

-컴퓨터 네트워크-

Application Layer II

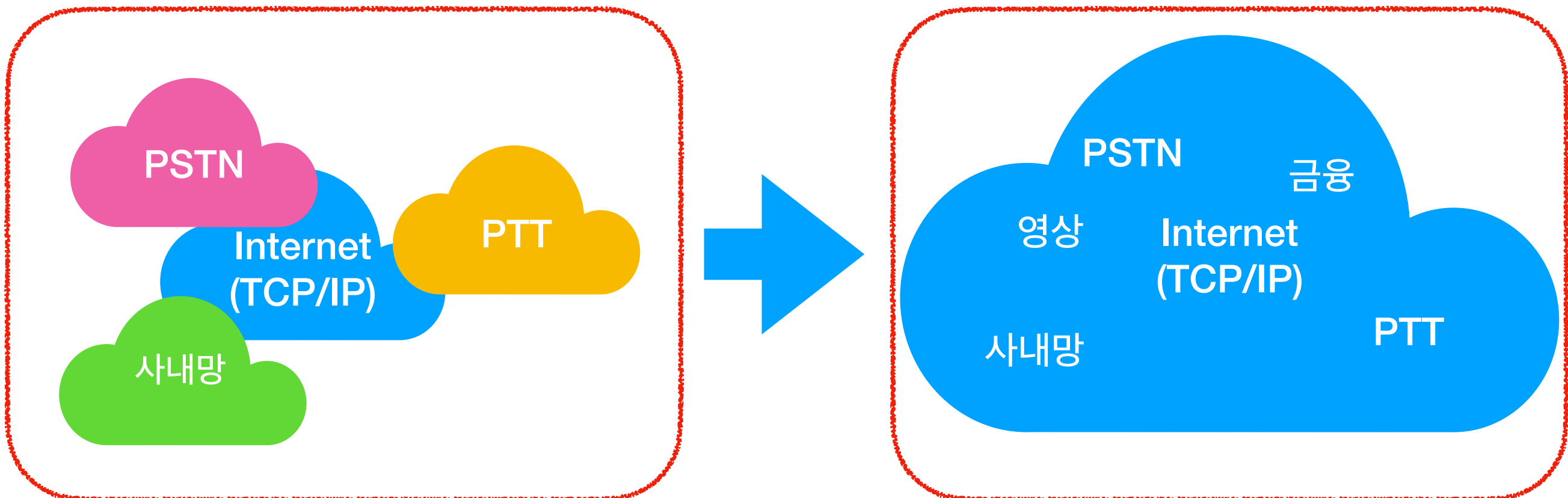
2021 Spring
Kyungseop Shin

Course Outline

- Multimedia traffic service의 특성 및 관련 전송 기법에 대해 이해
 - Multimedia traffic and QoS 개념
 - application level에서의 전송 기법 개념
 - transport/session protocol (RTP, SIP) 개념
 - Network scheduling 및 policing 개념

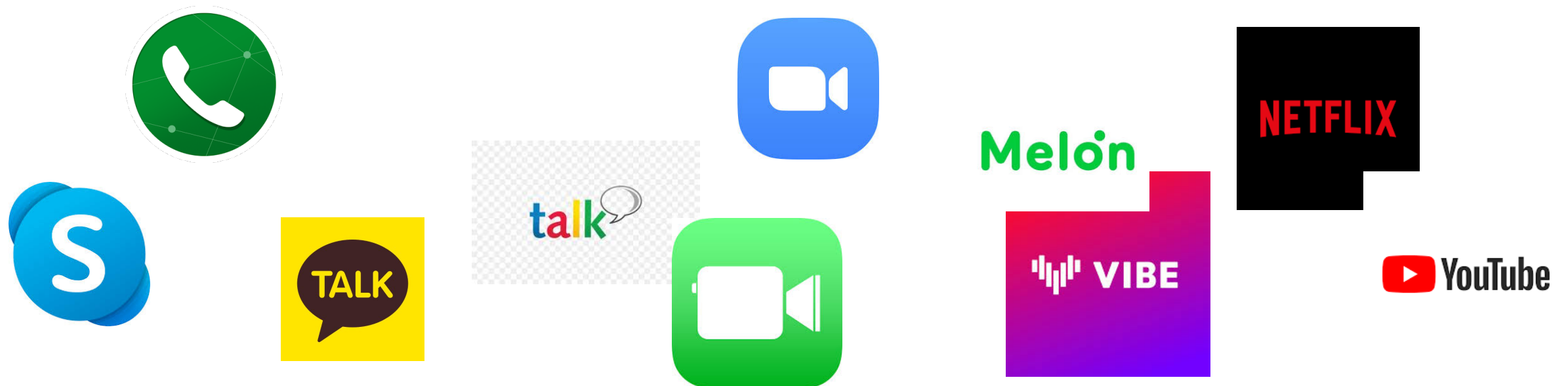
All IP

- IP를 기반으로 한 internet으로 모든 통신네트워크/서비스가 통합이 되는 개념
 - 모든 서비스는 인터넷 위에서, 인터넷은 모든 서비스의 통신 통로
 - internet이 궁극적으로 통신 서비스를 천하통일하는 현상



Multimedia Service

- All IP 추세에 따라 정보전달의 난이도가 높은 서비스가 internet을 기반으로 launching되고 있음
 - 고전적인 internet service : HTTP, mail, FTP, ...
 - Multimedia service : streaming, conversation
 - 실시간성이며, 정보 전달에 대한 요구조건 존재



Multimedia Traffic : Video

- Video : 영상 + 음성 정보
 - 기본적으로 연속된 이미지를 저장한 영상이 보통 용량이 큼
 - 영상 품질에 따라 100kbps ~ 3Mbps (HD급)로 가변
 - 사진/음악 대비 압도적으로 높은 용량을 필요로 함

	Bit rate	Bytes transferred in 67 min
Facebook Frank	160 kbps	80 Mbytes
Martha Music	128 kbps	64 Mbytes
Victor Video	2 Mbps	1 Gbyte

Multimedia Traffic : Video

- Video는 고용량만큼 전송 시 압축 기술이 동반됨
 - video compression은 기본적으로 redundancy를 줄이는 방향으로 동작
 - spatial redundancy : 공간적으로 유사한 pixel은 비슷한 정보를 가짐
 - temporal redundancy : 시간적으로 동일 pixel은 느리게 변함
 - 압축 품질에 따라 같은 영상을 multiple version으로 만들기도 함
 - 무선인터넷 속도에 따라 적응적으로 서로 다른 용량의 영상 전송 가능

Multimedia Traffic : Audio

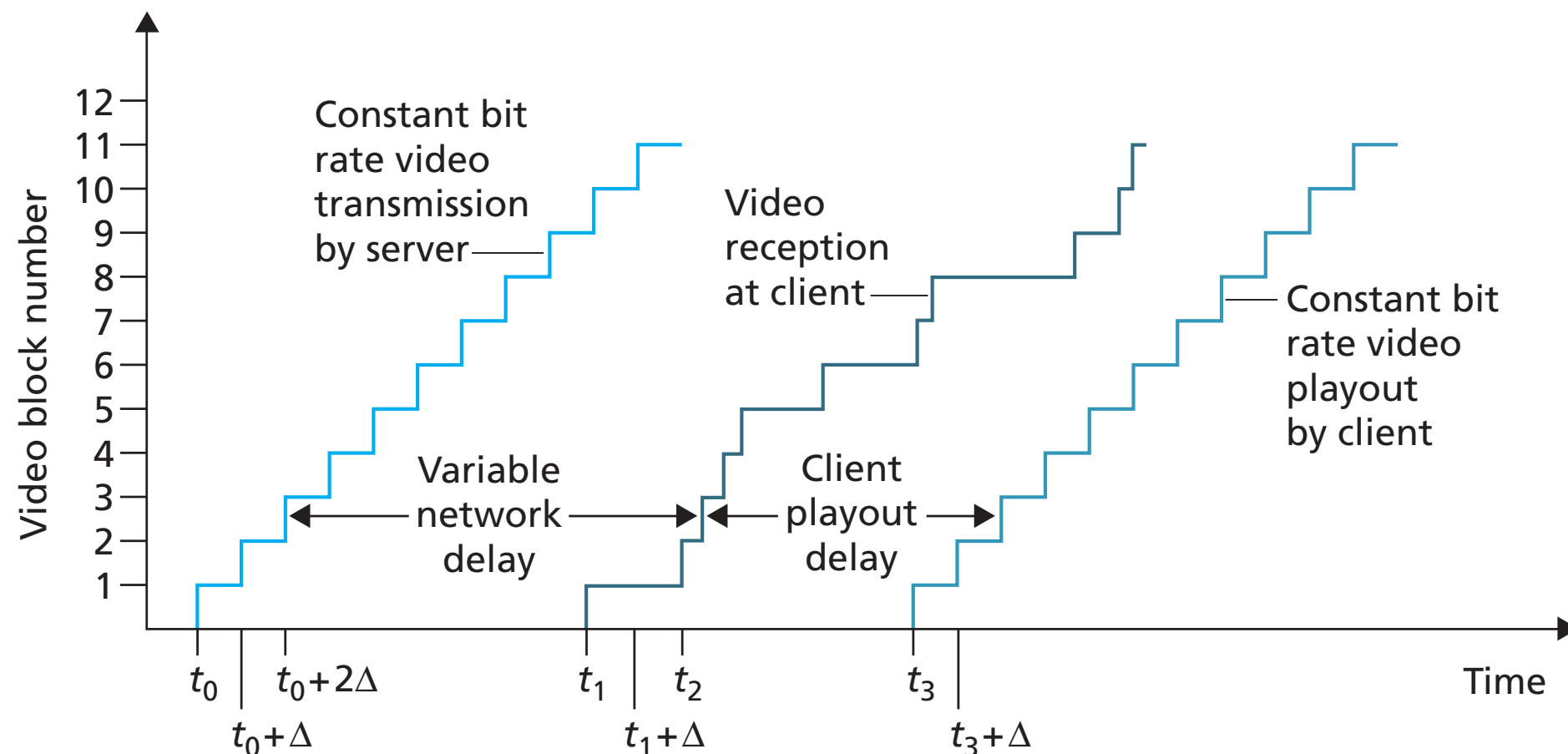
- Video에 비해 상대적으로 낮은 용량을 가짐
 - 음성 신호를 특정 sampling rate로 sampling
 - sampling된 것을 quantization해서 bit로 표현
- $\text{bit rate} = \text{sampling rate (Hz)} \times \text{bits per sample}$
 - bit rate이 높을수록 sampling rate이 높거나 quantization에 의한 왜곡이 적으므로 음성 품질이 더 우수함
- MPEG 1 layer 3 (MP3) 혹은 Advanced Audio Coding (AAC)등의 압축 기술을 활용해서 전송률을 최대한 낮춤
 - 음성의 경우 G.721, G.729, AMR 등 음성에 특화된 압축 기술 사용
- 사람들의 체감 상 Video 대비 왜곡에 더 민감함

