Computer Networks 대학원 기출 예상 문제

< Chapter 1: Computer Networks and the Internet >

Q1) Circuit switching과 Packet switching에 대해서 설명해보세요.

Circuit switching과 Packet switching 모두 네트워크 링크와 스위치를를 통해 data packet을 전송하는 방식을 말합니다.

먼저 Circuit switching에 대해서 설명하겠습니다. Circuit switching에서는 메세지 전송하기 위해 virtual circuit이라는 리소스를 선점합니다. circuit을 선점할 때 일정 bandwidth를 함께 선점할 수 있기 때문에, 일정한 transmission rate 를 유지할 수 있습니다. Circuit Switching에서는 한번에 여러개의 서킷에서 메세지 전송하는 multiplexing을 위해 Frequency-Division Multiplexing과 Time-Division Multiplexing 방식을 지원합니다. FDM 방식에서는 주파수를 1/n로 나누어서 메세지 전송하고, TDM 방식은 시간을 일정한 프레임과 슬롯으로 나누어서 전송합니다.

여기서 Circuit Switching의 단점이 드러나게 되는데요, Circuit Switching의 경우 일정 타임 프레임 혹은 주파수를 선점하여서 사용하기 때문에, 메세지의 전송이 되지 않는 기간 동안은 선점된 자원을 낭비하는 형태가 되어 버립니다. 이를 Silence Period라고 합니다.

Circuit Switching에 반해, Packet Switching은 자원을 선점하는 프로토콜이 없습니다. Packet Switching에서는 메세지를 작은 단위인 data packet으로 쪼개어 전송하게 됩니다. 그리고 여러개의 패킷 스위치(라우터)와 커뮤니케이션 링크를 통해 source와 destination를 연결하고, 그를 통해 메세지를 전송합니다.

Packet Switching의 경우 수요가 있을 때만 링크를 이용하기 때문에, Silence Period가 있는 Circuit Switching에 비해 bandwidth를 효과적으로 공유할 수 있습니다.리소스를 선점하여 사용하는 것이 아니기 때문에, 데이터 패킷을 전송하는 데에 있어서 delay가 생기고, 또 패킷이 누락되는 단점을 가지고 있기도 합니다.

Q2) Packet Switching에서의 Delay에 대해서 설명해보세요.

Packet Switch는 커뮤니케이션 링크와 패킷 스위치를 통해 데이터 페킷을 전송하기 때문에, 그로부터 발생하는 Delay를 가지고 있습니다. 여기서 Delay는 크게 네가지로 나뉘게 됩니다: Nodal Processing, Queuing, Transmission, Propagation Delay

Nodal Processing Delay는 패킷 스위치에서 패킷을 분석하는데 소요되는 딜레이를 말합니다. 라우팅을 함에 있어서 헤더를 분석하고 링크를 걸어주는 작업 등이 이에 속합니다. Queuing Delay는 패킷이 전송되기 전까지 큐에서 기다리면서 소요하는 시간을 의미합니다. Transmission Delay는 패킷 전체를 링크로 밀어넣는데 걸리는 시간을 의미합니다. 마지막으로 Propagation Delay는 커뮤니케이션 링크를 통해 패킷이 전송되는 시간을 의미합니다.

Q3) OSI 7-Layer에 대해서 설명해보세요.

OSI Model에서는 네트워크 시스템의 프로토콜을, 모듈화를 통한 관리의 편의성을 위해, 잘 정의된 일곱가지 레이어로 나누었습니다: Application, Presentation, Session, Transport, Network, Link, Physical Layer

Application Layer는 네트워크 애플리케이션과 그에 해당하는 프로토콜이 존재하는 공간입니다. 이는 HTTP, FTP, DNS, SMTP 등의 애플리케이션을 포함하는데요, application layer protocol은 이러한 애플리케이션들이 메세지를 주고받는 방식을 정의해놓은 프로토콜입니다.

Presentation Layer는 data compression, data encryption, 그리고 data description 등의, 교환되는 데이터의 의미 분석에 대한 communication service를 제공합니다.

Session Layer에서는 각 데이터들에 대한 Synchronization service를 제공합니다.

Transport Layer는 application-layer에서 전달된 메세지를 상대방 endpoint의 application에 전달해주는 역할을 합니다. Transport Layer protocol에는 TCP와 UDP가 있고, TCP를 통해서는 data segment를 reliable하게 전송하고, UDP를 통해서는 비록 unreliable하지만 오버헤드와 딜레이가 적은 전송을 하게 됩니다.

Network Layer는 transport layer segment를 datagram으로 캡슐화 해서 상대편 host로 전송해주는 역할을 합니다. IP Protocol, ICMP, 그리고 Routing protocol로 이루어져 있으면서 network에서 datagram을 구체적으로 어떻게 전송할지에 대한 프로토콜을 정의하고 있습니다.

Link Layer는 노드와 노드 사이의 링크와 관련된 계층으로, 각 링크에서 어떤 포워딩 방식을 사용하는지에 대해서 말을 하고, 링크마다 다른 프로토콜을 채택할 수 있습니다.

Physical Layer는 각 링크들의 Physical medium과 관련된 서비스를 제공합니다.

< Chapter 2: Application Layer >

Q1) Application Layer Protocol의 예와, 각 프로토콜이 어떤 일을 하는지 말해보세요.

대표적인 Application Layer Protocol로는 HTTP, FTP, SMTP, DNS가 있습니다.

먼저 HTTP는 웹클라이언트가 웹서버에 어떤 식으로 웹 페이지를 요청해야 하고, 또 웹서버가 어떤 식으로 웹클라이언트에 요청된 페이지를 전송해주어야 하는지를 정의하는 프로토콜입니다.

그 다음은 FTP인데요, 이는 클라이언트가 서버의 파일 시스템에서 데이터를 찾아 전송받는 방법을 명시해둔 프로토콜입니다.

SMTP는 각 메일서버간의 메시지 전달방식을 정의하는 프로토콜로서, 각 유저가 애플리케이션을 통해 전자메일을 주고받을 수 있도록 해주는 프로토콜입니다.

마지막으로 DNS에 대해 설명하겠습니다. 각 호스트의 identity를 정의하기 위해서는 host name과 IP Address 두가지의 방법론이 있는데, user-side에서는 hostname을 원하고 router에서는 ip address가 필요합니다. 그렇기 때문에 hostname을 ip address로 변환해주는 directory service가 필요한데, 이를 구현해둔 것이이 DNS입니다.

Q1-2) 그럼 각 프로토콜의 차이점을 설명해보세요.

먼저 HTTP와 FTP를 비교해보면, 가장 큰 차이점은 HTTP는 하나의 TCP connection을 사용하는 반면 FTP는 TCP control connection과 TCP data connection 두개의 TCP connection을 사용한다는 점입니다. 그리고 HTTP는 stateless 서비스로, 서버측에서 상대방과 연결에 대해 어떠한 정보도 저장하고 있지 않는 반면, FTP는 특정 user account에 대한 control connection 등의 state를 저장하고 있다는 점 등이 다릅니다.

다음으로 SMTP와 HTTP를 비교해보면, HTTP는 웹페이지를 불러오는 pull protocol인 반면, SMTP는 메시지를 전송하는 push protocol입니다. 두번째로 SMTP의 메시지는 7-bit ASCII 포멧이어야 한다는 제약조건이 있는 반면 HTTP는 그런 제약조건이 없다는 차이점이 있습니다. 하지만 이후 SMTP에서는 MIME 헤더가 추가되면서 Non-ASCII 포멧의 메시지도 전송할 수 있게 되었습니다.

Q2) 인터넷에서 Cooky가 사용되는 메커니즘을 설명해보세요.

유저의 Identity와 관련된 서비스를 제공하기 위해서는 HTTP의 Statelessness를 보완하기 위해 사용되는 것이 Cooky입니다. 이는 주로 상업적인 용도의 웹서비스에서 많이 사용되는데요, 그 메커니즘은 다음과 같습니다.

유저가 사이트에 접속을 하면 서버는 고유한 identification number를 만들고, 또 그를 통해 indexing되는 entity를 만들어 back-end database에 저장합니다. 그리고는 생성된 identification number를 담은 쿠키를 HTTP 헤더에 담아 유저에게 보내줍니다. 그 이후부터 유저가 해당 서버에 접속을 할 때마다 유저의 id number는 서버에 보내지게 되고, 해당 서버는 이를 이용하여 유저에게 다양한 맞춤식 서비스를 제공할 수 있습니다.

Q3) Web Caching의 이점에 대해 말해보세요.

Web Caching의 이점은 크게 두 가지로 나누어 볼 수 있습니다.

먼저, Web Caching을 이용하게 되면 전송거리를 단축시키면서 response time을 줄일 수 있습니다. 특히나 client-server의 bottleneck bandwidth보다 client-cache의 bottleneck bandwidth가 훨씬 넓을 경우 응답시간을 훨씬 더 단축시킬 수 있습니다.

두번째로, 클라이언트에서 보내지는 메세지가 서버에 까지 갈 필요가 없기 때문에, 네트워크 전체의 트래픽을 줄일 수 있는 효과가 있습니다.

Q4) 전자메일을 주고 받는 메커니즘에 대해 설명해보세요.

인터넷 메일 시스템은 크게 user agent, mail server, 그리고 SMTP로 구성되어 있습니다.

먼저 예전 모델을 먼저 살펴보자면, 한쪽의 user agent가 상대방의 user agent에게 전자메일을 보내게 되면, 그 메세지는 먼저 자신의 mail server에 있는 message queue에 저장됩니다. 그 이후 SMTP를 통해 상대방 user agent가 속해있는 mail server로 전송된 message는 mail server 안에 있는 message queue에 저장되었다가, 각 user agent의 mail box로 이동하게 됩니다. 그럼 각 user agent는 전송된 전자메일을 자신의 mail box에서 찾아볼 수 있는 것입니다.

