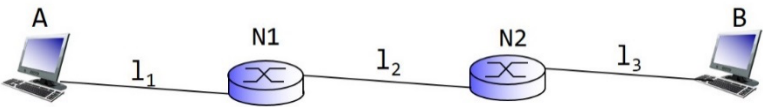


컴퓨터네트워크 중간고사 대체 활동

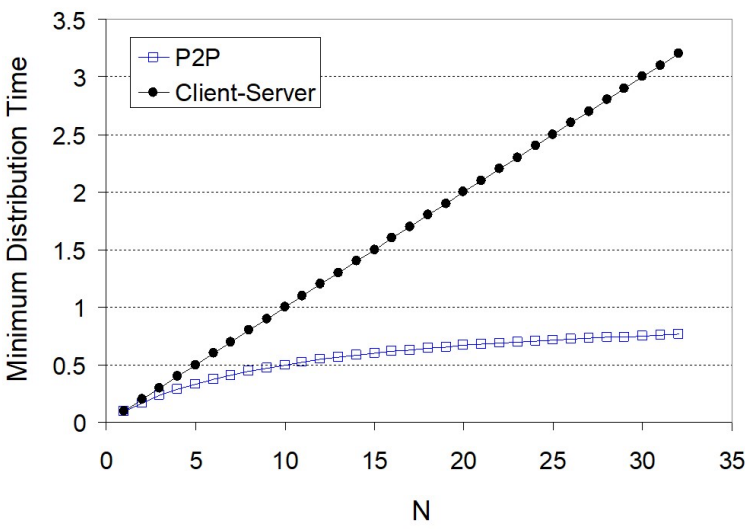
학과: 미디어학과

학번: 201821048

이름: 이서영

문제 1 (3월 20일)	packet switching으로 인해서 얻는 이익과 생기는 부담에 대해서 본인이 이해하는 범위 안에서 설명하시오.
기존 제출 답안	데이터를 패킷으로 잘라 라우터에 전달하면서 회선을 독점하지 않아 자원을 효율적으로 사용할 수 있다 부담 : 패킷이 가는 도중에 사라지거나 지연될 수 있다.
교수 feedback	예, 틀린 얘기는 아닌데, 조금 더 문장에 내용을 채워서 기술하면 좋겠군요.
수정 답안	이익 : 1. 간헐적으로 데이터를 보낸다고 가정할 때 같은 양의 자원을 가지고 circuit switching 보다 많은 사용자를 감당할 수 있다. 2. 그물망 구조로 되어있기 때문에 중간에 하나의 라우터가 고장나는 경우가 생겨도 다른 라우터를 통해 패킷을 보낼 수 있다. 3. 패킷단위로 자르기 때문에 중간에 라우터가 쉬는 경우를 최소화해서 네트워크 자원을 효율적으로 사용할 수 있다. 부담 : 1. 라우터의 저장 공간을 넘어서 패킷이 도착할 경우 처리하지 못한 패킷이 버려지거나 손상될 수 있다. 2. circuit switching 처럼 연결된 상태를 유지하지 않아서 실시간 비디오 전송 같은 경우에 신뢰성을 비교적 보장하지 않는다. 3. 각 패킷이 어떻게 움직여야 하는지 데이터를 추가해야하기 때문에 내용보다 형식에 자원을 더 소모할 수 있다.
문제 2 (3월 25일)	 <p>위 그림과 같이 A와 B가 두 개의 라우터를 걸쳐 연결이 되어 있다. L bytes 크기의 packet을 A에서 전송하기 시작해서 B에 도착하기까지의 지연 시간을 계산하는 데에, 각 전송 단계에서 nodal delay 네 가지 요소들 중 어떤 요소들이 고려되어야 하는지 적으시오. 즉, A→N1, N1→N2, N2→B 각 단계에 대해서 기술하시오.</p>
기존 제출 답안	기존제출 없음
교수 feedback	

