

# QoS(Quality Of Service )

---



# Part 1.

---

**QoS( Quality Of Service )**



# 01 QOS

## QoS 개념

qos란 다른 응용 프로그램, 사용자, 데이터의 흐름의 우선순위를 정하거나 데이터 전송에 특정 수준의 성능을 보장하기 위한 서비스의 품질이다

중요한 패킷(voip, iptv) 같은 것들을 망에 혼잡이 생겼을 때 먼저 보내주는 역할이라고 할 수 있다. 한마디로 서비스들의 중요순위를 매겨 그것들을 교통 정리 하는 것이 qos 이다

## QoS 종류

qos에는 크게 2가지 패킷이 분류가 되는데 먼저 첫 번째는 Multi-field(MF) Classification 이라고 한다. 이 패킷은 패킷 헤더 내에 있는 다양한 필드들을 보고 중요한 패킷인지 중요하지 않은 패킷인지 판단하고 분류한다.

두 번째는 Behavior-Aggregate(BA) Classification 이다. 이건 패킷에 이미 qos 정보가 포함되어 있어 간단히 그 필드만 보면 중요한지 아닌지를 판단할 수 있는 분류 방법이다. 이더넷 패킷의 경우 802.1p COS 필드이고, IP 패킷의 경우는 dscp 이다. 하나하나씩 알아보자

# 01 QOS

## 802.1p 우선순위

우선 순위	트래픽 종류
0 (가장 낮음)	최적 조건
1	배경
2	표준 (예비)
3	최적의 로드 (사업 결정)
4	제어 받는 로드 (스트리밍 멀티미디어)
5	소리 및 영상 (인터랙티브 미디어 및 소리) [100ms 이하의 레이턴시와 불안성]
6	계층 3의 네트워크 제어 보존 트래픽 [10ms 이하의 레이턴시와 불안성]
7 (가장 높음)	계층 2의 네트워크 제어 보존 트래픽 [가장 낮은 레이턴시와 불안성]

## 802.1p COS 필드

일단 802.1q 는 무엇이나 하면 VLAN Tag 이고 802.1p는 우선순위이다 (q 랑 p를 구별하자)  
802.1p (COS) 는 3비트로 8 단계의 우선순위 서비스를 할 수 있다 .

COS 는 Canonical Format Identifier 의 줄임말로 IEEE802.1q를 사용하는 이더넷 헤더 내 3비트의 필드로 0~7 사이의 값을 통해 전송 상의 우선순위를 나타 낸다.

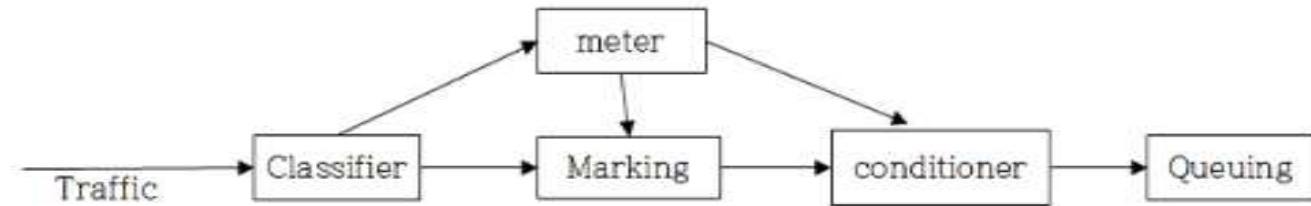
그러므로 802.1p 와 CoS 는 같은 말이다.

**\*중요\***

이더넷 망에서는 802.1p를 통해 패킷의 중요도를 판단할 수 있다.  
우선순위는 0 이 가장 낮고(꼴찌) 7이 가장 높다(1등)

# 01 QOS

## QOS 매커니즘



트래픽이 들어오면 분류자(Classifier)에 의해 비슷한 성질을 가진 트래픽끼리 묶어 클래스로 분류한다.

분류자를 통과한 후 미터를 통하여 트래픽 플로우를 측정한다.

미터에 의해 측정된 트래픽 플로우를 사전에 약속한 트래픽 특성과 비교하여 결과에 따라 마킹한다.

마킹된 패킷은 컨디셔너를 통해 약속된 트래픽 대역폭에 맞추어 조절한다.

컨디셔너를 통과한 패킷은 맨 처음 분류자에서 나눈 클래스에서 맞는 큐에 저장되어 스케줄링에 따라 출력된다.

분류자 : 해당 패킷을 클래스 별로 구분하는 단계(ACL, class-map)

마킹 : 구분한 클래스에 대해 Policy-map을 통해 마킹을 해 우선순위를 지정

컨디셔너 : 버퍼를 이용하여 조절하는 Shaping 과 Drop을 이용한 Policing (두 가지 방법)

큐잉 : 큐잉 기법으로 FIFO, WFQ, CQ, PQ, CBWFO, LLQ 등이 있다.

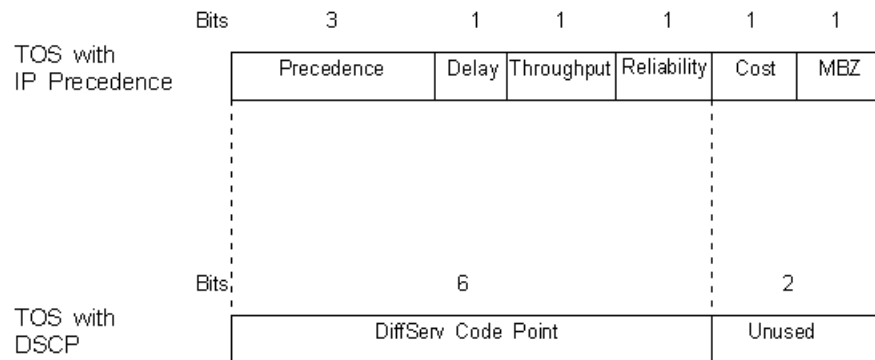
# 01 QOS

## QoS 매커니즘

SLA 는 service Level Agreement 로 VOIP나 멀티미디어 서비스는 높은 수준의 Qos가 보장되어야 하고, 사용자의 요구에 맞춰 서비스를 제공하기 위해 동적인 서비스 협약이 필요하다. SLA 는 이러한 서비스를 제공하는 데 있어 서비스 품질에 대한 판단 기준이 된다.