하지만 오늘날은 mail access가 client-server 구조를 사용하고 있습니다. sending user agent가 SMTP를 통해 자신의 mail server를 거쳐 상대방의 mail server로 메시지를 보내면, 그 메시지는 상대방 mail server의 message queue를 통해 user agent의 mail box로 이동하게 됩니다. 여기서, push operation인 SMTP를 통해서는 user agent가 mail box에 있는 메시지를 읽어오지 못합니다. 그렇기 때문에 user agent는 Post Office Protocol (POP)나 Internet Mail Access Protocol (IMAP)을 사용하여 자신의 mail box에서 메시지를 읽어오게 됩니다.

Q5) DNS의 메커니즘에 대해 설명해보세요.

High level에서의 Overview는 다음과 같습니다. 먼저 Application이 특정 hostname과 함께 user-side의 DNS를 invoke하면, user-side의 DNS는 그것을 받아 UDP 기반으로 네트워크로 쿼리 메세지를 보내게 됩니다. 그럼 잠시후 user-side의 DNS는 그에 맞는 메세지를 받게 되고, 그를 자신을 호출한 application에 전달하게 됩니다.

좀 더 구체적으로 들어가자면, DNS server는 크게 root DNS server, top-level domain (TLD) DNS server, authoritative DNS server 이렇게 세개의 클래스로 이루어져 있습니다. 그리고 DNS hierarchy에 속해있지 않은 local DNS server라는 것이 그 사이에서 중심적인 역할을 하게 됩니다. 유저가 local DNS server에 목적지 hostname을 보내면, local DNS server는 root DNS server를 통해 top-level domain DNS server의 주소를 반환받게 됩니다. 그렇게 반환된 주소의 TLD DNS server에 목적지의 hostname을 요구하게 되면, com, edu, net, 그리고 각 나라의 top-level domain을 포함하고 있는 TLD DNS server는 authoritative DNS server의 주소를 반환하여 주게 되고, 그를 통해 유저는 각 기관의 공공에 공개된 서버에 접속함으로써 해당 기관의 서비스를 제공받을 수 있게 되는 것입니다.

여기서 DNS의 속도를 개선하고 DNS 결과값을 받는 데에 필요한 메세지의 횟수를 줄이기 위해서 local DNS server를 통한 DNS chaching을 사용하기도 합니다.

Q6) P2P Architecture의 self-scalability에 대해 설명해보세요.

File distribution에 있어서, client-server architecture의 경우 하나의 서버가 파일을 업로드하면, 나머지 클라이언트들이 인터넷의 bandwidth를 공유하면서 업로드 된 파일을 다운받게 됩니다. 혹은 하나의 서버가 각각의 클라이언트에게 일일이 파일을 전송해주어야 합니다. 결국 하나의 서버가 모든 클라이언트의 요구를 수용해야 하기 때문에, Distribution time = Dcs >= max(NF/u s, F/d min) 가 됩니다.

하지만 P2P architecture의 경우, 각각의 peer가 server의 file distribution을 도와줄 수가 있습니다. 다시 말해, 각각의 peer는 자신이 파일을 다운받으면, 자신의 업로드 capacity를 서버와 함께 그 파일을 업로드 하는 것에 보탤 수 있게 되는 것입니다. 결국, Distribution time = D p2p >= max{F/u s, F/d min, NF/ (u s + ui 들의 합)} 이 됩니다.

이러한 P2P 의 효과를 self-scaling이라고 합니다.

Q7) P2P Community에서 정보를 검색하는 방법에 대해 설명해보세요.

P2P community에서의 정보 검색 방법은 크게 세가지 방법이 있습니다.

먼저, Centralized Index가 있는데요, 이 구조에서는, 하나의 index server가 있어서 각 peer들이 어떤 정보를 갖고 있는지에 대한 index 정보를 담고 있습니다. 그래서 각 peer는 index server를 통해 자신이 원하는 정보를 담고 있는 peer를 찾게 되고, 그를 찾게되면 해당 peer와 client-server model을 구축하면서 파일을 전송받게 되는 것입니다.

그 다음은 Query Flooding인데요, 자신이 원하는 정보가 있으면, abstract한 local network인 overlay에 해당 쿼리를 broadcast하는 방식입니다. 그래서 해당 정보가 있는 peer는 query hit을 날려주게 되고, 그럼 두 peer간의 파일 전송이 이루어지는 방식입니다. 하지만 이 방법은 네트워크에 너무 많은 query message가 아무런 제어 없이 방사가 된다는 단점이 있습니다. 이를 보완하기 위해서 query count를 설정하여 일정 횟수 이상의 hop이 전송이 된 query message는 자동

으로 누락되게 하는 메커니즘을 도입할 수 있습니다.

마지막으로 Hierarchical Overlay가 있는데요, 이 네트워크는 ordinary peer, super peer, 그리고 super peer를 이어주는 edge 이렇게 세가지의 구조물로 이루어 져 있습니다. 각 super peer는 자신의 overlay에 속해있는 ordinary peer들의 index server 역할을 하게 됩니다. ordinary peer는 super peer를 통해 외부로 query message를 보내게 되고, 해당 정보가 있는 ordinary peer가 속해있는 overlay를 찾으면 그를 통해 파일을 전송받는 형식입니다.

< Chapter 3: Transport Layer >

Q1) Transport Layer와 Network Layer, Link Layer 모두 연결에 관련된 것인데, 각 계층이 어떻게 다른가요?

간단하게 말한다면, Transport Layer에서는 host에서 돌아가고 있는 process간의 logical communication을 제공하고, Network Layer에서는 host간의 logical communication을 제공합니다. 우체국의 예를 간단하게 들면, 우리는 우편물을 보낼 때 우체국에 가서 등기로 보낸다든지, 특송으로 보낸다든지 주문을 합니다. 그렇게 주문을 하고 우편물을 전달하면 목적지까지 메세지가 전달된다는 것을 알기 때문인데요, 이것이 Transport Layer에 속한다고 볼 수 있습니다. 그리고 나면 구체적으로 어떤 경로를 통해 갈지는 우체국에서 담당을 하게 되는데요, 이와 같은 과정을 거쳐서 실제로 배달원에게 우편물을 전달해주는 역할을 해주는 것이 Network Layer라고 볼 수 있습니다.

그리고 network Layer가 end-to-end 연결을 담당하고 있다면, link layer는 node-to-node 연결을 담당하고 있습니다. 뿐만 아니라 Network layer에서는 IP address를 사용하지만 link layer에서는 MAC address를 사용한다는 차이도 가지고 있습니다.

Q2) Multiplexing과 Demultiplexing에 대해 설명해보세요.

Application Layer와 Transport Layer 사이에는 문과 같은 역할을 하고 있는 Socket이라는 통로가 존재합니다. 그래서 Application Layer와 Transport Layer가 메세지를 주고 받을 때는 이런 Socket을 통해서 communication이 이루어지는데요, segment가 Transport Layer에 도착하게 되면 Transport Layer에서는 header field를 분석해서 그 segment를 해당 process의 socket에 전달하게 됩니다. 이러한 작업을 Demultiplexing이라고 합니다. 그와 반대로, 각 socket으로부터 data chunk를 받아 이를 캡슐화 하고, 그것들을 Network Layer로 전달하는 작업을 Multiplexing이라고 합니다. 여기서 각 소켓은 port number 라는 각각의 identification number를 부여받게 됩니다.

Q2-1) 그럼 TCP에서와 UDP에서의 각 작업이 차이점이 있나요?

있습니다. UDP Socket은 destination IP address와 destination port number로 식별이 되는 반면, TCP Socket의 경우 Source IP address와 Source port number를 포함한 네가지의 조합으로 식별이 됩니다. 그렇기 때문에, UDP Socket의 경우 source가 다르더라고 destination IP address와 destination port number만 같으면 같은 소켓으로 메세지가 전달이 되는 반면, TCP Socket의 경우 destination 정보가 일치한다 하더라도 source 정보가 일치하지 않으면 다른 소켓을 통해 메세지가 전달이 됩니다.

Q3) Transport Layer의 프로토콜 중 UDP에 대해서 설명해보세요.

Transport Layer는 segment를 정확한 대상 프로세스에 전달하기 위해서 Multiplexing/Demultiplexing 서비스를 제공해야 하는데요, 그 기능과 함께 간단한 error checking 서비스를 제공하는 프로토콜이 User Datagram Protocol (UDP) 입니다. TCP 와는 달리 Reliable한 data transfer를 제공하지는 않지만 그래도 DNS 등 다양한 어플리케이션에서 사용하고 있는데요, 그 이유는 다음과 같습니다.

먼저, TCP의 경우 Congestion control, Flow control 등의 제어가 많기 때문에 데이터를 보내는 시점 등에 있어서 제약이 있는 반면 UDP는 그런 제약이 하나도 없습니다. 그렇기 때문에 Message를 받자마자 바로 전송이 가능합니다. 뿐만 아니라 ,TCP와는 달리 UDP는 Connection establishment가 필요 없기 때문에, 그를 위한 Delay가 없다는 것 또한 하나의 장점입니다. 그로 인해 Connection state를 저장하기 위해 필요한 메모리 또한 필요하지 않다는 것도 하나의 이점이라고 볼 수 있습니다. 하지만 Reliable Data Transfer를 지원하지 않기 때문에 Data Loss rate가 높다는 단점 또한 안고 있습니다.

