

Data Link Control (DLC)

DLC Services

- DLC는 인접한 두 노드 사이의 커뮤니케이션의 절차를 다룸
- Link가 전용인지 Broadcast인지 상관없음
- Framing, Flow and Error Control 포함

Framing

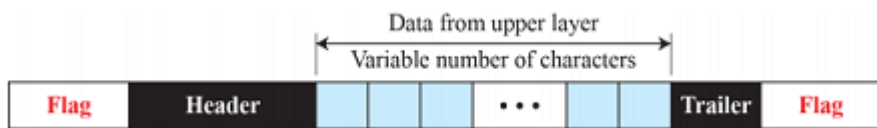
- 비트들을 Frame 단위로 묶음
 - Frame은 서로 구분가능해야함
- Sender/Destination Address를 통해 메시지를 SRC/DEST로 구분지음
- Destination Address는 패킷이 어디로 갈것인지를 정의
- Sender Address는 받는쪽에서 Receipt를 Acknowledge하는데 도움을 줌

Frame Size

- 전체 메시지를 하나의 Frame으로 묶을 수 있지만, 그렇게 하지 않음
- 프레임의 사이즈가 커지면 Flow/Error Control이 비효율적으로 됨
- 사이즈가 너무 크면 하나의 비트만 틀려도 다 다시 보내야됨
- Fixed Size Framing : 경계를 구분지을 필요가 없음
 - ex) ATM WAN : fixed (called cells)
- Variable Size Framing
 - 프레임의 끝/시작을 표기할 무언가가 있어야 함

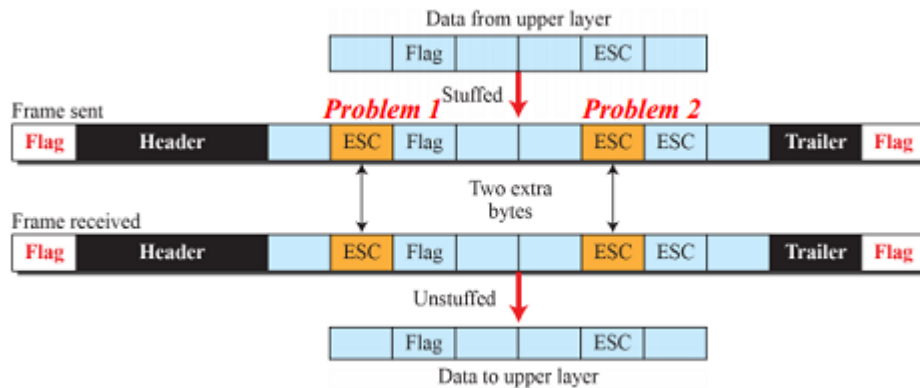
Character-Oriented Framing

- Byte-Oriented Framing으로도 불림
- Flag : 1 바이트
 - 프레임을 구분하기 위해 프레임의 시작/끝 부분에 Flag 하나씩 추가
 - Flag가 프로토콜 의존적인 Special Character로 이루어짐
- Header : Src/Dst 주소, 컨트롤 정보
- Trailer : 에러 탐지 쓸모없는 비트들



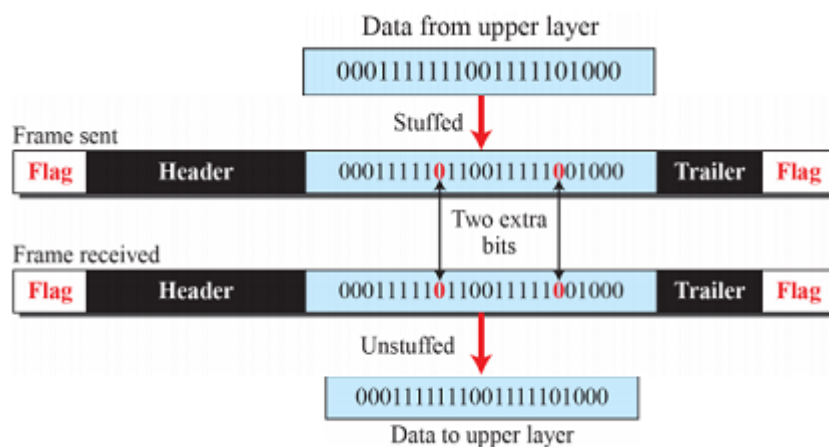
- 문제점 1 : 데이터에 Flag Character가 포함 될 수 있음
 - Byte Stuffing(Character Stuffing) 사용
 - ESC(이스케이프 문자)를 사용해서 데이터섹션에서 Flag Character 전에 삽입
 - Receiver가 ESC를 만나면, Data section에서 삭제하고 다음 Character를 데이터로 인식

- 문제점 2 : 데이터에 하나 이상의 ESC가 Flag character 전에 나오면?
 - ESC 캐릭터 앞에도 ESC 캐릭터가 추가적으로 하나 삽입되어야 함