## QoS Marking

.DiffServ 마킹 기법은 예전 방식은 IP Precedence 방식과 이를 보완한 DSCP 가 있다.



## QoS Marking

IP Precedence에서 DSCP 로 변경된 것을 알 수 있을 것이다.

TOS는 우선순위를 나타내는 3비트의 Precedence 및 4비트의 서비스 유형 지정 비트, 그리고 사용되지 않은 1 비트이다. 현재는 DSCP 필로 정의되어 있다.

# 01 QOS

## QoS Marking

### IP Precedence

TOS 필드의 상위 3비트를 사용한다. 숫자가 클수록 우선순위가 높다.

#### *Precedence Values*

Value	Description
000 (0)	Routine or Best Effort
001 (1)	Priority
010 (2)	Immediate
011 (3)	Flash - mainly used for Voice Signaling or for Video.
100 (4)	Flash Override
101 (5)	Critical -mainly used for Voice RTP.
110 (6)	Internet
111 (7)	Network

6과 7은 인터넷용과 네트워크용으로 예약되어있다. 즉, 사용자가 임의로 마킹하지 않고, 특정한 프로토콜에 예약되어 써지고 있다는 것을 알 수 있다.

실제로 우선권의 값 중 가장 높은 것은 5 (critical) 이다.

0 은 BE(best effort) 를 뜻한다. BE는 ‘일단 최선을 다해서 보내긴 보내보겠다.’는 의미로 제일 낮은 등급의 우선 순위 값이다. 이때 6, 7에서 예약되지 않고, Qos 지정을 받지 않은 모든 패킷들은 이 값을 가진다

따라서 5는 실질적인 최우선 순위 값으로 voip 와 같이 가장 우선적으로 처리되어야 할 패킷들에 마킹 시켜주게 된다.

# 01 QOS

## QoS Marking

Dscp

Differentiated Services Code Point 의 약자이며 RFC 2474에서 상위 7비트를 DSCP 라고 정의하였고, 각 값마다 Qos 의 중요도에 대한 표준을 잡아놓았다.

그래서 라우팅 망에서는 이 DSCP를 통해 패킷의 중요도를 판단할 수 있다.

6비트는 위와 같은 형식으로 이루어져 있으며, 앞 3비트로 클래스를 나누고 나뉜 클래스에 대해 Drop 가능성을 3가지로 나누어 표현



DSCP 값 : 64개 코드 포인트

- . 우선순위 클래스 (3 비트)
  - .. 000 값 : 일반 최선 노력 인터넷 트래픽 (기본값)
  - .. 000 (보통: 우선순위 낮음) ~ 111 (네트워크 제어: 우선순위 높음)
- . 카테고리(그룹) 구분
  - .. 1 (xxxxx0) : 표준
  - .. 2 (xxxx11) : 실험용/로컬
  - .. 3 (xxxx01) : 실험용/예약

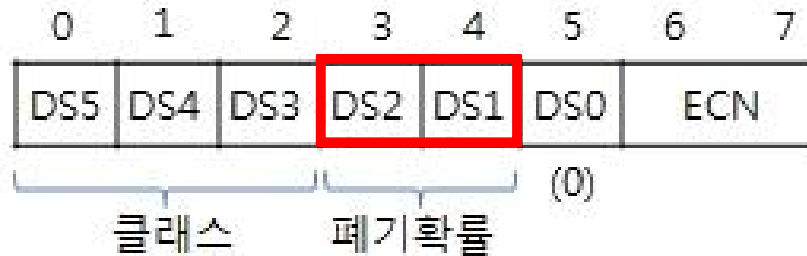


## 01 QOS

## QoS Marking

표 2 | IP 우선권과 DSCP의 비교

구분	이진수 값	이름	구분	이진수 값	이름
Precedence 0	000	routine	DSCP 0	000000	Best-effort
Precedence 1	001	priority	DSCP 8	001000	CS1
Precedence 2	010	immediate	DSCP 16	010000	CS2
Precedence 3	011	flash	DSCP 24	011000	CS3
Precedence 4	100	flash override	DSCP 32	100000	CS4
Precedence 5	101	critical	DSCP 40	101000	CS5
Precedence 6	110	internet	DSCP 48	110000	CS6
Precedence 7	111	network	DSCP 56	111000	CS7
CS = Class Selector AF = Assured Forwarding EF = Expedited Forwarding			DSCP 10	001010	AF11
			DSCP 12	001100	AF12
			DSCP 14	001110	AF13
			DSCP 18	010010	AF21
			DSCP 20	010100	AF22
			DSCP 22	010110	AF23
			DSCP 26	011010	AF31
			DSCP 28	011100	AF32
			DSCP 30	011110	AF33
			DSCP 34	100010	AF41
			DSCP 36	100100	AF42
			DSCP 38	100110	AF43
			DSCP 46	101110	EF



Drop 가능성에 대해 알아보자 (가능성 = Probabili)

01 = 낮음  
 10 = 보통  
 11 = 높음

6비트의 DSCP는 각 처리 노드에서 해당 패킷에 대한 처리 방법을 정의하는 PHB를 선택 할 때 사용한다.

# 01 QOS

## QoS Marking

PHB란 Per-Hop-Behaviour로 호별 행위이며 DiffServ가 구현된 라우터 등의 장비에서 다양한 등급으로 마킹되어 들어오는 일련의 패킷들의 흐름에 대해서 어떤 일관된 행위를 통해 다음 홉으로 전달하는 방식을 결정하는 것을 말한다.