수정 답안	<p>A→N1 : Prop A에서 N1까지 물리적 거리를 계산해야 한다. prop값은 $d(\text{length})/s(\text{speed})$로 $l(1)/s$이다. N1→N2 : proc, queue, trans, prop N1 라우터에서 N2라우터로 갈 때 proc에서는 패킷의 손상확인, 나가는 곳을 판단한다. proc값은 보통 msec보다 작다. queue에서는 전에 도착한 패킷들이 나가는 시간을 기다린다. Queue 값은 라우터의 혼잡 정도에 따라 달라서 예측하기 힘들다. Trans는 빠져나가는 시간이다. Prop와 비교했을 때 선을 타고 이동하는 시간이 아니라 이동하기위해서 라우터에서 움직이는 시간이다. Trans 값은 $L(\text{packet length})/R(\text{link bandwidth})$이다. N2→B : proc, queue, trans, prop 위에서 설명한 것과 동일하게 라우터에서 다음 목적지까지 이동하기 위해 nodal delay의 네가지 요소가 전부 고려되어야 한다.</p>
문제 3 (3월 25일)	Protocol들이 계층적 관계를 갖고 있어서 초래되는 한계는 무엇인가?
기존 제출 답안	기존제출 없음
교수 feedback	
수정 답안	<p>각 레이어에서 어떤 내용을 담고있는지 다른 레이어가 알지 못해서 비효율이 발생한다. 예를 들어 통역자가 통역하는 사람의 말이 무엇인지 생각하지 않고 그저 상대방이 알아 들을 수 있는 언어로 바꾸기만 한다. 만약 하위 레이어에서 상위 레이어의 상태를 인지 하고 데이터의 변형이 가능하다면 자원이 한정적일 때는 전달되는 내용을 바꾸거나 할 수 있다. 무선 네트워크 시 더욱 이 비효율성이 크게 느껴진다.</p> <p>데이터 전송 할 때 상위 레이어는 하위 레이어로 보낼 때 각 프로토콜에 맞게 데이터를 붙인다. Application에서는 상대 end application에서 볼 message를 보낸다. transport에서는 message에 헤더를 붙여 segment 형태로 만든다. network에서도 그 계층에 맞는 헤더를 붙여 datagram으로 만든다. link에서도 헤더를 붙여 frame 으로 만들고 물리적 선을 통해 보낸다. 이 반대의 순서로 받을 때는 하나하나 풀어보듯 각 계층에서 헤더를 제거한다.</p> <p>따라서 각 계층을 필수적으로 거쳐서 상위, 하위 계층을 갈 수 있기 때문에 효율적으로 사용하는데 한계가 발생한다.</p>
문제 4 (4월 1일)	Web cache가 유용한 경우가 언제인지 서술하시오.
기존 제출 답안	기존제출 없음
교수 feedback	

