

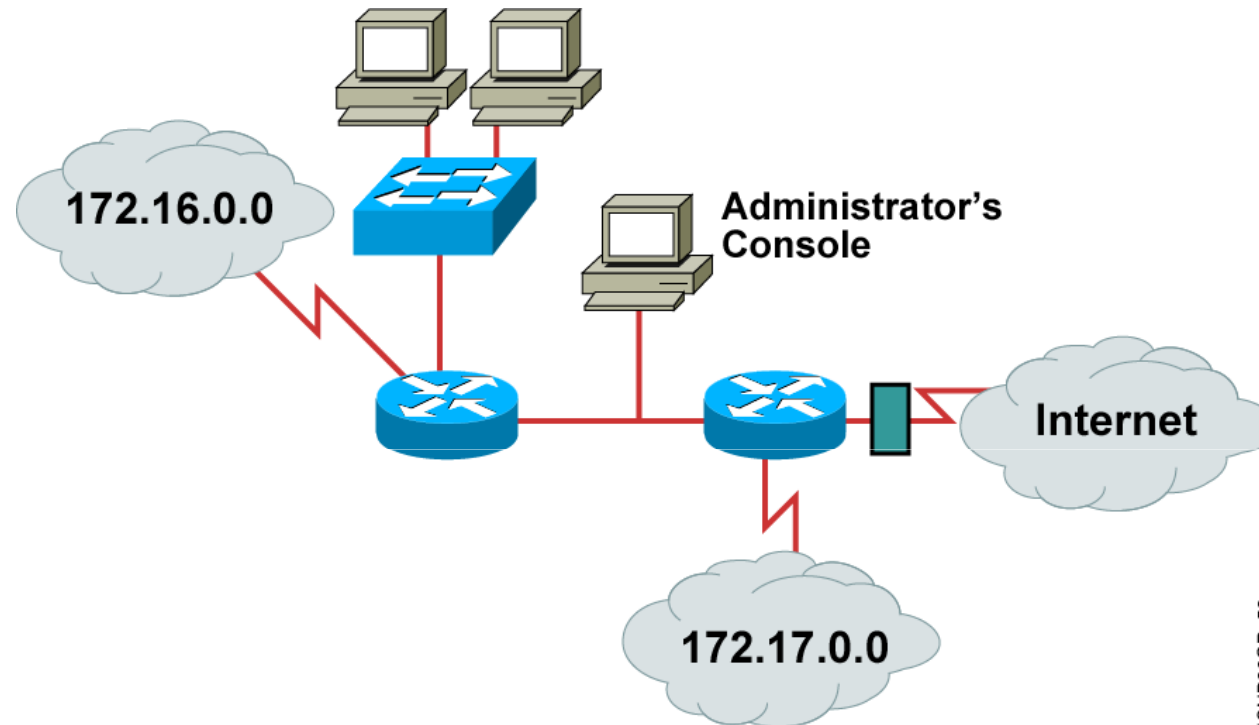


Module 05 : Managing IP Traffic with Access Lists and NAT



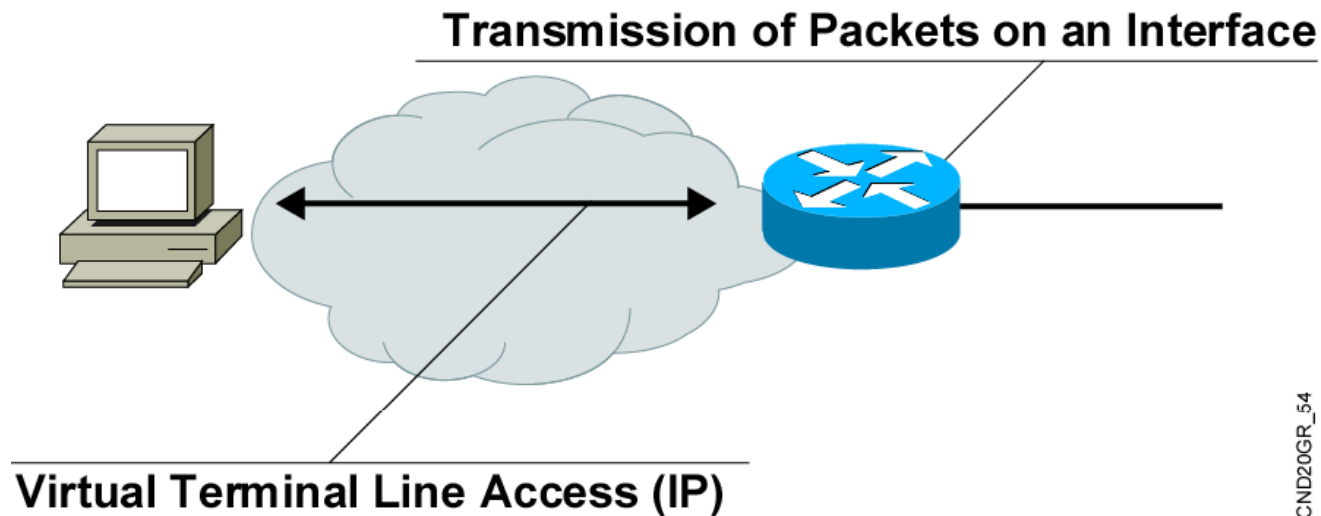
Access Lists and Their Applications

Why Use Access Lists?



- Router에서는 **ACL (Access Control List)**을 사용하여 트래픽 식별, 필터링, 암호화, 분류, 변환 작업을 수행할 수 있다.
- Router를 경유하는 **Packet**을 **Filtering** 한다.

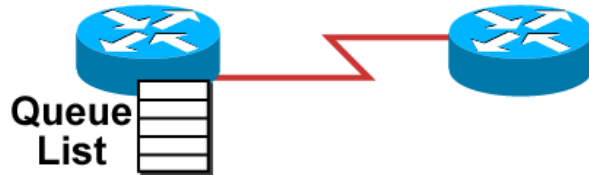
Access List Applications



- **Packet Filtering**을 활용하여 네트워크에서의 **Packet** 이동을 제어할 수 있다.
- **Router**에 **VTY** 포트로 들어오거나 **VTY** 포트에서 나가는 **Telnet** 트래픽을 허용 (**Permit**) 하거나 거부(**Deny**)할 수 있다.