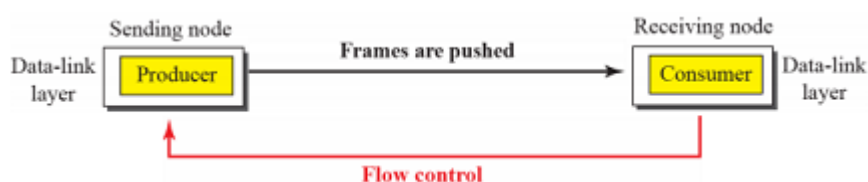
Bit-Oriented Framing (ex) LAN

- 8-bit 패턴 Flag : 01111110가 프레임의 시작/끝에 구분자로 삽입됨
- 문제점 : 데이터에 플레그 패턴이 나타날 수 있음
 - Bit Stuffing 사용
 - 데이터쪽에서 1이 5개 연속 나오면, 다음에 0을 하나 추가적으로 삽입



Flow And Error Control

- Sender가 Receiver가 받기 힘들 정도로 너무 빨리 데이터를 보내지 않도록 확인하는 것
- Sender의 Sending rate가 더 빠를 경우 버퍼가 필요함
- 문제점 : 버퍼의 크기에 제한이 있음
 - Discard : 버퍼가 가득차면 프레임을 버림
 - Feedback : Sender에게 속도를 늦추거나 멈추라고 요청



- Flow / Error Control이 합쳐질 수 있음

- Flow Control을 위한 ACK(Acknowledgement)를 Error Control에도 사용 할 수 있음
 - 제대로 도착했다는 정보를 보낼 수도 있음

Connectionless and Connection-Oriented

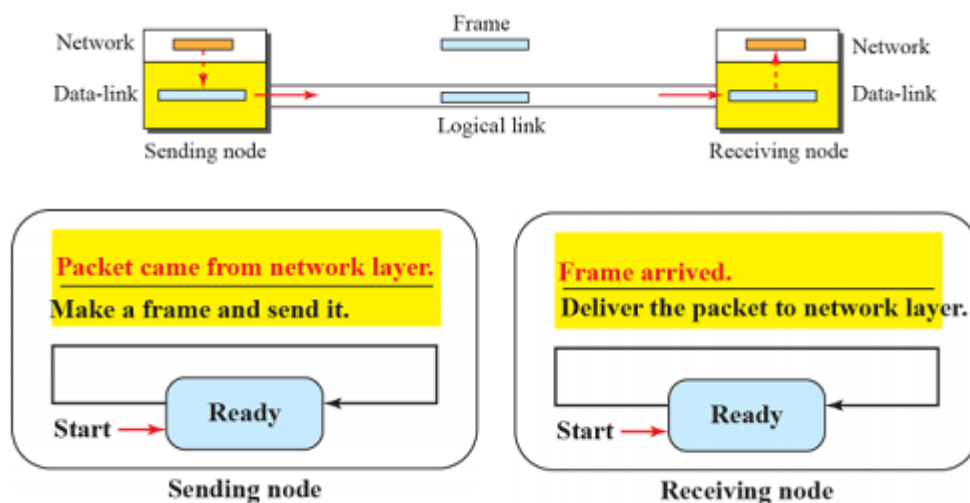
- DLC 프로토콜은 Connectionless 또는 Connection-Oriented 할 수 있음

Data-Link Layer Protocols

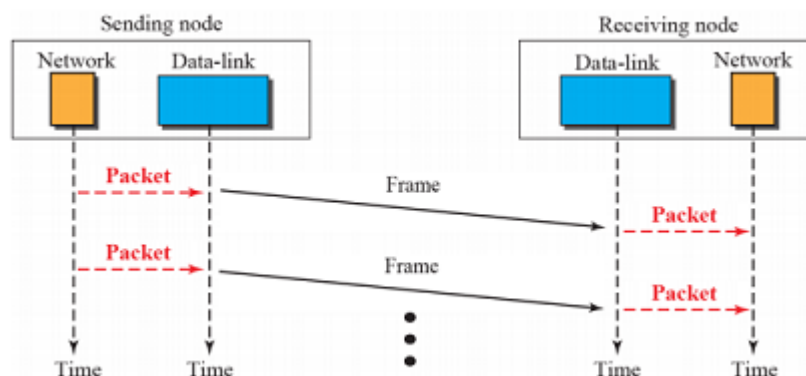
- Flow / Error Control을 위해 전통적으로 4개의 프로콜이 정의됨
- Simple / Stop-And-Wait는 현재까지 사용, Go-Back-N, Selective-Repeat은 사라짐