Multimedia Traffic : Service

- Streaming stored audio and video (Netflix, youtube 등)
 - On-demand로 수신
 - streaming / interactivity / continuous playout
 - 초기 재생을 위한 지연 시간은 보통 허용이 되며, pre-buffering을 통해 이후 재생은 웬만한 network delay 변화와 관계없이 순조롭게 됨
 - 성능 관점에서는 average throughput이 관건

Multimedia Traffic : Service

- Client-side application buffering
 - 일정 양의 영상정보를 받고 재생
 - network delay 변화에도 대처 가능



Multimedia Traffic : Service

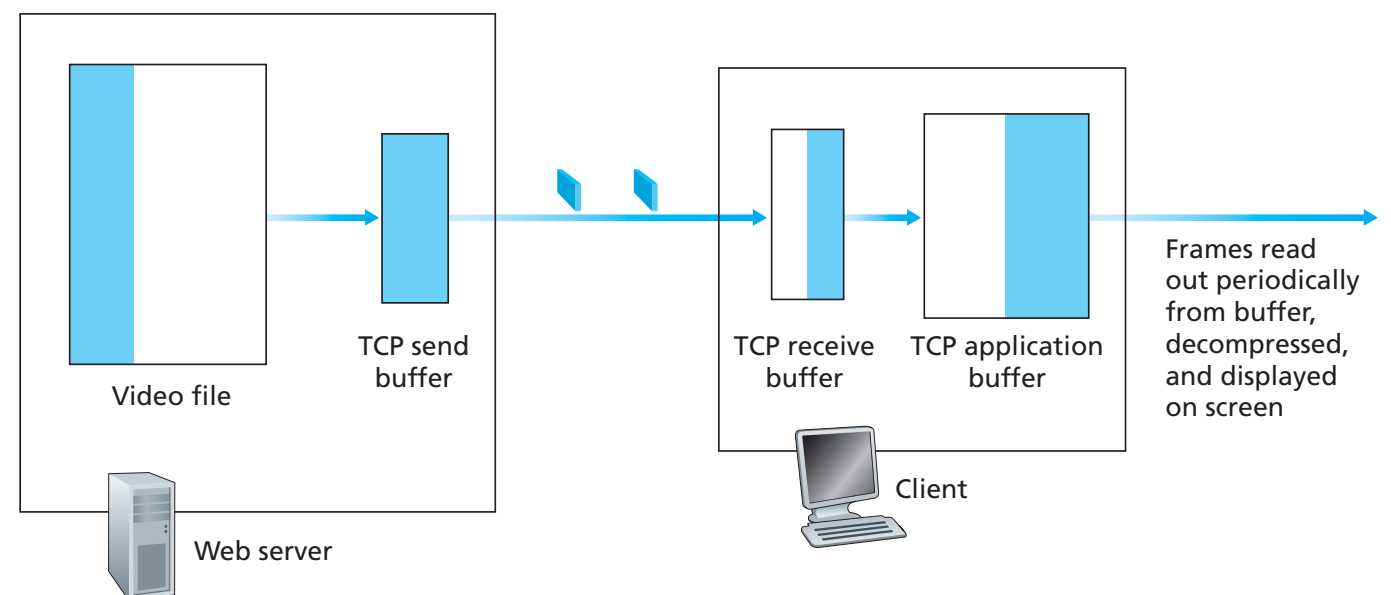
- Conversational Voice and Video
 - 실시간으로 서로 음성/영상으로 대화하는 형태
 - 내용이 미리 만들어진 것이 아닌 현재 실시간으로 생성되어 전달됨
- Delay-sensitive (delay, jitter)
 - 400ms 이상 걸리면 체감상 답답함을 느낌
- loss-tolerant : packet 몇개가 중간에 빠져도 OK

Streaming : UDP Streaming

- UDP 기반으로 해서 data를 실시간 전송하는 방식
 - Multimedia의 rate에 따라 송신 측 server에서 전송률을 조절
- Real-Time Transport Protocol (RTP)과 보통 연계
 - RTP : data encapsulation을 하며, sequence number & timestamp 정보를 실음
 - RTCP : streaming에 대한 client측 제어
- Issue
 - congestion control이 없으므로, network 환경에 따라 재생에 문제가 발생할 수 있음
 - RTCP에 의한 재생 제어에 대해 server에서 client별로 tracking을 해야 함
 - Firewall에서 UDP를 막는 경우가 많음

Streaming : HTTP Streaming

- HTTP server 형태로 multimedia 제공
 - URL에 의해 접근 (TCP connection / HTTP GET)
- Congestion control등으로 인한 TCP의 flow control 메커니즘이 streaming의 실시간 전송과 대치됨
 - 오늘날은 buffering & prefetching으로 해결
 - 재생 속도보다 빠르게 download해서 미리 client측에 저장

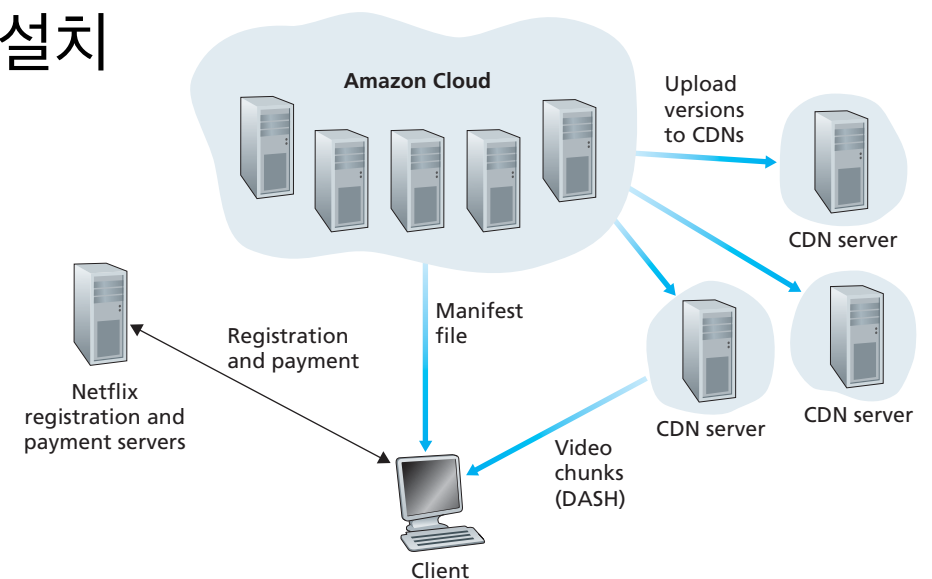


Streaming : Contents Distribution Network

- Video는 network load의 주 원인이 되어가고 있음
 - 영상 정보 자체가 size가 큼
 - 영상 서비스에 대한 사용자가 날이 갈수록 늘어나고 있음 (youtube, Netflix 등)
- Single massive data center 기반의 network streaming 방식
 - network상에서 client와의 거리가 일반적으로 큼
 - 여러 ISP를 거쳐야 하는 경우가 많으며, 이중 한 ISP의 용량이 적으면 bottleneck이 됨
 - 자주 보는 영상은 필요 이상으로 전송 빈도수가 높아짐
 - 고장나면 전체 service 이용불가로 이어짐 (fail-safety 측면에서 최악)

Streaming : Contents Distribution Network

- 따라서 보통 streaming service는 여러 대의 server를 분산적으로 두고 제공
 - 지역적으로 고르게 분포되며, 각종 video 정보를 복제해서 보유
 - private CDN (Google) vs. third-party CDN (Netflix)
- CDN 구축에는 두 가지 상반된 원칙이 존재함
 - Enter deep : 각 ISP network 내에 설치
 - Bring home : ISP 주변에 전용 회선으로 연결해서 설치



Streaming : Contents Distribution Network

- CDN을 둘러싼 기업 간 갈등
 - Netflix vs. SKB
 - Netflix vs. LGU+

‘망 이용료’ 갈등 넷플릭스, SK브로드밴드에 소송

남혜현 | 2020년 4월 14일

소송의 발단은 통신사의 망 이용료 지급 요청이다. 트래픽이 늘어난 만큼 회선을 증설해야 하니 넷플릭스가 비용을 지불해야 한다는 주장이었다.

