node 통신을 위한 protocol을 포함함.

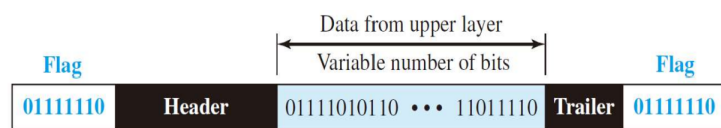
DLC에서 제공하는 service로는 framing, flow/error control 등이 있음.

3.1.2. Framing

1. Framing

상위 layer로부터 전달받은 datagram을 frame으로 encapsulation하고, frame을 datagram으로 decapsulation하여 상위 layer로 전달함. frame(bit-oriented)이 가진 data section의 bit열은 상위 layer에서 여러 가지 정보로 해석됨.

각 frame끼리의 구분 및 분리를 위해 구분자(Delimiter)를 사용해야 함. 대부분의 protocol에서는 8bit 크기의 flag(01111110)를 delimiter로 사용하여 frame의 시작과 끝을 나타냄.



2. Bit Stuffing

Bit Stuffing은 송신자 측에서 frame의 bit열을 확인하여 하나의 0과 다섯 개의 1이 등장(011111)하면 그 바로 뒤에 0을 추가하는 기법임. 수신자는 011111 뒤의 0은 제거함.

frame의 data section에서 flag에 해당하는 bit열이 등장할 수 있는데, 이를 end flag로 오인하지 않도록 bit stuffing을 적용하는 것.

이때 주의할 점은, 011111 뒤에 실제로 1이 오든, 0이 오든 0을 추가한다는 것임.

bit stuffing은 ethernet 등에서는 사용되지 않고, 속도가 매우 느려도 되는 값싼 네트워크에서 사용됨.

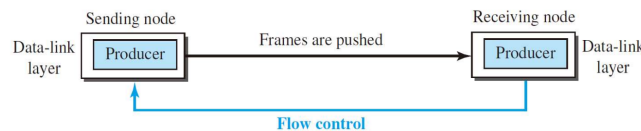
3.1.3. Flow/Error Control

Flow control과 Error control은 DLC에서의 가장 핵심적인 책임임. 이 둘을 통틀어 DLC라고 하는 것.

1. Flow Control

Flow Control은 송신자로부터 얼마나 많은 데이터를 전달받을 수 있을지를 조정하는 작업임.

일반적으로 송신보다 수신에 더 많은 자원이 필요하고, 처리 속도, 메모리 등의 측면에서 수신자의 처리 능력에는 한계가 있음. 수신자가 처리할 수 있는 것보다 더 빠르고 많은 데이터가 전송된다면 수신자는 포화 상태가 되고, 일부 데이터는 처리되지 못하고 소실됨. 반면에 수신자가 처리할 수 있는 것보다 더 느리고 적은 데이터가 전송된다면 수신자는 idle 상태에 빠지므로 비효율적임. 이를 잘 조정하는 것이 flow control임.



2. Error Control

Error Control은 error 처리 작업임.

data-link layer에서 수행하는 error control에서 error에 대한 detection에는 주로 CRC를 사용하며, 아래와 같은 2가지 작업 중 하나를 수행함.

- 1) frame에 error가 있으면 단순히 해당 frame을 무시하고, error가 없으면 단순히 상위 layer로 전달함.
- 2) frame에 error가 있으면 단순히 해당 frame을 무시하고, error가 없으면 송신자에게 acknowledgment를 전송함. 해당 ack는 flow/error control에 대한 정보로 활용됨.

Acknowledgment는 데이터가 전달되었거나, 연결을 허용함을 알리는 packet임. 간단히 ack로 표기함. 여기에서도 사용되지만, 다른 layer에서도 사용되는 개념임.

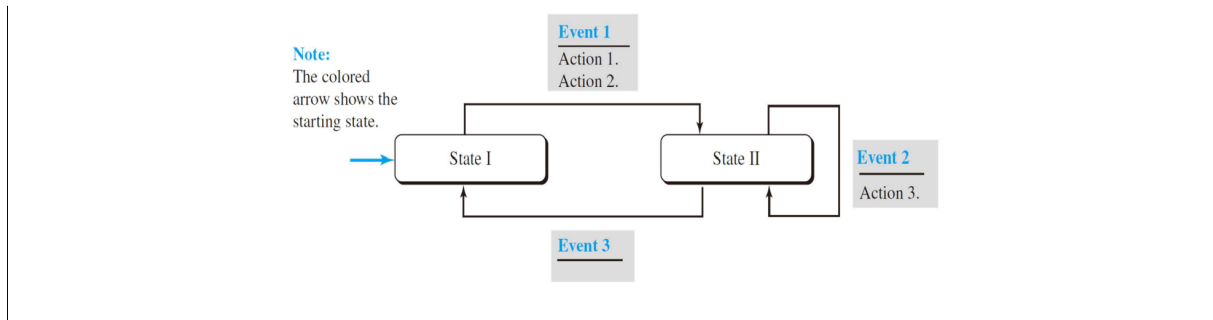
3.1.4. FSM

Finite State Machine(FSM)은 유한한 개수의 상태(State)를 가지는 장치(Machine)임. machine 어떤 state를 가지고, 정의된 event의 발생에 의해서만 state가 변화함.

각 Event에 대해서는 아래와 같이 2가지가 정의되어 있음.

- 1) 해당 event가 발생했을 때 수행할 작업의 목록.
- 2) 해당 event에 의해 결정되는 다음 state. (자기 자신으로 돌아올 수 있음.)

특히 이를 다이어그램으로 나타내면. 어떤 장치나 시스템의 동작을 직관적으로 나타낼 수 있음. 이를 DLC의 protocol을 나타내는 데에도 적용할 수 있음.



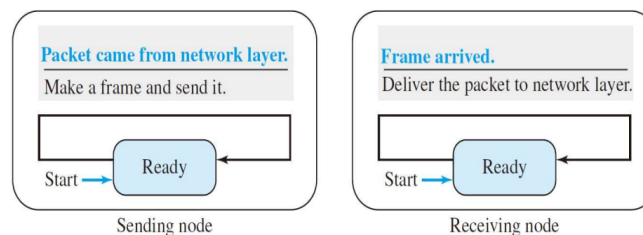
3.2. Protocols

DLC에서 flow/error control을 수행하는 protocol로는 simple, stop-and-wait, go-back-n, selective repeat이 있음. 이 중 오늘날 사용되는 것은 simple, stop-and-wait 뿐임.

3.2.1. Simple

*Simple*은 flow/error control을 수행하지 않는 protocol로, 별다른 이름이 존재하지 않음.

송신자는 상위 layer의 packet으로 frame을 구성해서 계속해서 보내기만 하고, 수신자는 해당 frame을 받아서 상위 layer로 전달하기만 함.

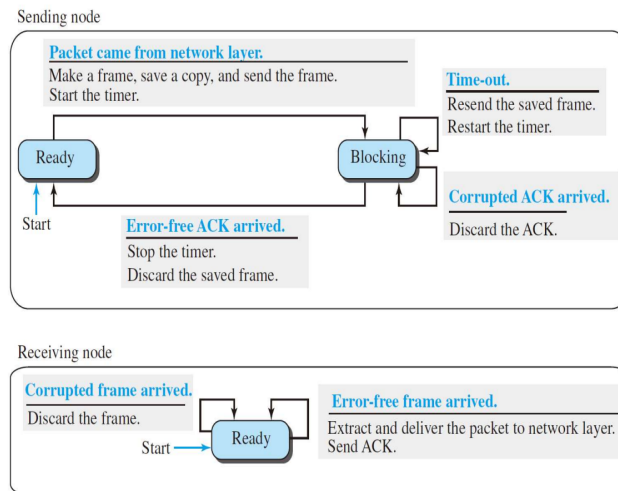


3.2.2. Stop-and-Wait

1. Stop-and-Wait

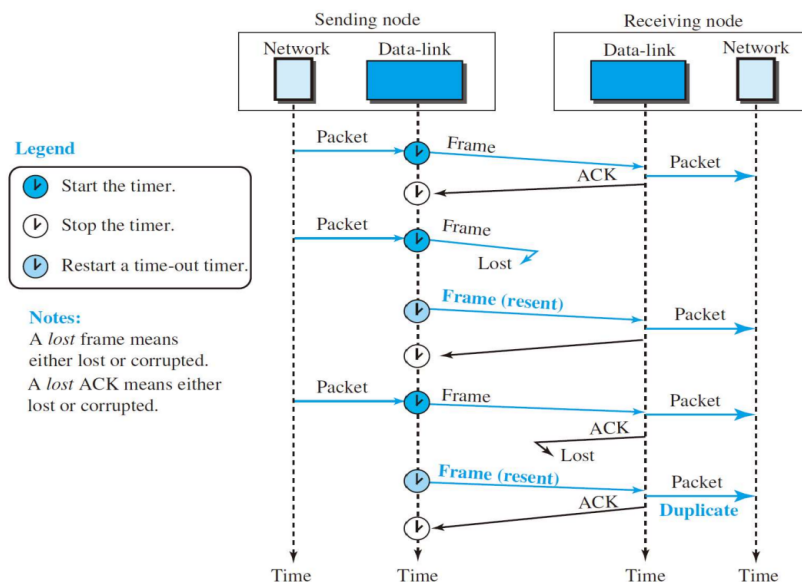
*Stop-and-Wait*는 송신자가 하나의 frame을 전송한 뒤 수신자로부터 ack를 전달받을 때까지 대기하는 protocol임.

송신자는 한 번에 하나의 frame을 전송하고, 즉시 timer를 시작하여 시간을 측정함. timer가 expire되기 전에 ack가 도착하면, 다음 frame을 전송하고 timer를 초기화함. timer가 expire되면, 송신자는 frame이 소실되었거나 error가 발생한 것으로 판단하여 해당 frame을 한번 더 전송함.



송신자는 *frame*을 여러 번 보내야 할 수 있으므로 *ack*를 받기 전까지는 해당 *frame*의 복사본을 저장하고 있음.

참고로 이때 *error detection*을 위해 *frame*에는 *CRC*를 추가함.



2. Sequence/Ack Number

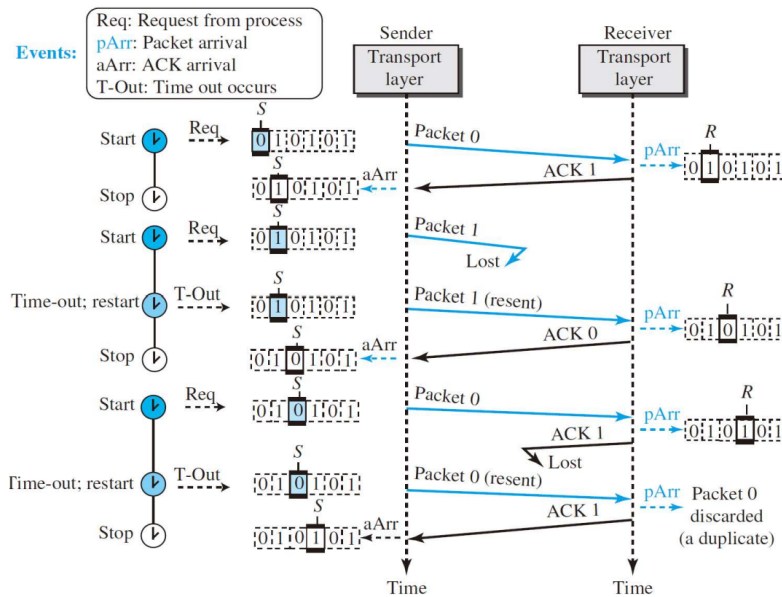
*frame*과 *ack* 모두에 소실 또는 *error*가 발생할 수 있음. 즉, *error*에 대한 경우는 아래와 같이 3가지가 있음.