Simple Protocol

- Flow / Error Control이 없는 간단한 프로토콜
- Receiver가 모든 받는 프레임의 바로 handle 할 수 있다고 가정

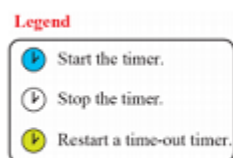
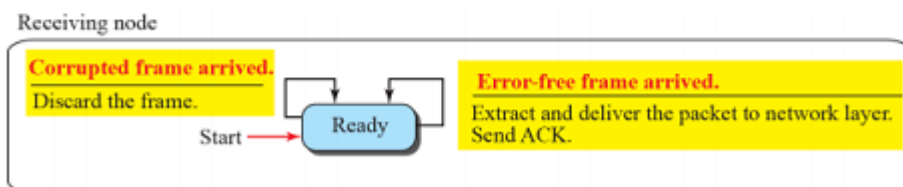
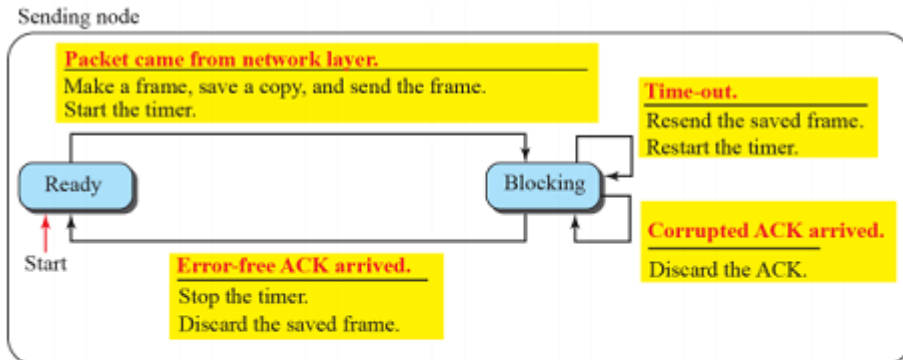
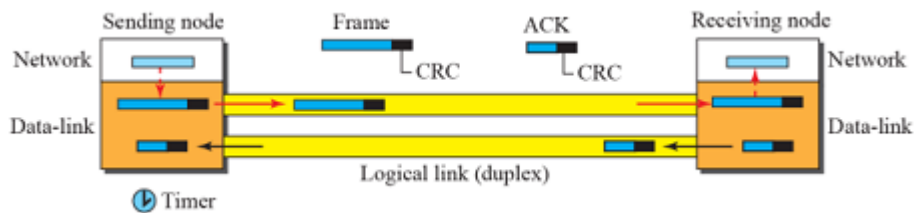


- Sender가 Receiver가 뭘하든 상관안하고 강 보냄

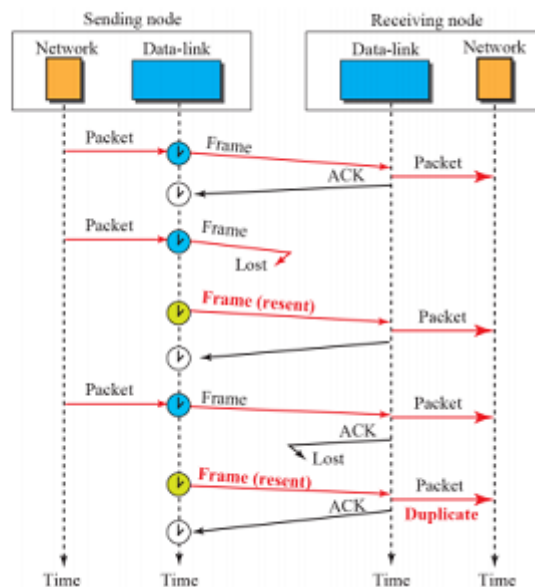


Stop-and-Wait Protocol

- Flow / Error Control 모두 사용
- 지금 볼건 Primitive Version
- Sender가 한번에 하나의 프레임을 보내고, 다음 프레임을 보내기 전 ACK를 기다림
- 손상된 프레임을 탐지하기 위해 각 프레임에 CRC를 더해야 함



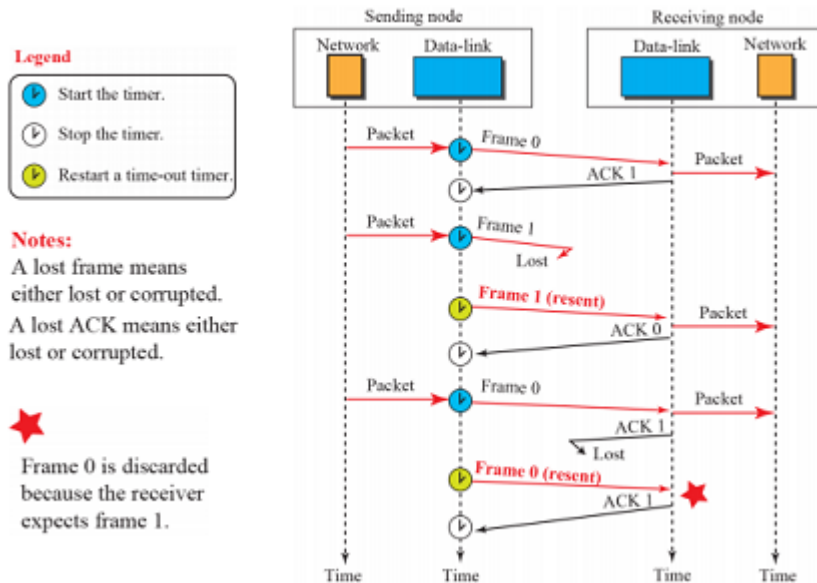
Notes:
 A lost frame means either lost or corrupted.
 A lost ACK means either lost or corrupted.