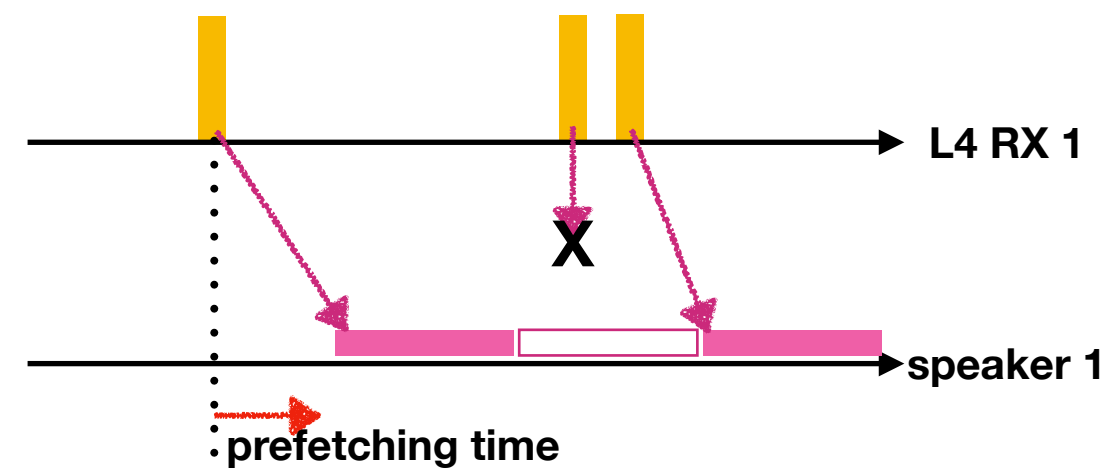
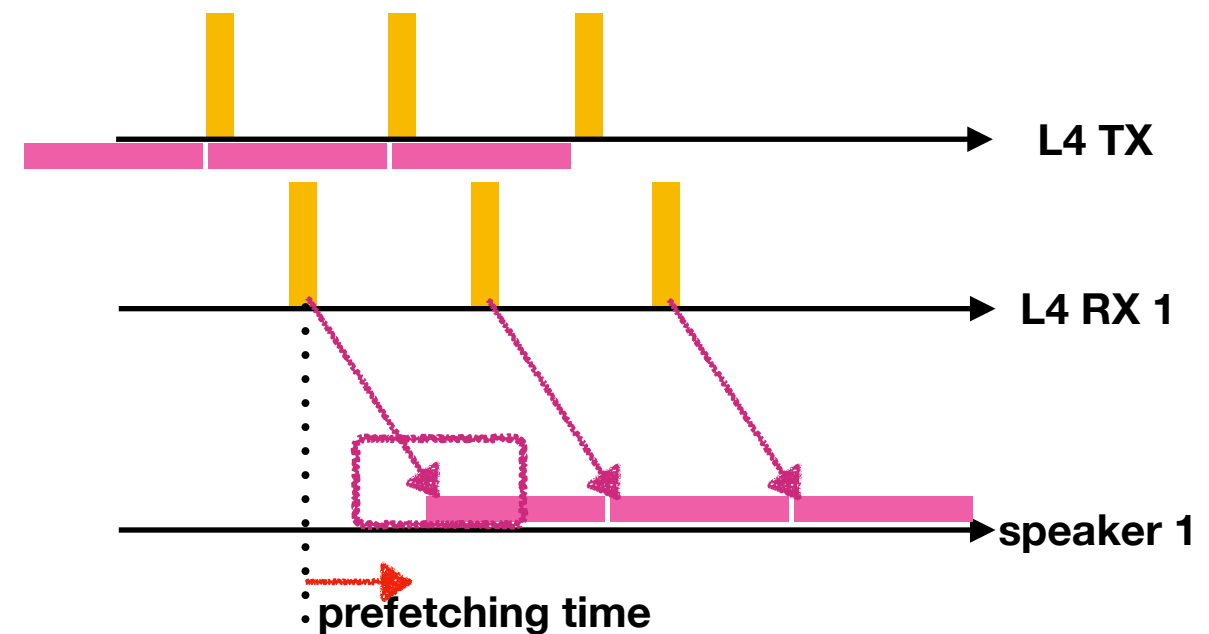
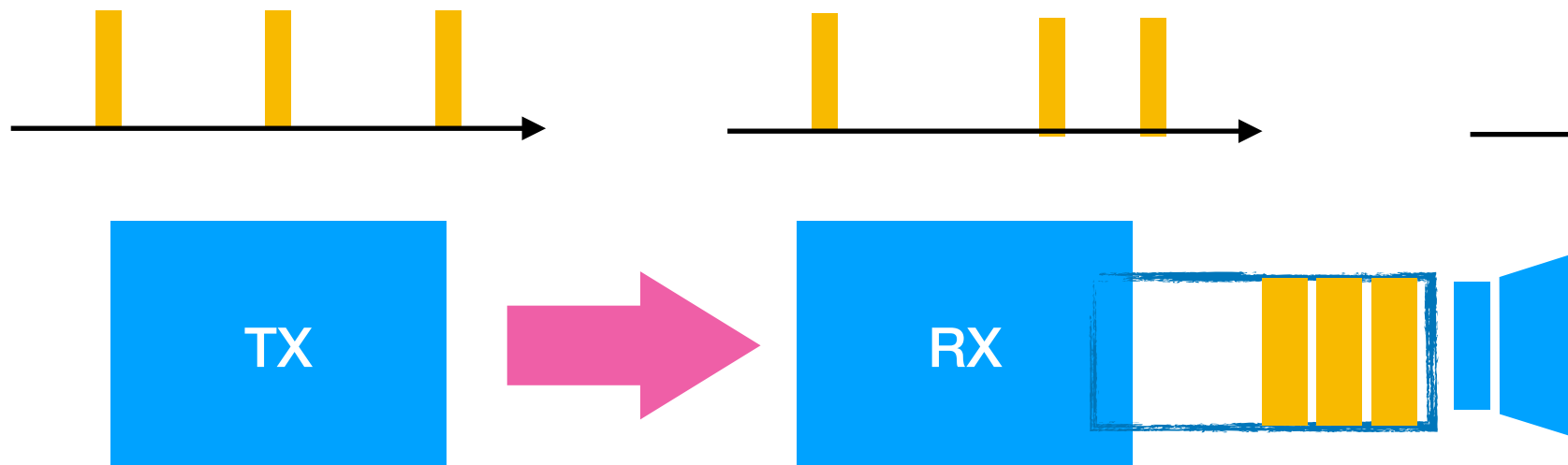
협상에서 넷플릭스 측은 국내에 캐시서버를 설치, 트래픽의 수요를 미리 예측해 관련 콘텐츠를 가져다 두는 ‘오픈 커넥트’를 무료로 제공하겠다고 제안했다. 트래픽이 적은 새벽 시간대에 한국에서 많이 보는 콘텐츠를 미리 가져다놓는 방안을 쓰기 때문에 국제 회선의 증설이 필요가 없다는 이야기다. 이는 넷플릭스와 협력하고 있는 LG유플러스, LG헬로비전, 딜라이트 등이 선택해 활용하고 있는 방안이다.

Voice over IP

- Conversational service에서 가장 기본적인 음성 서비스에 해당
 - IP 상에서 voice를 전달하는 일련의 application protocols
 - Video만큼 network load가 크진 않음
 - 고전적인 서비스이긴 하지만 delay 관점에서 상당히 민감한 서비스임
 - delay 관련 성능 여러가지 중 한가지만 열화가 일어나도 체감상 service quality가 크게 떨어짐

Voice over IP

- VoIP example : 8kHz sampling, 8bit quantization
 - 초당 bit 생성률 : $8 \times 8000 = 64\text{kbps}$
 - 20ms마다 bit를 모아서 data로 전송
 - $64\text{kbps} \times 0.02 = 160\text{ Byte}$ 를 20ms마다 전송
 - 수신측 재생 상황
 - 20ms마다 160B data가 일정하게 들어와야 함