1. *frame*과 *ack*가 올바르게 전송된 경우.
2. *frame*이 올바르게 전송되지 않은 경우. *frame*이 재전송됨.
3. *frame*은 올바르게 전송되었는데 *ack*가 올바르게 전송되지 않은 경우.

문제는 *ack*가 제대로 전송되지 않은 경우임. 송신자는 *ack*를 받지 못했으므로 이전에 전송된 *frame*이 다시 전송되게 되는데, 수신자 측에서는 이 *frame*이 재전송된 것인지를 알 방법이 없음. 그래서 *sequence/ack number*를 사용하여 각 *frame/ack*의 순서 정보를 나타냄.

*Sequence Number*는 *frame*의 순서에 대한 번호이고, *Ack Number*는 *ack*의 순서에 대한 번호임. 3번 경우에 대해 *frame*이 재전송된 것인지만 파악하면 되므로, *sequence number*는 0과 1만을 값으로 가지면 됨. *ack number*는 수신자 입장에서 다음에 받아야 하는 *frame*의 *sequence number*를 나타냄. 예를

들어, 현재 전달받은 frame의 sequence number가 1이면, ack number는 0이 됨.
0과 1만 사용하므로 number 표기에 사용하는 bit는 1개임. 즉, number 값은 mod-2에서 정의됨.



3. 성능

stop-and-wait은 channel이 두껍고 길수록 비효율적임.

channel이 두껍다(thick)는 것은 channel이 큰 bandwidth(높은 data rate)를 가졌다는 것이고, 길다(long)는 것은 수신자까지의 왕복 지연(round-trip delay. 시간.)이 길다는 것임.

Bandwidth-Delay Product는 channel이 ack를 기다리는 동안 전송할 수 있는 data(bit)의 개수임. data rate과 delay time(여기서는 왕복 시간)을 곱해 bandwidth-delay product를 계산할 수 있음.

당연하게도 channel이 두껍고 커서 bandwidth-delay product가 클수록, ack의 수신을 기다리며 발생하는 낭비가 심함.

stop-and-wait은 window size가 1인 GBN임.

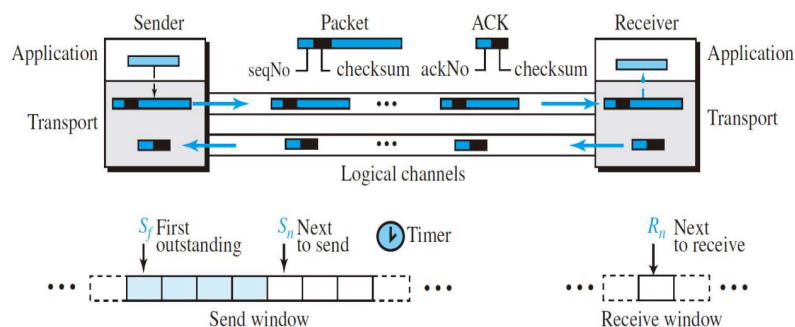
3.2.3. Go-Back-N

1. Go-Back-N

Go-Back-N(GBN)은 송신자가 여러 개의 frame을 전송하다가 수신자로부터 ack를 받지 못하면 ack를 받지 못한 지점부터 다시 전송하는 protocol임.

GBN에서는 window를 사용함. Window는 frame의 sequence number를 저장하여 송신자/수신자 각각에서 frame 송수신에 사용하는 개념임.

여러 개의 frame과 ack가 channel에 동시에 존재할 수 있음.



구체적인 동작 과정은 아래에 다시 정리함.

2. Sequence/Ack Number

sequence/ack number로 m bit를 사용한다면, sequence/ack number는 $\text{mod-}2^m$ 에서 정의됨. 즉, $0 \sim 2^m - 1$ 의 값을 가지고, 해당 값을 넘어가면 다시 0부터 시작함. GBN에서는 여러 개의 frame을 구분해야 하므로 여러 개의 bit를 사용함.

ack number는 수신자 입장에서 다음에 받아야 하는 frame의 sequence number를 나타냄.

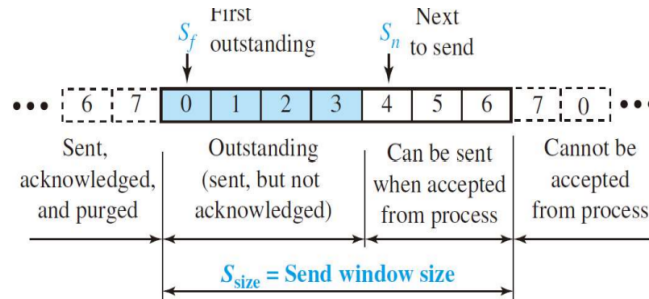
3. Send Window

Send Window는 송신자가 사용하는 window로, 아래와 같이 총 4가지 영역으로 나뉨.

- 1) window 왼쪽 : 전송되었고 ack를 받은 frame들.
- 2) window에서 색칠된 부분 : 전송되었고 ack를 받지 못한 frame들. 해당 frame의 복사본을 저장하고 있음.
- 3) window에서 색칠되지 않은 부분 : 전송이 준비되지 않아 아직 전송하지 않은 frame들.
- 4) window 오른쪽 : window가 오지 않아 고려 대상이 아닌 frame들.

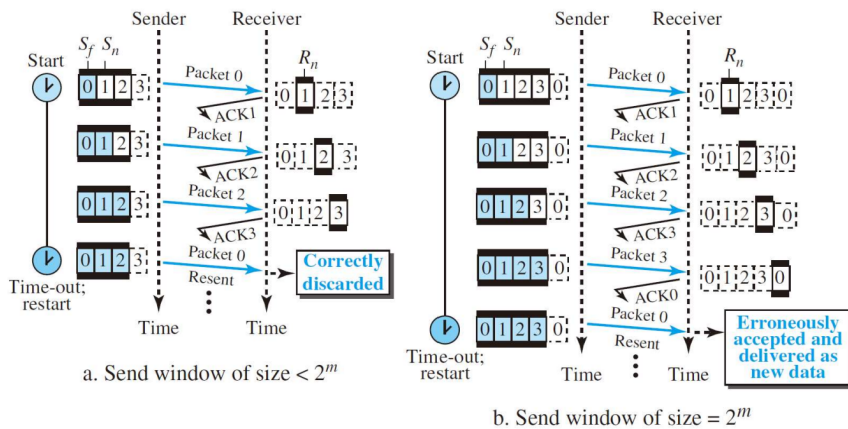
send window의 사용을 위해서는 3가지 변수가 필요함. S_f 는 window의 시작 위치를 가리키고, S_n 은 다음에 전송할 frame을 가리키고, S_{size} 는 window의 size를 나타냄.

ack를 받으면 send window는 오른쪽으로 해당 ack만큼 slide 됨.



send window의 최대 크기는 $2^m - 1$ (전체 sequence number 개수보다 1 작음.)임. 만약 window의 크기가 2^m 이상이라면, send window 안에 전체 sequence number가 모두 들어오는 상황이 발생할 수 있음. 이 경우 window 내의 frame을 전부 정상적으로 전송했는데 ack가 전달되지 못해 timer가 expire 되면 window의 처음부터 다시 전송됨. 이때 수신자 입장에서는 바로 수신받은 다음 number의 frame이 전송되는 것이므로, 처음부터 다시 전송되는 것인지, 그 다음 frame이 전송되는 것인지를 구분할 수 없음. 즉, window의 크기가 전체 sequence number의 개수보다는 작아야 이런 상황을 방지할 수 있음.

여기에서도 문제가 발생하는 지점은 ack가 제대로 전달되지 않는 경우임.



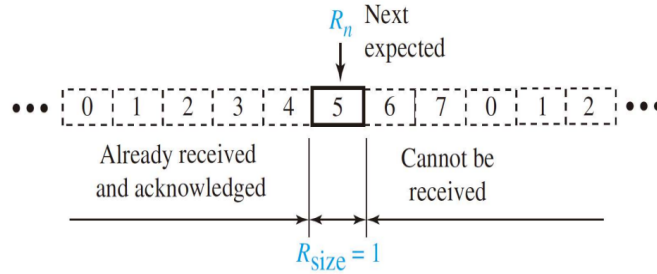
4. Receive Window

Receive Window는 수신자가 사용하는 window로, 항상 크기가 1임. 아래와 같이 구성되는데, 왼쪽은

frame을 수신받아 ack를 전송한 부분, 오른쪽은 frame을 수신받지 못한 부분임.

receive window의 사용을 위해서는 현재 수신받아야 하는 frame을 가리키는 변수 R_n 하나만을 사용하면 됨.

receive window에서는 R_n 에 해당하는 frame일 때에만 R_n 을 한 칸 slide하고, 그렇지 않은 경우에는 frame을 무시함.

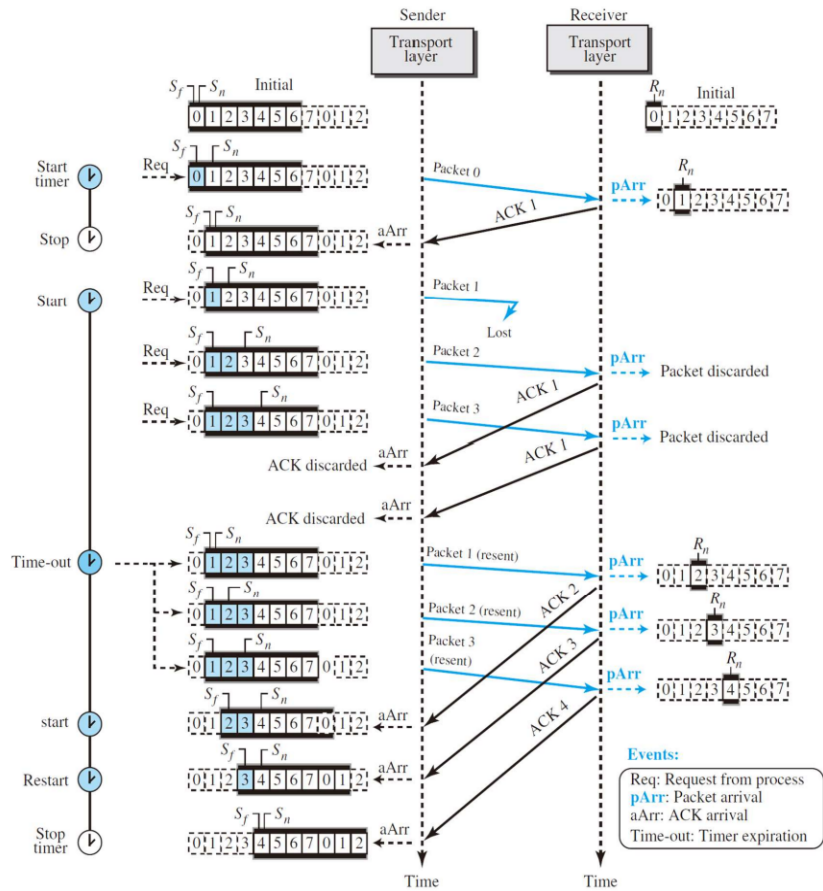


5. 동작 과정

송신자는 send window를 사용해 전송할 frame을 관리하며, ack의 도착에 상관없이 window의 frame을 계속 전송함. 이때 timer를 사용함. 유효한 ack(이미 과거에 받은 ack는 무시함.)를 받으면 timer를 초기화하고, n 에 대한 ack를 받았다면 $n-1$ 까지 정상적으로 전송된 것이므로 해당 부분까지 window를 slide함. timer가 expire되면 table의 처음부터 다시 전송함. 이때 주의할 점은 timer는 유효한 ack를 받을 때 초기화된다는 것임. 즉, window를 slide할 때 timer가 초기화됨.