- ACK가 Lost되면, Sender가 같은 프레임의 재송신하고 Receiver가 이를 받아들여 똑같은 패킷을 두개 받게 되는 오류 발생

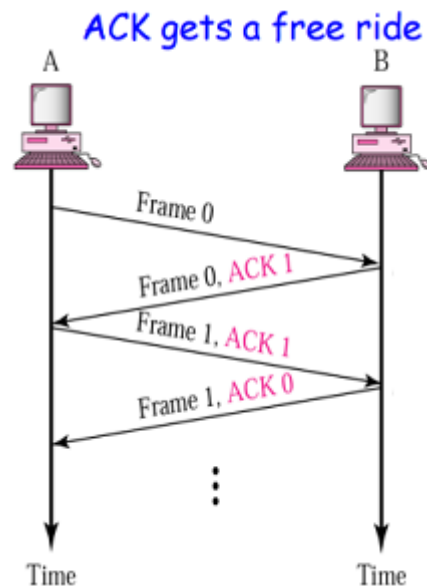
Sequence and Acknowledge Numbers

- Receiver 쪽의 Network Layer가 중복 패킷을 받는 상황을 피하기 위해, 각 패킷은 자신이 몇번째로 받아져야 하는 패킷인지를 알고 있어야 함
- 데이터 프레임에 Sequence Number 추가, ACK 프레임에 ACK number 추가
 - Sequence는 0으로 시작해 0,1 왔다갔다
 - ACK는 1으로 시작해 0,1 왔다갔다



Piggybacking

- 위의 두 프로토콜은 **단방향** 커뮤니케이션을 위해 설계됨
 - ACK를 제외하면 한쪽방향으로만 데이터가 전송됨
- 양방향으로 데이터가 왔다갔다 하도록 설계된 프로토콜이 있음
 - 그러나 통신의 효율성을 위해, 한쪽 방향의 데이터가 ACK와 함께 피기백킹되어 반대편으로 보내짐
- 데이터와 ACK를 합치는 개념으로, Bandwidth를 아끼는 효과가 있음

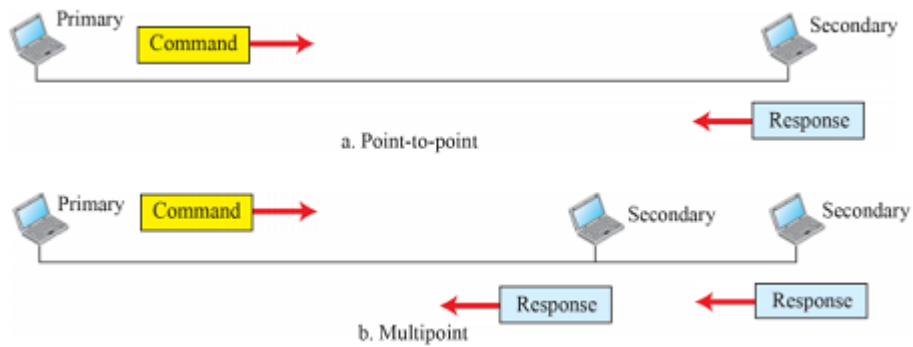


HDLC

- High-level Data Link Control
- Point-to-Point, Multipoint 링크를 통한 통신을 위한 Bit-oriented 프로토콜
- Stop-and-Wait 프로토콜을 구현함
- 실용적인 목적으로 구현된 프로토콜이지만, 이 프로토콜은 다른 실용적인 프로토콜의 Base가 되었음 ex) PPP, Ethernet, wireless LANs

Configuration and Transfer Modes

- 두개의 Transfer Mode 제공
 - Normal Response Mode (NRM)
 - Station Configuration is **Unbalanced**
 - One primary station(for sending commands)
 - Multiple secondary station(for only respond)
 - Point-to-Point, multipoint link 둘다에 쓰임

