

-컴퓨터 네트워크-

Protocol Functions II

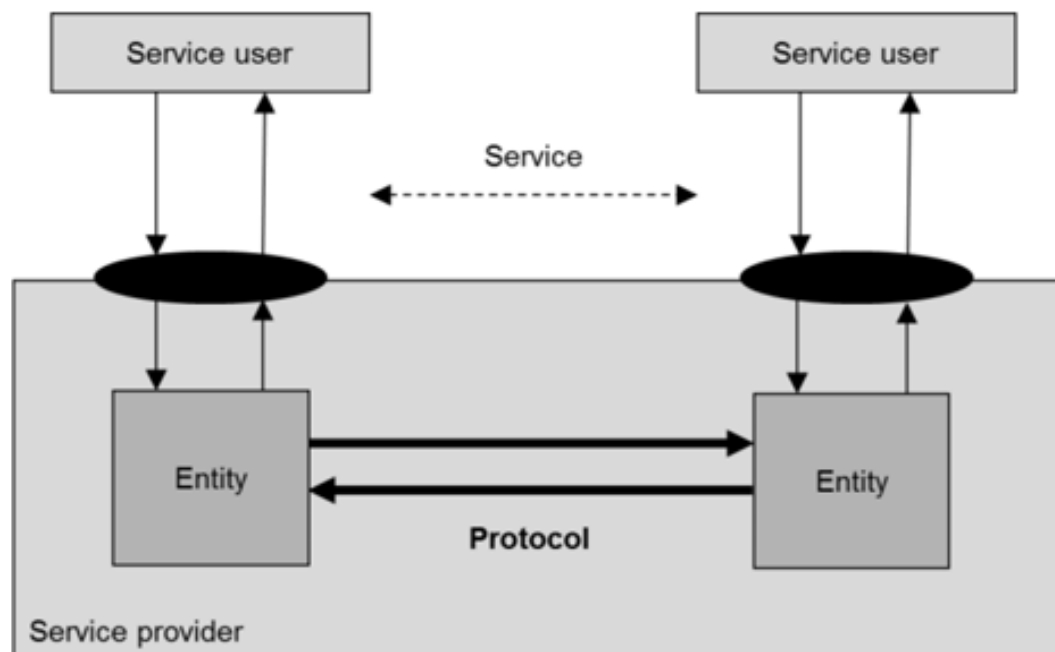
2022 Spring
Kyungseop Shin

Course Outline

- 다양한 Protocol function들에 대해 이해
 - Synchronization
 - PDU handling / flow control

Protocol Function 개념

- protocol model은 일반적인 protocol 모양새에 대한 내용
 - 실루엣은 보이나, 실제 생김새는 아직 실감이 안감
 - 실제 생김새 = protocol function

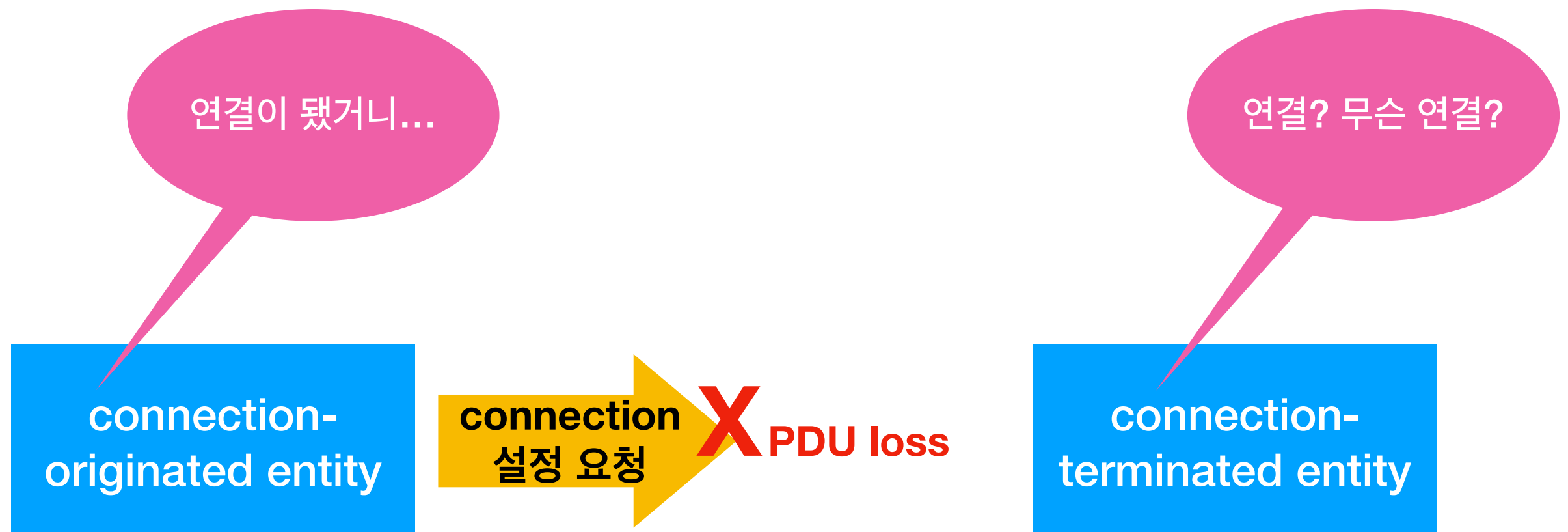


Protocol Function 개념

- 여러 protocol에서 두루 사용되는 특정 procedure / mechanism
 - 여기저기서 공통적으로 등장하는 주연/조연 같은 존재
- 예시 :
 - error control
 - PDU가 정상적으로 전달되지 않은 상황에서의 protocol entity의 동작
 - fragmentation, flow control
 - entity간 data를 서로 주고받는 속도 및 형태 조절