수신자는 receive window를 사용해 frame을 전달받는 frame을 관리함. R_n 에 해당하는 frame이 아니면 전부 무시하고, R_n 에 해당하는 frame이면 해당 frame 바로 다음 frame의 sequence number를 ack로 보냄.

이때 ack는 누적(cumulative)으로 처리됨.



6. 성능

stop-and-wait에서의 ack의 수신을 기다리며 발생하는 낭비는 해결하지만, timer가 expire될 때 window의 모든 frame을 전부 다시 전송하므로 noise가 많은 channel에서는 비효율적임.

3.2.4. Selective Repeat

1. Selective Repeat

Selective Repeat(SR)는 GBN과 유사하지만, window의 모든 packet을 전부 다시 전송하는 대신 손상된 packet만을 선택적으로 재전송하는 protocol임.

GBN과 유사하므로, 모든 내용 대신 GBN과의 차이점을 중심으로 정리함.

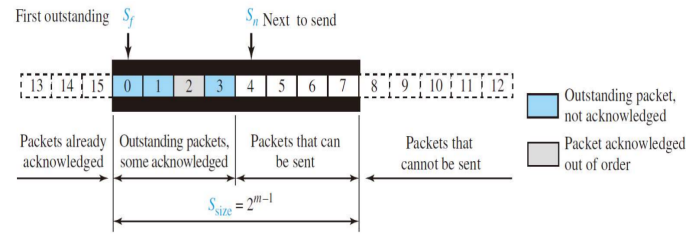
2. Window

send window와 receive window를 사용함. 여기서 send/receive window는 크기가 같음.

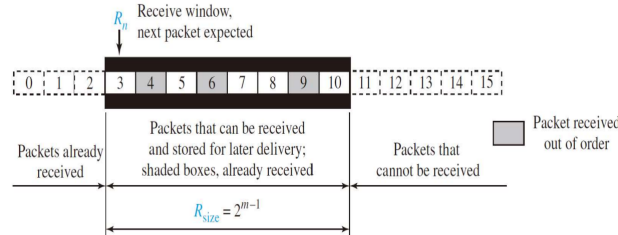
send window는 최대 크기가 달라진 것 외에는 GBN과 형태는 동일함. 다만, ack를 받으면 해당 packet만을 ack된 것으로 처리함. window의 시작점부터 연속적으로 ack를 받았다면 window를 slide함.

receive window는 최대 크기가 send window와 맞춰짐. 수신자는 순서에 상관없이 packet을 전달받으면 일단 window에 넣음. receive window의 크기가 send window와 같기 때문에 packet의 순서가 뒤바뀌어 오더라도 전부 모일 때까지 적절히 저장해 둘 수 있음. window의 시작점부터 연속적으로 packet이 모두 모이면 해당 부분을 상위 layer로 전달하고 window를 slide함.

Send window

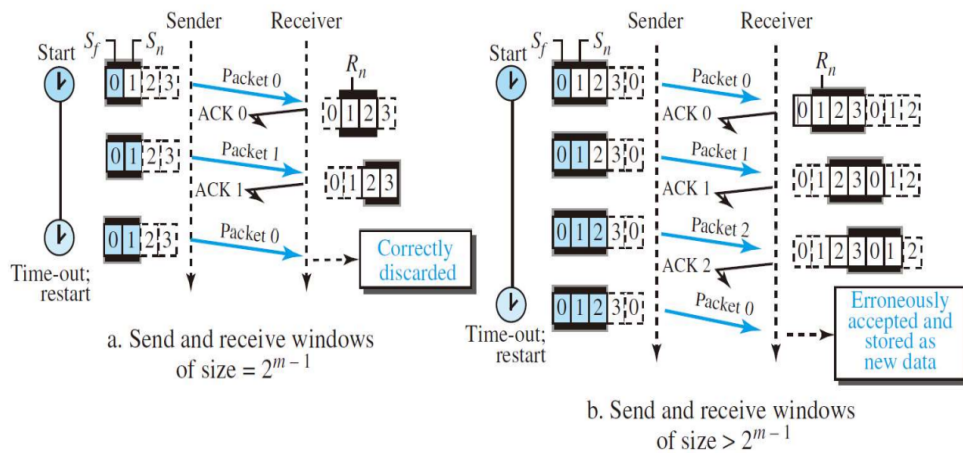


Receive window



sequence/ack number로 m bit를 사용한다면, window의 최대 크기는 2^{m-1} (전체 sequence number 개수의 절반)임. 만약 window의 최대 크기가 이 값 이상이라면, 송신자 측에서 timer의 expire에 의해 packet을 맨 앞부터 재전송하는 경우에 receiver window가 이미 slide되어 있어 해당 packet이 처음부터 다시 전송되는 것인지, 그 다음 frame이 전송되는 것인지를 구분할 수 없음. 즉, window의 크기가 전체 sequence number 개수의 절반보다 같거나 작아야 이런 상황을 방지할 수 있음.

여기에서도 문제가 발생하는 지점은 ack가 제대로 전달되지 않는 경우임.



3. Acknowledgment

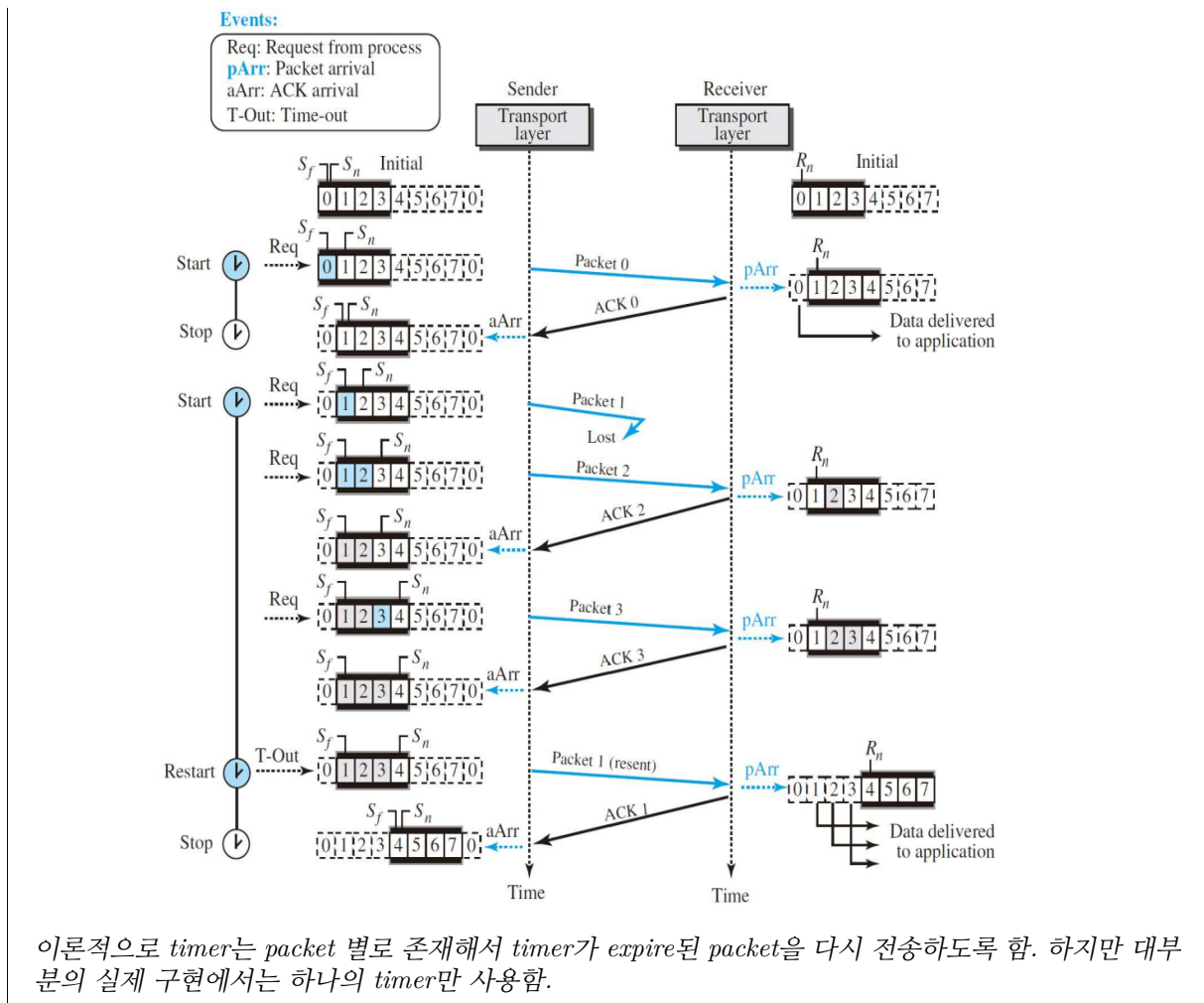
SR에서 ack number는 정상적으로 전달된 packet의 sequence number임.

SR에서는 ack가 cumulative하게 처리되지 않고, packet마다 모두 전송됨.

4. 동작 과정

송신자는 timer를 시작하면서 window 내의 packet을 하나씩 전송함. 수신자로부터 ack를 받으면 해당 number를 ack 처리함. window의 시작점부터 연속적으로 ack를 받았다면 window를 slide하고 timer를 초기화함. timer가 expire되면 window의 앞쪽부터 ack되지 않은 것들을 다시 전송함. 이때 주의할 점은, ack를 받을 때 timer를 초기화하는 것이 아니라, window를 slide할 때 timer를 초기화한다는 것임.

수신자는 window에서 전달받은 packet을 받은 것으로 표기하고, 해당 sequence number를 ack로 보냄. window의 시작점부터 연속적으로 packet이 모두 모이면 해당 부분을 상위 layer로 전달하고 window를 slide함.



3.3. 기타 개념들

3.3.1. Piggybacking

Piggybacking은 packet에 ack를 담아서 전송하는 기법임.

실세계에서는 주로 양방향(Bidirectional) 통신이 빈번하므로 상호 간에 packet과 ack를 주고받아야 하고, 이때 piggybacking을 사용할 수 있음.

3.3.2. HDLC

1. HDLC

High-level Data Link Control(HDLC)은 point-to-point/broadcast link에서 모두 사용되는 bit-oriented 통신을 지원하는 protocol임. 유선/무선 LAN에서 모두 사용함.

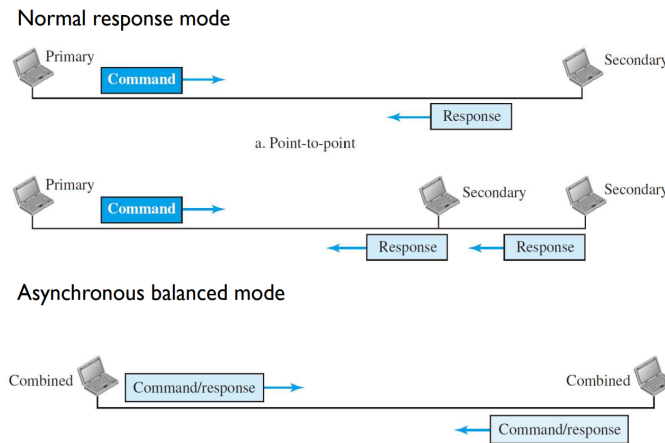
HDLC에서는 두 가지 주요 transfer mode를 제공함.

1) Normal Response Mode(NRM)

station은 primary와 secondary로 분류되고, 그 구성(Configuration)은 균형이 맞춰져 있지 않은 mode임. primary station은 command를 보내고, secondary station은 응답만 할 수 있음.