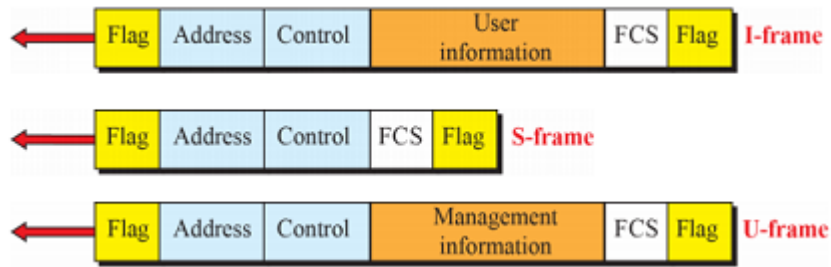


- Asynchronous Balanced Mode (ABM)
 - Station Configuration is **Balanced**
 - One primary station(for sending commands)
 - One secondary station(for only respond)
 - 링크가 Point-to-Point
 - 현재 가장 흔한 모드



Framing

- 모든 가능한 모드와 Configuration에서의 유연성 지원을 위해 세개의 프레임 타입 지원
- I-frames(Information Frames)
 - 유저 데이터를 data-link 하고, 유저 데이터와 관련된 컨트롤 정보(piggybacking)
- S-frames(Supervisory Frames)
 - 컨트롤 정보를 전송하기 위해 사용
- U-frames(Unnumbered Frames)
 - 시스템 관리를 위해 Reserved
 - 데이터링크 자기자신을 관리하기 위함



Flag

- 프레임의 시작/끝 구분
- 여러개의 프레임을 보낼 때는, 한 프레임의 끝 플래그가 다음 프레임 시작 플래그로 작동
- 패턴 : 01111110

Address

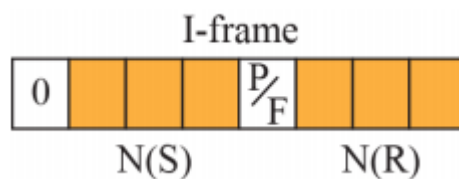
- 목표 스테이션의 주소를 저장
- Primary station이 보내고 있을 때, 주소는 Secondary Station의 주소(Target)
- Secondary Station이 보내고 있을 때, 주소는 Primary Station의 주소(Target)
- 사이즈 : 1 ~ n bytes

Control

- Flow / Error Control 위해 사용
- 사이즈 : 1 ~ 2 bytes

I-Frame

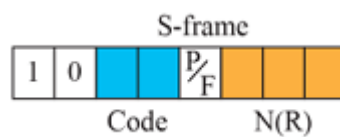
- 첫번째 비트 : 타입(0)
- N(S) 3비트 : 프레임의 Sequence Number(0~7)
- P/F 1비트
 - Poll(1) : Primary가 Secondary에게 보냄(Secondary의 주소)
 - Final(2) : Secondary가 Primary에게 보냄(Primary의 주소)
- N(R) 3비트 : 프레임의 ACK Number(**Piggybacking**이 사용됐을 경우)



S-Frame

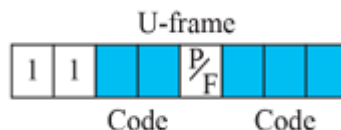
- Piggybacking이 불가능하거나 적절하지 못한 경우에만 사용
- 정보를 가지고 있지 않음
- 첫번째 비트 : 타입(10)
- 코드 2비트 : S-frame의 종류
 - 00 : Receive Ready(RR)
 - 받은 프레임(들)이 안전하고 단단하다는 뜻의 ACK

- N(R) 필드가 **ACK Number**를 뜻함
- 10 : Receive Not Ready(RNR)
 - Receiver가 바빠서 더 많은 프레임을 받을 수 없다는 뜻의 ACK
 - N(R) 필드가 **ACK Number**를 뜻함
 - Congestion Control 메커니즘의 역할을 함(Sender보고 느리게 해라)
- 01 : Reject(REJ)
 - Go-Back-N ARQ에서 사용 될 수 있는 NAK 프레임
 - N(R)이 **NAK Number**를 뜻함
- 11 : Selective Reject(SREJ)
 - Selective Reject ARQ에 사용되는 NAK 프레임
 - N(R)이 NAK Number를 뜻함
- P/F 1비트 : Poll/Final
- N(R) 3비트 : 프레임의 ACK/NAK Number(Piggybacking이 사용됐을 경우)