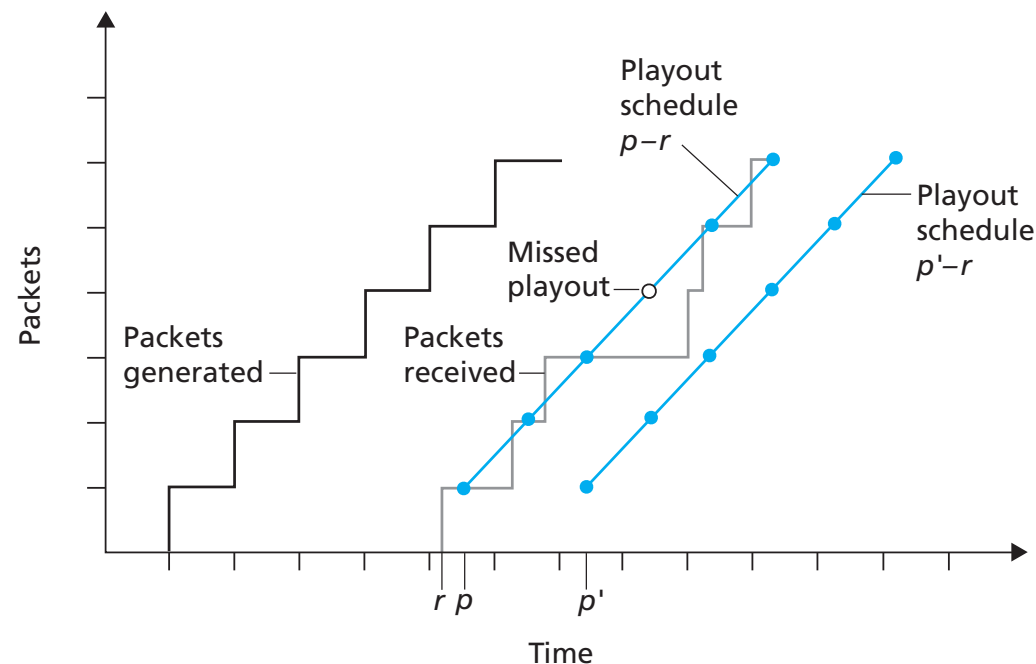


Voice over IP

- VoIP에 영향을 미치는 성능 요인
 - Packet loss : 일부 허용될 수 있음
 - FEC 혹은 interpolation으로 극복 가능
 - End-to-end delay : 300~400ms 이하는 OK
 - 평균적인 인간이 체감하는 최소 반응시간 길이
 - Packet jitter : packet inter-arrival time의 변화량 (분산으로 계산)
 - 일정 수준 이하로 유지가 되어야 함

Voice over IP

- 유무선 통신에서는 jitter는 불가피하므로
- 수신 재생 측에서 de-jittering을 위한 buffer를 운영
- 음성 재생 시작 시점에서 수신 직후보다는 좀더 marginal time을 두고 나서 시작 (playout delay)
- sequence/timestamp 정보를 통해 재생 조절



Real-Time Protocol (RTP)

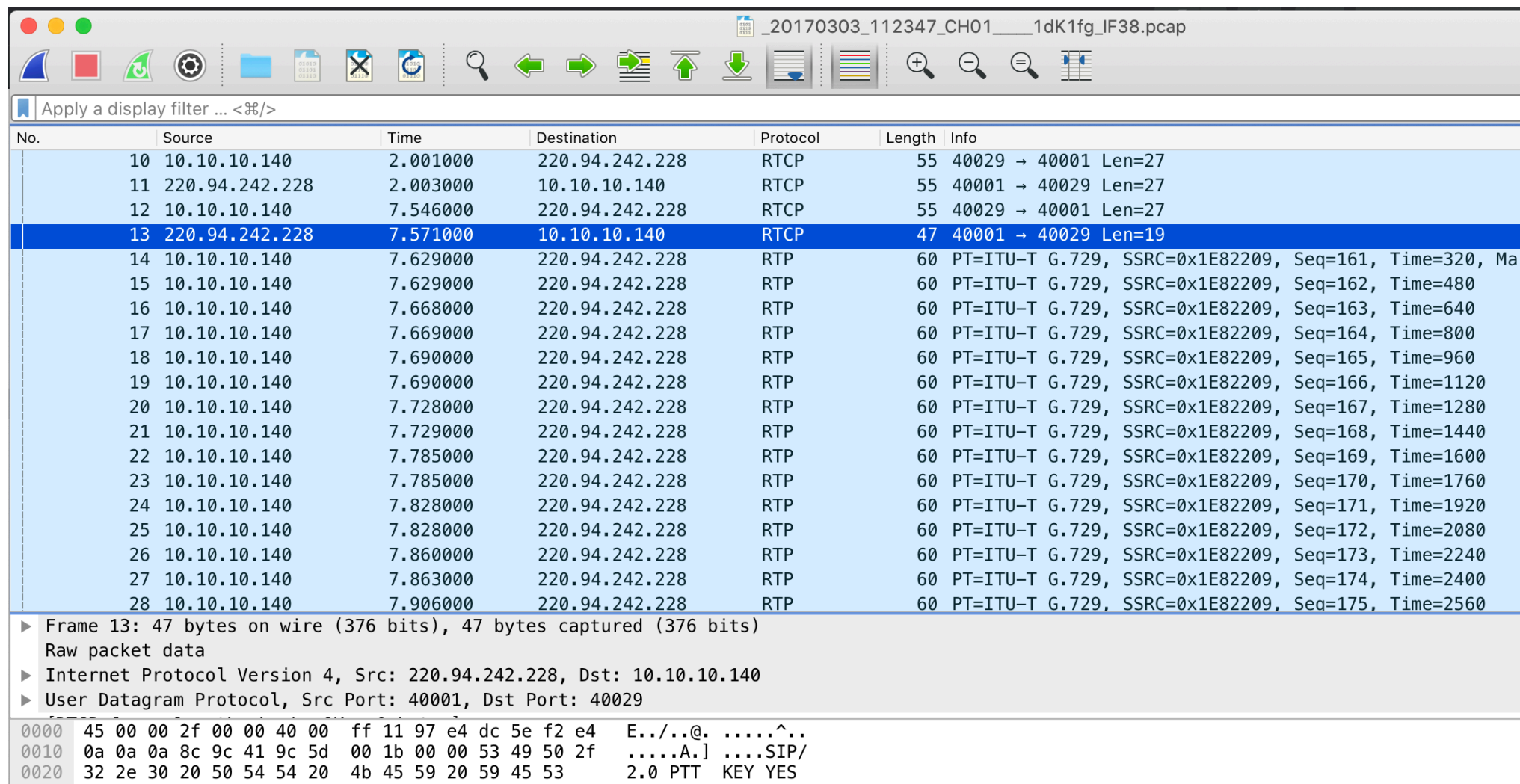
- multimedia data에 대한 추가적인 재생 정보를 담기 위해 data packet을 encapsulation하는 protocol
 - UDP 바로 위에서 동작하며, encapsulated PDU가 곧 UDP의 SDU가 됨
 - RTP header에 부가 정보가 포함됨
 - multimedia type, SN, timestamp, source ID

Payload type	Sequence number	Timestamp	Synchronization source identifier	Miscellaneous fields
--------------	-----------------	-----------	-----------------------------------	----------------------

Payload-Type Number	Audio Format	Sampling Rate	Rate
0	PCM μ -law	8 kHz	64 kbps
1	1016	8 kHz	4.8 kbps
3	GSM	8 kHz	13 kbps
7	LPC	8 kHz	2.4 kbps
9	G.722	16 kHz	48–64 kbps
14	MPEG Audio	90 kHz	—
15	G.728	8 kHz	16 kbps

Real-Time Control Protocol (RTCP)

- RTP와 더불어 같이 활용되는 control-plane protocol
- RTP에 대한 흐름 제어 및 각종 통계정보 제공



The image shows a Wireshark packet capture window for the file `_20170303_112347_CH01____1dK1fg_IF38.pcap`. The display filter is `<All>`. The packet list shows packets 10 through 28. Packets 10-13 are RTCP packets, and packets 14-28 are RTP packets. Packet 13 is selected, showing its details and raw data.

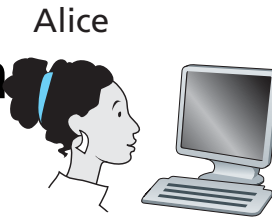
No.	Source	Time	Destination	Protocol	Length	Info
10	10.10.10.140	2.001000	220.94.242.228	RTCP	55	40029 → 40001 Len=27
11	220.94.242.228	2.003000	10.10.10.140	RTCP	55	40001 → 40029 Len=27
12	10.10.10.140	7.546000	220.94.242.228	RTCP	55	40029 → 40001 Len=27
13	220.94.242.228	7.571000	10.10.10.140	RTCP	47	40001 → 40029 Len=19
14	10.10.10.140	7.629000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=161, Time=320, Ma
15	10.10.10.140	7.629000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=162, Time=480
16	10.10.10.140	7.668000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=163, Time=640
17	10.10.10.140	7.669000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=164, Time=800
18	10.10.10.140	7.690000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=165, Time=960
19	10.10.10.140	7.690000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=166, Time=1120
20	10.10.10.140	7.728000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=167, Time=1280
21	10.10.10.140	7.729000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=168, Time=1440
22	10.10.10.140	7.785000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=169, Time=1600
23	10.10.10.140	7.785000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=170, Time=1760
24	10.10.10.140	7.828000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=171, Time=1920
25	10.10.10.140	7.828000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=172, Time=2080
26	10.10.10.140	7.860000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=173, Time=2240
27	10.10.10.140	7.863000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=174, Time=2400
28	10.10.10.140	7.906000	220.94.242.228	RTP	60	PT=ITU-T G.729, SSRC=0x1E82209, Seq=175, Time=2560