이런 PHB의 종류가 (4가지)

1 Default PHB (Best-effort Forwarding PHB) : 최소한의 자원량만 할당한다. 마킹 하지 않는다

2. CA PHB(Class Selector PHB) : 상위 3비트만 표시해서 기저존의 Precedence 값과 서로 호환해서 사용이 가능하고, 값이 클수록 우선 순위가 높다. - RFC 2475

3. AF PHB(Assured Forwarding PHB) : 4개의 클래스가 존재하며, 각 클래스에 대해 3개의 서로 다른 Drop 가능성 값을 갖는다.

4. EF PHB(Expedited Forwarding PHB) : DSCP 코드 값 중에서 우선 순위가 가장 높은 클래스, DSCP 값을 인식하지 못하는 장비에서도 IP Precedence 값으로 인식되어(5=Critical) 최고의 서비스를 보장한다.

→※DSCP 0번인 BE는 가장 낮은 우선순위이고, DSCP 46번인 EF는 가장 높은 우선순위이다

# 01 QOS

## QoS Marking

그림 6 | AF(Assured Forwarding) PHB

클래스	값	드롭 가능성	값
AF1	001 dd 0	낮음	01
AF2	010 dd 0	보통	10
AF3	011 dd 0	높음	11
AF4	100 dd 0		

\* 4개의 표준 클래스 (af1, af2, af3, af4)

\* DSCP 값의 범위 : 'aaadd0' 여기서 'aaa'는 클래스 값을 표현하고, 'dd' 드롭 가능성의 높고 낮음을 의미한다.

# 01 QOS

## Conditioner

Conditioner 는 할당 받은 임계값을 초과했을 때 처리하는 매커니즘으로

Shaping 방식과 Policing 방식이 있다.

이것들은 트래픽을 Classifier 하여 Marking 하고 Meter 에 의해 정의된 임계값과 비교 후 임계값 초과 시 발생 한다.

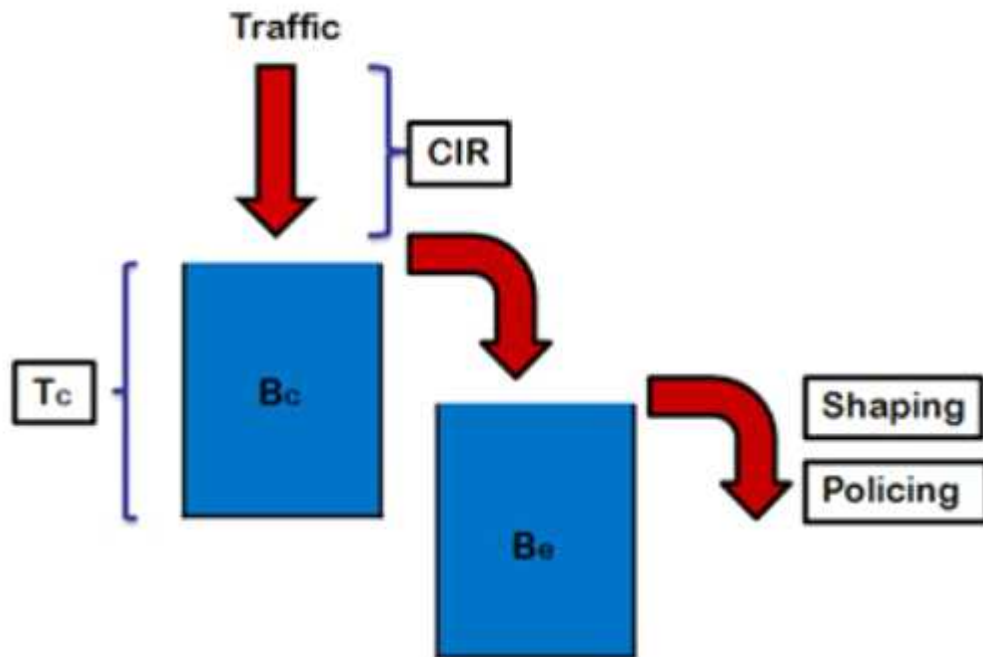
Shaping : 임계값을 초과할 때 버퍼에 저장을 해 두었다가 나중에 패킷을 처리하는 방식

Policing : 임계값을 초과할 때 초과한 패킷에 대해 Drop을 하는 방식

Token Bucket : Shaping 방식과 Policing 방식의 기본 알고리즘으로 일정 크기의 Bucket을 두어 트래픽이 들어올 때 Bucket 에 넣어두고, 처리하는 방식으로 Bucket을 넘어 갈 때 Shaping 이나 Policing 방식을 사용하여 처리한다.