Q4) Reliable data transfer에서의 pipelining에 대해서 설명해보세요

Reliable data transfer는 데이터 패킷을 전송하고 나면 ACK 신호를 기다렸다가 ACK 신호를 받고 나면 다음 패킷 전송하는, stop-and-wait protocol입니다. 그렇기 때문에, 한번에 하나의 패킷만을 전송할 수가 있다면, 해당 패킷의 RTT 동안 호스트는 아무것도 하지 않고 ACK 신호를 기다려야 하고, 이는 분명 퍼포먼스에 있어 치명적인 결점이 됩니다. 이를 극복하고 한번에 여러 개의 데이터 패킷을 전송함으로써 성능을 향상시키기 위해 고안된 것이 pipelining입니다.

pipelining을 지원하기 위해서는 많은 sequence number가 필요하고, 또 각 호스트에서 다량의 패킷을 저장할 buffer를 소유하고 있어야 합니다. 이러한 추가 자원이 할당되는 방식은 data loss와 data corruptiong, 그리고 packet의 time out에 대응하는 방법에 따라 크게 Go-Back-N과 Selective Repeat 방식으로 나누어집니다.

Q4-1) 그럼 GBN과 SR 방식에 대해서 설명해보세요.

GBN 프로토콜에서는 여러개의 데이터 패킷을 한번에 전송할 수 있지만, ACK 메시지를 받지 못한 데이터 패킷의 수에 대해 제한을 걸고 있습니다. GBN protocol에서는 base, nextseqnum, N 이렇게 세가지 변수를 가지는데요, 이는 flow control에 있어서 중요한 역할을 합니다. GBN sender host는 세가지 경우에 대해서 대처를 해야 하는데요, 가장 첫번째 경우는 상위 레이어에서 data message를 전송하였을 경우입니다. 이 경우 sender host는 window가 꽉 찼는지 확인한 이후, 꽉찼으면 메세지를 버퍼에 저장하거나 다시 돌려보내고, 그렇지 않았을 경우에만 이를 전송합니다. 두번째 경우는 ACK 메세지를 받았을 경우인데요, 이 경우, cumulative acknowlegment를 적용하여 그 이전의 패킷은 모두 성공적으로 전송이 되었다고 보고 그 이후부터 다시 전송을 시작한다. 이는 receiver측에서 순서와 맞지 않는 패킷은 과감하게 누락시키기 때문에 가능한 방법입니다. 마지막으로, time out에 대한 대응인데요, 네트워크의 congestion으로 인해서 전송이 지연되거나 패킷이 누락된 경우입니다. 이럴 경우, sender는 timeout 이벤트를 발생시킨 패킷부터 다시 패킷을 전송하게 됩니다.

GBN의 경우 많은 데이터 패킷을 한번에 재전송하기 때문에 성능에 있어서 단점을 초래할 수 있습니다. 그렇기 때문에 SR에서는 이를 보완하고자, receiver가 전송받지 못했다고 보이는 패킷만 선택하여 재전송함으로써, 불필요한 데이터 패킷 전송을 피하게 됩니다. receiver측에서는 순서에 맞지 않는 패킷이 도착했을 때, 그 패킷을 버퍼에 보관하고 있다가, 순서에 맞는 패킷이 오면 그를 같이 상위 계층으로 전송을 하게 됩니다. sender측에서는 base에 대한 ACK 메시지가 도착했을 때에만 window sliding을 함으로써 flow control을 해주게 됩니다.

Q5) TCP의 Reliable data transfer 에 대해서 설명해보세요

Transfer port layer의 프로토콜인 IP에서는 데이터의 전송에 있어서 데이터 전송, 순서, 무결성, 데이터 누락 등에 대해 어떠한 것도 보장하는 메커니즘을 지원하지 않습니다. 그렇기 때문에 TCP에서는 그러한 unreliability를 보완하기 위해서 Reliable data transfer 메커니즘을 지원하는데요, 이를 통해 corruption이 없고, 중복되지 않은 패킷이 순서대로 목적지에 올바르게 전달될 수 있도록 해줍니다.

TCP는 timeout과 duplicate acknowledgment를 통해서 이를 지원하게 됩니다. TCP 가 접하게 되는 이벤트는 크게 세가지로 볼 수 있는데요, 그 첫번째는 Application Layer에서 메세지를 전달받는 것입니다. 이 경우 TCP는 그를 segment로 캡슐화해서 Network Layer로 전달을 하고, timer를 시작합니다.

두번째 이벤트는 timeout인데요, TCP에서는 timeout을 발생시킨 segment를 재전송함으로써 이에 대응합니다. 여기서, timeout은 data packet의 누락때문에 발생하기도 하지만, 네트워크의 많은 트래픽때문에 전송이 지연되어 발생하기도 합니다. 그렇기 때문에 TCP는 한번 timeout으로 인해 재전송을 하게 되면 다음 time interval은 더블링을 해주게됩니다.

마지막 이벤트는 receiver로부터 ACK 메세지를 받는 이벤트인데요, 이 경우 TCP에서는 cumulative acknowledgment 메커니즘을 사용함으로써 segment가 순서대로 전송되었음을 알게 됩니다. ACK 메세지는 받았는데, 만약 timeout 이전에 똑같은 ACK 메세지를 세번을 받는다면, 그것은 TCP에서 패킷이 누락된 것으로 간주하고 fast retransmission을 해주게 됩니다.
Congestion control과 Flow control을 통해서 잠재적인 메세지 누락을 방지하는 것 또한 Reliable Data Transfer의 일환이라고 볼 수 있습니다.

Q5-1) 그럼 TCP는 GBN이에요 SR이에요?

TCP는 SendBase와 NextSeqNum을 가지고 있으면서 cumulative Acknowledgment를 사용한다는 점에서 GBN과 비슷합니다. 하지만 timeout 이후, 누락된 패킷을 포함하여 그 이후의 패킷들도 다같이 재전송하는 방식인 GBN과는 달리, fast transmit 메커니즘을 갖추고 순서에 맞지 않는 패킷들을 receiver측에서 버퍼에 저장해두고 오직 누락된 패킷만을 보낸다는 점에서는 SR과 비슷합니다. 그렇기 때문에 GBN과 SR의 하이브리드 버젼이라고 볼 수 있습니다.

Q5-2) TCP에서 three-way handshake는 왜 필요한거에요?

TCP가 지원하는 Reliable data transfer를 위해서는 buffer와 socket, 그리고 각종 변수 등의 파라메터들이 필요합니다. 그렇기 때문에 TCP에서는 데이터 패킷을 전송하기에 앞서서 이런 파라메터들을 설정해야 하는데, 그런 작업들을 하기 위해서 three-way handshake를 거치는 것입

니다.

Q6) Congestion Control에 대해서 설명해보세요

Network에서는 다양한 경우에 Congestion이 발생하게 됩니다. router의 capacity는 충분한데 packet-arrival rate가 outgoing rate를 초과하게 되는 경우, 패킷은 계속 라우터의 큐에 쌓이게 되고, 결국 congestion이 발생하게 됩니다. 라우터의 capacity가 유한할 경우에도 congestion은 발생합니다. 앞선 경우와 같이 arrival rate가 outgoing rate를 초과하는 경우, 이 시나리오에서는 메세지 누락이 발생하게 되는데요, 그렇게 되면 sender는 자연스럽게 재전송을 하게 됩니다. 이것은 부가적인 트래픽을 발생시키는 결과를 초래하게 됩니다. 하지만 무조건 timeout을 메세지 누락이라고 볼 수 만은 없습니다. congestion으로 인해서 over delayed packet이 존재할 경우, sender는 불필요한 재전송을 하게 됩니다. 이 또한 congestion에 일조하게 되는 것이죠.

이러한 Congestion을 제어하기 위해서는 네트워크의 도움을 받지 않는 end-to-end congestion control와 네트워크 구조의 도움을 받는 network-assisted congestion control이 있습니다. end-to-end congestion control에서는 timeout 혹은 data loss를 congestion의 근거로 봅니다. 그래서 이런 징후를 발견하게 되면, sender-side의 window size를 줄이던지, 아니면 TCP처럼 RTT를 늘리는 등으로 congestion control을 하게 됩니다. network-assisted congestion control에서는 라우터가 직접 choke packet을 날려서 자신이 congestion으로 인해 바쁘다는 피드백을 주는 방식을 취하기도 하고, 목적지 host에 resource management cell을 보내면서 NI, CI, ER 등의 control bit를 통해 sender측에 control을 요청하는 방식을 취하기도 합니다.

Q6-1) 그럼 TCP는 어떤 식으로 Congestion control을 하게 되나요?

TCP에서의 Congestion control은 크게 세가지의 congestion control로 구성되어 있습니다. 첫번째 control은 Additive-Increase, Multiplicative-Decrease 입니다. 이는, 평소에는 매 RTT마다 1 MSS씩 sending rate를 증가시키다가, timeout이나 packet loss의 이벤트를 만나게 되면 congestion window를 절반으로 줄이는 알고리즘입니다.

두번째 control은 slow start입니다. 이 알고리즘에서는 Congestion window가 처음에 1MSS에서부터 출발합니다. 그러다가 매 RTT마다 congestion widnwo가 두배로 증가하게 됩니다. 그러다가 loss event를 만나게 되면, 절반으로 줄어들고 그 이후부터는 linear하게 증가하는 알고리즘을 말합니다.

마지막으로, TCP Congestion control은 loss event의 종류에 따라 다르게 반응을 보이게 됩니다. 먼저, triple duplicate에 의한 loss detection에 대해서는 AIMD와 같이 반응을 하게 됩니다. 하지만 timeout에 의한 loss detection 이후에는 slow start phase로 들어가게 됩니다.

여기에, threshold라는 변수가 추가가 되면서 좀 더 동적인 congestion control이 가능하게 되는데요, 이 경우 timeout을 만나게 되면, 앞서 말한바와 같이 SS phase에 들어서게 되고, threshold는 congestion window가 떨어지기 직전값의 절반값으로 설정이 됩니다. 그럼 SS phase 동안 증가하던 congestion window는 congestion avoidance phase로 들어가게 되면서 AIMD 알고리즘을 사용하게 됩니다.