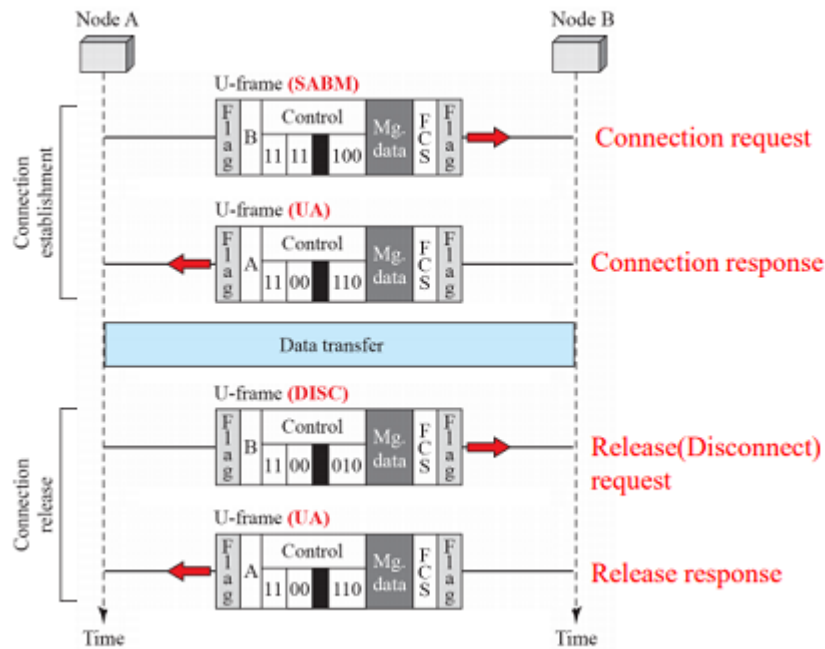


U-Frame

- 연결된 장치간의 세션 관리, 컨트롤 정보를 교환하기 위해 사용
- I/S와 다르게 Information Field(시스템 관리 정보, 유저데이터가 아님)가 있음
- 첫번째 2비트 : 타입(**11**)
- 코드 2비트(Prefix) : S-frame과 같음
- P/F 1비트 : I-Frame과 같음
- 코드 3비트(Suffix) : Prefix와 합쳐 5비트가 **32개의 다른 타입**을 뜻함



- 커넥션을 만들고/끊는곳에 사용

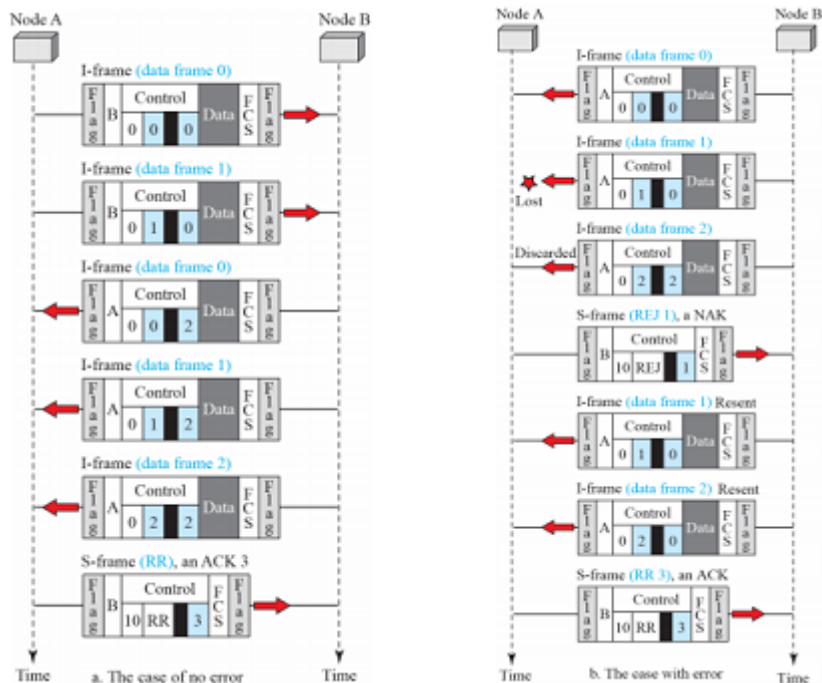


Information

- 유저의 데이터(I-Frame) 또는 관리 정보(U-Frame)

Frame Check Sequence (FCS)

- HDLC의 에러 탐지 필드
- 사이즈 : 2- ~ 4- 바이트 **CRC**



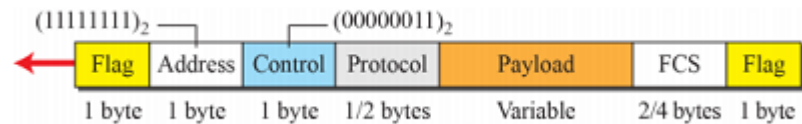
PPP

- Point-to-Point Protocol

- 컴퓨터 - 인터넷 서버 연결에 대부분의 인터넷 사용자들이 PPP 사용
- 데이터 전송을 관리하기 위해, Point-to-Point Protocol 필요 (Data-Link Layer)
- 가장가장 흔한 프로토콜