# 01 QOS

## Conditioner



CIR = ISP 와 계약한 전송 속도, 단위 : bps

Bc = 패킷을 담아두는 Bucket 으로 주 Bucket 이라고 보면 된다  
단위 : bit

Be = Bc를 초과할 시 담아두는 Bucket으로 보조 Bucket 이다  
단위 : bit

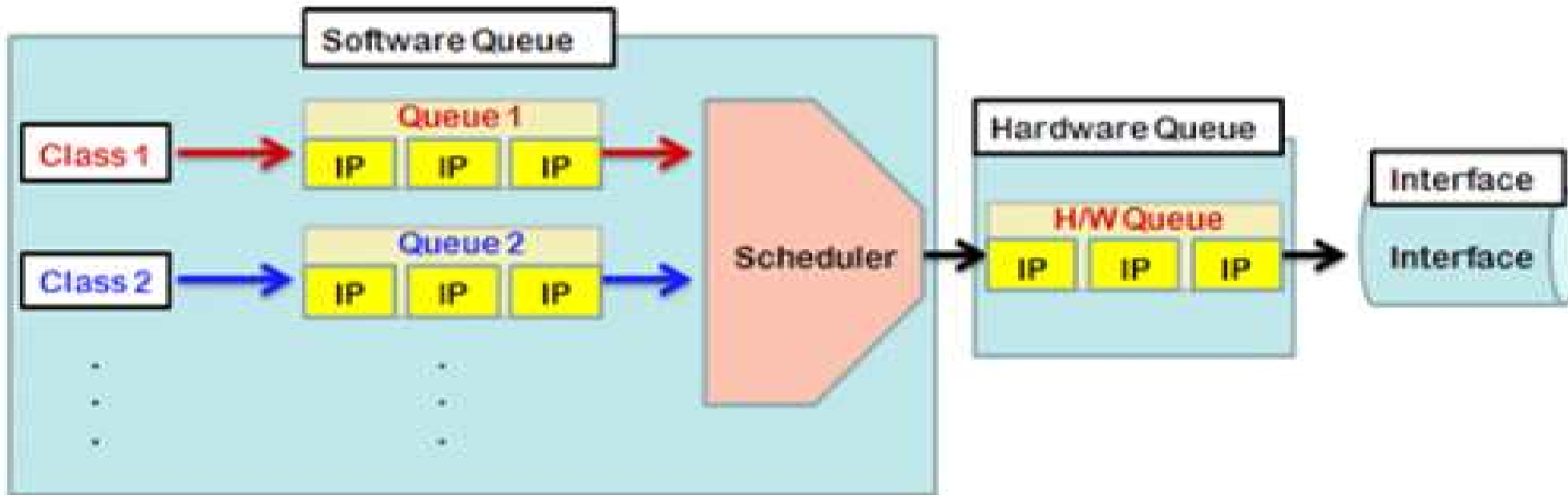
Tc = Bc를 가득 채우는 데 걸리는 시간, 단위 : bps,  $Bc = CIR * Tc$  이다.

# 01 QOS

## Queuing

큐잉이란 네트워크 장비가 처리할 수 있는 능력 이상으로 트래픽이 발생할 때 패킷들을 Queue 에 저장해 두었다가 나중에 서비스를 해주는 것이다.

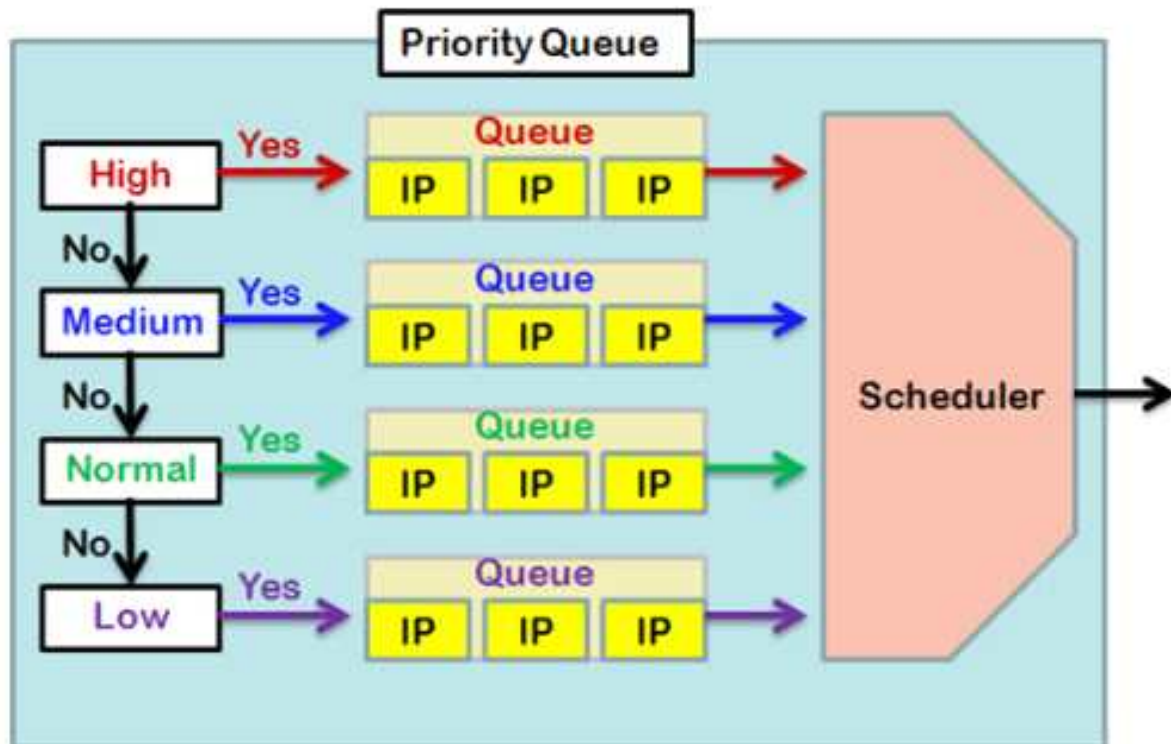
1. FIFO = 가장 기본적인 Queuing 구조로 먼저 들어온 패킷이 먼저 나간다는 스케줄링을 가진 Queue 이다.