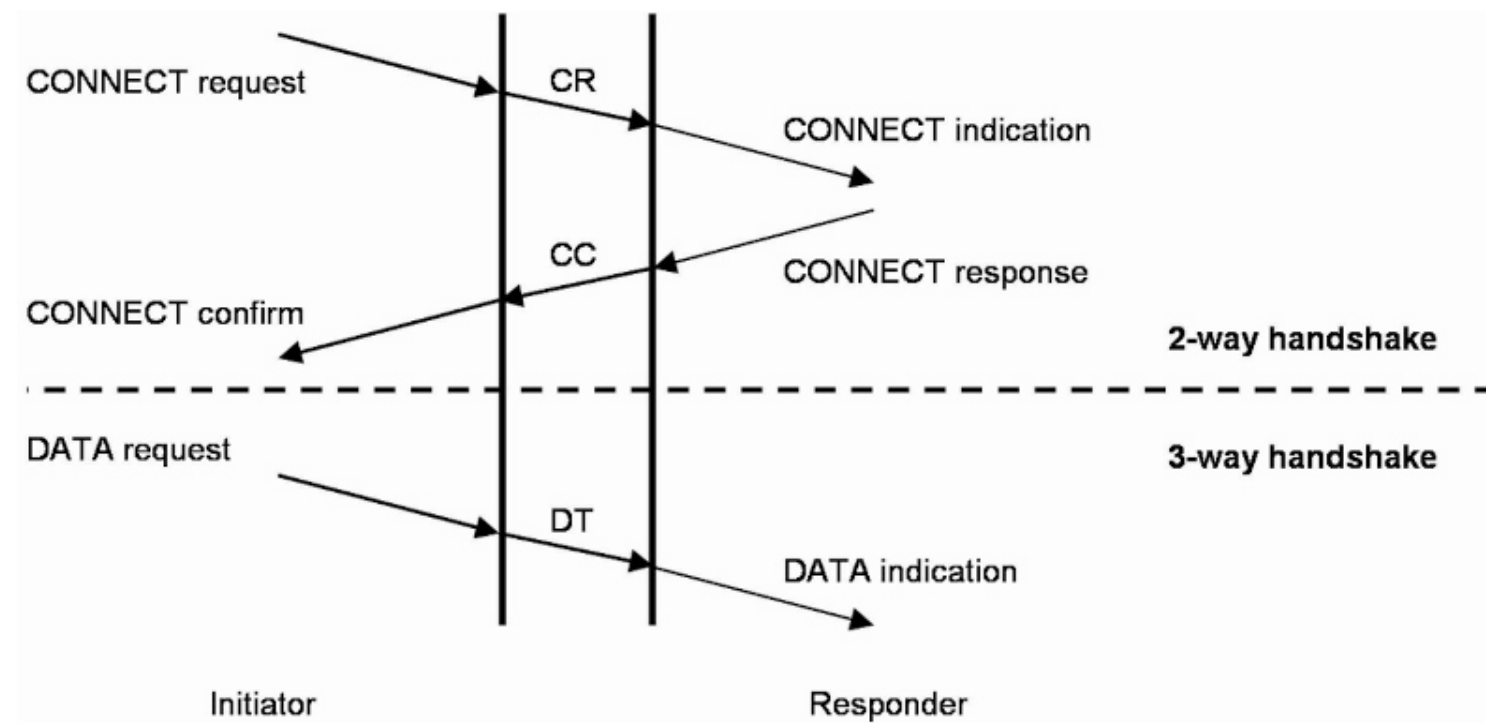
Synchronization

- entity 간 consistent한 protocol operation을 위해 서로의 state 를 맞추는 event 동작이 필요함
- connection setup / release에서 주로 활용됨



Synchronization

- peer entity간 connection (호 연결)
 - data를 보내기 전 connection이 맺어졌는지 여부 확인
- handshaking : 서로 합의하는 과정처럼 message를 주고받음
 - 2-way or 3-way

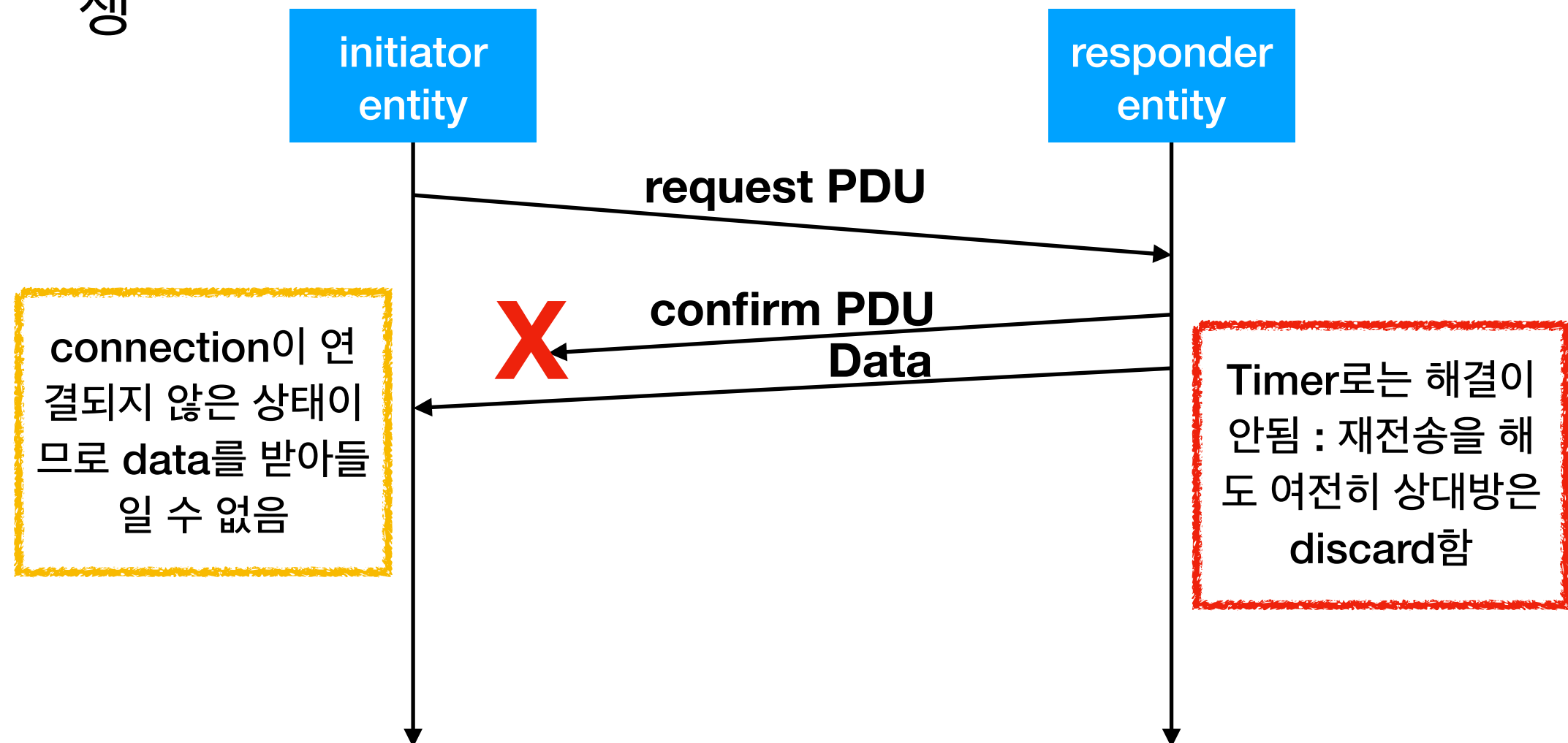


Synchronization

- 2-way handshaking
 - initiator entity : request PDU 전송
 - 관련 parameter를 모두 담아서 한번에 responder에게 전달
 - responder entity : confirm PDU 전송
 - confirm PDU 전송 직후 connection이 맺어졌다 가정
 - confirm PDU이 loss가 발생해도 마찬가지로 connection이 맺어졌다 가정