수정 답안	<p>40퍼센트는 local cache에서 60퍼센트는 원래 서버에서 사용자의 요구를 충족할 때 유용하게 사용할 수 있다. Cache는 Origin server 대신 proxy server에서 동일한 콘텐츠를 제공해줘서 사용자가 빠르게 서비스를 이용할 수 있도록 하는 기술이다. 사용자가 서버로 요청할 때는 괜찮지만 서버로부터 콘텐츠가 올 때 라우터에 queuing delay가 엄청나게 늘어난다. 왜냐하면 많은 사용자들이 요청한 크기가 큰 콘텐츠를 제공하기 위해서 비용의 한계가 있어서 한정된 라우터를 사용해야 하기 때문이다. 따라서 local web cache를 두어 라우터 용량을 늘리지 않고도 많은 사용자의 요구를 처리할 수 있도록 한다.</p> <p>여기서 사용자가 요청할 때 local cache에서는 origin에게 매번 페이지가 바뀌었는지 물어본다. 이 때 바뀌지 않았으면 304, 바뀌었으면 바뀐 데이터를 바로 보내준다. 사용자가 요청했을 때 가장 자원을 효율적으로 사용할 수 있는 적절한 비율이 local 4, origin 6이다.</p>																											
문제 5 (4월 8일)	<p>아래의 N명의 사용자들이 file을 획득하는 데에 걸리는 시간을 그래프로 나타낸 것이다. N이 증가함에 따라 client-server model은 시간이 일정한 비율로 지속적으로 증가하는데, p2p model은 client-server model에 비해 완만하게 증가한다. 그 이유가 무엇인지 강의 노트와 강의 동영상에서 사용하고 있는 표현들을 활용해서 설명하시오.</p> <div><table><caption>Estimated data points from the graph</caption><thead><tr><th>N</th><th>Client-Server (Time)</th><th>P2P (Time)</th></tr></thead><tbody><tr><td>0</td><td>0.0</td><td>0.0</td></tr><tr><td>5</td><td>0.5</td><td>0.2</td></tr><tr><td>10</td><td>1.0</td><td>0.3</td></tr><tr><td>15</td><td>1.5</td><td>0.4</td></tr><tr><td>20</td><td>2.0</td><td>0.5</td></tr><tr><td>25</td><td>2.5</td><td>0.6</td></tr><tr><td>30</td><td>3.0</td><td>0.7</td></tr><tr><td>32</td><td>3.2</td><td>0.8</td></tr></tbody></table></div>	N	Client-Server (Time)	P2P (Time)	0	0.0	0.0	5	0.5	0.2	10	1.0	0.3	15	1.5	0.4	20	2.0	0.5	25	2.5	0.6	30	3.0	0.7	32	3.2	0.8
N	Client-Server (Time)	P2P (Time)																										
0	0.0	0.0																										
5	0.5	0.2																										
10	1.0	0.3																										
15	1.5	0.4																										
20	2.0	0.5																										
25	2.5	0.6																										
30	3.0	0.7																										
32	3.2	0.8																										
기존 제출 답안	기존제출 없음																											
교수 feedback																												
수정 답안	<p>Client-Server 모델 : $D(cs)$는 $N \cdot F/u(s)$와 $F/d(\min)$에 영향을 받아 최소 시간을 구할 수 있다. $N \cdot F/u(s)$: n개의 copy 파일을 보내는 시간 $F/d(\min)$: min client가 다운받는 시간 P2p모델 : $D(p2p)$는 $F/u(s)$와 $F/d(\min)$, $N \cdot F/\{u(s) + 1 \text{부터 } n \text{까지 } u(i) \text{의 합}\}$에 영향을 받아 최소 시간을 구할 수 있다. $F/u(s)$: 하나의 파일을 보내는 시간 $F/d(\min)$: min client가 다운받는 시간 $u(s) + 1 \text{부터 } n \text{까지 } u(i) \text{의 합}$: 모든 peer가 전송에 참여한다고 가정할 때 업로드 속도</p> <p>여기서 증가 추세를 결정하는 부분은 $N \cdot F$ 밀의 분모이다. Client-Server 모델에서는 이</p>																											