Frame 13: 47 bytes on wire (376 bits), 47 bytes captured (376 bits)
Raw packet data
Internet Protocol Version 4, Src: 220.94.242.228, Dst: 10.10.10.140
User Datagram Protocol, Src Port: 40001, Dst Port: 40029

```
0000 45 00 00 2f 00 00 40 00 ff 11 97 e4 dc 5e f2 e4 E../..@. ....^..
0010 0a 0a 0a 8c 9c 41 9c 5d 00 1b 00 00 53 49 50 2f .....A.] ....SIP/
0020 32 2e 30 20 50 54 54 20 4b 45 59 20 59 45 53 2.0 PTT KEY YES
```

Session Initiation Protocol (SIP)

- Multimedia 전송을 위해 connection management를 해주는 프로토콜



Alice

167.180.112.24



Bob

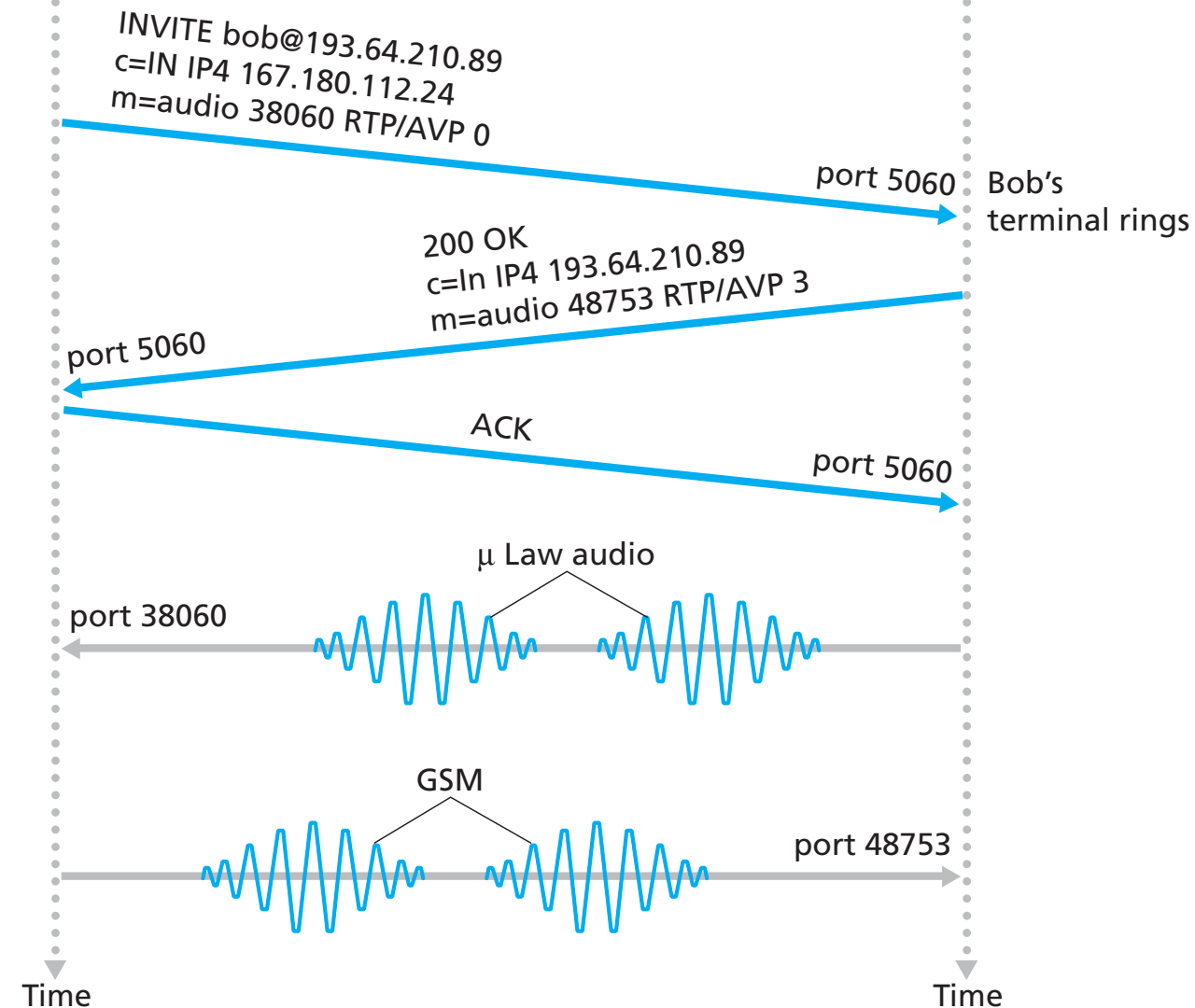
193.64.210.89

- IP 상에서 동작하면서 호 연결 서비스 제공

- 상대방 IP address에 대한 파악/관리 기능 포함

- HTTP와 유사한 ASCII 기반의 메시지 형태

- 호 연결 명령과 더불어 option parameter 를 같이 전송할 수 있음

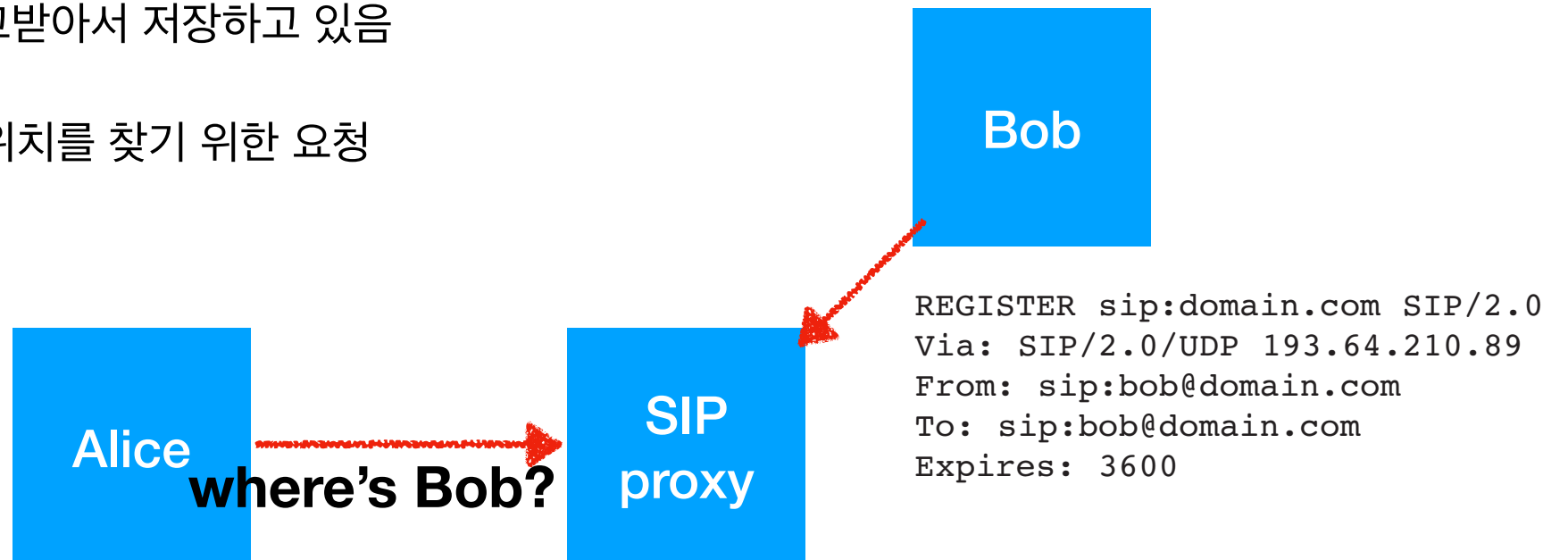


Session Initiation Protocol (SIP)

- SIP address : ID + 위치 정보
 - sip: xxxx 형태로 표현됨
 - IP address를 포함하고 있으면 곧바로 IP 전송이 가능하나, 보통 이런 유형이 아님
 - DHCP, mobile IP 등을 많이 활용하는 관계로, 요즘은 상대방의 IP가 고정적이지 않음
 - email 형태의 주소를 많이 활용 (SIP proxy를 통한 IP mapping 필요)
 - Bob의 위치를 보고받아서 저장하고 있음
 - INVITE - 상대방 위치를 찾기 위한 요청