2) Asynchronous Balanced Mode(ABM)

station의 configuration의 균형이 맞는 mode임. 각 link는 point-to-point이고, 각 station은 primary/secondary의 역할을 모두 수행함.



2. Framing

HDLC는 *framing*을 3가지로 나누어 처리함. 즉, 3가지 종류의 *frame*을 사용함.

- 1) *Information frame(I-frame)* : *user data*와 관련된 정보를 담은 *frame*.
- 2) *Supervisory frame(S-frame)* : *control* 관련 정보를 담은 *frame*(*ex. ACK*).
- 3) *Unnumbered frame(U-frame)* : 시스템 관리를 위한 *frame*. *link* 유지 등을 위해 전송함(*ex. WIFI*의 *beacon frame*).

wifi 연결을 하려고 할 때 목록에 뜰 수 있는 것은 beacon frame을 수신받았기 때문임.

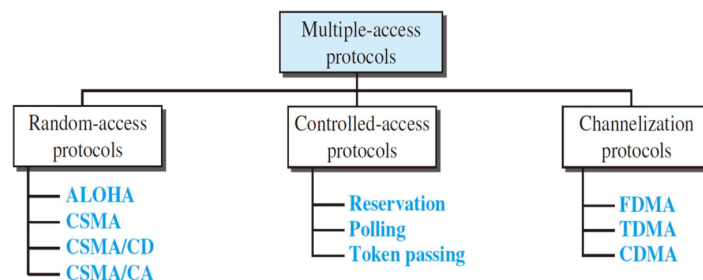
4. Media Access Control

4.1. 서론

4.1.1. Media Access Control

*Media Access Control(MAC)*은 *broadcast link*에서 추가적으로 처리되어야 하는 부분을 포함하는 *sub-layer*임. 즉, 하나의 *channel*에서 동시에 발생하는 여러 *access*를 조율(*coordinate*)하는 *protocol*들을 포함함.

해당 *protocol*들은 *channel*에서의 충돌(*Collision*)을 방지 또는 최소화하는 것을 목표로 함.



4.2. Random-access Protocols

4.2.1. Random-access

*Random Access*는 *station* 사이에 어떤 우선순위가 존재하지 않고, 어떤 *frame*이 어떻게 전송될 지는 단순히 *protocol*에 정의된 작업에 의해 결정되는 방식임. 즉, *station* 사이에서 전송이 어떻게 이루어질

지는 무작위하게 결정됨.

대신 frame의 전송은 주로 전송되는 시점에 각 channel의 상태가 idle이나, busy이나 등에 따라 결정됨. 어떤 station을 우선할 것인가에 대한 규칙은 존재하지 않기 때문에 각 station은 channel에 우선 접근하기 위해 경쟁함.

random-access protocol로는 ALOHA, CSMA, CSMA/CD, CSMA/CA가 있음.

4.2.2. ALOHA

1. ALOHA

ALOHA는 1970년대 하와이 대학에서 개발된 최초의 random access protocol임.

ALOHA에서 사용하는 기법들은 그 이후의 protocol에도 기본적으로 적용되는 것이 많음.

2. Pure ALOHA

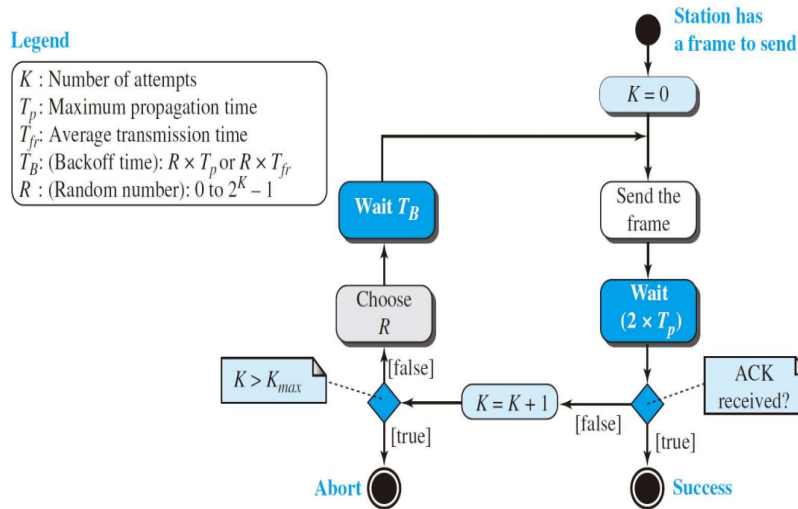
Pure ALOHA에서 각 station은 전송할 frame이 생길 때마다 단순 전송을 하고, 전송 여부는 수신자의 ack에 의존함.

이때 당연히 collision이 발생하므로, 아래의 규칙을 사용함.

1) 각 station에서는 timer가 expired(time-out)되면 back-off time(T_B)만큼 대기한 뒤에 해당 frame을 재전송함.

2) 재전송 시도가 미리 설정해둔 한계치(K_{max})를 넘어가면, 해당 station은 당분간 전송을 중지하고 나중에 다시 전송해야 함. (K_{max} 는 주로 15로 함.)

back-off time은 임의의 길이를 갖는 시간(random amount of time)을 의미함.

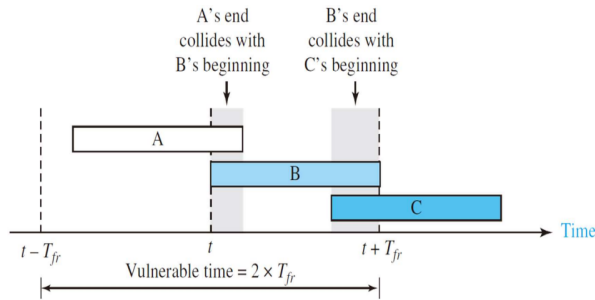


3. Vulnerable Time

취약 시간(Vulnerable Time)은 어떤 frame에 대해 collision이 발생할 가능성이 있는 시간 간격임.

이를 계산할 때 frame 하나가 차지하는 시간 간격(또는 평균 시간 간격)은 T_{fr} 으로 나타냄.

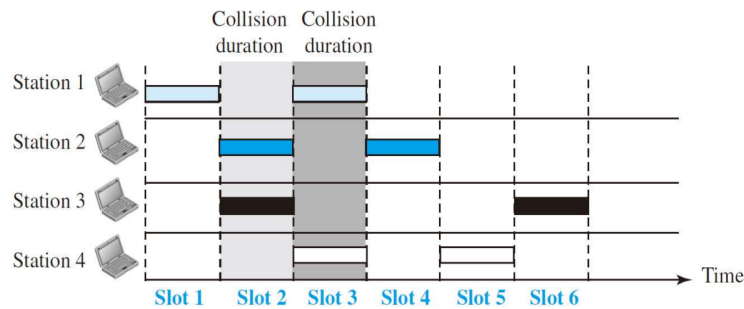
만약 pure ALOHA에서 여러 station에서 전송한 각 frame이 T_{fr} 의 시간 간격을 차지한다고 하자. 어떤 frame B를 시점 t 에 전송한다면 아래의 그림과 같이 $t - T_{fr} \sim t + T_{fr}$ 동안에는 collision이 발생할 가능성이 있음.



즉, pure ALOHA의 vulnerable time은 $2 \times T_{fr}$ 임.

4. Slotted ALOHA

Slotted ALOHA에서는 시간을 T_{fr} 의 길이를 가지는 time slot으로 나누어, 각 station은 frame을 time slot이 시작할 때에만 전송하도록 함.



slotted ALOHA에서 vulnerable time은 T_{fr} 임. 즉, slotted ALOHA는 pure ALOHA의 vulnerable time을 개선한 것임.

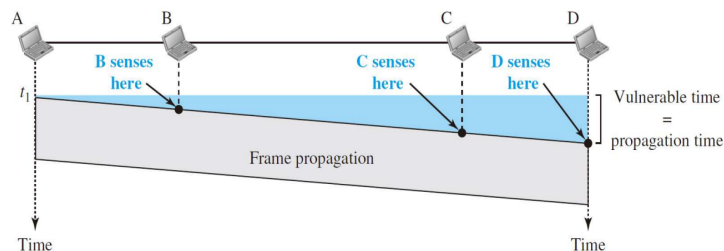
4.2.3. CSMA

1. CSMA

Carrier Sense Multiple Access (CSMA)는 각 station이 전송 전에 우선 channel이 비어 있는지를 검사하는 방식의 protocol임.

CSMA는 collision의 발생을 줄이지만, 전파 시간(Propagation Time, T_p)에 따른 delay로 인해 완전히 없애지는 못함.

CSMA의 vulnerable time은 T_p 임. 당연하게도 어떤 station에서 전송한 frame이 전파되기 전까지 다른 station은 그 발생을 알 수 없음.



2. Persistence Method

Persistence Method는 각 channel의 idle/busy 여부에 따른 동작 방식임. CSMA에는 아래와 같은 3가지 persistence method가 있음.

1) 1-persistence method

channel이 idle이면 즉시 전송함. channel이 busy이면 channel을 계속해서 감시하고 있다가 idle로 판단되는 순간 frame을 전송함.

collision 가능성이 높은 방식임.

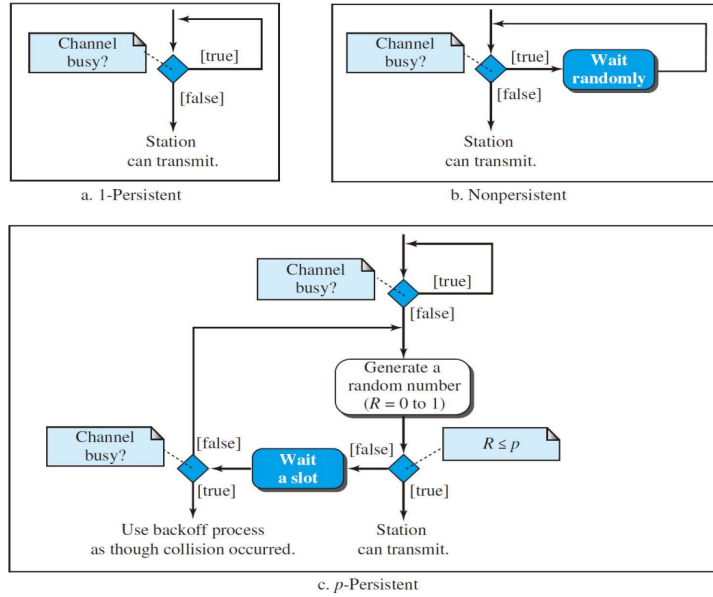
2) nonpersistence method

channel이 idle이면 즉시 전송함. channel이 busy이면 back-off time만큼 대기하다가 channel을 다시 검사함.

collision 가능성을 낮추지만, idle 상태임에도 대기하고 있을 수 있으므로 효율이 떨어짐.

3) p-persistence method

channel이 idle이면 p 의 확률로 전송하고, $q = 1 - p$ 의 확률로 다음 time slot까지 대기함. channel이 busy이면 back-off time만큼 대기하다가 channel을 다시 검사함.



4.2.4. CSMA/CD

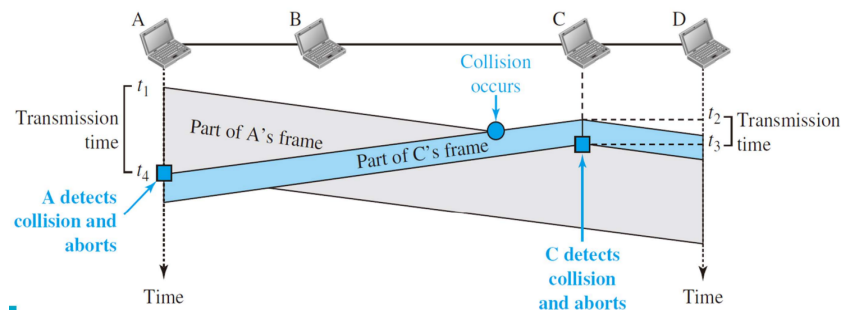
1. CSMA/CD

CSMA with Collision Detection(CSMA/CD)는 CSMA에 collision detection이 추가된 protocol임.