Framing

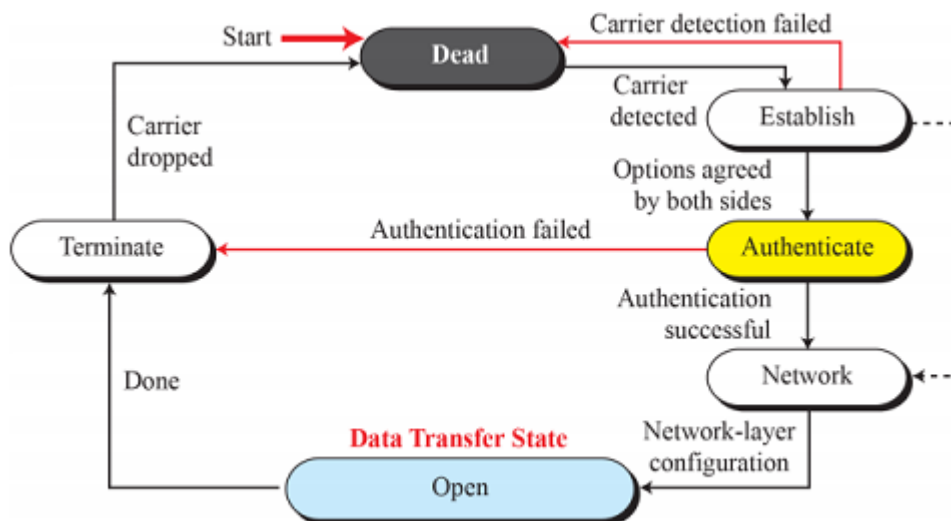
- Character-Oriented 프레임 사용



- Flag : 프레임의 시작/끝 구분, 패턴 01111110
- Address : 상수값 11111111 (FF) - 브로드캐스팅 주소
- Control
 - 상수값 00000011
 - PPP는 Flow Control을 제공하지 않음
 - Error Control 또한 탐지밖에 못함
- Protocol : 데이터 필드에 뭐가 있는지 정의(유저 데이터 또는 다른 정보)
 - 2 바이트 할당이지만 **1바이트만 사용**
- Payload : 유저 데이터, Byte Stuffing 필요
- FCS : 에러 탐지 필드
 - Size = 2- or 4- byte CRC

Transition Phases

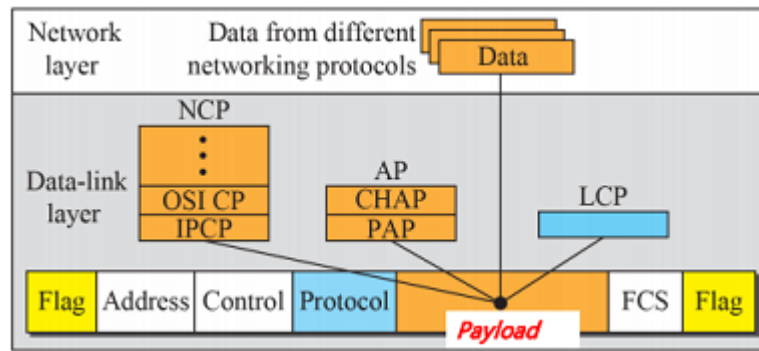
- PPP는 여러가지 Phase를 통과함



Multiplexing

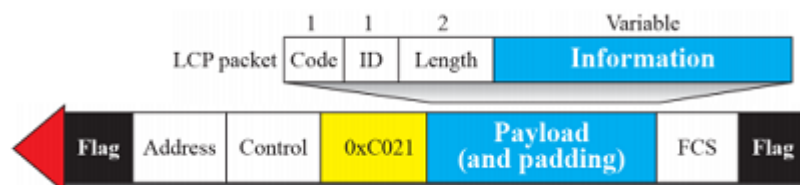
- PPP는 Link-Layer Protocol이지만, 다른 프로토콜의 집합도 사용함
 - 링크 Establishment, Party authentication, 네트워크 레이어 데이터 Carry
- PPP 패킷은 LCP/AP/NCP중 하나의 프로토콜로부터 데이터 Carry 가능

- 여러가지 네트워크 레이어로부터 데이터가 올 수 있음



LCP

- Link Control Protocol
- Protocol Field : 0xC021
- Code : LCP packet의 Type(11개 타입 있음)
 - 처음 4개 : link configuration / establish phase
 - 다음 2개 : link termination
 - 마지막 5개 : link monitoring / debugging
- ID : a value that matches a request with a reply
- Length : LCP packet의 길이
- Information : Option type, Option Length, Option Data등의 정보
 - 페이로드 필드 사이즈, 권한확인 프로토콜 등