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885
```

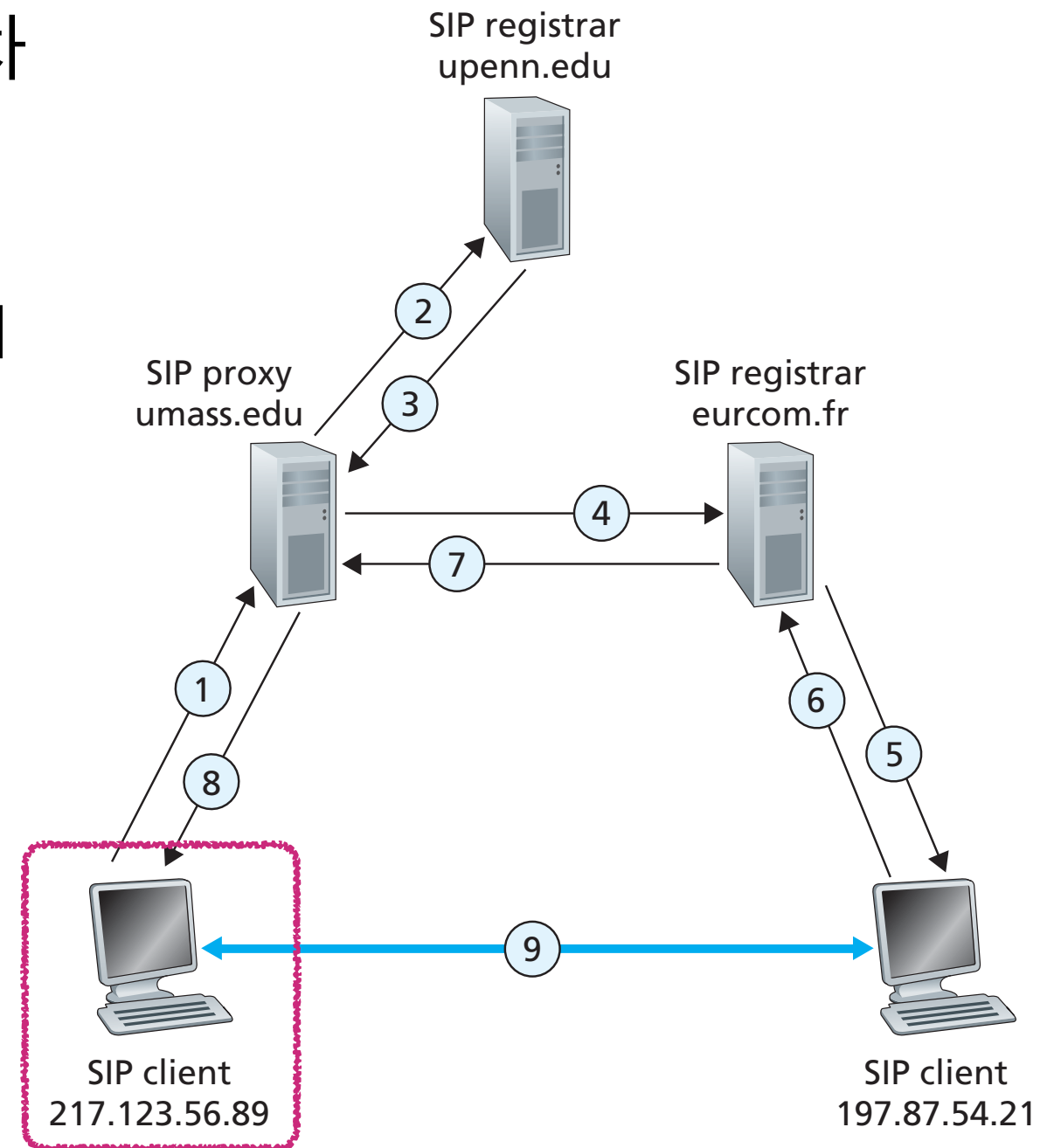
```
c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```



```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

Session Initiation Protocol (SIP)

- SIP를 통한 session initiation 절차 예시
- SIP proxy -> registrar -> end client
- 서로의 IP addr를 알면 이후 media 정보는 중간 server를 거치지 않고 직접적으로 주고 받음



SIP Message

- SIP INVITE message
 - session 시작 시 호 설정 요청을 위해 전달
 - 호 요청자 관련 정보 및 parameter

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885
```

```
c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Quality of Service

- Network 측면에서 service가 요구하는 성능의 품질

- data rate
- priority
- latency
- loss rate
- jitter

QCI	Resource Type	Priority	Packet Delay Budget (NOTE 1)	Packet Error Loss Rate (NOTE 2)	Example Services
1 (NOTE 3)	GBR	2	100 ms	10^{-2}	Conversational Voice
2 (NOTE 3)		4	150 ms	10^{-3}	Conversational Video (Live Streaming)
3 (NOTE 3)		3	50 ms	10^{-3}	Real Time Gaming
4 (NOTE 3)		5	300 ms	10^{-6}	Non-Conversational Video (Buffered Streaming)
5 (NOTE 3)	Non-GBR	1	100 ms	10^{-6}	IMS Signalling
6 (NOTE 4)		6	300 ms	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7 (NOTE 3)		7	100 ms	10^{-3}	Voice, Video (Live Streaming) Interactive Gaming
8 (NOTE 5)		8	300 ms	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
9 (NOTE 6)		9			

Network-level Multimedia Delivery