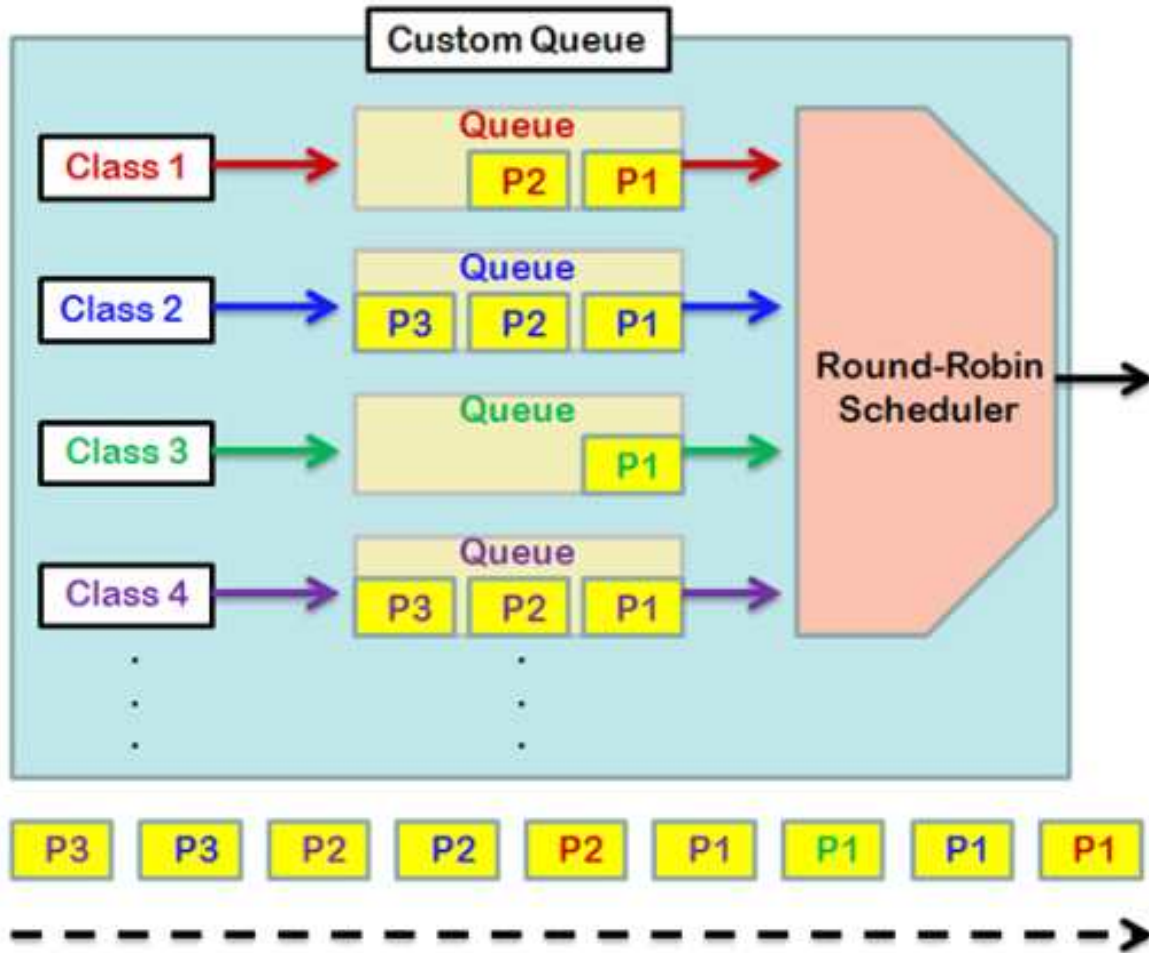
# 01 QOS

## Queuing

2. PQ = FIFO 단점을 해결하기 위해서 'High, Medium, Normal, Low' 4가지 클래스로 나눠 차등화된 서비스를 제공하는 Queuing 기법이다.



# 01 QOS



## Queuing

3. CQ = PQ의 문제인 우선순위가 높은 트래픽에 의해 우선순위가 낮은 트래픽이 서비스되지 못하는 현상을 해결하기 위해 각 클래스 별 Queue를 Round-Robin 방식으로 하나씩 돌아가며 처리한다.



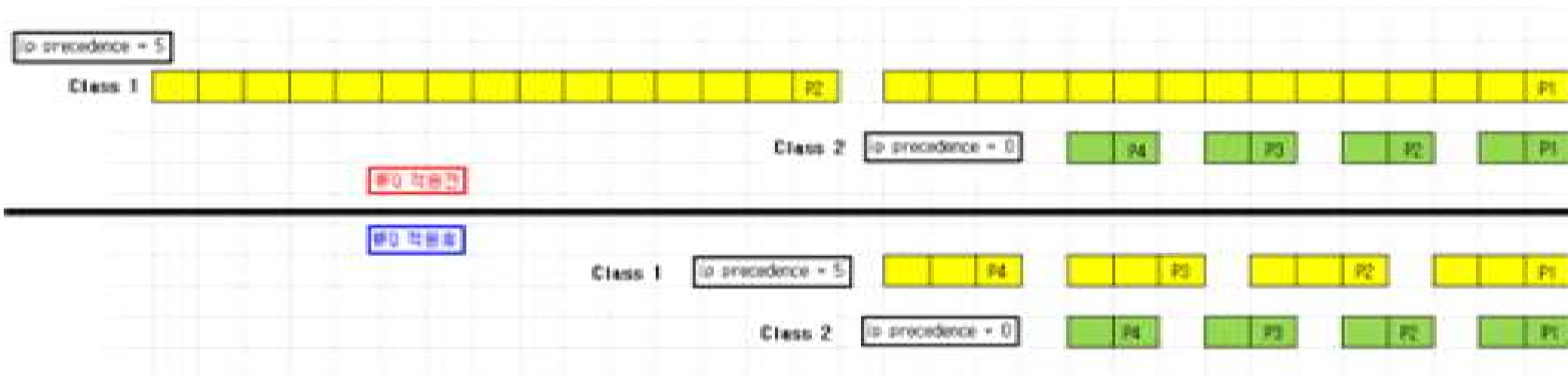
# 01 QOS

## Queuing

### 4. WFQ (Weighted Fair Queue)

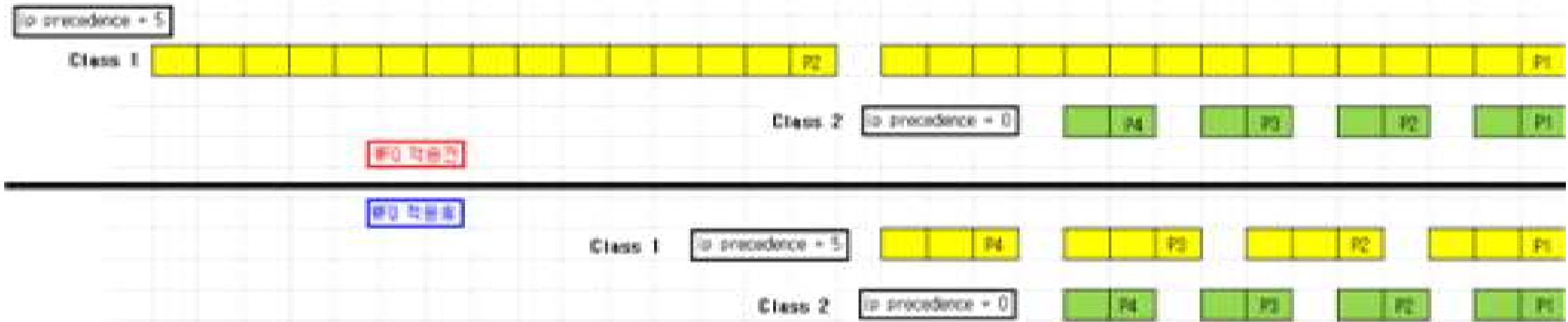
WFQ는 PQ의 우선순위가 높은 트래픽에 의해 우선순위가 낮은 트래픽 서비스가 불가능해 진다는 단점과 CQ의 클래스 별로 차등화 된 서비스를 받지 못한다는 단점을 해결하기 위해 만들어졌다.

WFQ는 4096개의 클래스가 존재하는데 각 클래스마다 IP Precedence로 가중치를 주어 실제 패킷 크기를 줄여 우선순위를 높여 처리하는 방식이다. (가상 패킷 크기 = 실제 패킷 크기 / ip precedence)



## 01 QOS

## Queuing



※ Class 1에 ip precedence가 5라면 실제 패킷 크기에 5로 나눠 가상 패킷으로 생각하게 된다. (C1\_P1 실제 크기 : 15 => 가상 크기 : 3)

적용전 패킷 처리 순서를 보면 C1\_P1 => C2\_P1 => C2\_P2 => C2\_P3 => C2\_P4 => C1\_P2 => C1\_P3 => C1\_P4 순 이다.

적용 후 패킷 처리 순서를 보면 C1\_P1 => C2\_P1 => C2\_P2 => C1\_P2 => C2\_P3 => C1\_P3 => C2\_P4 => C1\_P4 순 이다.

이처럼 가중치를 주어 패킷 크기가 커 우선 처리하기 힘든 경우를 잘 처리 해 준다.

```
R2#sh access-lists
Extended IP access list 100
  10 permit icmp host 1.1.12.1 host 1.1.3.3 (580 matches)
Extended IP access list 101
  10 permit icmp host 1.1.12.1 host 1.1.33.3 (260 matches)
```

ACL로 Ping 트래픽  
지정..

```
R2#
R2#sh class-map
Class Map match-all TELNET (id 3)
  Match protocol telnet

Class Map match-any class-default (id 0)
  Match any

Class Map match-all NET33 (id 2)
  Match access-group 101

Class Map match-all NET3 (id 1)
  Match access-group 100
```

class-map으로 TELNET과 PING 트래픽  
match 시킴.

```
R2#
R2#sh policy-map