CSMA/CD에서 각 station은 자신의 frame을 전송하면서 다른 station의 frame이 자신에게 전달되는지로 해당 frame에 대한 collision을 파악함. station이 collision을 detection했으면 즉시 전송을 중단하고 protocol에 정의된 persistence method에 따라 재전송함.

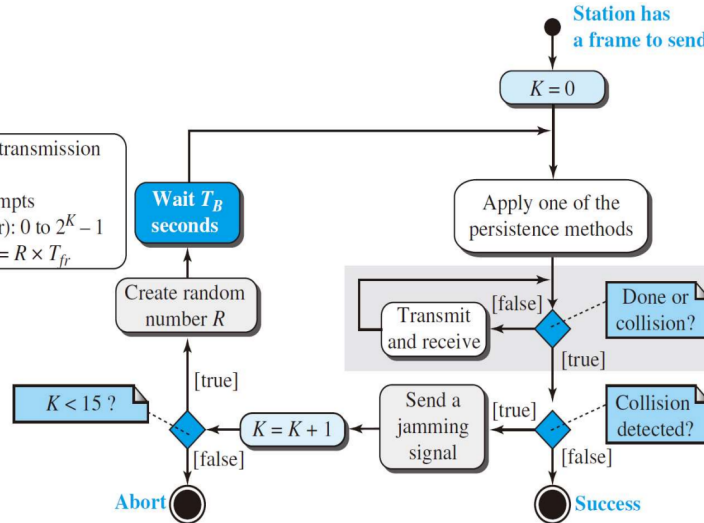
이때 ALOHA에서의 back-off time과 K_{max} 의 개념도 사용할 수 있음.

collision을 detection했으면 Jamming Signal을 다른 station들에게 전송하여 collision의 발생을 알리도록 하기도 함.



Legend

T_{fr} : Frame average transmission time
 K : Number of attempts
 R : (random number): 0 to $2^K - 1$
 T_B : (Backoff time) = $R \times T_{fr}$



2. Minimum Frame Size

CSMA/CD에서 특정 frame에 대한 collision detection은 해당 frame의 전송 중에만 수행됨. 즉, 해당 frame의 전송이 끝나기 전에 detection을 수행할 수 있어야 함.

가장 멀리 떨어진 두 station A, B 사이의 propagation time을 T_p 라고 하자. detection이 가장 늦게 수행되는 최악의 경우는 A로부터 전송된 데이터가 B에 도달하는 순간 B가 데이터를 전송하는 것임. A는 B가 전송한 데이터가 자신에게 돌아와야 collision detection을 할 수 있으므로, 만약 B의 데이터가 도착하기 전에 A에서의 frame의 전송이 끝난다면 collision이 detection되지 않음.

즉, frame에 대한 collision detection을 위해서는 minimum frame size가 $2T_p$ 이상이어야 함.

collision은 channel의 energy level의 측면에서도 확인할 수 있음. energy level이 zero인 경우 idle인 상황, normal인 경우 정상적으로 frame이 busy인 상황, abnormal인 경우(energy level이 2배인 경우 등) collision이 발생한 상황으로 볼 수 있음.

4.2.5. CSMA/CA

1. CSMA/CA

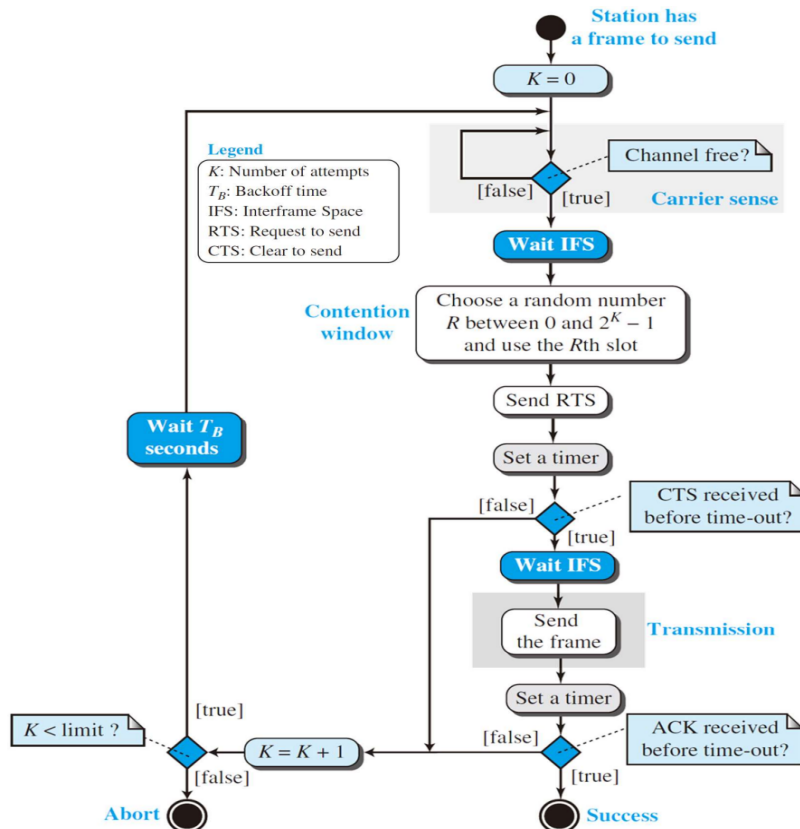
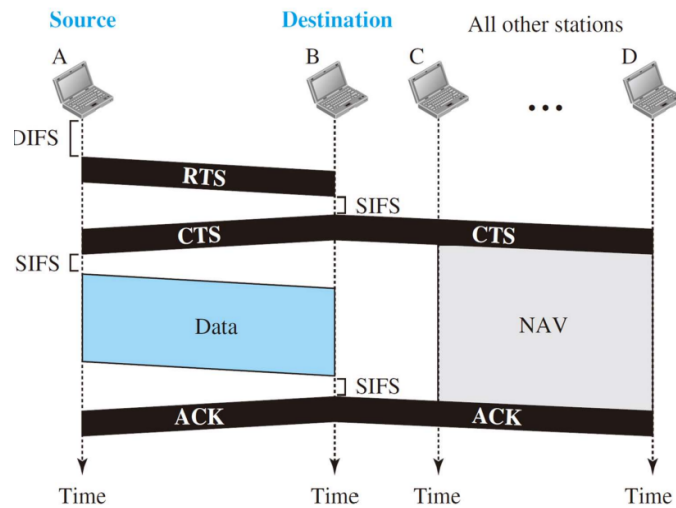
CSMA with Collision Avoidance(CSMA/CA)는 CSMA에 collision에 대한 회피가 추가된 protocol임.

이는 무선 통신으로의 적용을 위해 개발됨. detection에서는 자신이 전송하는 signal과 다른 station이 전송하는 signal이 섞인 signal을 감지하여, 자신이 보낸 signal과의 차이를 통해 collision의 발생을 확인함. 문제는, 무선 통신에서는 다른 station이 전송한 signal은 전송 도중에 에너지가 많이 유실되어 도착 지점에서는 그 에너지 크기가 작다는 것임. 즉, 자신의 signal과, 자신의 signal에 다른 signal이 섞인 signal이 유의미한 차이를 보이지 않을 수 있음. 그래서 무선 통신에서는 collision을 detection하는 대신 회피하는 방식을 주로 사용함.

2. 동작 과정

동작 과정은 아래와 같음. error 또는 소실에 의한 재전송 과정은 생략했음.

- 1) 송신자는 DIFS만큼 기다린 후, RTS를 보냄.
- 2) RTS를 받은 수신자는 SIFS만큼 기다린 후, CTS를 보냄.
- 3) CTS를 받은 송신자는 SIFS만큼 기다린 후, 데이터를 보냄.
- 4) 데이터를 전부 받은 송신자는 SIFS만큼 기다린 후, ack를 보냄.



4.2.6. CSMA/CA : 3가지 전략

CSMA/CA에는 아래와 같은 3가지 전략(기법)을 사용하여 collision을 회피함.

1. Interframe space

Interframe Space(IFS)는 channel이 idle 상태일 때 바로 전송하는 대신 잠시 기다리는 시간 간격임. 즉, 다른 station에서 전송을 시작했을 수도 있으니 기다리는 시간임.

IFS가 끝나면 station은 contention window에 따라 전송을 시작함.

각 station 별로 IFS를 다르게 해서 priority를 부여할 수도 있음. 당연하게도 IFS가 짧으면 priority가 높은 것임.

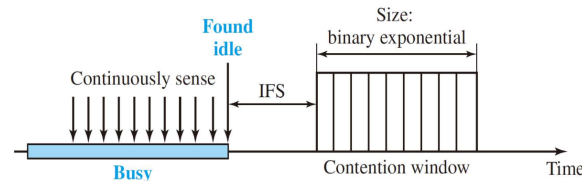
RTS를 보내기 전 최초로 등장하는 IFS를 *Distribute IFS(DIFS)*라고 하고, 그 이후 짧게 등장하는 IFS를 *Short IFS(SIFS)*라고 함.

2. Contention Window

*Contention Window*는 여러 개의 *slot*으로 나뉘어진 시간 간격임.

*station*은 IFS를 기다린 뒤, 특정 *slot*의 수만큼 기다린 뒤 전송을 시작함. 이때 *slot*의 수는 지수 대기 전략(*Binary Exponential Backoff Strategy*)에 따라 결정됨. *slot*에 개수에 대한 최소치(CW_{min} . 위의 그림에서는 0)와 최대치(CW_{max} . 위의 그림에서는 $2^k - 1$)를 저장해 두고, CW_{min} 과 CW_{max} 사이에서 무작위로 수를 하나 골라 해당 수 만큼의 *slot*을 기다림. 즉, 범위 내에서 무작위로 정해진 *slot*의 수만큼 기다렸다가 확인했는데 *busy*이면, 무작위의 범위를 두 배로 늘려 더 기다리게 됨.

이때 *station*이 정해진 *slot*만큼을 기다릴 때 *channel*이 *busy*임을 *detection*했다면, *slot*에 대한 대기 시간을 측정하는 *timer*를 잠시 멈춤. *channel*이 다시 *idle*임을 확인했다면 정지한 *timer*를 다시 실행시킴. 즉, 오래 기다린 *station*은 다시 그만큼 기다릴 필요가 없이 비교적 우선적으로 전송하게 됨.



3. Acknowledge

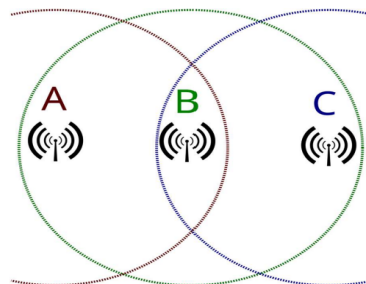
CSMA/CA에서는 RTS, CTS, ack를 제어(control) frame으로 활용하여 데이터를 전송함.

RTS(request to send)는 송신자가 수신자에게 데이터를 전송해도 되는지를 물어보는 frame이고, CTS(clear to send)는 수신자가 송신자에게 데이터를 전송해도 된다고 응답하는 frame임.

