목차

TCP와 UDP 기반 데이터 전송 과정 및 매체 접근제어 기술 비교 분석

- 26AI 조영규

1. TCP 기반의 데이터 전송과정(Slide 1~22)

- 기본적으로 패킷을 받을 때 하는 오류 검출 과정은 제외

Slide 1/22

- 상황: 애플리케이션에서 데이터를 보내기 위해 TCP 연결을 만드는 과정
    - 애플리케이션 계층에서 데이터를 전송하기 위해 192.168.3.2와 연결을 요청
    - 전송 계층에서 TCP 사용을 결정
    - TCP연결을 위해 192.168.3.2로 SYN을 보내려고 함
    - SYN 플래그를 1로 설정하고 헤더를 추가하여 하위 계층으로 보냄(캡슐화)

Slide 2/22

- 상황: L3에서 SYN을 보내기 위해 송신지 IP, 수신지 IP를 설정하는 과정
    - IP헤더를 추가하기 위해 자신의 IP주소와 보낼 상대의 IP주소를 설정한다.
    - 송신지 IP: 192.168.3.1
    - 수신지 IP: 192.168.3.2
    - 생성된 IP헤더를 추가하여 2계층으로 보낸다. (캡슐화)

Slide 3/22

- 상황: TCP 연결을 위해 192.168.3.2으로 패킷을 보내야 하지만, ARP Cache Table에 존재하지 않는다.
    - IP를 확인하여 수신지가 현재 192.168.3.0망에 존재하는 것을 파악할 수 있다. (내부 망)
    - 우선, 수신지 MAC 주소를 알기 위해 ARP request를 보낸다.
    - ARP request는 브로드 캐스트 패킷으로, 수신지 MAC 주소를 몰라도 보낼 수 있다.

Slide 4/22

- 상황: 192.168.3.2의 MAC 주소를 얻기 위해 ARP request 패킷을 보내는 과정
    - ARP를 보내기 위해 기존의 패킷을 임시 저장한다.
    - ARP 패킷을 보내기 위해 자신의 MAC 주소인 0800:0222:2222를 송신지 MAC으로 설정하고, 브로드 캐스트 MAC주소를 목적지 MAC 주소로 설정한다.
    - 브로드 캐스트 된 패킷은 내부 망 전체에 전달된다. (이더넷 헤더 추가 후 1계층으로 전달. 캡슐화)

Slide 5/22

- 상황: 192.168.3.2의 MAC 주소를 얻기 위해 ARP request 패킷을 보내는 과정
    - 생성된 ARP request 패킷을 브로드 캐스트로 전달

Slide 6/22

- 상황: 192.168.3.2에서 ARP request 패킷을 받음
    - 우선 수신지 MAC 주소를 확인하여 자신에게 온 패킷인지 확인한다.
    - 브로드 캐스트 패킷이므로 상위 계층으로 전달
    - 패킷의 type을 확인하여 0x0806을 보고 ARP 패킷임을 판별

Slide 7/22

- 상황: 2계층에서 ARP request를 확인하고 상위 계층(ARP)으로 전달
    - 이더넷 헤더를 제거한 후 상위 계층으로 전달(역캡슐화)

Slide 8/22

- 상황: ARP request를 받아 패킷에 있는 IP와 MAC주소를 ARP Cache Table에 등록
    - 자신의 IP주소로 온 것까지 확인한 후 송신지 IP주소와 MAC주소를 192.168.3.2의 ARP Table에 등록한다.
    - 응답을 돌려줄 수 있는 상태가 되었다.

Slide 9/22

- 상황: ARP reply를 보내는 과정
    - ARP reply에는 자신의 IP주소(192.168.3.2)와 MAC주소(0800:0222:1111)가 포함된다.
    - 3계층에서는 IP주소를 담은 헤더를 추가하여 2계층으로 보낸다. (캡슐화)
    - 2계층에서는 MAC주소를 담은 헤더를 추가하여 1계층으로 보낸다. (캡슐화)

Slide 10/22

- 상황: ARP reply를 보내는 과정
    - 192.168.3.2에서 192.168.3.1로 ARP reply 패킷을 송신한다.
    - 수신지 IP 주소와 MAC 주소를 알고 있어 Unicast 방식으로 전송한다.

Slide 11/22

- 상황: ARP reply를 수신하는 과정
    - 들어온 ARP reply 패킷의 수신지 MAC주소를 확인한다.
    - 자신의 MAC 주소와 일치하여 통과시키고, type을 통해 ARP 패킷임을 확인한다.
    - 2계층 헤더를 제거하고 3계층의 ARP로 보낸다.