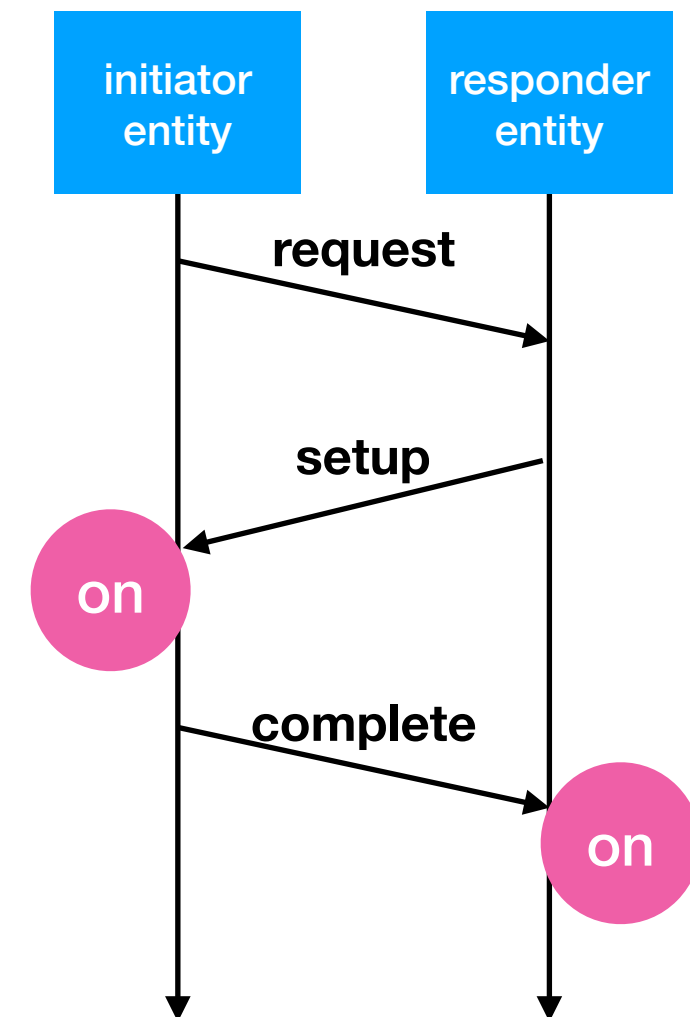
Synchronization

- unidirectional 형태 전송은 2-way handshake로 충분하나,
- bidirectional 형태 전송은 2-way handshake로 하면 문제 발생



Synchronization

- 3-way handshake : 주고 받고 또 주고
 - request / setup / complete
 - Initiator가 confirmation 수신에 대해 responder에게 confirm을 또 해줌 (complete)
 - responder는 complete을 수신한 뒤에 connection이 맺어진 것으로 간주
 - complete에 loss가 일어나도 responder는 connection이 맺어진 상태로 가정할 수 있음
 - 이전에 request는 받았으므로, connection을 맺고자 하는 initiator의 상태 확인
 - 후속 data PDU을 수신하면 complete을 수신한것으로 간주할 수 있음



Connection Management

- connection-oriented protocol에서 connection을 설정/관리/해제를 하기 위한 동작
 - Connection establishment / data transfer / connection release
- Connection establishment
 - Establish connection : entity간 connection 연결 수용/거부 상태를 맞추는 과정
 - handshake 과정을 통해 synchronization
 - Negotiate the quality of service (QoS) : 전송에 대한 품질 협의
 - throughput, delay, error/loss rate

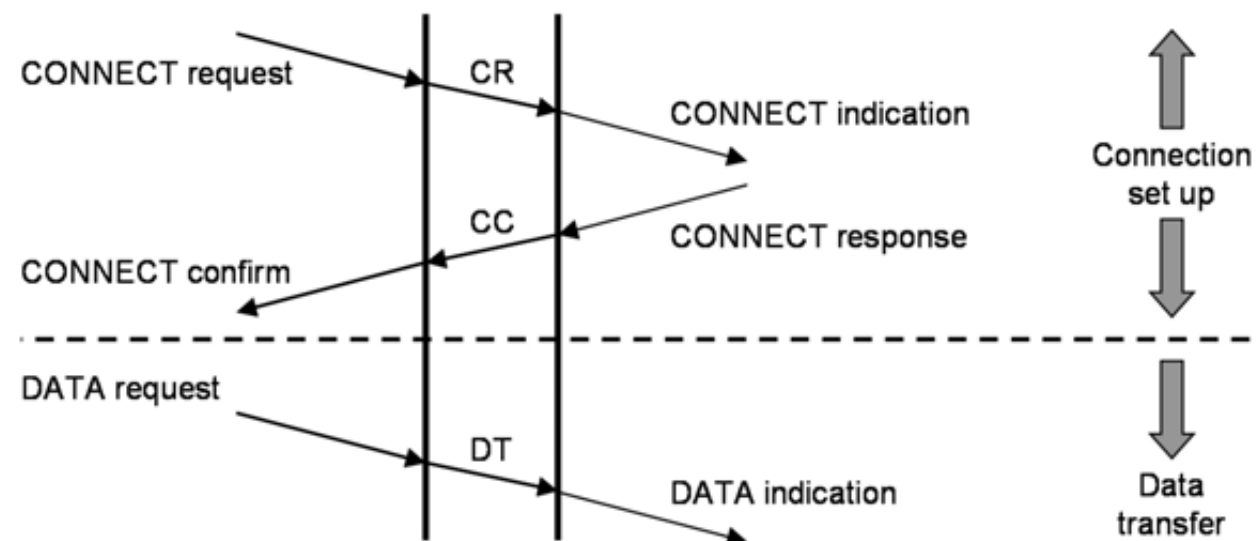
Connection Establishment

- responder는 connection request에 대해 reject(거절)할수도 있음
- resource 부족으로 connection을 다루지 못함
- request 메시지 내에 요청한 QoS가 받아들일 수 없을 때
 - reduce의 경우 responder가 역제안을 주고, initiator가 받을지 말지 판단