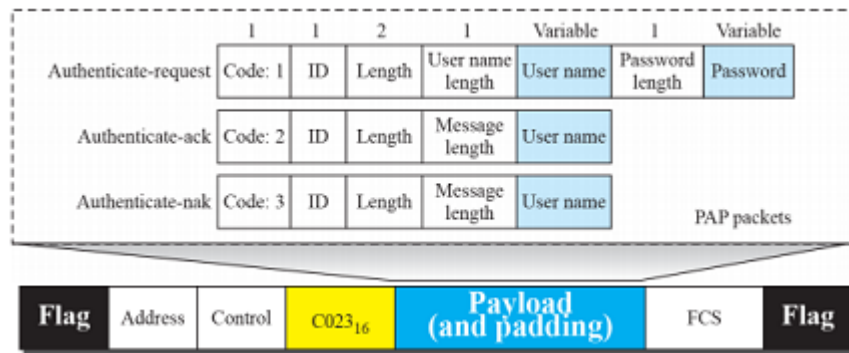


Authentication Protocols

- Authentication : 자원에 접근하려는 유저의 신분정보를 확인하는 과정

PAP

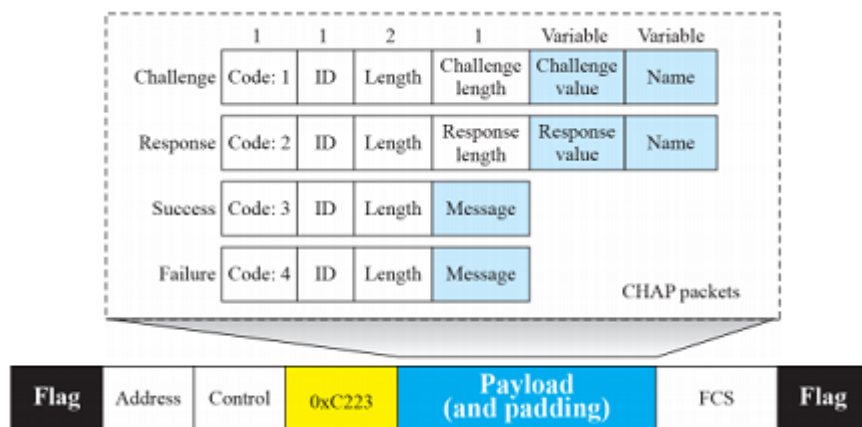
- Password Authentication Protocol
 - Protocol Field : 0xC023
1. Authentication Request : 유저가 ID/PW를 시스템에 보냄
 2. Authentication-ACK or NAK : 시스템이 유효성을 확인하고 Accept/Deny



CHAP

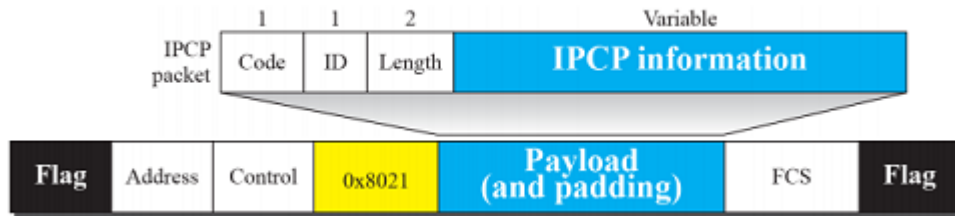
- Challenge Handshake Authentication Protocol
- Protocol Field : 0xC223
- Three-way

1. 시스템이 유저들에게 **Challenge** 패킷을 보냄(Challenge 값 저장)
2. 유저가 미리 정의된 함수에 Challenge Value + 자신의 패스워드를 넣어서 결과 생성, 시스템에 전송
3. 시스템이 2번과 같은 값을 넣어보고, 결과값이 같으면 접근 허용, 아니면 거절



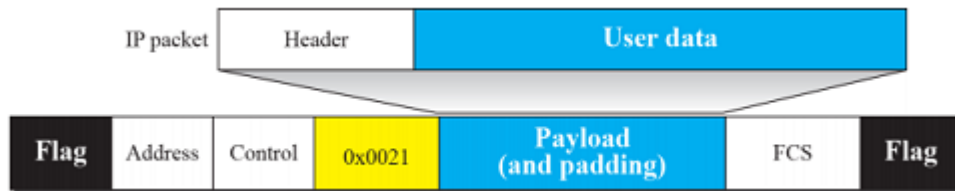
NCP

- PPP : Multi-Network-Layer 프로토콜
- 모두 다 네트워크 레이어 데이터를 가지고 있지는 않음
 - 그냥 들어오는 데이터를 위해 링크를 Configure 하는 역할
- Internet Protocol Control Protocol(IPCP)
 - Protocol Field : 0x8021
 - Configuring the link for carrying IP packets in the Internet
- OSI Network Layer Control Protocol
 - Protocol Field : 0x8023
- Xerox NS IDP
 - Protocol Field : 0x8025
-



Data From the Network Layer

- 네트워크 레이어 Configuration이 NCP에 의해 끝난 후, 사용자가 네트워크 레이어와 데이터 패킷을 주고받을 수 있음
- PPP가 IP 네트워크 레이어로부터 데이터를 Carry하는 경우
 - Field : 0x0021
- PPP가 OSI 네트워크 레이어로부터 데이터를 Carry하는 경우
 - Field : 0x0023



Multilink PPP

- 하나의 Point-to-Point 링크에 여러개의 채널이 가용하도록
- PPP frame의 논리적 구조가 여러개의 실질적 PPP 프레임으로 나누어져 있음