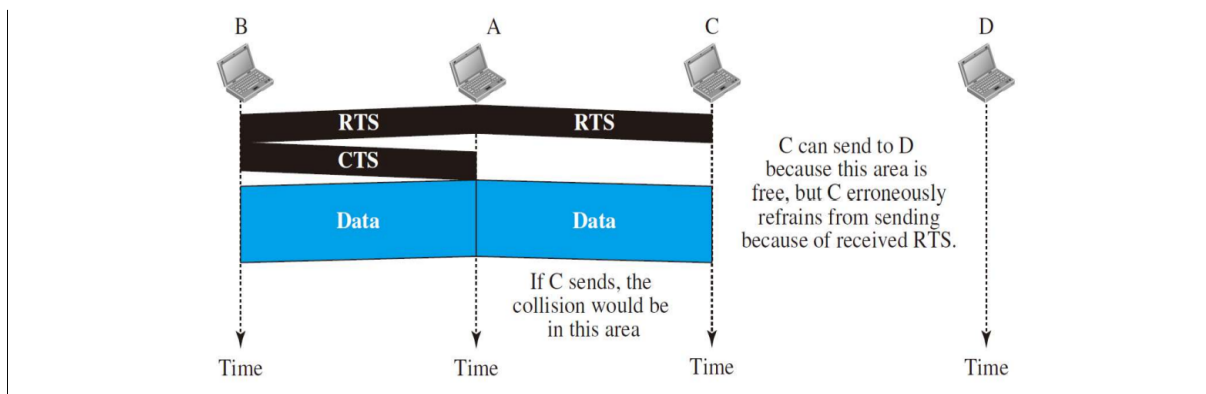
RTS에는 해당 통신이 *channel*을 점유해야 하는 시간 정보를 포함하고 있음. RTS를 전달받은 다른 *station*들은 *Network Allocation Vector(NAV)*라고 하는 *timer*를 시작하여 RTS에 적힌 시간만큼 대기한 뒤, *channel*에 접근을 시도함.

RTS와 CTS를 주고받아 두 *station*이 연결되는 과정을 *Handshaking*이라고 함. *handshaking* 과정에서 *error*나 소실, *collision*이 발생한 것으로 판단하면 송신자는 *back-off time*만큼 기다렸다가 RTS를 다시 전송함.

RTS와 CTS의 사용은 *Hidden-Station problem*을 해결함. 이는 *station a, b, c*가 존재할 때, *a*에서 *b*로 전송한 데이터가 *b*에는 전달되지만 *c*에는 전달되지 않아, *c*에서도 *b*에 데이터를 전송하여 *collison*이 발생할 수 있는 현상임. RTS/CTS를 사용하면 RTS가 다른 어떤 *station*에 전달되지 않더라도 CTS는 수신자와 연결된 모든 *station*에 닿을 수 있으므로 *hidden-station problem*에 따른 *collision*이 발생하지 않음.



반면, *hidden-station problem*과 유사한 문제인 *Exposed Station Problem*은 해결되지 못함. 이는 *station a, b, c, d*가 존재할 때, *a*와 *b* 사이에 발생하는 통신에 의해 *b*의 영향권 아래에 있는 *c*가 *b*의 영향권 밖에 있는 *d*와 통신하지 못하는 문제임.



4.3. Controlled-access Protocols

4.3.1. Controlled-access

Controlled-access는 각 station이 상의하여 어떤 station이 우선 전송할 것인지를 결정하는 방식임. 즉, 각 station은 다른 station들에게 허가(authorize)받기 전에는 전송할 수 없음.

controlled-access protocol로는 reservation, polling, token passing이 있음.

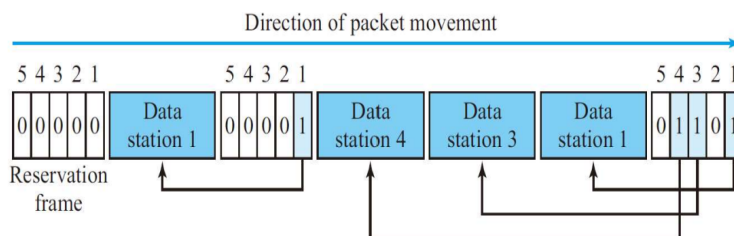
4.3.2. Reservation

Reservation은 각 station이 전송에 대한 예약을 한 뒤 데이터를 전송하는 방식의 protocol임.

reservation은 시간을 interval이라는 구간으로 나뉘고, 각 interval에는 Reservation Frame이 존재함. reservation frame은 각 station에 매핑된 Minislot으로 나뉘어져 있음.

frame을 전송하려는 station은 reservation frame에서 자신의 minislot의 값을 0에서 1로 지정하여 예약함. 이후 해당 interval에는 예약을 한 station의 frame들이 순서대로 전송됨.

당연하게도 minislot은 station의 개수만큼 존재함.



4.3.3. Polling

1. Polling

Polling은 하나의 primary station과 나머지 secondary station이 존재하는 topology에서, primary station이 전적으로 channel을 통제하는 방식의 protocol임.

모든 데이터 송수신은 primary station과 이루어지고(항상 primary station을 거침.), 어떤 secondary station이 channel을 사용하여 primary station으로 송수신할지는 primary station이 결정함.

당연하게도 primary station이 다운되면 channel 전체가 먹통이 됨.

2. 동작 과정

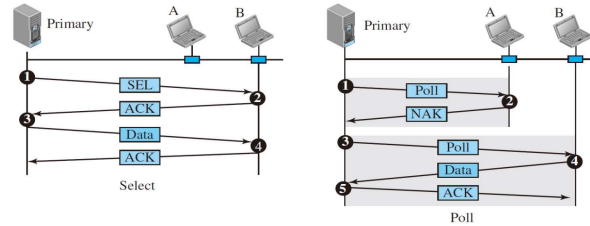
1) primary station에서 secondary station으로 전송하는 경우

primary station은 SEL(select) frame을 전송하고 ack를 받음. 이후 데이터를 전송하고 ack를 받음.

2) secondary station에서 primary station으로 전송하는 경우

primary station은 각 secondary station에 Poll frame을 보내 전송할 데이터가 있는지를 확인함. 각

station은 전송할 게 있으면 데이터를, 없으면 NAK frame을 보냄. primary station은 데이터를 받았으면 ack를 보냄.



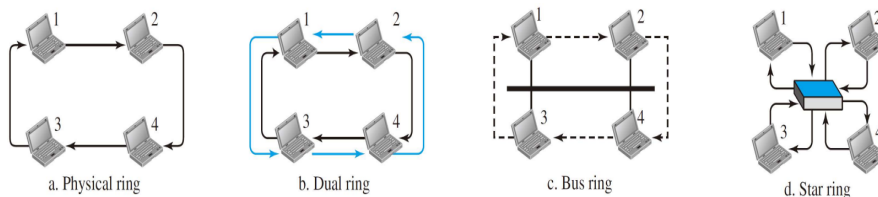
4.3.4. Token Passing

Token Passing은 각 station이 논리적으로 ring의 형태를 가지고, token을 해당 ring의 방향에 맞게 전달하며 권한을 부여하는 방식의 protocol임.

각 station은 token을 받았을 때 전송할 데이터가 있으면 channel을 점유하여 전송하고, 전송할 데이터가 없으면 ring의 다음 station에게 token을 넘김. 즉, token의 소유가 곧 channel 점유 권한이고, channel을 점유하려면 token이 올 때까지 기다려야 함.

이때 이전에 token을 가지고 있던 station을 전임자(predecessor), 뒤에 token을 받을 station을 후임자(successor)라고 함.

당연하게도 token을 붙들고 있는 시간에 제한을 두거나, 우선순위에 따라 token을 부여하는 식으로도 구현이 가능함.



4.4. Channelization Protocols

4.4.1. Channelization

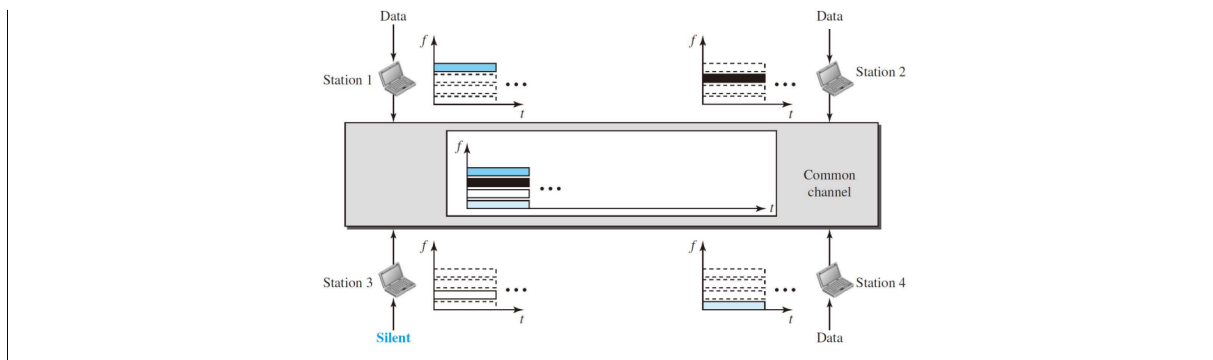
채널화(Channelization)는 link의 가용 bandwidth를 활용하여 시간, frequency, 부호화(code) 등으로 다중 접근을 하는 방식을 말함.

channelization으로는 FDMA, TDMA, CDMA가 있음.

4.4.2. FDMA

Frequency-Division Multiple Access(FDMA)는 link의 가용 bandwidth를 frequency band 단위로 나누고, 각 station은 할당받은 frequency band를 channel로 활용하여 데이터를 전송하는 방식의 protocol임.

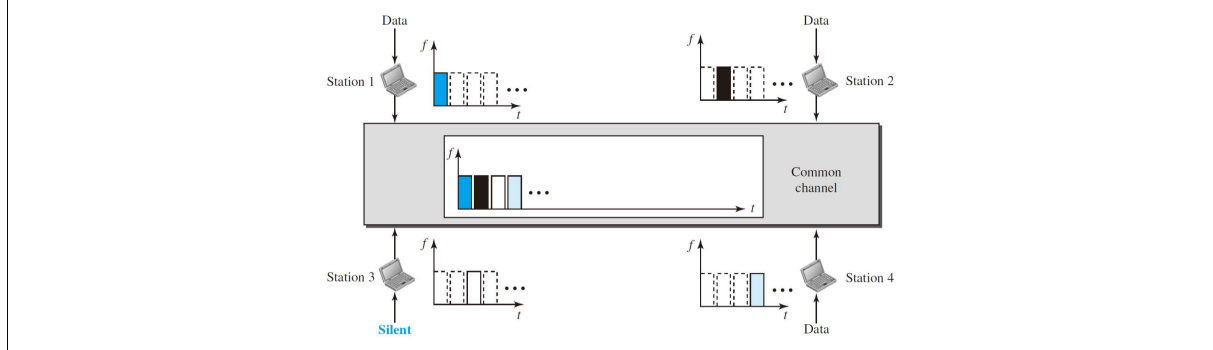
FDMA는 FDM과 유사한 방식을 사용하지만, 엄밀하게 보면 이 둘은 동일한 것이 아님. FDMA는 data-link layer의 protocol이고, FDM은 physical layer의 protocol임. FDM은 MUX를 사용하여 여러 signal들을 변조한 뒤 합쳐 하나의 복합 신호로 전송하는데 반해, FDMA는 각 station이 physical layer를 활용해 자신에게 할당된 bandwidth에 맞게 데이터를 변조하여 전송함.



4.4.3. TDMA

*Time-Division Multiple Access(TDMA)*는 시간의 관점에서 *bandwidth*를 활용하는 방식의 *protocol*임. *TDMA*에서는 시간을 *slot*으로 나누고, 각 *station*은 자신에게 할당된 *slot*에 데이터를 전송함.