Other Access List Uses

Priority and Custom Queuing



Dial-on-Demand Routing

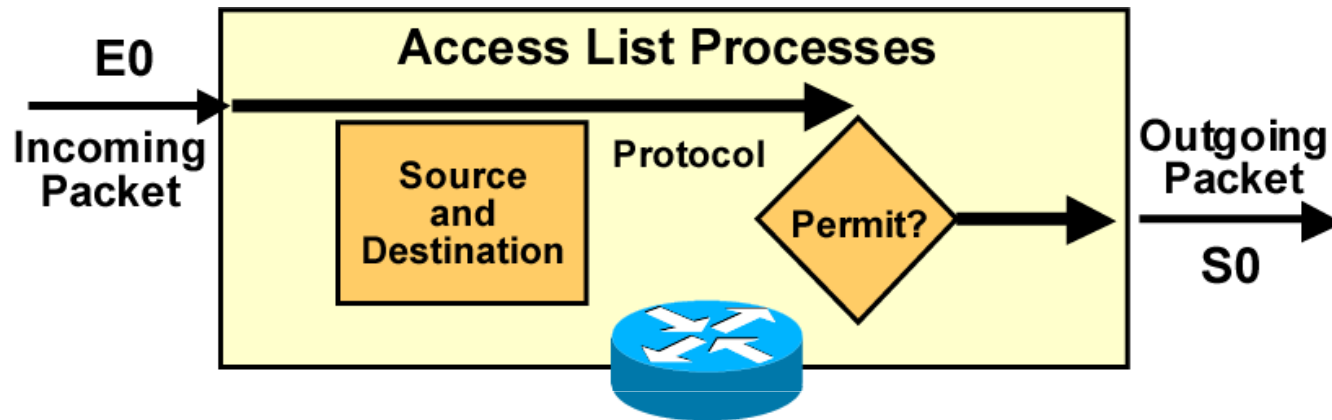


Route Filtering



- **Access list**를 다양한 방식으로 활용할 수 있다.
 - **Priority and Custom Queuing**
 - **DDR**
 - **Route Filtering**

Types of Access Lists



- **Standard Access list :**
 - **Source Address**를 검사한다.
 - 검사 결과에 따라 전체 **Protocol Suite**에 대한 **Packet** 출력을 허용하거나 거부한다.
- **Extended Access list :**
 - **Source Address** 와 **Destination Address**를 모두 검사한다.
 - 특정 **Protocol, Port** 번호, 다른 매개변수를 검사하여 유연하게 제어 가능하다.