Q7) TCP의 Flow Control에 대해서 설명해보세요

Flow Control은 sender 가 receiver의 버퍼가 넘치지 않도록 segment 전송 속도를 조절하는 것을 말합니다. 이것은 sender side의 LastByteSent, LastByteAcked 이 두가지 변수와, 그리고 receiver side의 window size를 이용합니다. 여기서 window size란, rcvBuffer에서 TCP segment가 차지하고 있는 공간을 제외한 빈공간을 말합니다. TCP Flow Control의 원리는 간단합니다. LastByteSent에서 LastByteAcked를 빼면 unacknowledged bytes가 나오는데요, 이 사이즈가 receiver side의 window size를 초과하지 않으면 receiver side의 버퍼가 넘치지 않는다는 것입니다. 이 원리를 바탕으로, TCP sender는 두 바이트의 차이가 0보다 작지 않을 때에만 메세지를 전송을 함으로써 flow control을 구현합니다.

< Chapter 4: The Network Layer >

Q1) Forwarding과 Routing의 차이점에 대해서 설명해보세요

data packet은 네트워크에서 라우터를 통해서 이동할 때, 라우터의 input link로 들어갔다가 forwarding table에 의해서 정해지는 output link를 통해 다시 다음 네트워크로 전송이 되는데요, 이처럼 라우터의 input link interface로부터 output link interface로 데이터 패킷을 전송시키는 router-local action을 forwarding이라고 합니다. 반면, routing은 패킷이 출발지로부터 목적지까지 전송될수 있는 end-to-end paths를 정하는 network-wide process 입니다.

Q2) Network Layer가 best-effort service를 제공한다고 말하는데요, 이게 무슨 말인지 설명해보세요.

네트워크 서비스의 가장 핵심은 데이터 패킷을 안전하게, 그리고 순서에 맞게 목적지에 전달하는 것에 있습니다. 하지만 네트워크 레이어의 경우에는 그 중 어떠한 것도 보장해주지 않기 때문에, best-effort service를 제공한다고 하는 것입니다. 즉, 노력은 하지만 보장하지는 않는다는 것입니다.

Q3) Network layer에서의 데이터 전송 방식에는 Virtual Circuit Networks와 Datagram Networks 두가지 방식이 있는데요, 이 두 방식이 어떻게 다른지 설명해보세요.

Network layer의 데이터 전송 방식은 Transport layer와 마찬가지로 Connection service와 connectionless service로 분류가 되는데요, VC network가 connection service, Datagram network가 connectionless service 에 속합니다.

Virtual-Circuit은 source-destination 사이의 path, VC number, 그리고 forwarding table 속의 entry로 구성되어 있습니다. 그리고 VC setup, Data transfer, VC teardown 세가지 phase를 거치면서 데이터를 전송합니다. VC setup phase에서는 transport layer가 network layer에 segment를 전달하고, receiver address를 지정한 후, VC가 생성되길 기다립니다. 여기서, route가 정해지고, 각 router의 forwarding table에는 그에 해당하는 entry가 추가가 됩니다. 그 이후 Data transfer phase동안 데이터를 전송하게 되고, VC teardown phase 동안에는 서로 전송이 끝났다는 신호를 하고, router의 forwarding table을 업데이트 함으로써 모든 전송을 마감합니다.

Datagram Network에서는 패킷들이 connection setup 없이 네트워크에 던져지게 됩니다. 그러면서 네트워크 라우터를 지나게 되는데, 라우터에서 destination address를 forwarding table과

매치시켜 보고 알맞은 다음 링크로 전송이 됩니다.

Q3-1) Transport Layer TCP의 connection setup과 Network Layer VC의 connection set up은 어떤 차이가 있나요?

3-way handshake의 유무, 그리고 router의 참여 여부가 아무래도 가장 큰 차이점이 아닐까 생각합니다. 우선 TCP connection set up에서 필수요소였던 3-way handshake가 VC의 connection setup에는 존재하지 않고 그 대신 signaling message를 보내게 됩니다. 게다가, TCP connection setup은 단순히 end system들만이 참여하는 것에 반해, VC connection setup은 end system을 포함한 router들 또한 참여하게 됩니다.

Q4) data packet이 라우터에 도착해서 forwarding 되는 과정을 설명해보세요.

router는 input port, switch fabric, output port, 그리고 routing processor로 구성되어 있습니다. data packet이 라우터에 들어오면 input port를 통해서 들어오게 되는데요, 여기서 physical, link layer에 대한 encaptulation 작업을 진행하게 됩니다. 그리고는 input queue에 저장된 채로 lookup과 forwarding 과정을 거치게 되는데요, routing processor에 저장되어 있는 forwarding table의 이미지를 각 input port의 메모리에 복사해둠으로써 bottle neck을 줄이고 라우팅 속도를 개선하는 이런 방식을 decentralized forwarding이라고 합니다. decentralized forwarding과정을 거친 데이터 패킷은 switch fabric을 통해서 output port로 이동을 하게 됩니다. output port에서도 마찬가지로 queue가 있어서 데이터 패킷이 여기에 쌓이게 되는데요, output port를 거치면서 다시 link layer, physical layer processing을 거치면서 데이터 패킷은 캡슐화가 되어 다시 다음 hop으로 이동하게 됩니다.

Q5) Internet의 Network Layer의 주요 세가지 구성요소에 대해서 간단하게 설명해보세요.

Internet Network Layer는 IP, Routing protocol, 그리고 ICMP로 구성되어 있습니다. IP는 구체적으로 addressing과 forwarding이 어떤식으로 일어나는지에 대해 정의를 내리고 있고, Routing Protocol은 이름 그대로 데이터 패킷이 어떤 경로를 이용해서 목적지까지 전송되는지에 대한 의사결정을 합니다. 그리고 마지막으로 ICMP는 datagram의 에러 리포팅과 네트워크 상태 정보에 대한 signaling을 담당하게 됩니다.

Q6) IP fragmentation에 대해서 설명해보세요.

Link-layer frame이 전송할 수 있는 데이터 패킷의 maximum size를 Maximum Transmission Unit이라고 해서 MTU라고 부르는데요, Link-layer의 프로토콜마다 MTU를 다르게 갖고 있습니다. Network Layer의 datagram의 경우 link layer의 frame에 담겨서 이동을 하기 때문에, 이런 MTU는 datagram의 사이즈에도 limit을 주게 됩니다. 여기서 문제는, 바로 link마다 다른 프로토콜이 적용될 수 있고 그로 인해 다른 MTU가 적용될 수 있다는 점입니다.

예를 들어서 1000 바이트짜리 datagram이 link-layer를 통해서 전송이 되다가, MTU가 600 바이트인 link-layer protocol을 만나면, 1000 바이트의 데이터 패킷은 전송이 될 수가 없게 됩니다. 이럴 경우 하나의 데이터 패킷을 둘 이상의 MTU보다 작은 단위로 쪼개서 각각을 fragment로 포장을 하여 다시 전송을 하게 되는데요, 이러한 과정을 fragmentation이라고 합니다.

destination host가 이 datagram을 받아보았을 때, 순서에 맞게 잘 전송이 되었는지, 그리고 아니라면 어떻게 순서에 맞게 다시 datagram들을 합칠건지에 대해서 고민을 해야 합니다. 여기서 사용되는 것이 IP header에 있는 identification, flag, 그리고 fragmentation offset 필드입니다. identification을 통해서 같은 source로부터 온 datagram인지 확인을 한 뒤, offset을 통해 원래 데이터 array에서 어떤 위치에 해당 datagram이 들어가야 하는지에 대한 정보를 얻을 수 있습니다. 그리고 flag가 0으로 설정된 데이터그램이 있는데요, 이는 해당 데이터그램이 마지막 바이트를 갖고 있는 데이터그램이라는 뜻입니다.

Q6-1) IP fragmentation은 어떤 단점을 가지고 있는지 설명해보세요.

router와 end system이 데이터그램을 fragmentation하고 재조합 하는 작업을 해야 하기 때문에 이는 그대로 각 요소에 overhead로 작용할 수 있습니다. 그리고 이를 이용한 DoS attack이 가능해진다는 것 또한 약점 중 하나입니다.

Q7) Internet address assignment strategy인 Classless Interdomain Routing (CIDR)에 대해서 설명해보세요

네트워크는 각각의 고립된 네트워크들이 인터페이스를 통해 연결되어 있는 구조를 하고 있습니다. 여기서 각각의 고립된 네트워크를 서브넷이라고 하는데요, 이 서브넷의 주소 구조를 보면 prefix와 subnetmask로 구성이 되어 있습니다. subnetmask에 명시되어 있는 숫자만큼의 비트가 prefix를 구성하게 되는데요, 32비트의 주소값중 prefix를 제외한 나머지 lsb들이 서브넷 안에 있는 각각의 host들에 할당이 됨으로써 IP address assignment가 이루어지게 됩니다. 이런 전략을 CIDR라고 합니다.

Q8) DHCP에 의해서 Host address가 할당되는 메커니즘에 대해 설명해보세요.

address assignment는 크게 네가지 단계로 이루어집니다. 가장 처음이 DHCP server 를 찾는 과정인데요, host는 DHCP discovery message를 만들어서 그것을 broadcasting하게 됩니다. 그럼 이 메세지를 받은 DHCP server는 IP address를 담은 DHCP offer message를 broadcasting해서 보내게 됩니다. 그럼 이제 host는 server에 DHCP request를 보내게 되고, 그를 받은 서버는 DHCP ACK message를 보내면서 주소를 할당받게 되는 것입니다.

Q9) IPv4에서 IPv6로 넘어오면서 어떤 점이 달라지게 됐나요?