Policy Map CBWFQ
  Class TELNET
    Bandwidth 30 (%) Max Threshold 64 (packets)
  Class NET3
    Bandwidth 20 (%) Max Threshold 64 (packets)
  Class NET33
    Bandwidth 20 (%)
```

Bandwidth를 전체에서 30%,  
20%로 설정 Drop은 Tail-Drop  
이 Default이다.

```
class NET33
  exponential weight 9
  class min-threshold max-threshold mark-probability
  -----
  0 - - 1/10
  1 - - 1/10
  2 - - 1/10
  3 - - 1/10
  4 - - 1/10
  5 - - 1/10
  6 - - 1/10
  7 - - 1/10
  rsvp - - 1/10
```

class NET33  
Bandwidth  
20%로 하고  
Drop을 WRED  
로 하였다.

```
Class class-default
  Flow based Fair Queueing
  Bandwidth 0 (kbps) Max Threshold 64 (packets)
```

나머지 Default는 WFQ로 한다.

## Queuing

5. CBWFQ (Class-Base Weight Fair Queue)  
CBWFQ는 WFQ의 확장판으로 각 클래스 마  
다 Bandwidth, Weight, Packet Limit 정책을  
정의할 수 있다.

또한 WFQ에서 혼잡 회피로 Tail Drop만을 사  
용한 반면 CBWFQ에서는 WRED도 사용할  
수 있다.

## 01 QOS

## Queuing

```
R2#sh policy-map int s1/2
Serial1/2
```

```
Service-policy output: CBWFQ
```

```
Class-map: TELNET (match-all)
 19 packets, 872 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
 Match: protocol telnet
 Queueing
  Output Queue: Conversation 265
  Bandwidth 30 (%)
  Bandwidth 463 (kbps) Max Threshold 64 (packets)
 (pkts matched/bytes matched) 2/97
 (depth/total drops/no-buffer drops) 0/0/0
```

```
Class-map: NET3 (match-all)
 580 packets, 776320 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
 Match: access-group 100
 Queueing
  Output Queue: Conversation 266
  Bandwidth 20 (%)
  Bandwidth 308 (kbps) Max Threshold 64 (packets)
 (pkts matched/bytes matched) 499/680496
 (depth/total drops/no-buffer drops) 0/0/0
```