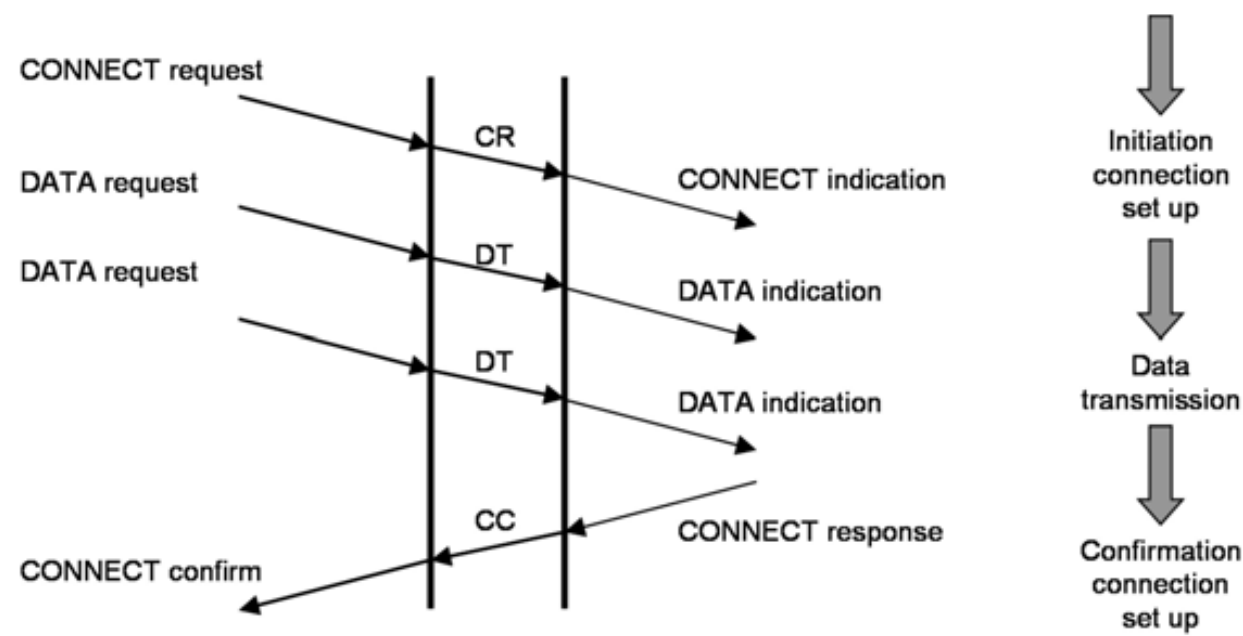
Connection Establishment

- peer entity들 간 connection을 설정하는 방법
 - explicitly : 메시지 교환을 마치고 서로 연결 상태로 전환
 - connection setup / data transfer phase 두가지로 명확하게 나뉨
 - 2-way 혹은 3-way handshake 활용



Connection Establishment

- implicitly : connection establishment 시도 시작과 동시에 connection이 맺어졌다고 가정
 - confirmation 메시지를 기다리지 않고 REQ 전송과 함께 data 전송
 - connection establishment delay가 성능 저하를 일으키지 않음
 - loss 혹은 connection reject시 PDU 재전송이 필요함



Connection Maintenance

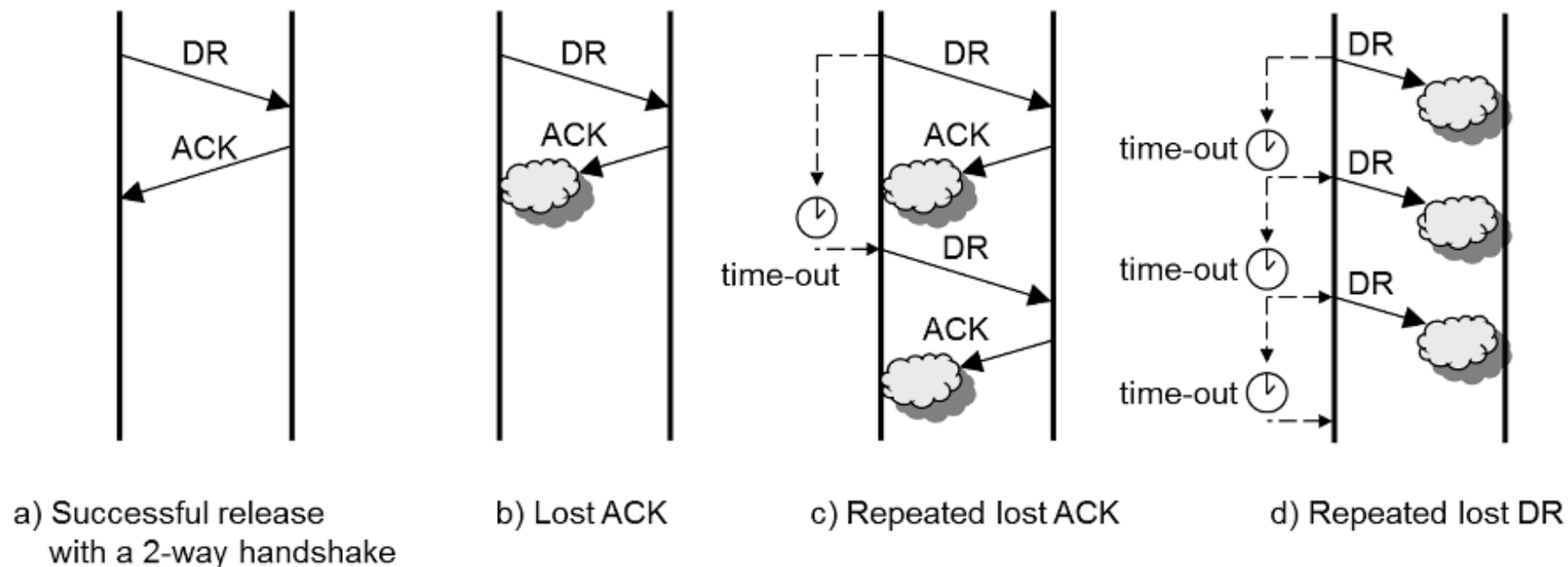
- Breakdown of a connection
 - 하위 layer에서 문제가 발생하면 상위 layer에게 이를 알림
- Re-establishment of connection
 - 상위 layer에서 하위 connection에 문제가 발생한 경우 취할 수 있는 조치
 - re-synchronization : broken (N-1) connection 에 대해 re-establishment를 함
 - reassignment : 새로운 (N-1) connection에 대해 establishment함

Connection Release

- SAP간 Connection이 더이상 쓸모가 없어졌을 때 수행
 - 불필요한 상황으로 인해 정상적으로 종료하는 경우 : explicit release
 - 비정상적으로 종료하는 경우 : abrupt release
- Explicit release는 entity간 서로 합의 하에 종료
 - peer간 release 상태에 대한 synchronization
 - release직전의 data 전달 동작

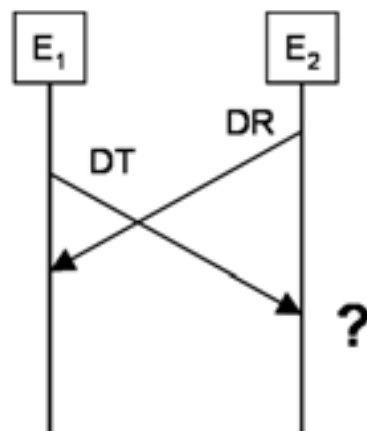
Connection Release

- Release 상태에 대한 synchronization은 peer 간 메시지 교환을 통해 가능
- 다만 아래 메시지 유실 상황을 대비해야 함
- 잘못하면 half-open connection problem 발생