먼저, 시간을 많이 소요하는 Fragmentation 과 Reassembly 작업을 할 필요가 없게 되었습니다. 그 대신, 사이즈가 너무 크면 패킷이 너무 크다는 메세지를 sender에 보내면 sender쪽에서 작은 사이즈의 ip diagram을 보내게 되었습니다. 그리고, error checking을 할 필요가 없어졌습니다. 이미 transport-layer와 link-layer에서 에러 체킹을 하고 있기 때문에, checksum을 통한 또한번의 에러검사는 할 필요가 없게 되었습니다.

Q9-1) IPv4에서 IPv6로 한번에 넘어가는 것은 쉽지 않은데요, 그럼 어떻게 두 버젼의 IP를 사용할 수가 있죠?

초기에는 dual-stack approach를 사용했는데요, 이 경우 두 버젼 다 사용가능해야 한다는

기능은 만족했지만, IPv6 node가 IPv4를 만나면 IP 버젼을 다운해서 보내야 한다는 단점이 있었습니다. 이 점을 극복한 것이 바로 tunneling입니다. tunneling을 이용하면, 만약 IPv6에 해당하는 datagram이 IPv4 노드를 만나게 되면, 그 datagram 자체를 data field에 넣어서 전송을 하게 됩니다. 그러다가 다시 IPv6 노드를 만나면 이제 데이터 필드에 저장해두었던 IPv6 datagram을 꺼내서 다시 전송하는 방식을 사용하는 것입니다.

Q10) Network Layer에서의 routing algorithm에 대해서 설명해보세요.

Network Layer에서의 routing은 크게 global routing과 decentralized routing으로 나눌 수 있습니다.

global routing에서는 각 노드가 network topology에 대한 전체적인 정보를 가지고 있는 상태에서 source와 destination간의 최단거리를 찾게 됩니다. link-state algorithm이라 부르기도 하고, Dijkstra algorithm을 주로 사용합니다.

decentralized routing에서는 각각의 노드가 네트워크 전반에 걸친 topology에 대한 정보를 모른 채, 주변 노드와 정보를 주고 받으면서 최단거리를 계산하는 방식입니다. Distance-Vector algorithm이라고도 부르고, Bellman-Ford algorithm을 주로 사용합니다.

Q10-1) Link-State routing algorithm에 대해서 좀더 구체적으로 설명해보세요.

Link-State routing algorithm은, 각 노드가 자신의 identity와 자신에게 붙어있는 link들의 cost를 담은 link-state packet을 broadcasting함으로써 network topology에 대한 전반적인 지식을 공유한 상태에서 진행됩니다. Link-State routing algorithm의 대표라고 할 수 있는 Dijkstra algorithm에 대해서 설명해보겠습니다.

Dijkstra algorithm에서는 자신 노드로부터 모든 노드에 이르는 least cost path를 계산하게 됩니다. 이 알고리즘은 세가지 데이터구조를 가지고 있는데요, 먼저 D(v)라고 해서 source node로 부터 destination node인 v까지의 최단거리의 cost를 모아놓은 집합이 필요합니다. 그 다음은 p(v)라는 데이터구조로써, 이는 node v로부터의 current least-cost path에서 이전 node에 대한 집합입니다. 그리고 마지막은 N'라는 집합으로, node v로의 최단거리가 계산이 되면 node v가 이 N'라는 집합의 원소로 추가가 됩니다.

알고리즘은 크게 initialization step과 loop step으로 나누어집니다. initialization step에서는 source를 설정하고, 이웃 노드들까지의 cost를 최단거리로 우선 설정하게 됩니다. 그리고 나머지 노드들로의 최단거리는 무한대로 설정이 됩니다. 그러면 loop step에 들어가게 되는데요, loop step에서는 N'에 없으면서 D(w)가 최소인 w를 찾게 됩니다. w를 찾은 이후에는 N'에 w를 포함시키고, s노드의 이웃들 중 각각의 v 노드들 중에서 D(v) = min(D(v), D(w) + c(w,v))를 계산하게 됩니다. 그리고 이런 작업은 N'가 노드 전체를 포함할 때까지 계속 진행이 됩니다.

Q10-2) Link-State routing algorithm은 어떤 문제를 안고 있고, 어떻게 해결 가능한가요?

LS routing algorithm은 oscillation이라는 문제를 안고 있습니다. 만약 link cost가 전송하게될 data load와 같다고 한다면, source로부터 destination까지의 최단거리 route가 계속 반복적으로 변하게 되는 현상이 발생할 수 있는데요, 이를 oscillation이라고 합니다.

이를 해결하기 위해서는 LS algorithm뿐만이 아니라 다양한 알고리즘을 사용하는 방법이

있습니다.

Q10-3) Distance-Vector routing algorithm에 대해서 좀더 구체적으로 설명해보세요.

Distance-Vector routing algorithm에서는 LS routing 과는 달리 네트워크 topology에 대해 아무런 지식 없이, 각각의 node들 스스로가 neighbor node와 정보를 교환하면서 least cost path를 계산해내는 방식입니다.

Distance-Vector routing의 대표적인 알고리즘으로는 Bellman-Ford algorithm이 있는데요, dx(y)가 node x에서 node y로의 최단거리라고 한다면, node x의 모든 neighbor node인 v에 대해서, dx(y) = minv{c(x, y) + dv(y)}, 로 설정이 됩니다. Bellman-Ford algorithm에서는 세가지의 데이터구조가 이용됩니다. 먼저, neighbor node와의 link cost, c(x,y), 가 필요합니다. 그리고 x로부터 네트워크 안의 모든 노드들에 대한 cost estimates를 담고 있는 Dx = [Dx(y): y in N], 마지막으로 neighbor들로부터 각 노드들에 대한 least cost를 담고 있는 Dv = [Dv(y): y in N] 가 필요합니다.

DV routing algorithm은 크게 initialization과 loop step으로 나누어집니다. initialization step에서는 Dx(y)를 c(x,y) 혹은 infinity로 설정을 하게 됩니다. 그리고 모든 neighbor node w에 대해서는 Dx(y) 값을 infinity로 설정을 하고, 이렇게 만들어진 Dx(y) vector를 모든 neighbor node에게 전달하게 됩니다. loop step에서는 link cost에 update가 오기까지 기다리게 됩니다. 그러다가 update 소식을 듣게 되면, Dx(y) = minv{c(x,v) + Dv(y)} 연산을 수행하게 됩니다. 그리고 그를 통해 업데이트를 하게 되면 다시 업데이트 된 내용을 neighbor node들에게 보내게 됩니다.

Q10-4) Distance-Vector routing algorithm은 어떤 문제를 안고 있고, 어떻게 해결 가능한가요?

Q10-5) LS와 DV을 성능 차원에서 비교해보세요.

LS의 경우에는 처음에 network상의 모든 node에 link state를 broadcasting해야 하지만, 자신과 붙어있는 노드와의 cost에 대해서만 정보를 전달하게 됩니다. 반면 DV에서는 자신과 붙어있는 노드들에게만 정보를 전달하면 되지만, 한번에 network상의 모든 node들에 대한 distance vector를 전달해야 합니다. 게다가, LS의 경우, 특정 router가 잘못됐을 경우 그 router 주변만이 잘못된 정보를 가지고 있지만, DV의 경우 그 정보가 퍼져나가서 네트워크 전체에 영향을 주게 된다는 단점을 가지고 있습니다.

Q11) Autonomous System (AS)가 생겨난 원인에 대해서 설명해보세요.

AS를 통해 얻을 수 있는 이점은 크게 두가지로 나누어서 볼 수 있습니다. 먼저, LS routing의 경우 network에 있는 모든 노드들에게 broadcasting을 해야 하는데, 그럴 경우, network bandwidth가 남아나질 않을 것입니다. 그리고, 회사 혹은 특정 조직에서는 자신들의 네트워크를 따로 관리하고 싶은 욕구가 있을 수 있습니다. 이 두가지 문제를 해결하기 위해서 Autonomouse System이라는 개념이 제안된 것입니다.

Q12) Internet에서 routing을 할 때, intra-AS routing에 대해서 설명해보세요.

Intra-AS routing은 크게 RIP와 OSPF로 나눌 수 있습니다.

먼저, Routing Information Protocol의 경우, 이는 DV routing에 속하는데요, link마다 cost

를 다르게 메기지 않고, hop count를 path cost로 생각하는 알고리즘입니다. 30초마다 한번씩 RIP response message를 통해서 distance vector를 주변노드와 교환을 하면서 RIP table과 forwarding table을 업데이트 하게 됩니다.

RIP와는 반대로 OSPF는 flooding을 통한 broadcasting과 Dijkstra algorithm을 사용하는 LS routing algorithm입니다. OSPF의 특징이라고 한다면 AS를 구조화할 수 있다는 점입니다. 하나의 AS는 boundary router를 시작으로 하는 backbone area와 그 속의 backbone router, 그리고 backbone area와 일반 area를 이어주는 area border router와 일반적인 internal router로 구성되어 있으면서, 각 area는 모두 자신들만의 routing algorithm을 구현하게 됩니다.

Q13) Internet에서 routing 할 때, inter-AS routing에 대해서 설명해보세요.

대표적인 Inter-AS routing으로는 Border Gateway Protocol (BGP)가 있습니다. BGP의 기본 개념은 다음과 같습니다. 먼저, neighboring AS로부터 subnet reachability information을 받습니다. 그리고 그런 정보를 AS 내부에 있는 모든 라우터들에게 전달해줍니다. 그리고 그런 정보를 가지고 subnet으로의 좋은 route를 모색하게 됩니다.

Q14) N-way unicasting이 뭐에요?

Broadcasting은 크게 두가지 방법에 의해서 이루어질 수 있습니다. 먼저, 하나의 node가 각각의 node에 point-to-point 방식으로 data를 전송하는 방식이 있을 수 있습니다. 두번째 방식은, source node가 neighbor node들에게 data를 전송하면 neighbor node들이 그 data의 copy를 다시 그 neighbor node들에 전송해주는 방법을 말합니다. N-way unicasting은 첫번째 경우를 말합니다.