*TDMA*는 *TDM*과 유사한 방식을 사용하지만, 엄밀하게 보면 이 둘은 동일한 것이 아님. *TDMA*는 *data-link layer*의 *protocol*이고, *TDM*은 *physical layer*의 *protocol*임. *TDM*에서는 느린 *channel*로부터 데이터를 모아 빠른 *channel*로 전송하는데에 반해, *TDMA*에서는 각 *station*이 *physical layer*로부터 할당받은 *slot*을 단순 사용함.



4.4.4. CDMA

1. CDMA

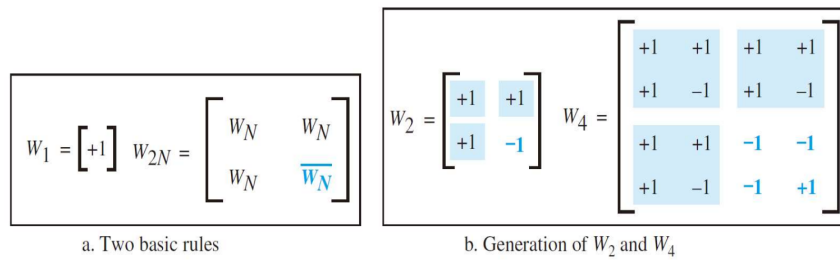
*Code-Division Multiple Access(CDMA)*는 각 *station*이 *chip*이라는 *code*(*bit*열)를 사용하여 하나의 *channel*로 동시에 전송하는 방식의 *protocol*임.

2. Chip

*Chip*은 각 *station*에 할당되는 서로 다른 *code*임. 이때 *chip*들은 아래의 조건을 만족함.

- 1) *station*의 개수가 N 이면, *chip*의 길이도 N 임.
- 2) *chip*은 순서쌍(벡터)으로 취급되며(스칼라 곱, 벡터 합 연산이 가능함.), 당연하게도 내적(*inner product*)이 가능함. 여기서 내적은 *dot product*임.
- 3) *chip*들은 서로 *orthogonal*(직교)함.

*CDMA*에서 *chip*의 *sequence*는 *station*의 개수만큼의 행/열을 가지는 *Walsh table*을 생성하여 얻음. 생성한 *walsh table*의 각 행 또는 열이 각 *station*의 *chip*으로 사용됨. *walsh table*은 아래와 같이 생성할 수 있음.



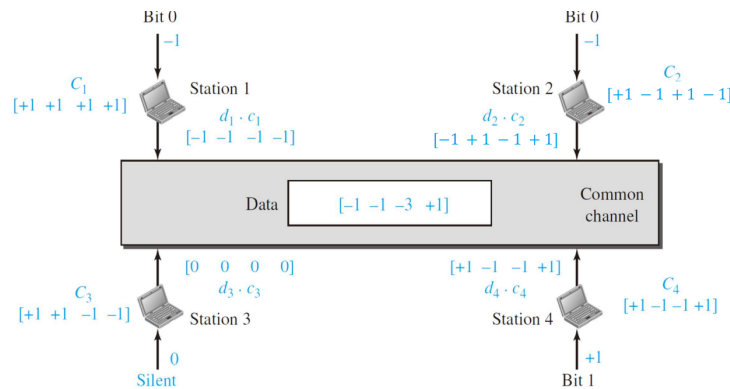
3. 동작 과정

각 station에서는 1을 전송한다면 +1을 값으로 하고, 0을 전송한다면 -1을 값으로 하고, 전송하지 않는다면 0을 값으로 하여 전송함.

각 station은 자신의 chip과 전송할 값을 곱해서 channel에 전송함. channel에는 각 station에서 전송한 chip×값이 모두 합쳐져 있게 됨.

각 station에서 특정 station에서 전송한 값을 확인하려면, channel에 존재하는 전체 값의 합과 송신자의 chip에 대해서 내적을 계산하고 station의 개수로 나누면 됨.

예를 들어, 4개의 station이 존재하고 각 chip을 $C_1 \sim C_4$, 전송하려는 값을 $d_1 \sim d_4$ 라고 하자. 각 station이 전송한 chip×값의 합이 $d_1C_1 + d_2C_2 + d_3C_3 + d_4C_4$ 이고, 송신자의 chip 값이 C_1 이면, 내적의 결과는 $d_1 < C_1, C_1 > \text{임}$. 이때 walsh table에는 1 또는 -1만 존재하므로, $d_1 < C_1, C_1 > = 4d_1$ 임. 즉, station의 개수로 나눈 d_1 을 확인할 수 있음.



참고로 여기서의 code는 복호화가 아니라 bit열을 의미함.

CDMA는 하나의 channel만을 활용한다는 점에서 FDMA와 다르고, 모든 station이 한꺼번에 전송한다는 점에서 TDMA와 다름.

5. Ethernet&wireless LAN

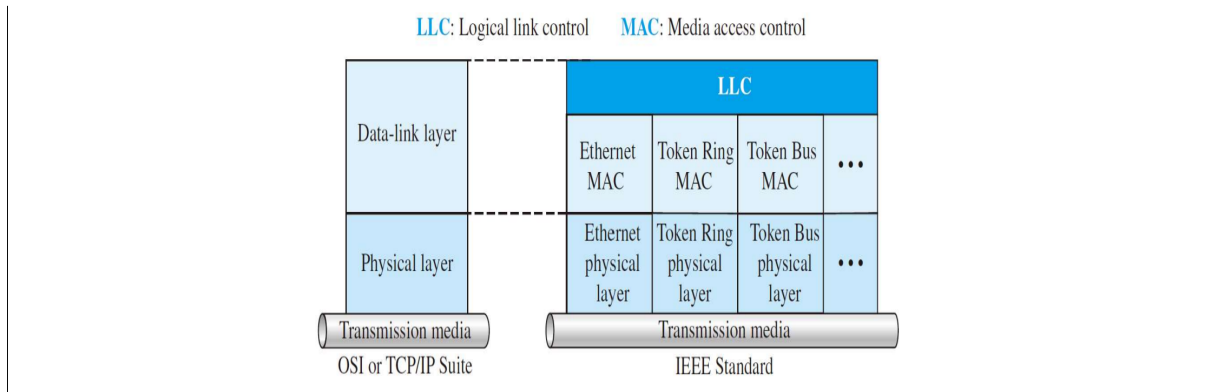
IEEE의 wired LAN protocol(Ethernet)과, wireless LAN protocol을 살펴보자.

5.1. Ethernet

5.1.1. IEEE project 802

IEEE project 802는 ethernet을 포함한 physical/data-link layer protocol의 구체적인 표준을 제공한 프로젝트임.

IEEE project에서는 data-link layer를 LLC(Logical Link Control. DLC와 대응되는 개념)와 MAC으로 나누었음. LLC는 framing, flow/error control 등 모든 IEEE LAN에서 단일 link 통신을 지원하는 부분임. MAC는 각 LAN에 따라 특정 기능을 수행하는 부분임.



5.1.2. Ethernet

Ethernet은 wired network에 대한 physical/data-link layer protocol임.

ethernet은 data rate에 따라 standard/fast/gigabit/10-gigabit ethernet으로 나뉨.

과거부터 여러 protocol들이 있었지만, ethernet을 제외한 대다수가 사라졌음.

5.1.3. Standard Ethernet

Standard Ethernet은 가장 표준적인 ethernet으로, 아래와 같은 특징들을 가짐.

1. Connectionless/Unreliable

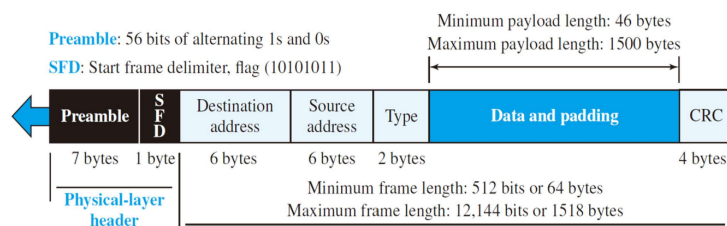
standard ethernet에서 frame을 주고받는 통신은 connection되어 있다거나, reliable하지 않음.

각 frame은 독립적으로 전송되며, 송신자는 실제로 전송되었는지를 확인하지 않음. 수신자도 frame이 제대로 전달되지 않았으면 이를 단순히 무시함. standard ethernet에서는 이런 책임을 상위 layer에게 넘김.

2. Frame Format

standard ethernet의 frame format은 아래와 같음.

- 1) Preamble : synchronization을 위한 부분. 010101...로 구성됨.
- 2) Start Frame Delimiter(SFD) : frame 시작점에 대한 delimiter. 01010111임.
- 3) DA/SA : destination과 source에 대한 link-layer address임.
- 4) Type : frame이 encapsulation한 packet이 상위 layer에서 사용한 protocol 정보임. (ex. IP)
- 5) Data : 상위 layer로부터 받은 packet. packet 길이에 따라 길이가 가변적임.
- 6) CRC : CRC의 redundant bit.



3. 구현

아래와 같은 구현이 정의되어 있음.

Implementation	Medium	Medium Length	Encoding
10Base5	Thick coax	500 m	Manchester
10Base2	Thin coax	185 m	Manchester
10Base-T	2 UTP	100 m	Manchester
10Base-F	2 Fiber	2000 m	Manchester

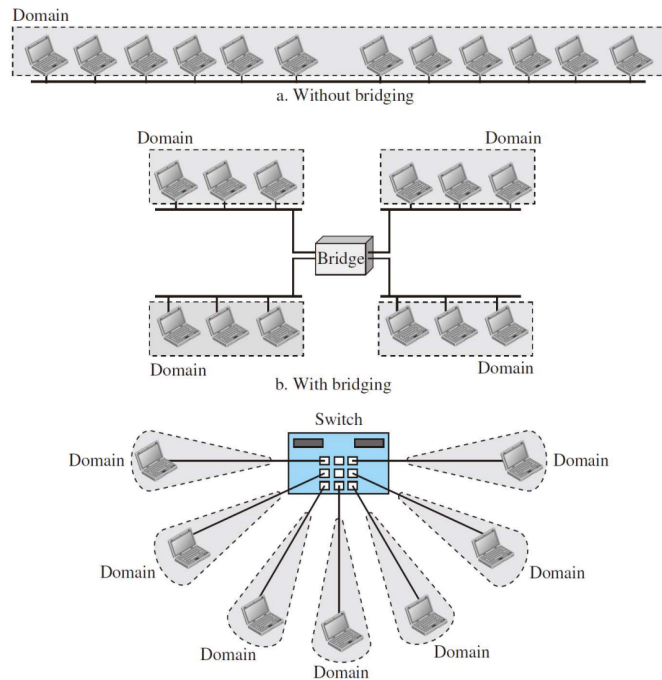
10Base5를 예로 들면, 10은 10Mbps를, Base는 baseband(digital)를, 5는 500m등을 의미함. 각 이름 별

의미를 가지고 있음.

4. Bridged to Switched

초기 ethernet은 bridge를 사용하여 network를 나누어 구성했음. bridge를 사용하면 단순히 모든 station을 하나의 link로 연결하는 것보다 더 빠르고 collision을 줄일 수 있음.