Slide 12/22

- 상황: ARP reply를 수신하는 과정
    - 데이터 링크 계층(L2)에서 네트워크 계층(L3)으로 패킷을 전송

Slide 13/22

- 상황: ARP reply를 수신하는 과정
    - 패킷의 수신지 IP주소를 확인하여 자신의 것이 맞는지 확인
    - 확인 후 해당 패킷의 송신지 IP와 송신지 MAC 주소를 자신의 ARP Cache Table에 등록한다.

Slide 14/22

- 상황: TCP 3-WHS 과정
    - 목적지 MAC 주소를 알아내어 TCP 3-WHS 과정을 재개
    - 4계층에서 TCP헤더(송/수신지 포트 번호, SYN)
    - 3계층에서 IP헤더(송/수신지 IP주소)
    - 2계층에서 이더넷 헤더(송/수신지 MAC주소)
        - 위 캡슐화 과정을 진행 후 전송

Slide 15/22

- 상황: TCP 3-WHS 과정
    - 목적지에 도착한 패킷의 수신지 MAC주소, 수신지 IP주소, 수신지 포트 번호를 통해 원하는 애플리케이션 간에 연결을 시작한다. (역캡슐화 진행)
    - 받은 SYN에 대한 ACK를 보낼 준비를 한다.

Slide 16/22

- 상황: TCP 3-WHS 과정
    - SYN에 대한 ACK를 보낸다.
    - 슬라이드 14번과 같은 캡슐화 과정을 거쳐 SYN, ACK를 전송한다.

Slide 17/22

- 상황: TCP 3-WHS 과정
    - 192.168.3.2에서 192.168.3.1로 SYN, ACK를 보낸다.
    - 역캡슐화 과정을 통해 TCP에서 SYN, ACK를 확인한다.
    - 192.168.3.2이 보낸 ACK를 받았다는 ACK를 보낼 준비를 한다.

Slide 18/22

- 상황: TCP 3-WHS 과정
    - 연결이 수립되었음을 알리는 ACK를 보낸다.
    - 캡슐화 과정을 거쳐 패킷을 전송한다.

Slide 19/22

- 상황: TCP 3-WHS 이후 데이터 전송
    - L4 계층에서 L7 계층으로 연결이 만들어진 것을 알려 데이터 전송을 시작한다.
    - L7(Application Layer)에서 L4(Transport, TCP)로 데이터를 보낸다.
        - 데이터는 TCP 3-WHS시 결정된 MSS를 기준으로 Fragmentation을 진행한다.

Slide 20/22

- 상황: 데이터 전송
    - Application Layer에서 받은 데이터를 캡슐화를 거쳐 전송한다.
    - 192.168.3.1 → 192.168.3.2
    - Sequence Number = 3
    - 수신지 Port, IP, MAC 주소를 다 알고 있어 Unicast 방식으로 전송할 수 있다.

Slide 21/22

- 상황: 데이터 전송
    - 192.168.3.2에서 받은 패킷을 역캡슐화를 거쳐 상위 계층(Application layer)으로 전송한다.
    - SN은 아직 바뀌지 않는다.

Slide 22/22

- 상황: 데이터 전송
    - 데이터 수신 후 정상적으로 받았다는 ACK를 보낸다.
    - 이 때, ACK는 4가 된다. (SN + data size)
    - 캡슐화 과정을 거쳐 192.168.3.1로 패킷을 전송한다.

- **캡슐화**
    - 송신자측
    - 상위 계층에서 하위 계층으로
    - 제어 정보 추가

- **역캡슐화**
    - 수신자측
    - 하위 계층에서 상위 계층으로
    - 제어 정보 제거

2. **UDP 기반의 데이터 전송과정(Slide 1~17)**

Slide 1/17

- 상황: 애플리케이션에서 데이터를 보내는 과정
    - Application 계층에서 Transport 계층으로 데이터를 전송함
    - 192.168.3.1에서 192.168.4.2로 라우터를 경유하여 외부망으로 전송
    - Transport 계층에서 UDP 사용을 결정
    - Application 계층에서 데이터 전송

Slide 2/17

- 상황: 데이터를 전송하기 위해 UDP헤더와 IP 헤더를 추가하는 과정
    - Transport 계층에서 송신지 포트 번호와 수신지 포트 번호가 담긴 헤더를 추가
    - Network 계층에서 송신지 IP 주소와 수신지 IP 주소가 담긴 헤더를 추가