	<p>용자 n이 많아져도 서버가 모든 것을 처리해야하기 때문에 처리해야할 n에 따라 linear 하게 증가한다. 하지만 P2p모델에서는 서버가 한 copy를 보낸 후 peer가 다른 peer에게 파일을 주기 때문에 server 혼자서 파일을 보내지 않는다. 따라서 처리해야할 n에 따라 log scale로 커지는 모습을 볼 수 있다.</p>
<p>문제 6 (4월 10일)</p>	<p>본인이 이해하는 범위 안에서 CDN과 web cache의 유사점 혹은 차이점을 기술하시오.</p>
<p>기존 제출 답안</p>	<p>유사점 : 클라이언트가 자신에 대한 데이터를 보내 효율적이고 편리하게 서비스를 이용할 수 있도록 한다. 차이점 : CDN은 가까운 서버가 서비스 할 수 있도록 해주는 것이고 cache는 클라이언트의 상태를 기록해 두는 것이다.</p>
<p>교수 feedback</p>	<p>음... 이번에도 CDN이나 web cache나 제대로 학습이 되어 있지 않습니다.</p>
<p>수정 답안</p>	<p>유사점 : 사용자가 빠르게 서비스를 이용할 수 있도록 한다. 클라이언트가 원래 서버에 서비스를 요청하지만 그 서버가 바로 데이터를 보내주지 않고 다른 서버 ex)proxy(cache), CDN(CDN)서버를 통해 데이터를 처리한다. 차이점 : CDN은 대용량 콘텐츠를 빠르게 전송하기 위해 만들어졌고 cache는 동일한 콘텐츠를 반복적으로 전송하지 않고 local server에 저장할 수 있도록 만들었다. CDN은 content distribution network의 약자이다. 복제된 내용이 분산되어 실제 서버처럼 운영하는 것이다. 비디오 콘텐츠를 제공하는 유튜브나 넷플릭스 같은 기업이 대용량 데이터를 전세계의 이용자에게 서비스를 제공할 때 서버가 위치한 곳이 멀리 떨어져 있다. 그래서 여러 장소에 서버를 두고 분산된 각 서버에서는 가까운 클라이언트의 요청을 복사된 비디오를 보내 처리한다. Cache는 이와 다르게 이전에 받았던 콘텐츠를 기록해 두었다가 사용자에게 반복적으로 제공해 주는 점에서 차이가 있다.</p>
<p>문제 7 (4월 17일)</p>	<p>Transport layer protocol implementation과 application process 사이에 multiplexing과 demultiplexing이 발생합니다. Multiplexing과 demultiplexing이 각각 source(sender)와 destination(receiver) 중 어디에서 일어나며 왜 일어나는지 설명하세요.</p>
<p>기존 제출 답안</p>	<p>multiplexing at sender : 보내는 쪽에서 헤더를 붙인다. demultiplexing at receiver : 받는 쪽에서 받은 것을 해석해 사용한다.</p>
<p>교수 feedback</p>	<p>음... 맞다고 해야 할까요...? 어떻게 실현되는가를 설명하고 있는데, 왜 일어나야 하는지를 이해하는 것이 더 중요합니다.</p>

수정 답안	<p>Multiplexing : sender가 데이터를 보낼 때 모으기 위해서이다. 여러 어플리케이션이 보내는 데이터가 각 소켓을 통해 오는데 이것을 처리하는 것이다. 왜냐하면 하나의 transport 계층에서 처리해야 하기 때문이다.</p> <p>Demultiplexing : receiver가 데이터를 받을 때 분배를 하기 위해서이다. Multiplexing과 반대로 헤더 정보를 이용해 받은 세그먼트를 올바른 각각의 소켓 application에 전달한다. 왜냐하면 segment를 message로 바꿔줘야 application 에서 이해할 수 있기 때문이다. 각 헤더에는 어디로 가야하는지 socket port 정보가 있는 destination port가 적혀있다. 또한 어디에서 왔는지 source port가 적혀있다. 이에 맞게 multiplexing과 demultiplexing을 한다.</p>
문제 8 (4월 22일)	Host A와 host B 사이의 RTT가 200 ms이고 A와 B 사이의 path의 bandwidth가 10 Mbps이다. Packet의 크기는 1000 bytes이다. Pipelined protocol (sliding window 기법)을 사용하는데, path의 bandwidth를 100% 활용하기 위해서는 sender의 window 크기는 얼마이어야 할까? 중간 계산 과정을 글로 적어 주세요.
기존 제출 답안	윈도우의 크기는 $1 + (RTT / (L / R))$ 을 구하면 최대 bandwidth를 활용할 수 있다. RTT는 0.2(sec), L은 8000, R은 $10 * 10^6$ 이다. L/R 은 0.0008이다. $(RTT / (L / R))$ 은 250이다. 따라서 자원을 최대 사용할 때 윈도우의 크기는 251이다.
교수 feedback	예, 잘 했어요.
수정 답안	변경 없음
문제 9 (4월 22일)	<p>Sender가 순서번호 1,2,3,4,5인 packet들을 전송했다.</p> <p>Sender가 ACK 5를 수신했다고 가정하자. Go-Back-N 방식일 때 sender가 어떻게 조치하는지, selective repeat 방식일 때 어떻게 조치하는지 적으시오.</p>
기존 제출 답안	<p>1. Go-Back-N : 2, 3, 4, 5, 6 을 sender 가 전송한다. selective repeat : 2 를 sender 가 재전송해서 ack 이 올 때까지 기다린다.</p> <p>2. Go-Back-N : 5, 6, 7, 8, 9 sender 가 전송한다. ack 이 올 때까지 기다리고 전송한 것 중 ack 이 안 온 것부터 윈도우 크기만큼 다시 전송한다. selective repeat : 5, 6, 7, 8, 9 sender 가 전송한다. ack 이 올 때까지 기다리고 전송한 것 중 ack 이 안 온 것만 다시 전송한다.</p>
교수 feedback	<p>음... 답을 꼼꼼하게 읽어보고 제출했을까요?</p> <p>문제에서 5 번까지 전송했다고 했습니다. 1)에서 6 번 packet 을 전송할 이유가 없지요. 그리고, 2)에서 ACK 5 를 받았는데 5 번 packet 을 보내야 하는 이유가 무엇일까요?</p> <p>그리고, selective repeat 에 대해서는 2)에 작성한 답이 오류입니다. 강의 노트나 강의 동영상상을 보고 다시 공부하기 바랍니다.</p>