Q14-1) N-way unicasting의 단점에 대해서 말해보세요.

N-way unicasting의 경우 n개의 패킷이 만들어져서 network를 돌아다니게 되는데요, 이는 어찌보면 굉장히 비효율적인 방법이라고 볼 수 있습니다. 그리고, N-way unicasting을 통해서 broadcasting을 하려면 source node가 각 노드의 IP address를 알아야 하는데, 그 방법이 굉장히 무겁다는 것입니다.

Q15) flooding의 문제와, 그 해결방법에 대해서 설명해보세요.

먼저, flooding은 크게 uncontrolled flooding과 controlled flooding이 있습니다. uncontrolled flooding에 대해 먼저 설명을 하고 그 다음에 controlled flooding에 대한 설명을 하도록 하겠습니다.

flooding의 개념은, 어떤 노드가 broadcast packet을 받았을 때, 그 broadcast packet을 복사하여 그 복사된 패킷을 다시 모든 이웃 노드들에게 broadcasting하는 것을 말하는데요, 여기에서 데이터를 복사하고 전송하는 것에 있어서 아무런 제어를 하지 않는 것을 uncontrolled flooding이라고 합니다. 그 결과, 무한히 많은 broadcasting package이 생성이 되어 broadcast storm을 만들고, 그는 곧 네트워크 자체를 무용지물로 만들어버리게 됩니다.

이런 단점을 극복하기 위해 만든 것이 controlled flooding입니다. 이에는 크게 sequence-number-controlled flooding과 reverse path forwarding이 있는데요, sequence-number-controlled

flooding의 경우는 각 노드가 자신이 받은 데이터 패킷의 SN을 가지고 있어서, 만약 그 패킷을 한번더 받을 경우 그 패킷을 누락시키는 방법을 말합니다. 그리고 RPF에서는, router가 패킷을 받았을때, 그 패킷이 source로부터 최단거리를 통해서 왔는지 확인해보고, 그랬을 경우에만 모든 outgoing link로 패킷을 전송하게 됩니다.

controlled flooding의 경우도, redundant broadcast packet을 완벽하게 피하지는 못하는데요, 그를 위해 고안된 flooding mechanism이 바로 spanning-tree broadcast입니다. cycle을 포함하지 않고 모든 node를 포함하는 tree 구조를 spanning tree라고 하는데요, 그 cost의 값이 최소값이 되는 tree를 minimum spanning tree라고 합니다. 이 spanning tree를 만들기 위해서 center-based approach가 사용됩니다. 이 spanning tree를 사용해서 broadcasting을 하게 되면 controlled flooding에서 겪었던 redundant 문제까지 해결할 수가 있습니다.

< Chapter 5: The Link Layer and Local Area Networks >

Q1) Link Layer에서 지원하는 서비스에는 어떤 것이 있나요?

Transport layer에서 내려온 datagram을 frame으로 캡슐화 하여 MAC protocol을 이용해서 실제 링크 위에 올려놓는 역할을 합니다. 뿐만 아니라 에러 없이 데이터그램을 전송하기 위한 reliable delivery도 가능하고, flow control, error detection, error correction 등의 서비스도 함께 제공합니다. 마지막으로, half-duplex와 full-duplex를 함께 지원합니다.

Q2) Link Layer에서 error detection을 위해 사용하는 기술에 대해서 말해보세요.

Link layer에서는 에러를 발견하기 위해서 parity check, checksumming method, 그리고 cyclic redundancy check라는 세가지 기술을 사용합니다.

먼저, parity check는 크게 single parity scheme과 two-dimensional parity scheme 두가지로 나눌 수 있습니다. parity check는 parity bit를 추가하면서 1 비트의 개수를 세어 보는 방식인데요, 이를 통해서 에러 유무를 파악할 수 있습니다. 거기에, two-dimensional parity scheme을 사용하면 어디에서 에러가 발생했는지 또한 알 수가 있어 이를 수정하는 것 또한 가능해집니다.

checksum method에서는 헤더에 있는 필드를 더한 뒤 그의 1의 보수를 구하는 방식으로 진행이 되는데요, 이 때 receiver측에서 checksum 필드와 다른 필드들을 더했을 때, 모든 비트가 1이면 error가 없는 것이고, 0인 비트가 있으면 그곳에서 에러가 났다는 것을 발견할 수 있게 됩니다.

Link Layer에서는 이제 다음 방법인 Cyclie Redundancy Check (CRC) 를 주로 사용합니다. Transport layer는 소프트웨어에서 error detection을 처리하기 때문에 그 간편성이나 속도가 매우 중요한 데 반해, Link layer에서는 전용 hardware를 사용하기 때문에 속도가 빨라 CRC를 사용할 수 있는 것입니다. 여기서는 r-bit의 CRC bit와, D*2^r을 나눌 수 있는 r+1 bit인 Generator를 추가적으로 사용하게 되는데요, D*2^r = nG XOR R 일 때 R이 0이면 에러가 없는 것이고, R이 0이 아니면 에러가 있다고 판단하는 것입니다.

Q3) Link Layer에서는 Multiple Access Protocol을 사용하게 되는데요, MAP가 왜 필요한건지 설명해보세요.

Link Layer는 크게 point-to-point linke와 broadcast link로 나눌 수 있습니다. broadcast link의 경우 각 노드는 데이터를 전송하기 위해서 broadcasting 방식을 사용하게 되는데요, 모든 sender와 receiver가 모두 이런 방법을 사용하다보니, low-level에서 신호가 서로 충돌되어 손상되는 문제가 발생하게 되고, 결과적으로 데이터가 제대로 전달되지 못하게 됩니다. 이를 극복하기 위해서 각 노드가 데이터 전송을 제어하기 위한 방법이 필요했고, 그를 위한 multiple access protocol이 필요하게 된 것입니다.

Q3-1) 그럼 Multiple Access Protocol의 종류에는 어떤 것들이 있나요?

Multiple Access Protocol은 크게 Channel Partitioning protocol, Random Access protocol, 그리고 Taking-Turns protocol 로 나눌 수 있습니다.

Q3-2) Channel Partitioning protocol에 대해서 설명해보세요.

channel partitioning protocol은 크게 TDM과 FDM, 그리고 CDMA 방식으로 나눌 수 있습니다. TDM 방식에서는 bandwidth를 time frame별로 나누고, 또 그 frame을 time slot으로 나누어서 각 time slot에 데이터 전송 을 할당하는 방식을 말합니다. 그리고 FDM은 bandwidth를 주파수별로 나누고 각 주파수별로 데이터 전송을 할당하는 방식을 말합니다.

마지막 방식인 CDMA는 FDM방식과 TDM 방식의 단점을 보완한 방식으로도 볼 수 있습니다. FDM과 TDM이 주파수와 time-slot을 할당하는 방식이라면, CDMA는 각 노드에 code를 할당하는 방식입니다. 그래서 이 코드가 잘 선택이 된다면, 많은 노드가 한번에 전송을 하면서 간섭이 일어난다고 할지라도 receiving node가 그 결과를 다시 원래의 데이터로 복구할 수 있게 되는 것입니다.

Q3-3) Random Access protocol에 대해서 설명해보세요.

Random Access protocol은 channel partitioning 방식과는 달리 데이터를 전송할 때 항상 Full transmission rate를 이용해서 전송하게 됩니다. 그러다가 다른 node와의 충돌을 감지하게 된다면 자신의 전송을 멈추게 되고, 네트워크가 idle state에 접어들면 random하게 정해진 시간을 기다린 이후에 다시 전송을 시작하는 방식입니다. Random Access protocol에는 크게 ALOHA protocol과 CSMA protocol이 있습니다.

ALOHA protocol의 가장 기본적인 형태는 slotted ALOHA입니다. 이 방식에서는 모든 frame이 L bit로 나누어지고 이렇게 나누어진 frame은 각 time-slot에 할당이 됩니다. 그리고 각 노드는 time slot의 시작지점에만 frame을 전송할 수 있습니다. 그리고 만약 노드간의 충돌이 일어난다면, 각 노드는 자신의 데이터 전송을 취소하고 랜덤한 시간을 또 기다리게 됩니다. 여기서는 이제 데이터 전송이 full rate로 된다는 장점이 있지만, 반면 종종 time-slot이 낭비된다는 단점 또한 안고 있습니다. 그 효율은 0.37R 정도라고 조사되어져 있습니다. pure ALOHA는 slot이 나누어져있지 않아 앞뒤를 모두 신경써야 하기 때문에 그 확률이 절반으로 떨어진다는 단점이 있습니다.

커다란 cocktail party에서 사람들이 좀 더 효과적으로 의사소통을 하기 위해서는 몇가지 매너를 지키게 되는데요, 이러한 방법론을 선택하여 데이터 전송방식에 적용시키는 것이 CSMA 방식입니다. 이 protocol에서는, 각 노드가 데이터를 전송하기 이전에 먼저 주변을 살펴보게 됩니

다. 그래서 만약 다른 노드가 데이터를 전송하고 있으면, 임의의 시간을 기다리다가 다시 channel을 sense해보게 됩니다. 만약 channel이 idle하다는 것이 발견되면 바로 데이터 전송을 시작하는 방식입니다. 그리고 CSMA/CD에서는 collision detection scheme이 도입되는데요, 만약 자신이 데이터를 전송하고 있을 때 다른 노드와의 충돌을 감지하게 되면 바로 자신의 전송을 취소하는 방식이 되겠습니다.

Q3-4) Taking-Turns protocol에 대해서 설명해보세요.