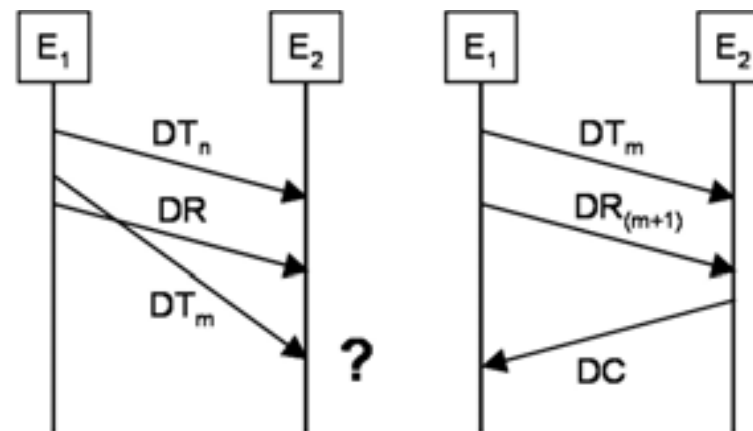


Connection Release

- Unidirectional connection에 대한 release를 하려면
 - TX entity 쪽에서 release 요청을 해야 함
 - RX entity는 PDU 전송 상황을 모르므로 release를 요청하면 PDU 유실 여지가 있음
 - release 요청 PDU 역시 sequence number를 가져야 함
 - Reordering이 필요한 상황에서 마지막 PDU가 유실될 여지가 있음
- Release 절차는 2 way handshake 형태로 이루어져야 함
 - release 요청 PDU의 유실 상황에 대한 대비



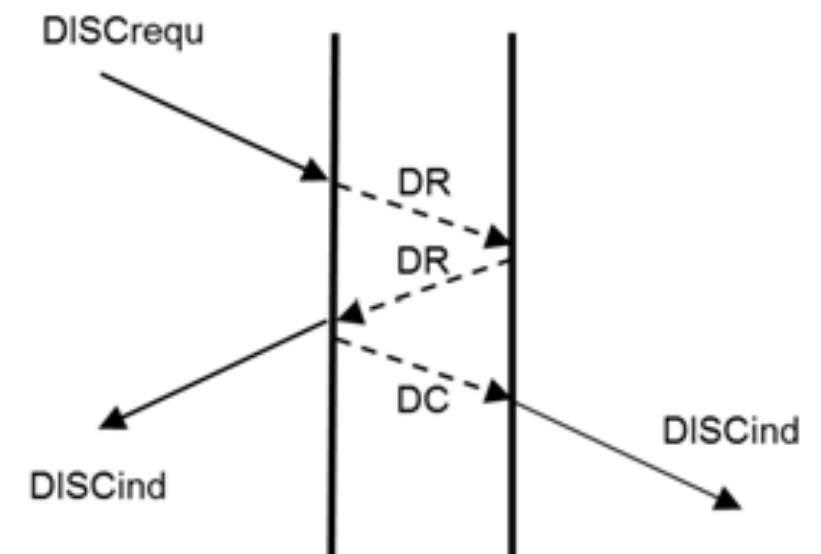
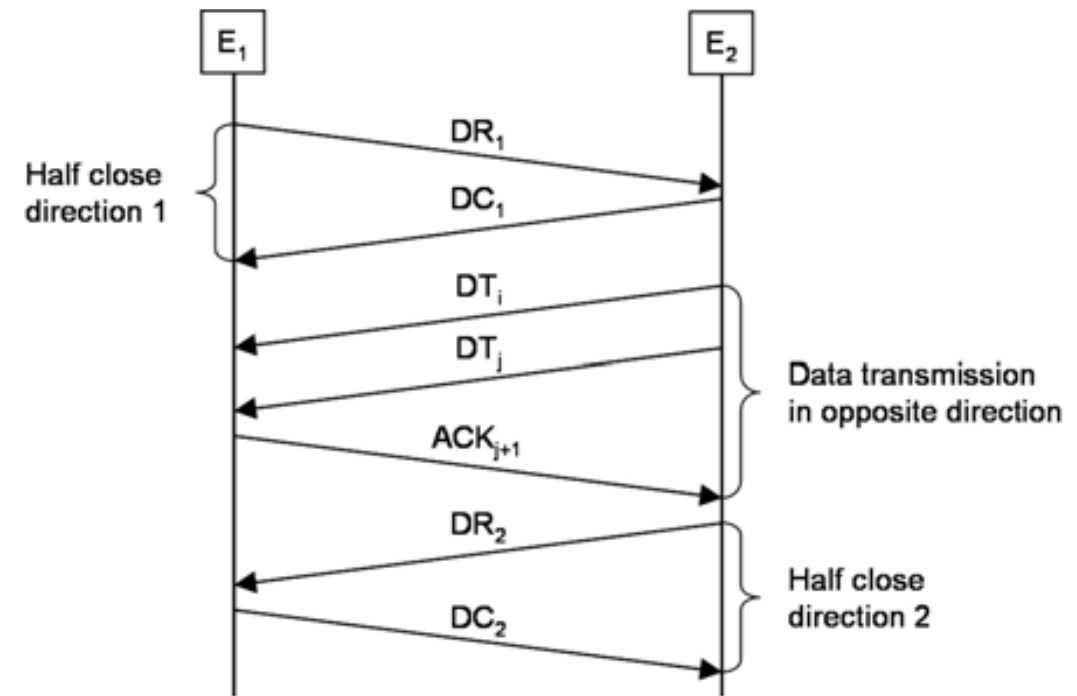
a) Who may close the connection?



b) Connection release with send information (plus 2-way handshake)

Connection Release

- Bidirectional connection에 대한 release를 하려면
 - 기본적으로 unidirectional connection 두개에 대해 각각 release를 하는 것과 같음 (=half close)
 - 3-way handshake 절차를 활용하면 한 번에 양방향에 대해 동시에 release 가능
 - PDU loss에 대한 대비를 통해 synchronization 문제 해결을 할 필요가 있음