Slide 3,4/17

- 상황: 데이터를 전송하기 위해 헤더를 추가하는 과정
    - 게이트웨이의 주소가 ARP Cache Table에 있는지 확인
    - 외부 망이므로 게이트웨이 MAC주소로 보내야 함
    - 없다면 ARP reply가 올 때까지 패킷 임시 저장

Slide 5/17

- 상황: 게이트웨이 MAC 주소를 등록하는 과정
    - 3계층에서 IP를 보고 내부망과 외부망을 구분
    - 외부망으로 패킷을 보내야 하기 때문에 게이트웨이인 192.168.3.2주소를 확인
    - ARP Cache Table에 없기 때문에 ARP request를 보내야 함

Slide 6/17

- 상황: 게이트웨이 MAC 주소를 등록하는 과정
    - ARP request를 보내 게이트웨이의 MAC 주소를 물어본다.
    - 캡슐화 과정을 거쳐 만들어진 ARP request를 브로드 캐스트로 전송

Slide 7/17

- 상황: 게이트웨이 MAC 주소를 등록하는 과정
    - 라우터(게이트웨이)가 ARP request를 받는다.
    - 패킷의 수신지 MAC, 수신지 IP를 확인하여 자신의 패킷임을 확인한 후 역캡슐화를 통해 상위 계층으로 올린다.
    - 자신의 ARP Cache Table에 받은 주소를 등록한다.

Slide 8/17

- 상황: 게이트웨이 MAC 주소를 등록하는 과정
    - ARP reply 패킷에 자신의 IP와 MAC 주소를 담아 캡슐화를 거쳐 전송한다.
    - reply 패킷은 수신지 MAC 주소를 알기 때문에 Unicast 방식으로 전송한다.

Slide 9/17

- 상황: 게이트웨이 MAC 주소를 등록하는 과정
    - 192.168.3.1에서 ARP reply 패킷을 받는다.
    - 수신지 MAC, 수신지 IP를 순서대로 확인하여 자신에게 온 패킷인지 확인한다.
    - ARP Cache Table에 전달받은 게이트웨이 주소를 등록한다.
    - 전송하려던 데이터의 수신지MAC 필드에 게이트웨이 MAC 주소를 설정한다.

Slide 10/17

- 상황: Media Translation 1
    - 외부망에 존재하는 목적지로 전송하기 위해 내부망의 게이트웨이로 전송한다.
    - 이더넷 헤더의 DST MAC이 게이트웨이의 MAC

Slide 11/17

- 상황: Media Translation 후 패킷 전송
    - 게이트웨이(라우터)에서 전송 받은 패킷의 MAC 주소를 보고 L3으로 전달한다.
    - 자신의 IP 주소가 아닌 것을 확인 후, 라우팅 테이블을 조회하여 경로 제어를 한다.
    - Forwarding 방식으로 전송
    - 목적지 MAC 주소가 0800:0222:1111로 설정

Slide 12/17

- 상황: Media Translation 후 패킷 전송
    - 라우터가 라우팅 테이블을 조회하여 192.168.4.0망은 Gi 0/1로 나가면 되는 것을 확인
    - 목적지 IP 주소가 192.168.4.2로 설정
    - 수정한 패킷을 L2로 전달

Slide 13/17

- 상황: Media Translation을 위한 수신지 MAC 주소 얻기
    - 수신지 MAC 주소를 얻기 위해 ARP request 전송
    - 캡슐화된 패킷이 브로드 캐스트로 192.168.4.0망에 전송된다.

Slide 14/17

- 상황: Media Translation을 위한 수신지 MAC 주소 얻기
    - 192.168.4.2가 전송 받은 패킷을 상위 계층으로 전달(역캡슐화)
    - 수신지 MAC(브로드 캐스트이기에 수신), 수신지 IP 확인

Slide 15/17

- 상황: Media Translation을 위한 수신지 MAC 주소 얻기
    - 자신의 IP로 온 것을 확인 후 ARP reply 패킷 송신
    - 캡슐화 과정을 거쳐 게이트웨이(라우터)에게 전송

Slide 16/17

- 상황: Media Translation을 위한 수신지 MAC 주소 얻기
    - L2, L3에서 수신지 주소를 확인하여 자신에게 온 패킷인지 확인한다.
    - 전달받은 주소를 자신의 ARP Cache Table에 등록한다.