- L4 이하에서 multimedia data에 대한 전송 철학/개념
 - 보통은 best-effort service임 (최선을 다하겠음)
 - 모든 traffic들이 network에서는 동일하게 취급이 됨
 - 흔히 application-level에서 multimedia 전송 기법을 고려

- Differentiated service : 서로 특성의 traffic이 class로 구분되어 priority가 다르게 취급됨

- Per-connection Quality-of-Service Guarantees : 연결 별로 서로 다른 요구 사항이 존재하며, 이에 맞게 network에서 전달

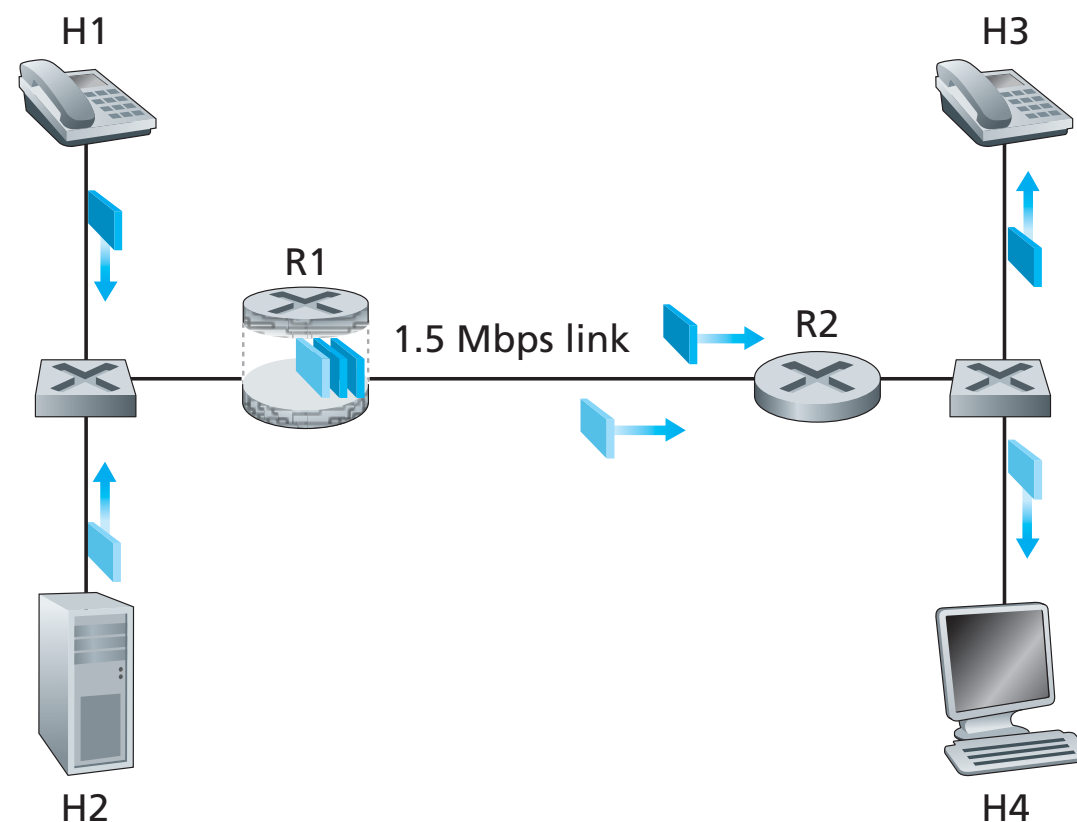
Approach	Granularity	Guarantee	Mechanisms	Complexity	Deployment to date
Making the best of best-effort service.	all traffic treated equally	none, or soft	application-layer support, CDNs, overlays, network-level resource provisioning	minimal	everywhere
Differentiated service	different classes of traffic treated differently	none, or soft	packet marking, policing, scheduling	medium	some
Per-connection Quality-of-Service (QoS) Guarantees	each source-destination flows treated differently	soft or hard, once flow is admitted	packet marking, policing, scheduling; call admission and signaling	light	little

Best-Effort Network

- 최선을 다하는게 미덕이며, 그 이상에 대한 보장은 없음
 - Multimedia application의 strict requirement를 꼭 만족시키려 하지 않고, network 성능을 최우선으로 고려
 - 모든 multimedia connection들을 만족시키려 하면 network는 부하가 커지고 congestion 발생
- Network 용량이 수요보다 충분히 크도록 network 설계가 이루어져야 함
 - bandwidth provisioning : 주어진 network topology하에서 특정 link가 요구되는 capacity
 - network dimensioning : end-to-end 성능 요구사항을 만족시키기 위한 network 설계 문제 (router 위치 및 연결, 각 link의 capacity 결정 등)