Connection Release

- 3-way handshake에서의 synchronization 문제

- 대부분 PDU Timer로 해결

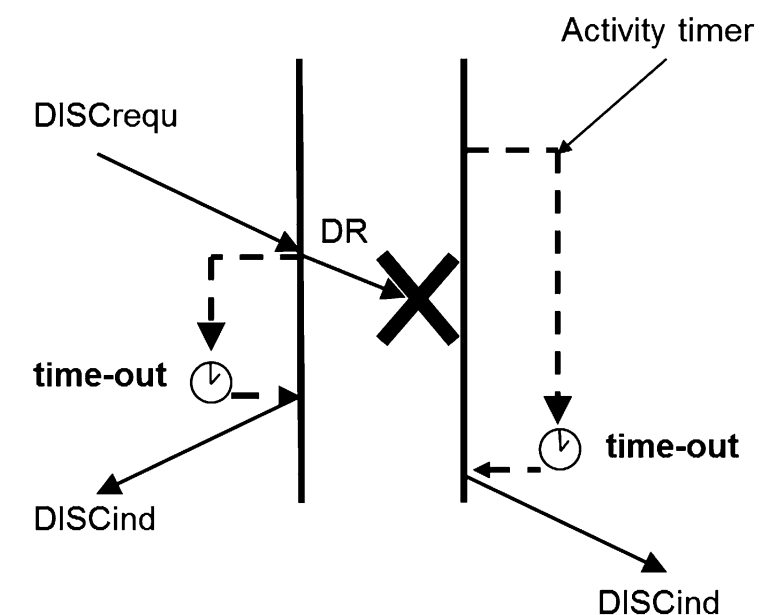
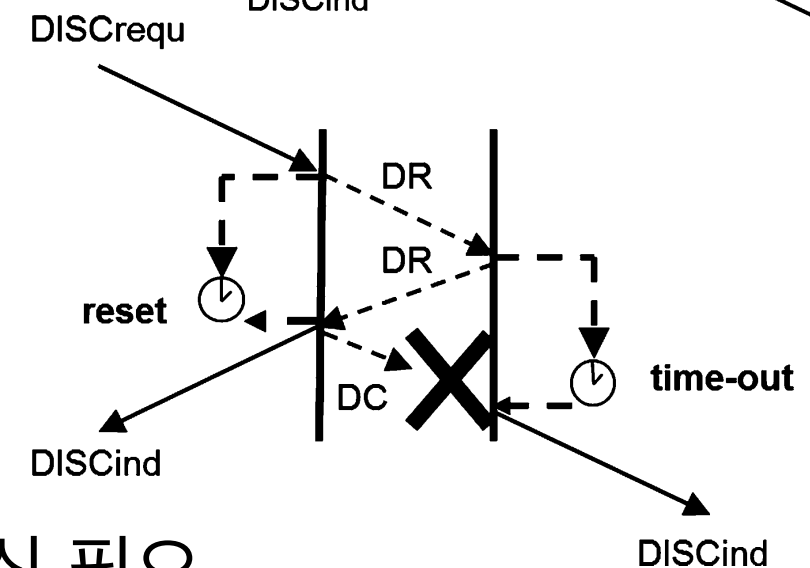
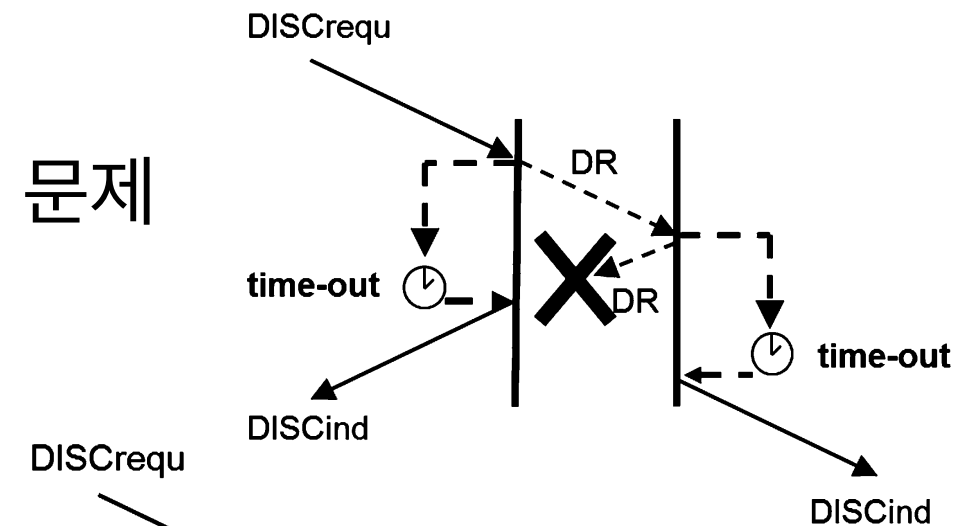
- msg2 loss : 양쪽 모두 expiry

- msg3 loss : 수신 쪽만 expiry

- msg1 loss : activity를 감시하는 별도 방식 필요

- TX : PDU Timer expiry

- RX : activity timer expiry

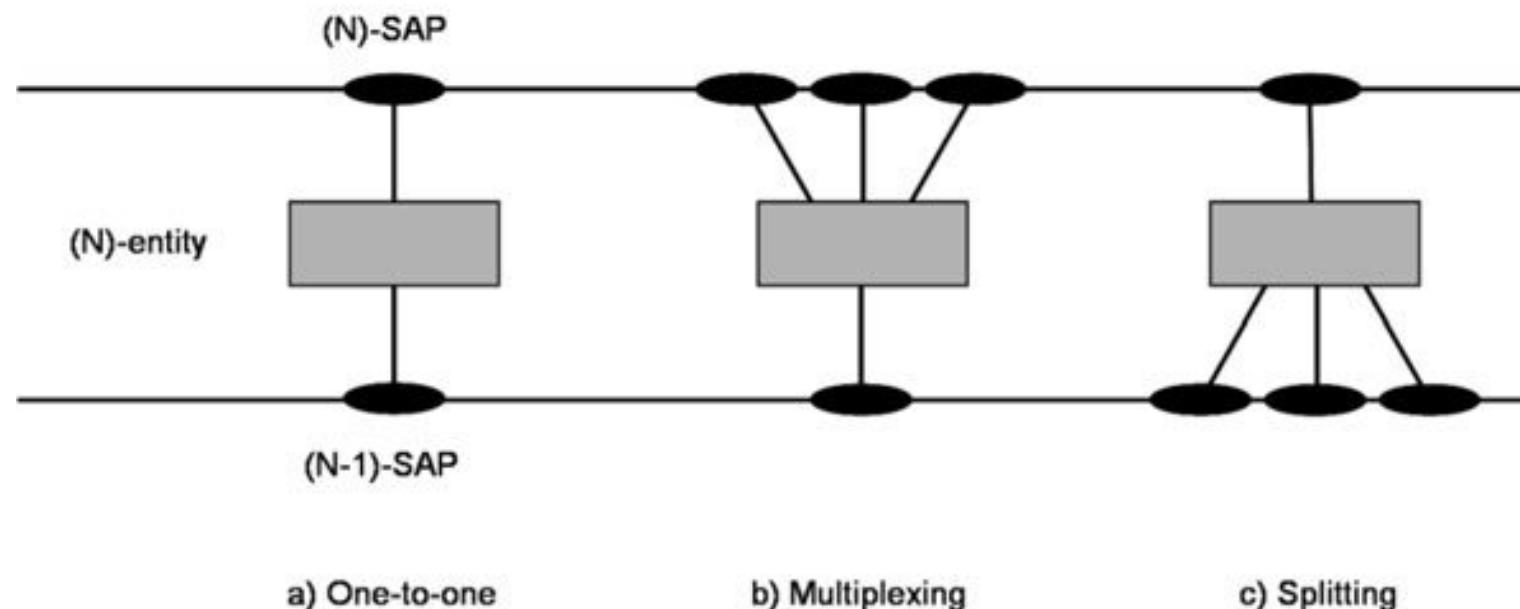


Connection Release

- Abrupt connection release : immediate break-up
 - 비정상적인 상황에서 곧바로 connection 해제
 - data transfer loss가 필연적으로 발생할 수 밖에 없음
 - exceptional case
 - irreversible transmission error
 - security issue
- Reuse of connection reference 문제에 대한 방지
 - freezing of connection reference : 일정 시간동안 해당 연결을 사용하지 않음