Multiple Access Protocol에서는 두가지 ideal requirements를 가지고 있습니다. 먼저, 하나의 node만이 active할 때에는, 그 노드는 full rate로 데이터 전송을 할 수 있어야 한다는 점. 그리고 두번째는 M개의 노드가 active 하다면 각 노드는 R/M의 속도로 데이터 전송을 할 수 있어야 한다는 점입니다. channel partitioning과 random access protocol에서는 첫번쨔 조건은 만족시키지만 두번쨔 조건은 만족시키지 못하기 때문에, 그를 보완하는 새로운 protocol이 필요했는데요, 그것이 바로 taking-turns protocol입니다. taking-turns protocol의 예로 들 수 있는 것으로는 polling protocol과 token-passing protocol이 있습니다.

polling protocol에서는 master node를 하나 두고, master node가 각각의 노드에게 poll을 제공하면 해당 노드가 데이터 전송을 시작할 수 있는 방식입니다. 이 방식에서는 collision을 피하고 full rate로 전송할 수 있다는 장점이 있지만, polling delay와 master node에 문제가 발생할 경우 전체 시스템이 동작하지 않을 수 있다는 단점을 가지고 있습니다.

그 다음은 token-passing protocol인데요, 이 방식에서는 하나의 토큰을 다른 여러 노드가 일정한 패턴으로 주고 받는 형식입니다. 그러면서 토큰을 받은 노드는 데이터 전송을 시작하고, 만약에 전송할 데이터가 없으면 토큰을 다음 노드에게로 넘겨주는 방식을 말합니다. 이 방식에서는 하나의 노드가 오작동을 하면 전체 시스템에 문제가 발생할 수 있다는 단점이 있습니다.

Q4) MAC Address가 뭐에요?

Link Layer 에서는 host간에 서로 데이터를 주고받는다기 보다는 host에 붙어있는 network adapter끼리 데이터를 주고 받는다고 볼 수 있습니다. 그렇기 때문에 IP address를 사용하기 보다는 각 adapter의 주소를 사용해야 하는데요, 6바이트로 이루어진 이 어뎁터의 물리적인 주소를 MAC address라고 합니다. 이 address의 경우 항상 고유한 번호를 가지고 있어야 하기 때문에 이 주소는 IEEE에서 관리하고 할당해주는 것을 원칙으로 합니다.

Q4-1) sending 노드가 어떻게 receiving 노드의 MAC address를 알아서 데이터를 전송할 수 있는 건가요?

Network layer까지 각 노드는 IP address를 가지고 데이터를 전송하기 때문에 MAC address를 알아낼 방법이 없습니다. 그래서 여기서 적용되는 protocol이 바로 Address Resolution Protocol (ARP) 입니다. 각 노드는 메모리에 ARP Table이라는 데이터구조를 가지게 되는데요, 이 구조 안에는 IP address와 MAC address간의 맵핑 정보가 담겨 있습니다. 그래서 이를 통해 맵핑을 하여 데이터를 보내게 되는 것입니다. 만약 ARP table 안에 엔트리가 존재하지 않을 경우에는 sender side에서 ARP packet이라는 특수한 패킷을 만들어 broadcasting을 하게 되고, 해당되는 IP address를 지닌 host쪽에서는 그 응답을 보내게 되어 entry가 자동으로 업데이트 될 수 있는 것

입니다.

Q5) Ethernet이 뭔지 설명해보세요.

Ethernet은 현재 가장 주목받고 있는 wired LAN technology로, 다른 기술들에 비해 높은 데이터 전송 속도를 꾸준히 보여줌으로써 현재 표준으로 사용되고 있는 기술입니다. CRC check를 제공하지만 connection과 ACK 메세지 없이 데이터를 전송하기 때문에 unreliable하다고 볼 수 있지만, 이 점이 Ethernet을 간단하고 보다 싸게 만들어주는 이점으로 작용하기도 했습니다. 과거 Ethernet은 hub에 연결된 star topology 방식이었는데요, hub의 경우 broadcasting 방식을 사용하기 때문에 Multiple Access Protocol이 반드시 필요했고, 그를 위해 CSMA/CD 방식을 채택하여 사용하였습니다.

오늘날에는 hub-based가 아닌 switch-based start topology를 사용하기 때문에 broadcast link에서 point-to-point link로 전환이 되었습니다. 뿐만 아니라 store-and-forwarding packet switching 방식을 사용하고 있습니다. point-to-point 방식을 사용하고 있기 때문에 더이상 multiple access protocol이 적용될 필요가 없지만, hub와 switch가 동시에 공존하는 이상, multiple access protocol은 적용이 되어야 합니다.

Q6) router와 switch, 그리고 hub가 어떻게 다른지 설명해보세요.

먼저, router, switch, 그리고 hub는 각각 transport layer, network layer, 그리고 link layer에 속한 디바이스 입니다. router의 경우 IP address를 사용하여 데이터를 포워딩 하지만 switch의 경우 MAC address를 이용해서 포워딩을 한다는 차이가 있습니다. 그리고 switch의 경우 filtering기술을 적용하고 있고 데이터 포워딩을 할 때 두개의 OSI layer만 거치면 되지만, routing의 경우 filtering이 없고 세개의 layer를 거침으로 인한 delay를 갖는다는 단점이 있습니다. 반면, router의 경우 spanning tree scheme을 적용함으로써 broadcast storm에 대한 방안을 제시하고 최적의 포워딩 및 라우팅 경로를 제시하고 있지만, switch의 경우 그에 대한 어떠한 방안도 제시하지 않고 있다는 점 또한 차이점으로 볼 수 있습니다.

< Chapter 6: Wireless and Mobile Networks >

Q1) Wireless 환경은 infrastructure mode와 ad hoc mode로 나눌 수 있는데요, 두 모드의 차이에 대해서 말해보세요.

Infrastructure mode는 각각의 wireless host가 base station을 통해서 기존의 wired network에 접근할 수 있는 구조를 말합니다. 그렇기 때문에 전통적인 addressing, routing 방식이 적용되는 환경입니다. ad hoc mode란 wireless host가 그와같은 기존의 infrastructure를 가지고 있지 않은 환경을 말합니다. 그렇기 때문에 ad hoc mode에서는 host들이 스스로 routing 및 forwarding 등의 기능을 만족시켜야 합니다.

Q2) Wired link에 비해 Wireless가 안고 있는 문제점이 있다면 어떤 것이 있나요?

먼저, sender와 receiver간의 거리가 멀어질수록 그 신호의 강도가 떨어진다는 단점이 있습니다. 뿐만 아니라, 다른 신호로 인한 간섭을 받음으로써 신호가 손상된다는 단점도 있고, host

가 이동함에 따라 속해있는 subnet이 변하게 되고, 그에 따라 path가 계속 변하기 때문에 multipath propagation의 단점도 안고 있습니다.

앞서 말한 세가지 단점으로 인해 wireless 인터넷은 높은 bit error rate을 갖게 된다는 risk를 안게 됩니다. 그를 극복하기 위해서 Signal-to-Noise Rate를 높이는 방법도 있지만 power를 많이 소모한다는 단점이 있구요, physical layer에서의 모듈을 transmission rate가 높은 것으로 바꿔가며 사용하는 방법 또한 가능한 방법 중 하나입니다.

bit errer rate가 높다는 것 외에도 hidden terminal problem과 fading이라는 문제점 또한 안고 있습니다.

Q3) CDMA는 어떤 식으로 multi access protocol을 구현하게 되나요?

아무래도 multi access protocol에서 가장 중요하게 다뤄져야 하는 것은 신호간의 충돌입니다. 이는 신호간의 충돌이 일어나면 신호가 손상을 입게 되고 그로 인해 제대로 된 데이터가 전송되지 않기 때문인데요, 신호가 같이 합쳐져서 그 데이터가 손상되더라도 그를 복구시키면서 올바른 데이터를 전송할 수 있는 multi access protocol이 code division multiple access (CDMA) 입니다.

이는 random access protocol의 한 종류인데요, sender와 receiver는 데이터를 주고받기 이전에 코드비트인 chipping sequence를 공유하게 됩니다. 그럼 sender측에서 데이터를 전송하면서 이 코드를 통해서 기존의 데이터를 encoding하게 되는데요, 이렇게 전달된 데이터는 receiver 측에서 똑같은 code를 이용해서 decoding이 됩니다. 그로 인해 원래의 데이터로 복구를 하는 방식입니다.

이 방식을 이용하면 아무리 신호가 다른 신호와 합쳐져서 잘못된 신호가 전달된다 하더라도, 잘 디자인된 코드를 통해서 원래의 신호를 복구할 수 있기 때문에, 신호간 충돌로 인한 데이터 손상을 극복하기 위한 multiple access protocol로 사용되고 있습니다.

Q4) WiFi 환경에서 wireless host가 AP에 접속하는 방법에 대해서 설명해보세요.

WiFi 환경에서 wireless host와 AP가 연결되는 것을 association이라고 하는데요, BSS 안에 있는 host가 접근 가능한 AP를 발견하고 그에 association request frame을 보내면 AP측에서 response frame을 보내주는 식으로 association은 성립이 됩니다. 여기서 알아야 할 것은 host가 접근 가능한 AP를 찾는 방법인데요, 그에는 크게 두가지의 방법이 있습니다.

먼저, passive scanning이 있는데요, passive scanning에서는 AP가 SSID와 MAC address가 담긴 beacon frame이라는 frame을 broadcasting 하게 됩니다. 이는 광고같은 것인데요, 이 frame을 받은 host는 그 AP가 접근가능하다는 것을 알게 되어, request와 response를 주고 받으면서 association을 맺게 되는 것입니다. 그 외에도 active scanning이 있는데요, 여기에서는 host가 probe frame을 broadcasting하게 됩니다. 그래서 이를 감지한 AP에서probe response를 보내면, 이것을 받은 host와 함께 association request와 association response frame을 주고받으면서 association이 성립하게 됩니다.

Q5) 그럼 WiFi 에서는 어떤 Media Access Control (MAC) Protocol을 사용하나요?