수정 답안	<p>Go-Back-N : sender는 receiver가 패킷을 전부 다 받았다고 생각한다.</p> <p>selective repeat : timer가 지나면 sender는 receiver가 ack이 오지 않은 패킷을 못 받았다고 생각하고 1,2,3,4를 재전송한다.</p> <p>Go-Back-N에서는 receiver가 패킷이 제대로 전달받은 가장 마지막 번호로 ack을 보낸다.</p> <p>여러 개의 ack신호대신 최종적인 하나의 ack을 보내는 것이다. 하지만 selective repeat에서는 각 패킷이 ack을 전송해서 ack이 없다면 각 패킷이 도착하지 않았다고 판단한다. 그래서 선택적으로 그 패킷들만 재전송을 해준다.</p>
문제 10 (4월 24일)	congestion이 어떤 상황을 의미하는지 기술하고, congestion을 방지하기 위한 해결 방안은 무엇인지 간략하게 제시하시오.
기존 제출 답안	<p>혼잡이 발생해서 전송한 내용을 정상적으로 받지 못하는 경우, 패킷을 잃어버리거나 지연시간이 발생한다.</p> <p>end system 과 네트워크 상에서 어떤 상태인지 알려주는 것이다.</p>
교수 feedback	2)의 답만 적었군요. 1)의 congestion이 어떤 상황인지 기술하지 않았어요.
수정 답안	<p>Congestion : 많은 이용자가 네트워크가 처리할 수 있는 능력을 초과해서 많은 양의 패킷을 보내는 경우이다. 라우터가 초당 처리할 수 있는 데이터 양은 정해져 있다. 하지만 sender는 라우터와 소통을 하지 못해 처리가 불가능할 정도로 많은 양을 보낼 수 있다. 이럴 경우 패킷이 버려지거나 한참을 기다려야 하는 상황이 발생한다.(lost packets and long delay) 혼잡이 발생할 경우 패킷이 버려져서 다시 패킷을 보내야 하고 사용자가 한참을 기다려야 하기 때문에 큰 문제가 된다. 서로 다른 레이어 간에 정보를 주고받지 않기 때문에 효율적으로 문제를 해결하기 힘들다.</p> <p>Congestion을 피하기 위해서 slow start, Congestion avoidance, window 사이즈 감소를 이용한다. slow start 구간은 1부터 시작해서 윈도우 사이즈를 2배씩 늘리는 구간이다. Ssthresh 까지 늘려서 기하급수적으로 빠르게 늘린다. Congestion avoidance에서는 어느 정도 크기가 되었으면 $cwnd = cwnd + MSS \cdot MSS / cwnd$로 linear하게 증가한다. 왜냐하면 계속 지수적으로 늘리면 혼잡이 빠르게 발생할 수 있기 때문이다. window 사이즈 감소는 첫 번째 경우는 타임아웃이 발생하면 cwnd가 1이 되고 ssthresh가 1/2로 줄어든다. 그리고 slow start를 한다. 두 번째 경우는 동일한 번호의 ack이 4개가 갈 경우 cwnd를 1/2하고 Congestion avoidance로 작동한다.</p>