PDU Adjustment

- N -> N-1 layer mapping 형태는 크게 세가지가 존재
 - one-to-one
 - Multiplexing : 상위 layer의 여러 SAP을 모아서 처리
 - Splitting : 하위 layer의 여러 SAP으로 분배
- SDU에 대한 처리도 구조에 맞게 이루어짐



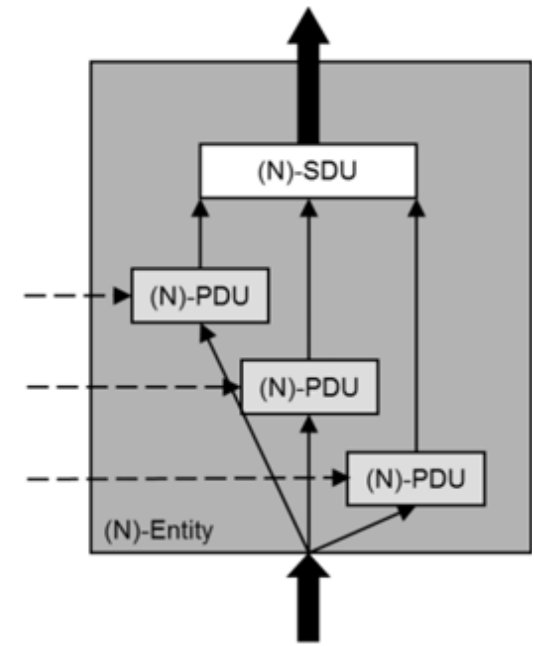
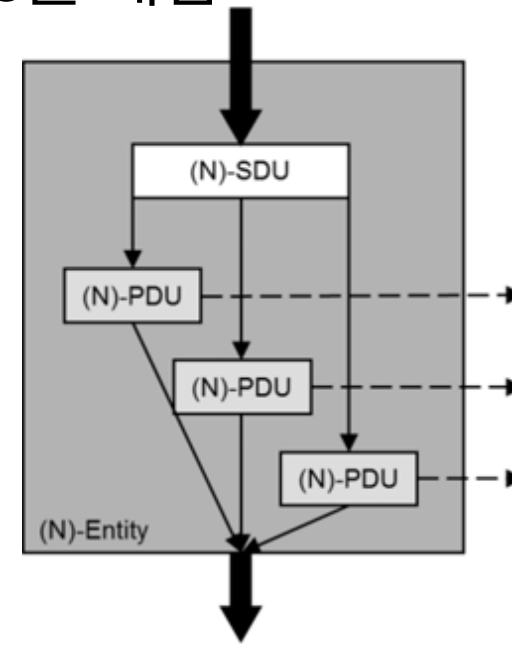
PDU Adjustment

- Multiplexing 시 필요한 protocol functions
 - Scheduling : N layer로부터 동시에 PDU 수신 시 (N-1) layer 로 우선 보낼 PDU 선택
 - Flow control : (N-1) connection capacity 에 대한 regulation
 - Assignment : 수신한 (N-1) PDU를 N layer에 적절히 올림
- Splitting 시 필요한 protocol functions
 - Scheduling : (N-1) layer의 SAP으로 보낼지 결정
 - aggregation : 수신 측에서 (N-1) connection들로부터 N PDU 생성

PDU Adjustment

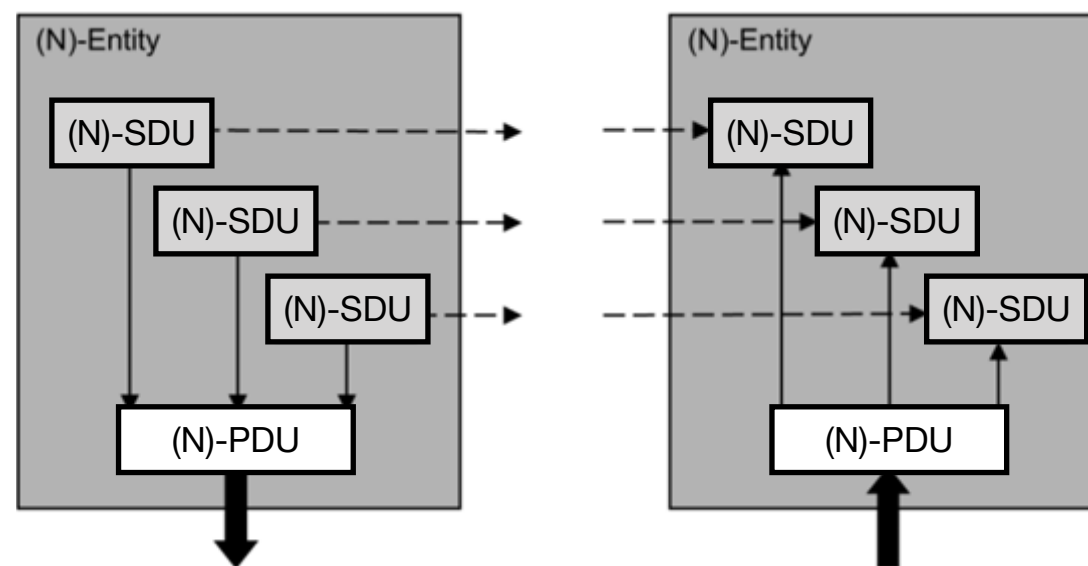
- SDU에 대한 길이 조정
 - maximum transfer unit (MTU)에 따라 윗단의 SDU size를 조정해야 하는 경우
 - SDU를 자르거나 합쳐서 PDU를 만듦
- fragmentation : SDU가 너무 크면 잘라서 여러 PDU 생성
 - Segmentation : SDU를 자르고 sequence를 매김
 - reassembling : reorder하고 합침

overhead가 증가하긴 하나, error control, resynchronization 등 관점에서도 잘게 쪼개는게 유리할 수 있음



PDU Adjustment

- concatenation : 작은 SDU를 하나로 합쳐서 보냄
- Chaining : 여러 SDU를 하나의 PDU로 합침 (합친 정보를 header에 포함)
- separating : 다시 분리