Slide 17/17

- 상황: Media Translation2
    - 수신지(192.168.4.2)의 MAC 주소도 알게 되어 패킷의 이더넷 헤더의 수신지 MAC주소 필드를 설정하여 192.168.3.1 → 192.168.4.2로의 패킷 전송이 완료된다.

3. 매체 접근 제어 기술(CSMA, CSMA/CD, CSMA/CA)들의 기술적 차이를 정리한다.

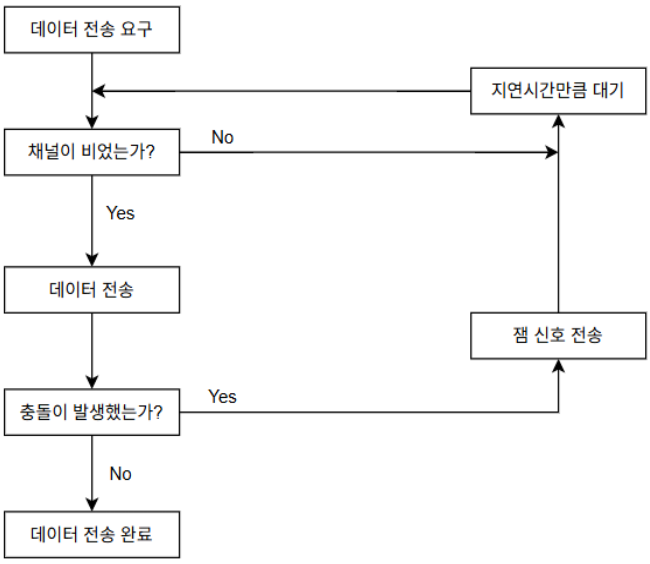
3.1 CSMA

- 기본 개념
    - 네트워크 매체를 여러 노드가 공유하는 환경에서 채널이 비어 있는지 감지하고 전송하는 것을 말한다.
    - 여러 노드가 하나의 전송 매체를 공유하므로 서로의 전송 간에 충돌이 발생할 수 있다.
    - Carrier Sense: 패킷을 전송하기 전에 노드는 네트워크가 사용중인지 확인한다. 다른 장치가 이미 데이터를 보내고 있는지를 확인하는 과정이다.
    - 충돌: 여러 Station에서 데이터를 동시에 보낸 경우 데이터가 충돌하여 손상된다. 네트워크가 혼잡하면 성능이 급격히 저하되는 단점이 존재.
    - 데이터 송신권을 경쟁을 통해 획득하는 방식으로, 충돌 감지나 회피가 불가능하다.
    - 이론적인 기본 모델. 실제로는 사용하지 않는다.

상황	설명
동시에 두 호스트가 전송	매체가 비어 있다고 오인하고 동시에 전송 시작
충돌 지점	패킷들이 전송 매체 상에서 겹치며 손상된다.