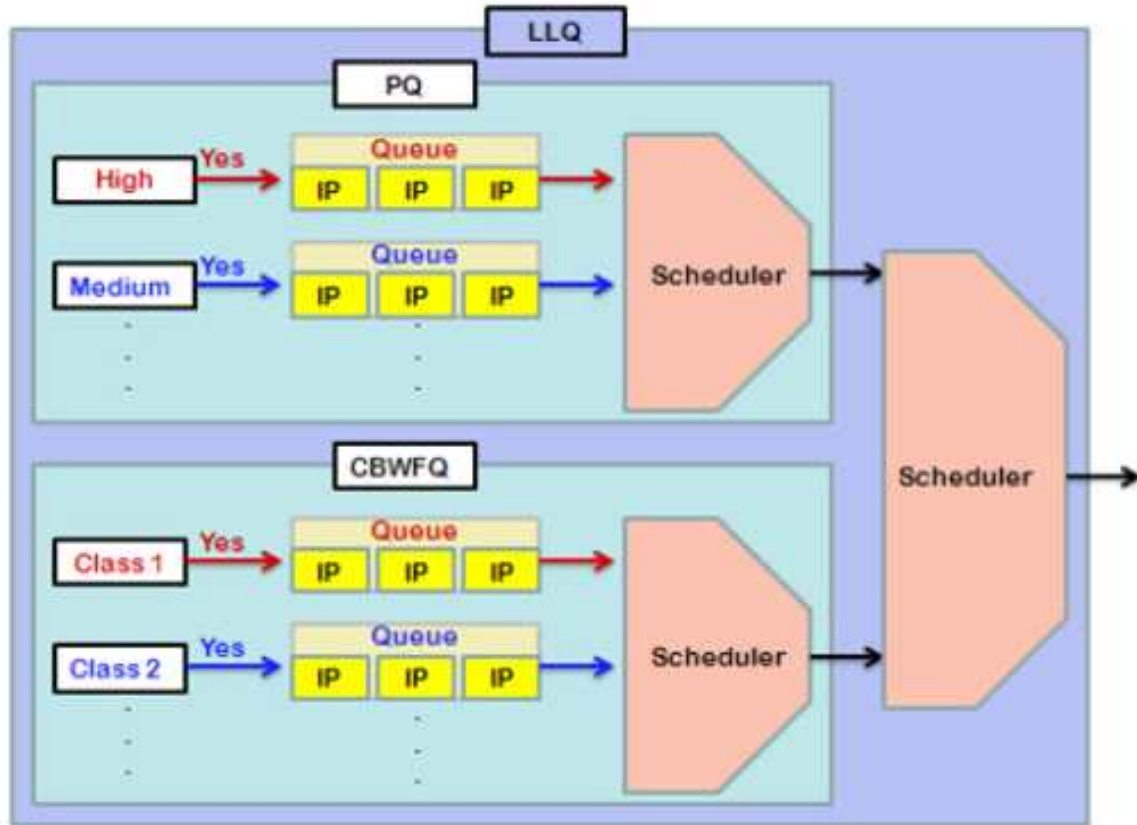
```
Class-map: NET33 (match-all)
 260 packets, 365840 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
 Match: access-group 101
 Queueing
  Output Queue: Conversation 267
  Bandwidth 20 (%)
  Bandwidth 308 (kbps)
 (pkts matched/bytes matched) 240/335760
 (depth/total drops/no-buffer drops) 0/0/0
 exponential weight: 9
 mean queue depth: 0
```

class	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
0	260/365840	0/0	0/0	20	40	1/10
1	0/0	0/0	0/0	22	40	1/10
2	0/0	0/0	0/0	24	40	1/10
3	0/0	0/0	0/0	26	40	1/10
4	0/0	0/0	0/0	28	40	1/10
5	0/0	0/0	0/0	30	40	1/10
6	0/0	0/0	0/0	32	40	1/10
7	0/0	0/0	0/0	34	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10

```
Class-map: class-default (match-any)
 154 packets, 10133 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
 Match: any
 Queueing
  Flow Based Fair Queueing
  Maximum Number of Hashed Queues 256
 (total queued/total drops/no-buffer drops) 0/0/0
```

```
R2#
```

# 01 QOS



## Queuing

### 6. LLQ (Low Latency Queue)

LLQ는 CBWFQ에 PQ를 혼합한 Queuing 기법으로 우선 처리해야 할 트래픽은 PQ로 수행하고 나머지는 CBWFQ로 수행한다.

# 01 QoS

## 패킷트레이서에서 QoS 설정



어떤 트래픽인지 acl 을 통해 지정한다.

class-map을 생성하고, 지정한 acl을 match 시킨다.  
트래픽 분류가됨(classifier)

policy-map을 생성하고, 생성한 각 class 마다 DSCP를 세팅한다.

이때 나머지 트래픽은 default 인 BE방식으로 설정한다.

R1->R2로 가므로 output 으로 policy-map을 적용시켜준다.

# 01 QoS

## 패킷트레이서에서 QoS 설정

### 1.Access-list

```
access-list 100 permit icmp host 20.20.20.1 host 20.20.20.2  
access-list 101 permit tcp host 20.20.20.1 host 20.20.20.2 eq telnet
```

# src : 20.20.20.1, dst : 20.20.20.2 로 향하는 icmp, telnet 패킷을 허용한다.

### 2.Class-map

```
class-map match-all telnet  
match access-group 101
```

```
class-map match-all ping  
match access-group 100
```

# 101번과 100번 access-list에 포함되는 패킷은 각각 telnet, ping 클래스 맵에 포함시킨다.

# 01 QoS

## 패킷트레이서에서 QoS 설정

### 3. Policy-map

```
policy-map dscp_marking
class ping
set ip dscp ef # icmp 에 대해서는 ef를 마킹한다.

class telnet
set ip dscp af11 # telnet 에 대해서는 af11을 마킹한다.
```

### 4. PING

```
QoS_Router(config-if)#do ping 20.20.20.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.20.20.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/6 ms



## 결과