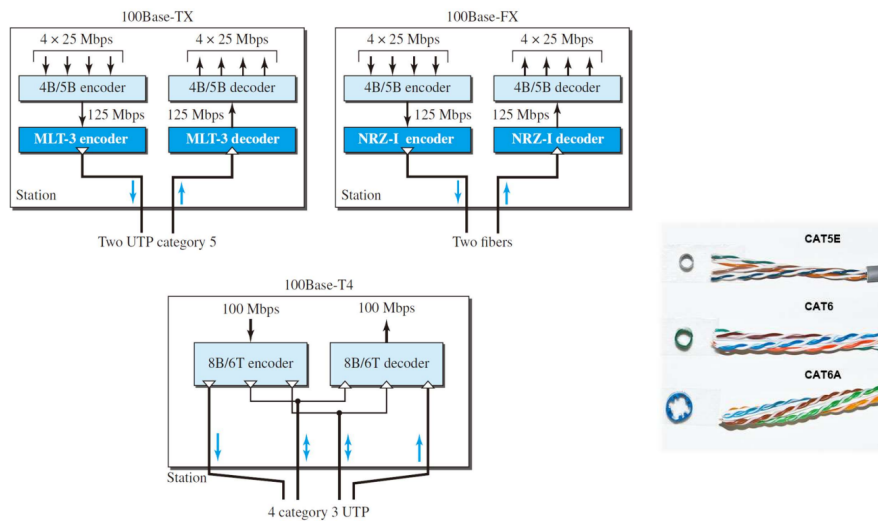
이후에는 switch를 사용하여 bridge에서처럼 몇 개의 network로만 나누는 것을 넘어서서, 더 많은 수로 나누어 구성했음.



5.1.4. 기타 ethernet들

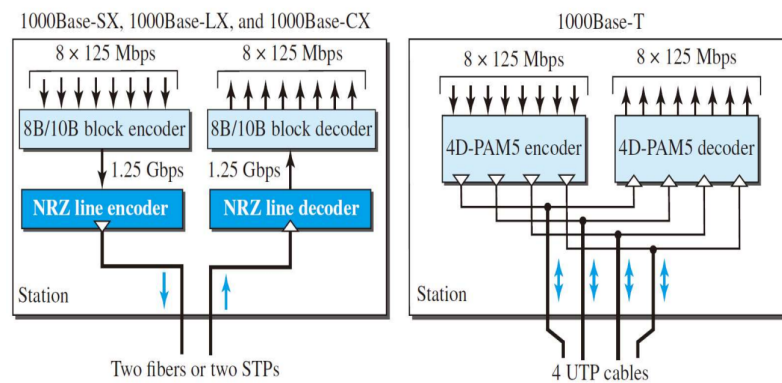
기타 ethernet들에 대한 구현은 아래와 같음.

1. Fast Ethernet



Implementation	Medium	Medium Length	Wires	Encoding
100Base-TX	UTP or STP	100 m	2	4B5B + MLT-3
100Base-FX	Fiber	185 m	2	4B5B + NRZ-I
100Base-T4	UTP	100 m	4	Two 8B/6T

2. Gigabit Ethernet



Implementation	Medium	Medium Length	Wires	Encoding
1000Base-SX	Fiber S-W	550 m	2	8B/10B + NRZ
1000Base-LX	Fiber L-W	5000 m	2	8B/10B + NRZ
1000Base-CX	STP	25 m	2	8B/10B + NRZ
1000Base-T4	UTP	100 m	4	4D-PAM5

3. 10-Gigabit Ethernet

현재 10-gigabit ethernet은 광섬유로만 구현이 가능함.

Implementation	Medium	Medium Length	Number of wires	Encoding
10GBase-SR	Fiber 850 nm	300 m	2	64B66B
10GBase-LR	Fiber 1310 nm	10 Km	2	64B66B
10GBase-EW	Fiber 1350 nm	40 Km	2	SONET
10GBase-X4	Fiber 1310 nm	300 m to 10 Km	2	8B10B

5.2. wireless LAN

5.2.1. wireless LAN

wireless LAN이 가지는 특징은 아래와 같음.

1. Architecture

wireless LAN은 wired LAN, wireless LAN, infrastructure network 등 router같은 장비를 사용하는 network들과 연결될 수 있음.

2. Access Control

wireless LAN에서 가장 주요한 고려 사항 중 하나는 access control임. 즉, 공기 등의 공유 medium을 사용한 통신에서 각 station이 어떻게 소통할 것인지를 정의해야 함.

무선 통신에서는 다른 station이 전송한 signal은 전송 도중에 에너지가 많이 유실되어 도착 지점에서는 그 에너지 크기가 작다는 것임. 즉, 자신의 signal과, 자신의 signal에 다른 signal이 섞인 signal이 유의미한 차이를 보이지 않을 수 있음. 그래서 무선 통신에서는 signal의 송신과 수신을 동시에 하기 어려움.

거리에 따른 에너지 유실 등으로 hidden station problem이 있을 수 있음.

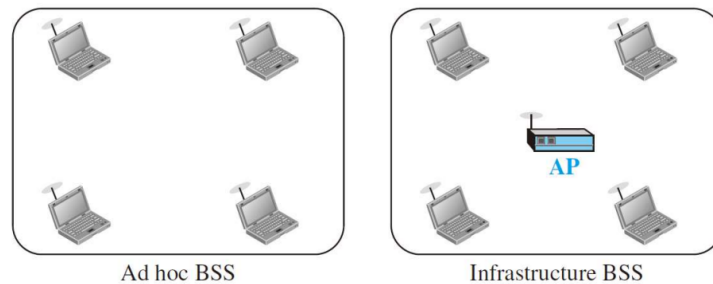
5.2.2. IEEE 802.11

IEEE 802.11은 wireless LAN에 대한 physical/data-link layer protocol의 구체적인 표준임. WIFI 등이 정의됨.

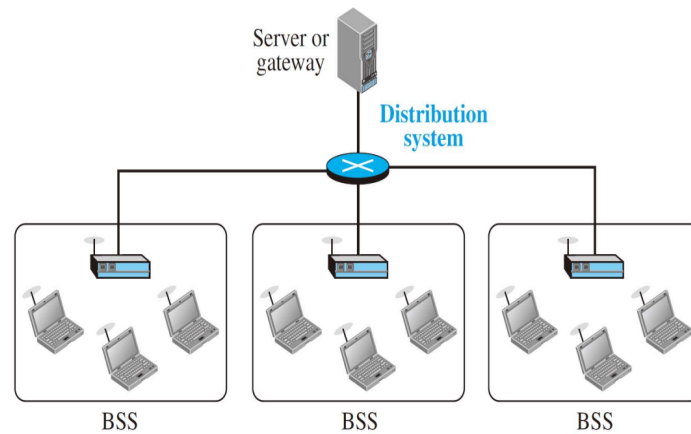
1. Architecture

architecture는 basic service set과 extended service set으로 나눌 수 있음.

basic service set은 station끼리 통신하거나(ad hoc BSS), station들에 추가로 Access Point(AP)라고 하는 중앙 station을 뒤서 통신함(infrastructure BSS). 이때 station들을 묶은 단위를 BSS라고 부름.



extended service set은 AP를 통해 여러 BSS를 distribution system으로 연결함. distribution system 자체는 wired/wireless LAN일 수 있음.



2. MAC Sublayer

IEEE 802.11에서는 MAC을 DCF와 PCF라는 sublayer로 나누어 access method를 정의함.

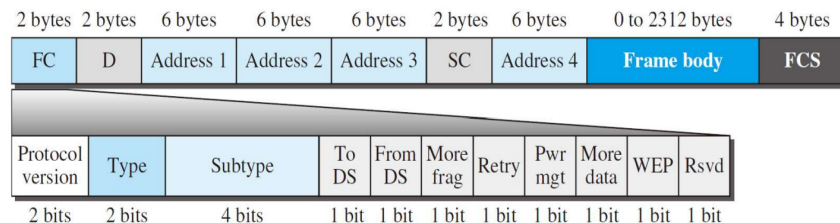
Distributed Coordination Function(DCF)는 access method로 CSMA/CA를 사용함.

Point Coordination Function(PCF)는 infrastructure network(AP가 포함된 BSS)에서 사용될 수 있는 optional access method임. ad hoc network에서는 사용되지 않음. AP와의 통신 방법이 정의되어 있음. 시간에 민감한 환경에서 사용함.

3. Frame Format/Type

MAC의 Frame Format은 아래와 같음.

- 1) Frame Control(FC) : frame 자체에 대한 정보를 담은 부분. from/to DS를 포함함.
- 2) Duration(D) : NAV 지정에 대한 전송 기간 등.
- 3) Address : to/from DS에 따른 address 정보.
- 4) Sequence Control(SC) : flow control에 사용되는 frame의 sequence number.
- 5) Frame Body : frame이 담고 있는 내용.
- 6) FCS : CRC 관련 부분.



frame의 type은 아래와 같음. frame에서 type이라는 field에 의해 지정되고, 해당 type 내에서도 subtype을 구체적으로 지정하기도 함.

- 1) Management Frame : 통신 초기에 관리를 위한 frame.
- 2) Control Frame : access/ack 등에 사용되는 frame.
- 3) Data Frame : 전송할 data를 담고 있는 frame.

4. Addressing

frame format에는 총 4개의 address field가 존재하는데, FC의 to DS와 from DS의 값에 의해 각 address field가 가지는 의미가 달라짐. 이때 address 1은 항상 해당 frame이 전송될 device이고, address 2는 항상 해당 frame을 전송한 device임. field 값이 0이면 source/destination을 의미하고, 1이면 AP를 의미함.

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	Destination	Source	BSS ID	N/A
0	1	Destination	Sending AP	Source	N/A
1	0	Receiving AP	Source	Destination	N/A
1	1	Receiving AP	Sending AP	Destination	Source

1) to DS가 0, from DS가 0인 경우

BSS 내에서 source가 destination으로 전송함.

2) to DS가 0, from DS가 1인 경우

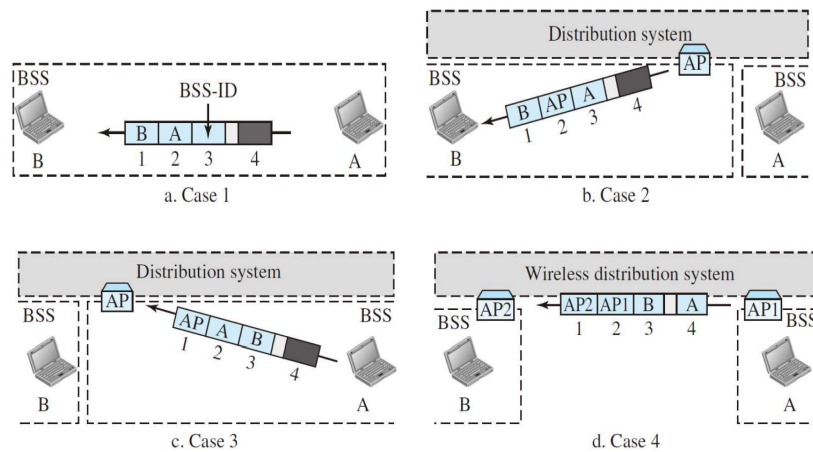
AP가 distribute system으로부터 받은 것을 destination에 전송함. 이때 AP에게 전송한 source의 address를 함께 전송함.

3) to DS가 1, from DS가 0인 경우

source가 AP를 통해 distribute system으로 전송함. 이때 해당 frame을 전달받을 destination의 address를 함께 전송함.

4) to DS가 1, from DS가 1인 경우

한 AP가 distribute system을 통해 다른 AP로 전송함. 이때 해당 frame을 전송한 source와 해당 frame을 전달받을 destination address를 함께 전송함.



5. 구현

아래와 같은 구현이 정의되어 있음. 이때 사용하는 대역은 산업/과학/의료 등을 위한 무면허 대역임.

IEEE	Technique	Band	Modulation	Rate (Mbps)
802.11	FHSS	2.400-4.835 GHz	FSK	1 and 2
	DSSS	2.400-4.835 GHz	PSK	1 and 2
	None	Infrared	PPM	1 and 2
802.11a	OFDM	5.725-5.850 GHz	PSK or QAM	6 to 54
802.11b	DSSS	2.400-4.835 GHz	PSK	5.5 and 11
802.11g	OFDM	2.400-4.835 GHz	Different	22 and 54
802.11n	OFDM	5.725-5.850 GHz	Different	600