 - end point에서의 traffic 요구 모델 정립, 성능 요구사항 정의 및 단대단 성능 예측 등이 필요
 - + 경제성 조건

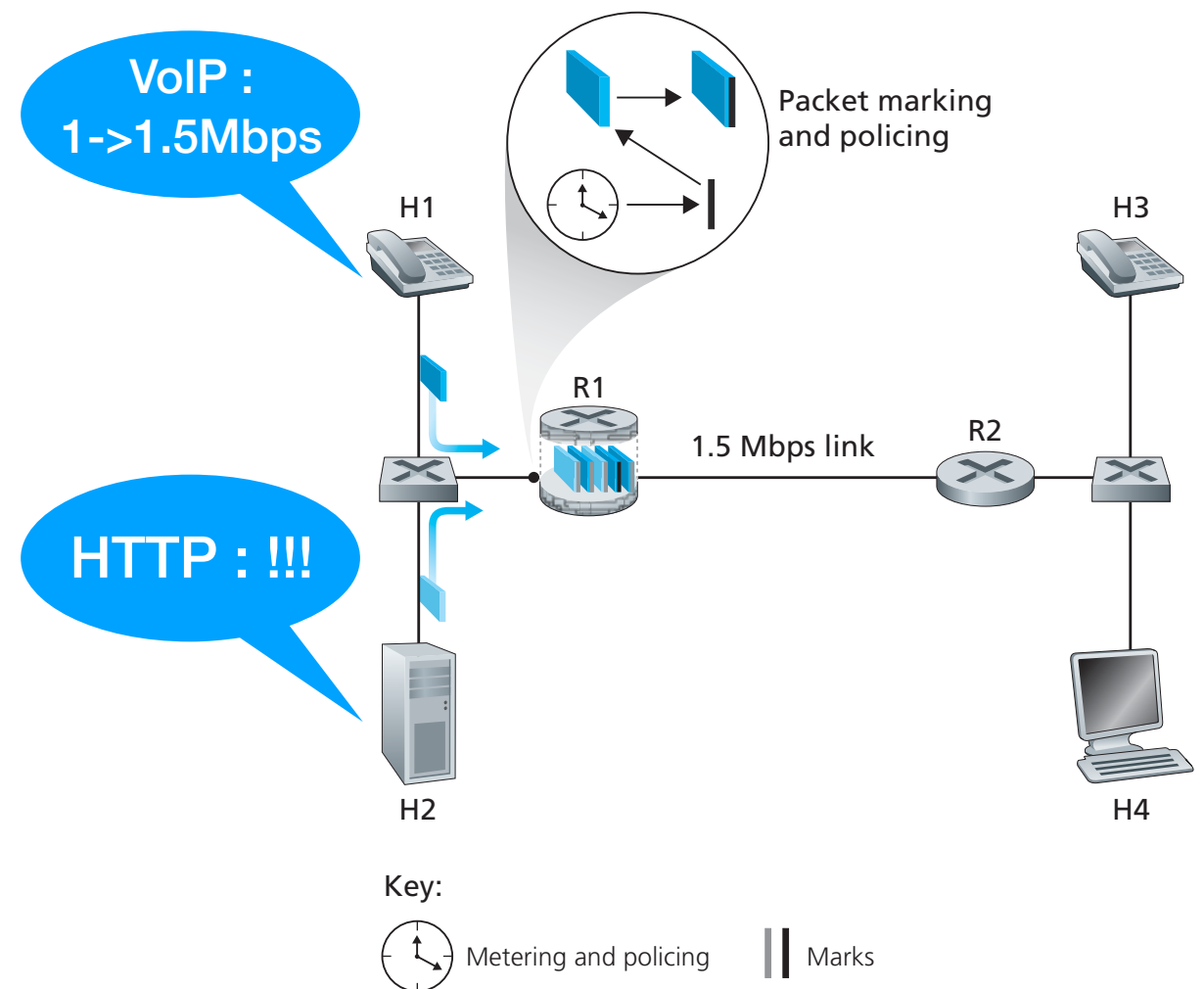
Multiple Classes of Service

- 다양한 application service에 대해 'class'로 분류해서 전달
 - class에 따라 전달에 대한 우선순위를 다르게 둠
- Packet marking : IP header의 type-of-service (ToS)로 인식



Multiple Classes of Service

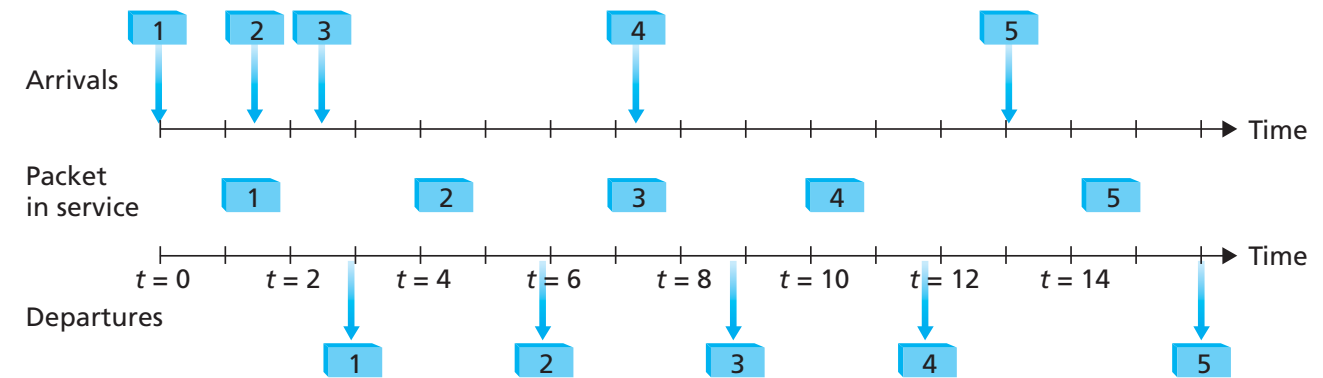
- Traffic isolation : 한쪽 서비스가 다른 한쪽 서비스에 품질 측면에서 영향을 일으키지 않도록 제어
- Traffic policing : 각 service별로 제한을 둠
- Efficient resource usage : 주어진 Network resource는 아끼지 말고 최대한 다 쓰는 것이 좋음



Scheduling Mechanism

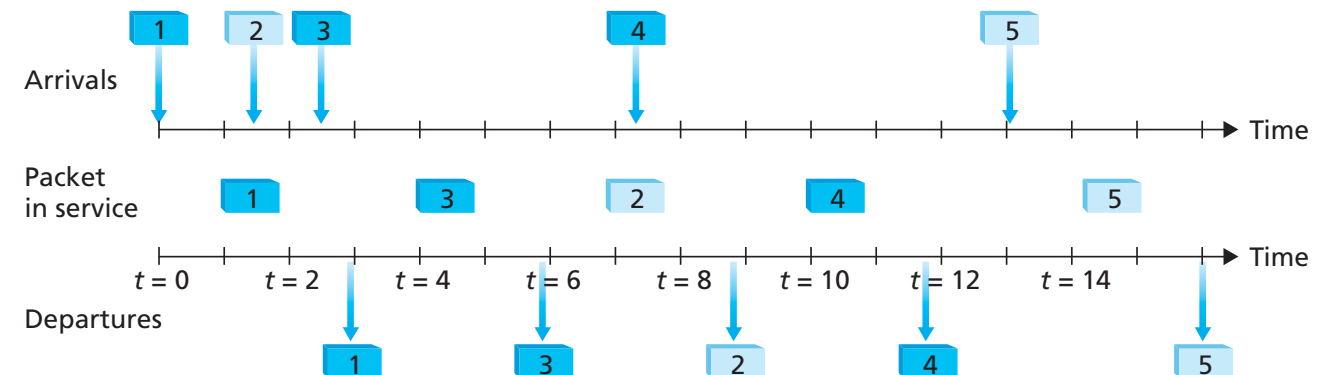
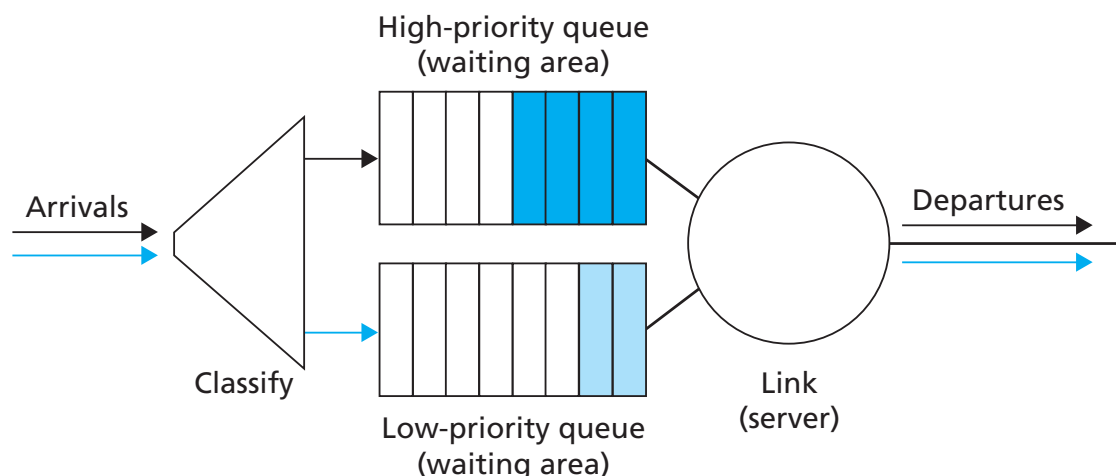
- router에서 어떤 packet을 먼저 내보낼지 정하는 과정

- First-In-First-Out (FIFO) :



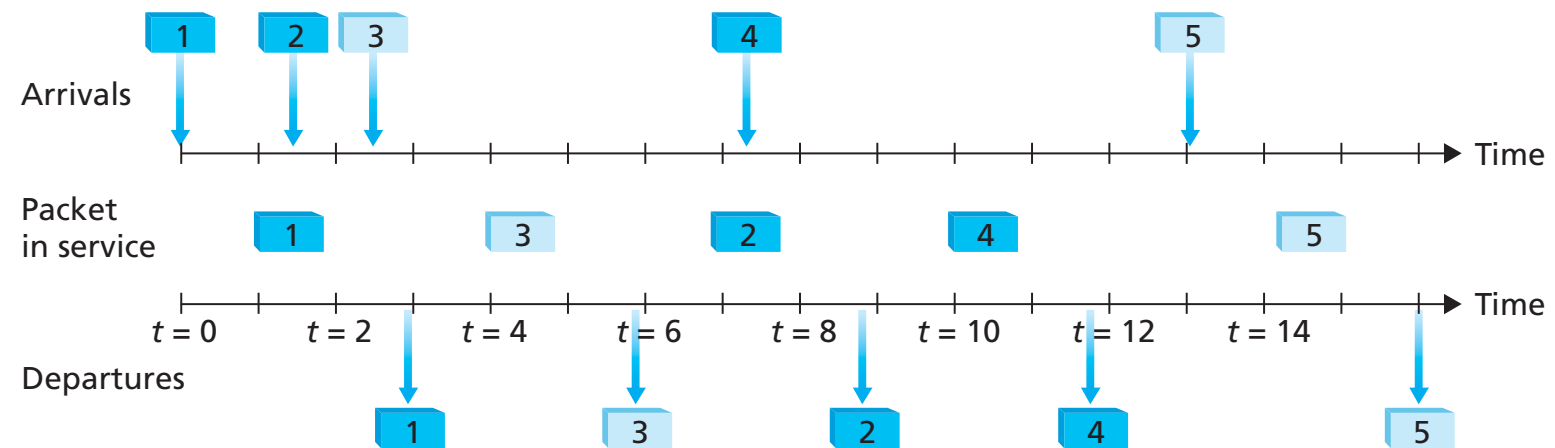
- Priority Queuing :

- class별로 queue를 별도로 둬



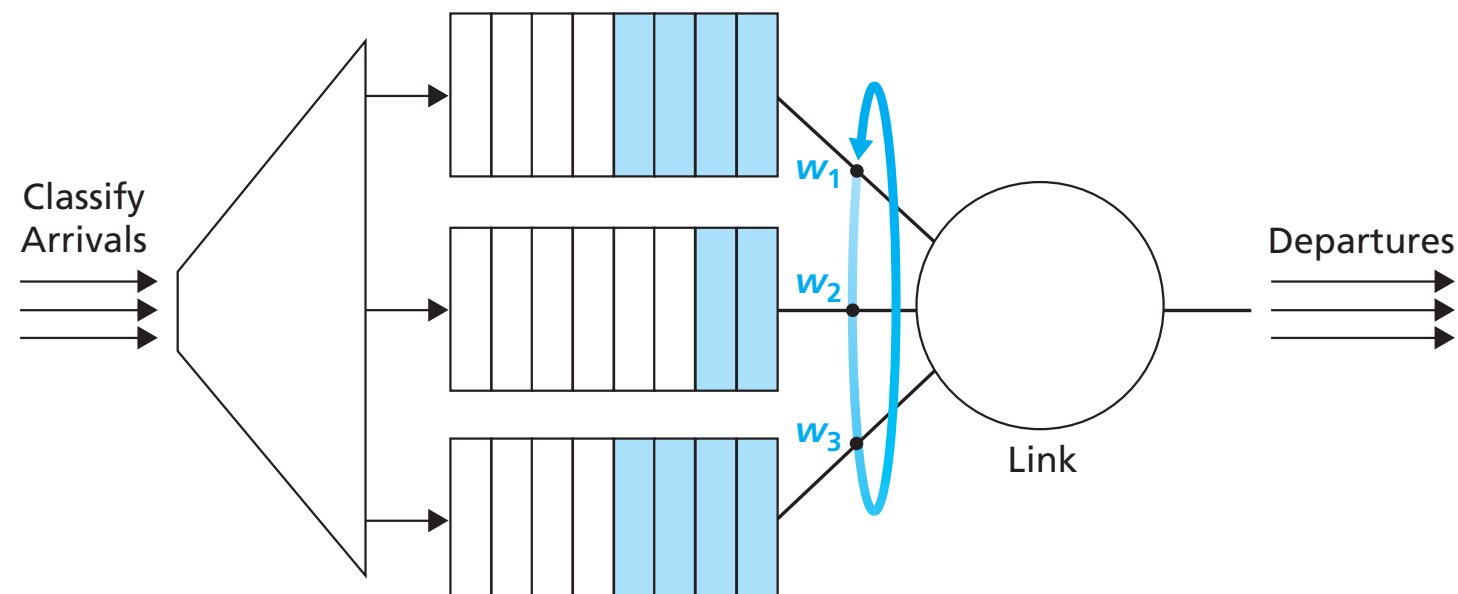
Scheduling Mechanism

- Round Robin
- Queue별로 공평하게



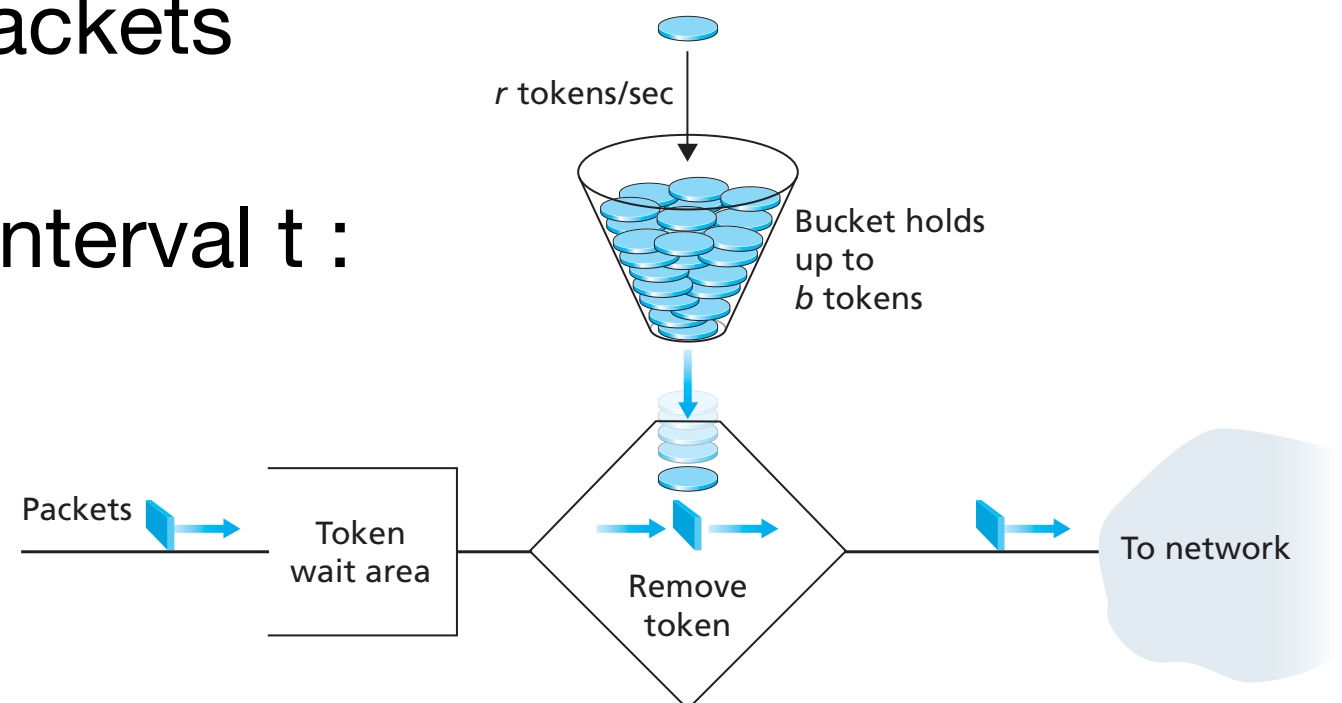
- Weighted Fair Queuing : Queue별로 weight에 맞게 공평하게

$$\frac{w_i}{\sum w_j}$$



Policing : Leaky Bucket

- bucket에 담긴 token 양에 따라 packet 전송을 허용
 - bucket은 r 의 비율로 token이 쌓이며, b 개 이상은 담지 못함
 - 1개의 packet을 보내려면 token 1개를 써야 함
- maximum burst size : b packets
- maximum packet in time interval t :
 - $rt+b$



Policing + Scheduling

- Leaky bucket + Weighted Fair Queuing
- bucket내 token에 따라 queue에 packet 축적
- 축적된 packet을 weight에 따라 순차적으로 전송