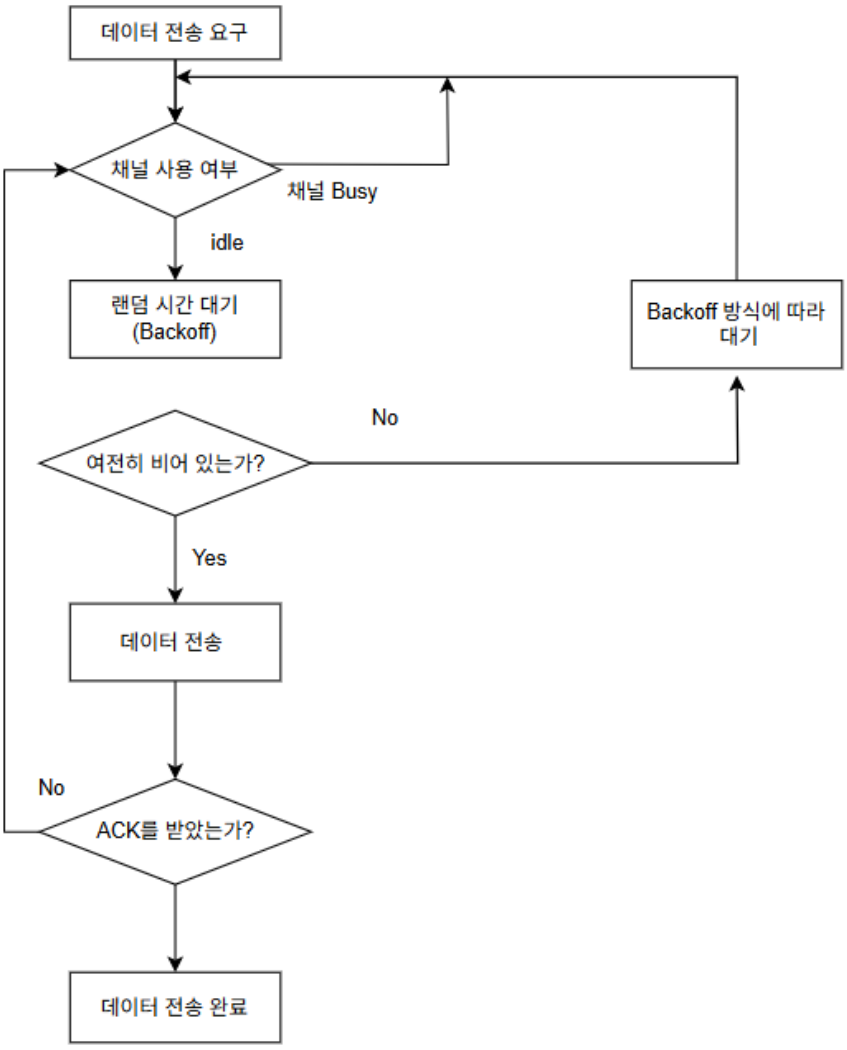
3.2 CSMA/CD

- **충돌 해결**
    - CSMA를 개선한 방식으로, 충돌을 조기에 검출하여 통신로를 빠르게 해방시키는 제어가 추가된 방식이다.
- 동작 방식 및 특징
    1. Carrier Sense를 통해 망이 데이터를 전송 중인지 확인한다.
    2. 데이터가 흐르고 있지 않으면 데이터를 송신한다.
    3. 충돌이 발생했는지를 검출하고, 충돌이 발생한 경우에는 송신을 즉각 중지하여 Media를 점유하지 않는다.
    4. 잼 신호를 전송하여 네트워크 전체에 충돌이 있었음을 알린다.
    5. 난수 시간만큼 기다린 후 송신을 재개한다. (충돌을 일으킨 쌍방이 즉시 재전송하려고 하면 또다시 충돌이 발생하기에 이를 방지한다.)
    6. 반 이중 전송 방식을 사용한다.
        - 반 이중 통신: 동시에 통신 불가. 한 번에 한 쪽씩 통신할 수 있는 방식으로, 상대적으로 느리다.
    7. 유선 이더넷에서 주로 사용한다
    8. 충돌 자체를 허용하기 때문에 혼잡 가능성이 존재한다.



3.3 CSMA/CA

- 기본 원리
    - 무선 LAN에서 사용되는 방식으로, 충돌을 회피하는 특징이 있다.
    - 무선 LAN은 여러 단말기가 같은 주파수대를 공유하는 매체 공유형 네트워크이다.
    - 무선 네트워크에서는 충돌 감지가 **불가능**에 가깝기 때문에 충돌을 감지하지 않고 회피하는 것이 필수이다.
    - 전 이중/반 이중 모두 가능하다.
        - 전 이중 통신: 반 이중 통신과는 다르게 동시에 송수신이 가능한 통신 방식이다.
    - 무선 환경에 최적화되어 안정성을 높이지만, 지연 가능성도 높아진다.
- 동작 과정
    - Carrier Sense로 통신 매체가 사용 중인지 확인한다.
    - 사용 가능한 상태(idle state)를 확인한 후 랜덤 시간(backoff)만큼 대기한 뒤에 데이터 전송을 시작한다. 이 대기 시간을 통해 충돌을 회피할 수 있다.
    - ACK(확인 응답) 수신 대기를 한다. 수신 측이 데이터를 잘 받았는지 대기하는데, 만약 수신에 실패한다면 데이터가 충돌했거나 손상됐다고 판단하여 Backoff 후 재전송을 시도한다.



3.4 CSMA/CD와 CSMA/CA 비교표

항목	CSMA/CD	CSMA/CA
주 사용 환경	유선 네트워크(예: 이더넷)	무선 네트워크(예: Wi-fi)
충돌 처리 방식	충돌 발생 후 감지 -> Jam 신호 전송	충돌 발생을 피하기 위해 사전 대기 전략 사용
충돌 감지 가능 여부	가능(충돌 감지 방식)	불가능(충돌 회피 방식)
충돌 후 동작	전송 중단 -> Backoff -> 재시도	ACK 미수신 시 -> Backoff -> 재시도
데이터 전송 시점	채널 감지 후 즉시 전송	감지 후에도 랜덤 대기 후 전송
통신 방식	반 이중 통신	전 이중/반 이중 통신