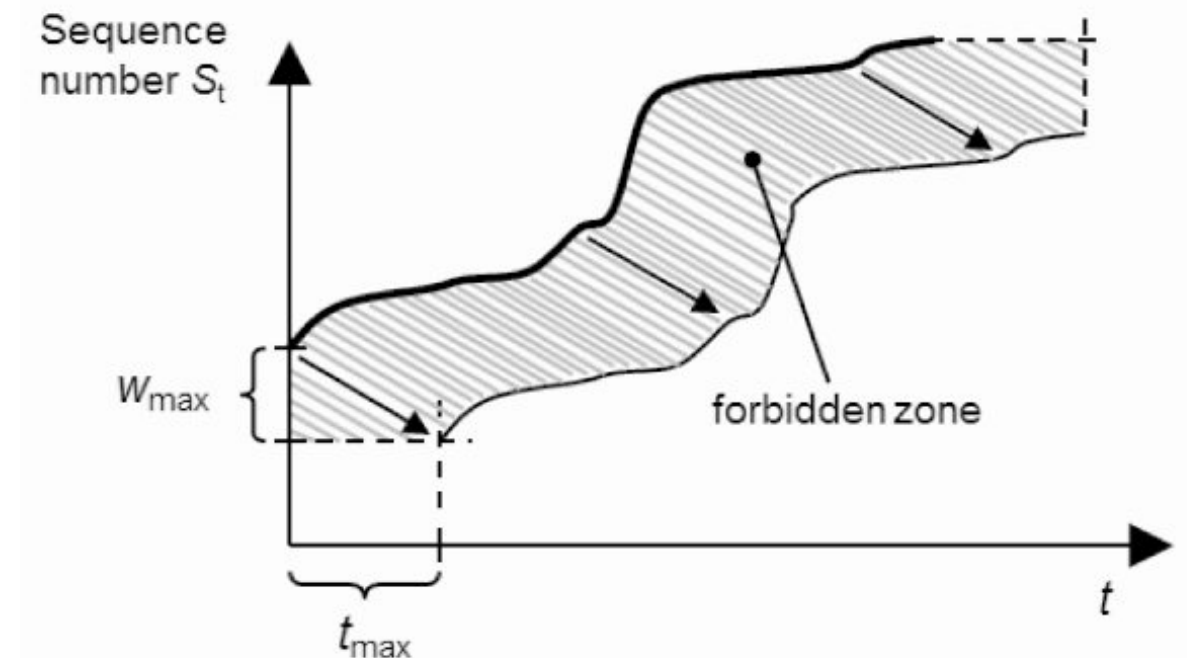


Sequence Number

- Sequence number : connection management, PDU adjustment를 하는 대부분의 프로토콜은 번호 붙이기를 함
 - PDU header에 붙임
 - connection 상에서 전달되는 PDU의 전송상태 확인 (synchronization)
 - reordering / duplication
 - fragmentation / concatenation

Sequence Number

- Sequence number를 관리하는데 있어서 두가지 고려 사항
 - 서로 다른 PDU에 동일한 sequence 부여는 피해야 함
 - freezing connection reference후 sequence reset
 - forbidden zone 정의
- sequence number overflow
 - 최대값이 충분히 커야 함
 - wrap around



Flow Control

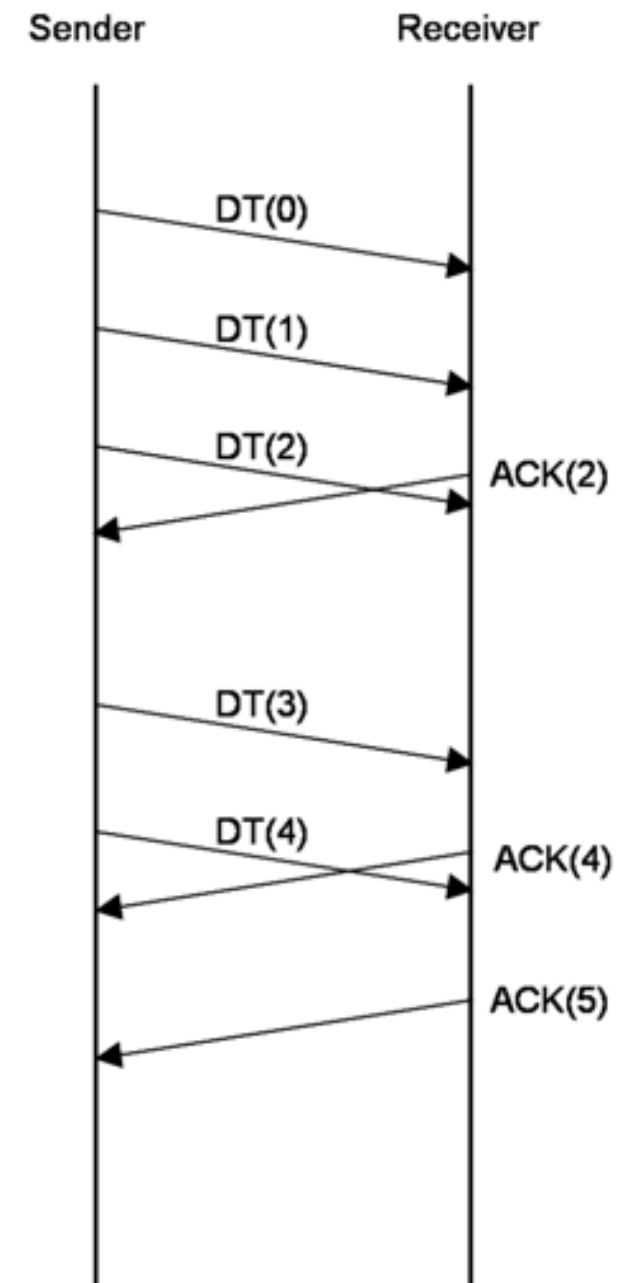
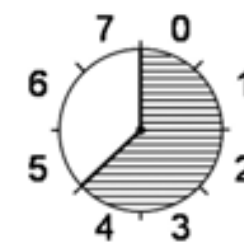
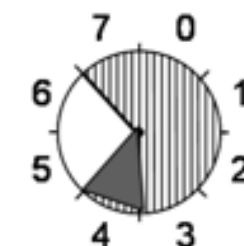
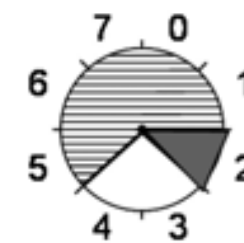
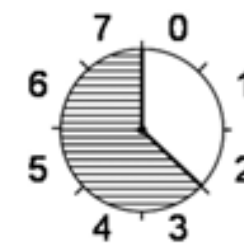
- 송수신 entity 사이에 교환되는 PDU 개수 조정
 - receiver entity에서의 processing overload
 - channel의 용량
- window-based flow control : “window” (PDU range)내에서만 전송
 - first transmission에 대해서 PDU range내에서 제한적으로 전송
 - retransmission은 제한 없이 함

Flow Control

- start/stop procedure
 - 수신 측에서 stop signal을 보내서 제어
 - 잦은 stop signal로 인해 discontinuous, bursty data flow 발생
- credit procedure
 - 수신 측에서 credit을 주고, sender는 허용된 credit (보통 sequence number 범위)에 따라 전송
 - 송신 측은 허용된 범위의 PDU를 보내고 나서 credit을 새로 받을때까지 기다림

Flow Control

- Sliding window protocol
 - 가장 보편적인 credit procedure
 - 수신측이 제공한 credit 정보는 곧 window를 sliding해주는 양



Flow Control

- Rate-based flow control
 - end-to-end 제한 뿐 아니라 network load 도 고려하는 흐름 제어
 - network의 overload 상황을 방지할 수 있음
 - 송신 측에서 전송률에 대한 제한을 두고 전송
 - 허용된 burst (특정 시간동안 보내는 data) size를 제어