IEEE 802.11에서는 CSMA/CA 프로토콜을 사용합니다. 이는 CSMA/CD와 그 기본원리는

비슷합니다. CSMA/CD에서는 Carrier Sensing이라는 기법을 사용하는데요, 이는 사람과 사람간의 대화에서 말하기 전에 먼저 상대방의 말을 경청해주는 것과 같은 원리입니다. 각 host는 주변에서 데이터 신호가 전송이 되고 있다는 것을 알게 되면 잠시 자신의 전송을 미루게 됩니다. 그러다가 주변 netowrk가 idle한 상태가 되면 그제서야 전송을 하게 되는 것입니다. 뿐만 아니라, 자신이 전송을 하고 있는데 다른 host가 전송한 데이터신호와의 충돌이 일어났음을 감지하게 되면 그 즉시 전송을 멈추고 random period 동안 자신의 전송을 미룬 이후 다시 전송을 하게 되는 방식입니다.

CSMA/CA는 Collison Avoidance로, Collison Detection과는 조금 다른 방법을 사용합니다. CSMA/CA는 wireless 환경에서 사용이 되는 것이기 때문에, wireless 라는 환경의 특성의 단점을 그대로 소화해야 합니다. 말하자면, wireless 환경에서는 신호가 전송될수록 그 강도가 약해져서 sender와 멀리 떨어져있는 host는 그 데이터 신호가 전송되고 있는지, 혹은 자신이 보내는 데이터 신호와 충돌을 일으키고 있는지 알기 힘들 수 있습니다. 이를 hidden terminal problem 이나 fading이라고 하는데요, 이를 극복하기 위해서는 그를 감지하기 위한 섬세한 hardware를 구축해야 하는데 그는 비용이 너무 많이 들게 됩니다.

따라서, CSMA/CD처럼 다른 host의 전송이 끝나자마자 데이터 신호를 전송하게 되면 다른 신호와의 충돌이 일어났을 때 그를 인식하지 못해서 bandwidth를 낭비할 수 있기 때문에, CSMA/CA에서는 다른 network가 idle state에 들어선 이후에 각 host는 Distributed Inter-Frame Space라는 임의의 시간동안 기다렸다가 데이터를 전송하게 됩니다. 이를 통해 동시에 신호를 전송하여 collison을 일으킬 가능성을 최대한으로 줄이게 되는 것입니다.

그리고 WiFi라는 bit가 변질되기 쉬운 환경에서 사용되고 있기 때문에, link-layer acknowledgement scheme 또한 사용하고 있습니다.

Q6) Hidden Terminal Problam이 왜 일어나고, 어떤 점에서 문제가 되는 것인가요?

wireless환경에서 host가 신호를 전송하면 그 신호는 sender로부터 멀어질수록 강도가 약해지는 fading이라는 문제를 겪게 되는데요, Hidden Terminal problem은 이런 특성때문에 발생합니다. 이 것이 왜 문제가 되는지는 하나의 시나리오를 들어가며 설명을 하겠습니다. 하나의 AP가 있고, 그 BSS에 속해있는 두개의 host가 그 AP에 동시에 데이터를 전송하기 시작했다고 가정을 하겠습니다. 이런 상황에서, 전송되는 데이터 신호는 각각의 sender로부터 멀어질수록 그 신호가 약해지기 때문에 AP에 데이터신호가 도달했을 때, 두 신호는 충돌을 일으킴에도 불구하고 양쪽의 host는 그 신호를 감지하지 못하는 경우가 발생할 수 있습니다. 이럴 경우 전송은 했지만, 계속 충돌만을 일으키면서 데이터도 제대로 전송하지 못하고 네트워크 자원만 낭비한 결과를 초래하게 됩니다. 그렇기 때문에 hidden terminal problem이 wireless internet 환경에서 문제가 되는 것입니다.

Q6-1) 그럼 Hidden Terminal Problem의 해결책으로는 어떤 것들이 있나요?

RTS와 CTS가 그 해결책으로 사용되고 있습니다. sender가 데이터를 전송하고 싶을 때, sender는 먼저 해당 AP에 예상 전송시간과 ACK 메세지를 담고 있는 RTS frame을 보내게 됩니다. AP가 이 RTS 프레임을 받고 나면 CTS frame을 broadcasting해줌으로써 응답을 해주는데요, 여기서 CTS는 sender에게 전송이 가능하다고 알려주는 동시에, 다른 host들에게 지금은 데이터를 전

송하는 host가 있으니 전달된 예상 전송시간만큼 기다리라는 신호를 보내주는 두가지 역할을 수행하게 됩니다. 여기서도 물론 CSMA/CA방식이 적용이 됩니다.

이런 RTS/CTS 방식은 특정 delay가 있고, 또 네트워크 자원을 소모하게 된다는 단점이 있기 때문에, 그런 overhead를 감수할만한 가치가 있는, 대용량의 데이터 전송을 위한 channel reservation에 적합한 메커니즘이라고 볼 수 있습니다.

Q7) Cellular Internet Access에서는 FDM/TDM이 조합된 시스템과 CDMA 방식이 두 주류를 이루고 있습니다. 이 두 방식에 대해 설명해보세요.

FDM/TDM이 조합된 시스템에서는, 먼저 FDM 방식을 이용해서 주파수별로 대역폭을 나누게 됩니다. 그 이후 각 주파수별로 TDM 방식을 적용하는 방식을 말합니다. 그리고 CDMA의 경우는 주파수를 나누지 않고, sender 와 receiver끼리 chipping sequence를 교환함으로써 collisong을 피하면서 데이터를 전송하는 방식을 말합니다. 한국과 북미지역에서 주로 CDMA가, 그리고 유럽지역에서 FDM/TDM의 혼합방식이 사용되는데요, CDMA의 경우 주파수 할당을 할 필요가 없을 뿐 아니라, TDM/FDM 방식에서는 일정 거리 이상 떨어져있는 지역에서만 해당 주파수를 재사용할 수 있는데 이런 점을 고려할 필요가 없다는 장점을 가지고 있습니다.

Q8) 최근 3G방식이 큰 주류를 이루게 되었는데, 2G와 3G의 차이가 뭐에요?

가장 큰 차이라고 한다면 2G는 음성 통신에 초점을 맞추었고, 3G는 음성과 데이터 통신 모두에 초점을 맞추었다는 것을 들 수 있습니다. 예전의 1G도 있었는데, 1G의 경우는 음성 통식에만 초점을 맞춘 아날로그 방식의 시스템이었습니다.

먼저 2G에 대해서 구체적으로 설명해보겠습니다. 2G의 경우에는 아날로그 신호를 디지털 신호로 전환을 하여 전송하는 방식을 사용합니다. 이는 크게 GSM과 IS-95 WCDMA 방식을 표준으로 채택하였습니다. 먼저 GSM방식은 유럽지역에서 주로 사용되는 방식인데요, FDM과 TDM을 혼합한 인터페이스를 사용하고 있습니다. 그리고 북미지역과 한국에서는 CDMA 방식을 사용하고 있습니다.

2세대에서 3세대로 넘어가기 이전에는 2.5세대의 통신이 존재했는데요, 여기서는 GPRS방식과 CDMA 2000의 phase 1 방식이 표준으로 채택되었습니다. GPRS방식은 기존의 방식에서는 TDM time slot을 한 호스트당 하나만 할당해준 것을 on-demand 방식으로 바꿔서, 필요에 따라 할당량을 변경해가는 방식을 채택하여 사용했습니다.

3세대 통신에서는 CDMA-2000과 UMTS방식이 채택되어 사용되고 있습니다. UMTS에서는 TDMA의 타입 슬롯 안에 WCDMA 기술을 접목시킨, 기존의 GSM 방식과는 별도의 방식을 채택하였습니다.

Q9) Wireless 환경에서 Mobile User가 하나의 subnet에서 다른 subnet으로 이동했을 경우, IP 할당은 어떻게 이루어지게 되죠?

많이 사용되는 방법이라고 한다면, home agent와 foreign agent를 두는 방법을 들 수 있습니다. 이 방식에서 mobile node는 permanent address를 가짐과 동시에, foreign agent로부터 COA라는 foreign address를 할당받게 됩니다. 여기서 COA는 foreign subnet의 주소 할당 방식을 따르게 되기 때문에, foreign agent는 COA-Permanent address 간 맵핑 정보를 가지고 있어야 합

니다.

Q9-1) 그럼 Mobile node에 대한 라우팅은 어떻게 이루어지나요?

Mobile node의 라우팅은 indirect routing과 direct routing의 두가지로 나눌 수 있습니다. indirect routing의 경우 요청하는 host가 home agent에 데이타를 전달하면, home agent는 그 데이터를 캡슐화 하여 mobile node가 속해있는 subnet의 foreign agent에게로 전달하게 됩니다. 그렇게 전달된 데이터는 foreign agent가 COA를 사용해서 해당 node로 전달을 하게 되고, 해당 노드는 그를 통해 데이터를 보낸 correspondent에게 답변을 보내는 방식입니다.

indirect routing의 경우 triangle routing problem을 겪게 되는데요, direct routing을 사용하면 이를 겪지 않고 효과적으로 routing을 할 수 있습니다. direct routing에서는 correspondent 또한 correspondent agent를 갖게 됩니다. correspondent agent는 먼저 home agent에 control message를 보내서 mobile node가 home subnet에 있는지 확인을 합니다. 만약에 해당 노드가 home subnet에 존재하지 않을 경우, correspondent agent는 foreign agent가 home agent에 알려준 COA로 다시 컨트롤 메세지를 보내 확인하는 해당 노드의 유무를 확인하는 방법을 거치게 되는데요, 이를 통해 해당 노드의 위치를 파악하게 되면 그제서야 correspondent는 데이터를 해당 노드로 전송을 하게 됩니다.

참고 문헌:

Computer Networking: A Top-Down Approach (4th E). James F. Kurose, Keith W. Ross.