```
QoS_Router#show policy-map interface serial 0/0/0
Serial0/0/0
```

```
Service-policy output: dscp_marking
```

```
Class-map: ping (match-all)
  5 packets, 640 bytes
  5 minute offered rate 20 bps, drop rate 0 bps
  Match: access-group 100
  QoS Set
    dscp ef
    Packets marked 5
  Queueing
    Output Queue: Conversation 265
    Bandwidth 0 (kbps)Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0
```

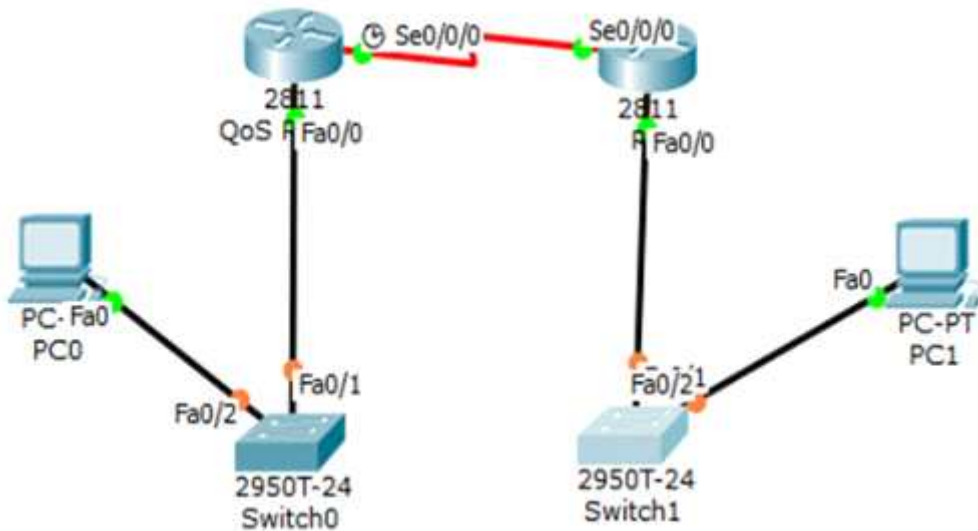
```
Class-map: telnet (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group 101
  QoS Set
    dscp af11
    Packets marked 0
  Queueing
    Output Queue: Conversation 266
    Bandwidth 0 (kbps)Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0
```

```
Class-map: class-default (match-any)
  6 packets, 1908 bytes
  5 minute offered rate 34 bps, drop rate 0 bps
  Match: any
```

# 01 QOS

## Policing ,Shaping 설정

acl 로 해당 네트워크를 지정한다.  
class-map 으로 지정한 acl 로 매치시킨다.  
policy-map을 생성하여 클래스에 Shaping을 적용한다.  
생성한 Policy-map을 인터페이스에 output 으로 적용한다



# Policing ,Shaping 설정

## 1. Access-list

```
QoS_Router(config)#access-list 100 permit icmp host 10.10.10.2 host 30.30.30.2
```

## 2. Class-map

```
QoS_Router(config)#class-map match-all shaping
QoS_Router(config-cmap)#match access-group 100
```

## 3. Policy-map

```
QoS_Router(config)#policy-map shap
QoS_Router(config-pmap)#class shaping
QoS_Router(config-pmap-c)#shape average 80000
```

# [단위 : bps, CIR 값을 입력해야함]

## 4. Service-policy

```
QoS_Router(config-pmap-c)#int se0/0/0
QoS_Router(config-if)#service-policy output shap
```

## 5 . Ping

```
QoS_Router(config-if)#do ping 20.20.20.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.20.20.2, timeout is 2 seconds:
!.....
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/341/563 ms
```

## 결과

```
QoS_Router(config-if)#do show policy-map interface se0/0/0
Serial0/0/0
```

Service-policy output: shap

Class-map: shaping (match-all)

13 packets, 1664 bytes

5 minute offered rate 52 bps, drop rate 0 bps

Match: access-group 100

Traffic Shaping

Target/Average Rate	Byte Limit	Sustain bits/int	Excess bits/int	Interval (ms)	Increment (bytes)
80000/80000	2500	10000	10000	125	1250

Adapt	Queue	Packets	Bytes	Packets	Bytes	Shaping
Active	Depth			Delayed	Delayed	Active
-	0	10	1370	3	411	no

Queueing

Output Queue: Conversation 265

Bandwidth 0 (kbps)Max Threshold 64 (packets)

(pkts matched/bytes matched) 6/822

(depth/total drops/no-buffer drops) 0/0/0

Class-map: class-default (match-any)

3 packets, 422 bytes

5 minute offered rate 13 bps, drop rate 0 bps

Match: any

# CBWFQ 설정

ACL로 ICMP 트래픽을 지정한다.  
class-map 으로 트래픽을 매치시킨다  
policy-map 으로 각 클래스의 대역폭을 보장한다.  
생성한 Polic-map을 인터페이스에 output 으로 적용한다.

## 1. Access-list

```
QoS_Router(config)#access-list 100 permit icmp host 20.20.20.1 host 20.20.20.2
```

## 2. Class-map

```
QoS_Router(config)#class-map match-all ping  
QoS_Router(config-cmap)#match access-group 100
```

## 3. Policy-map

```
QoS_Router(config)#  
QoS_Router(config)#policy-map queue  
QoS_Router(config-pmap)#class ping  
QoS_Router(config-pmap-c)#bandwidth percent 20  
QoS_Router(config-pmap-c)#exit  
QoS_Router(config-pmap)#class class-default  
QoS_Router(config-pmap-c)#fair-queue # WFQ 설정
```

## 4. Service-policy

```
QoS_Router(config-pmap-c)#int se0/0/0  
QoS_Router(config-if)#service-policy output queue
```

## 결과

```
QoS_Router(config-if)#do show policy-map interface se0/0/0
Serial0/0/0
```

Service-policy output: queue

Class-map: ping (match-all)

0 packets, 0 bytes

5 minute offered rate 0 bps, drop rate 0 bps

Match: access-group 100

Queueing

Output Queue: Conversation 265

Bandwidth 20 (%)

Bandwidth 308 (kbps)Max Threshold 64 (packets)

(pkts matched/bytes matched) 0/0

(depth/total drops/no-buffer drops) 0/0/0

Class-map: class-default (match-any)

2 packets, 370 bytes

5 minute offered rate 11 bps, drop rate 0 bps

Match: any

Queueing

Flow Based Fair Queueing

Maximum number of Hashed Queues 256

Bandwidth 1158 (kbps)Max Threshold 64 (packets)

(total queued/total drops/no-buffer drops) 0/0/0

# END

---

고생하셨습니다