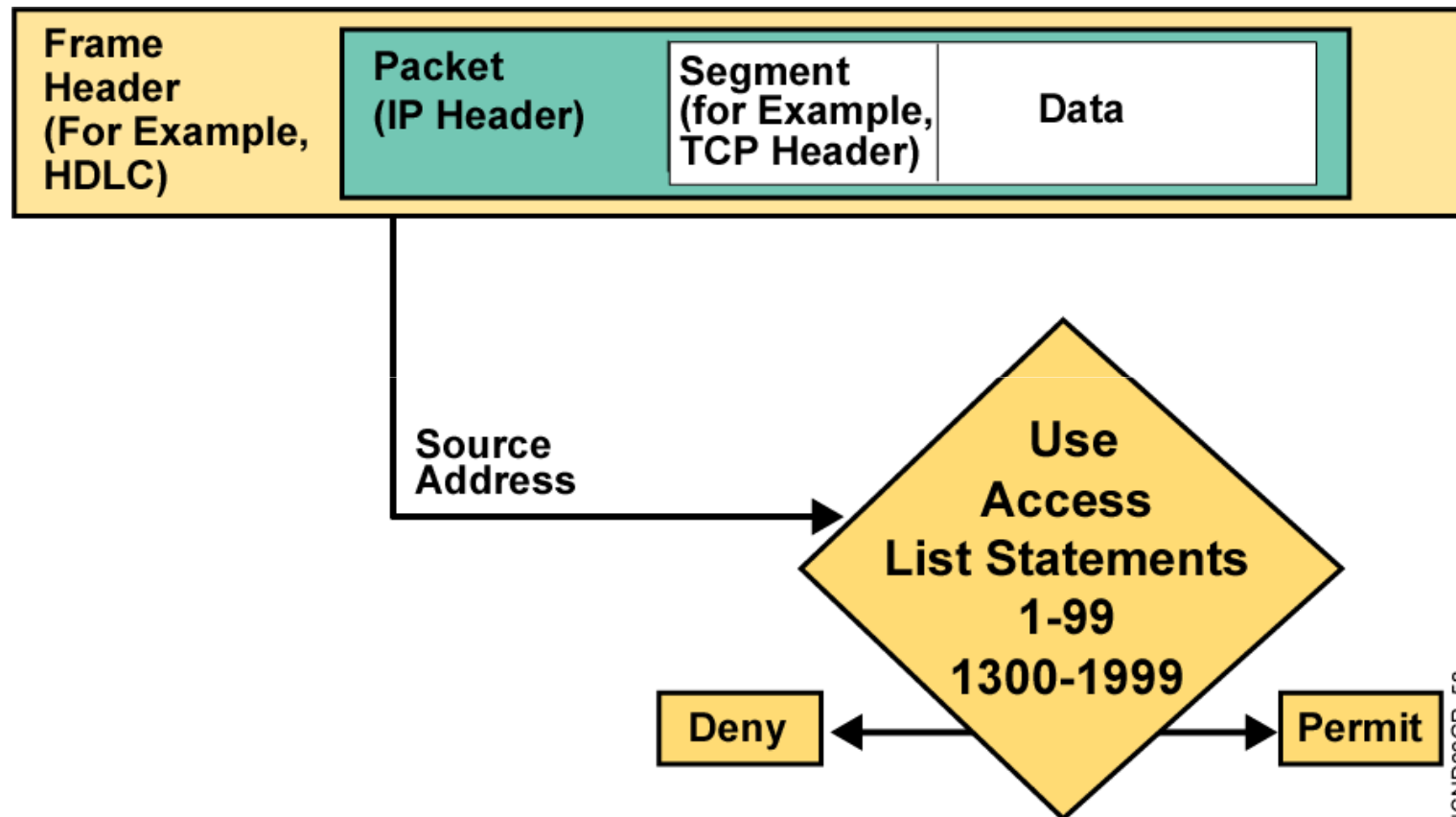


How to Identify Access Lists

Access List Type		Number Range/Identifier
IP	Standard	1-99, 1300-1999
	Extended	100-199, 2000-2699
	Named	Name

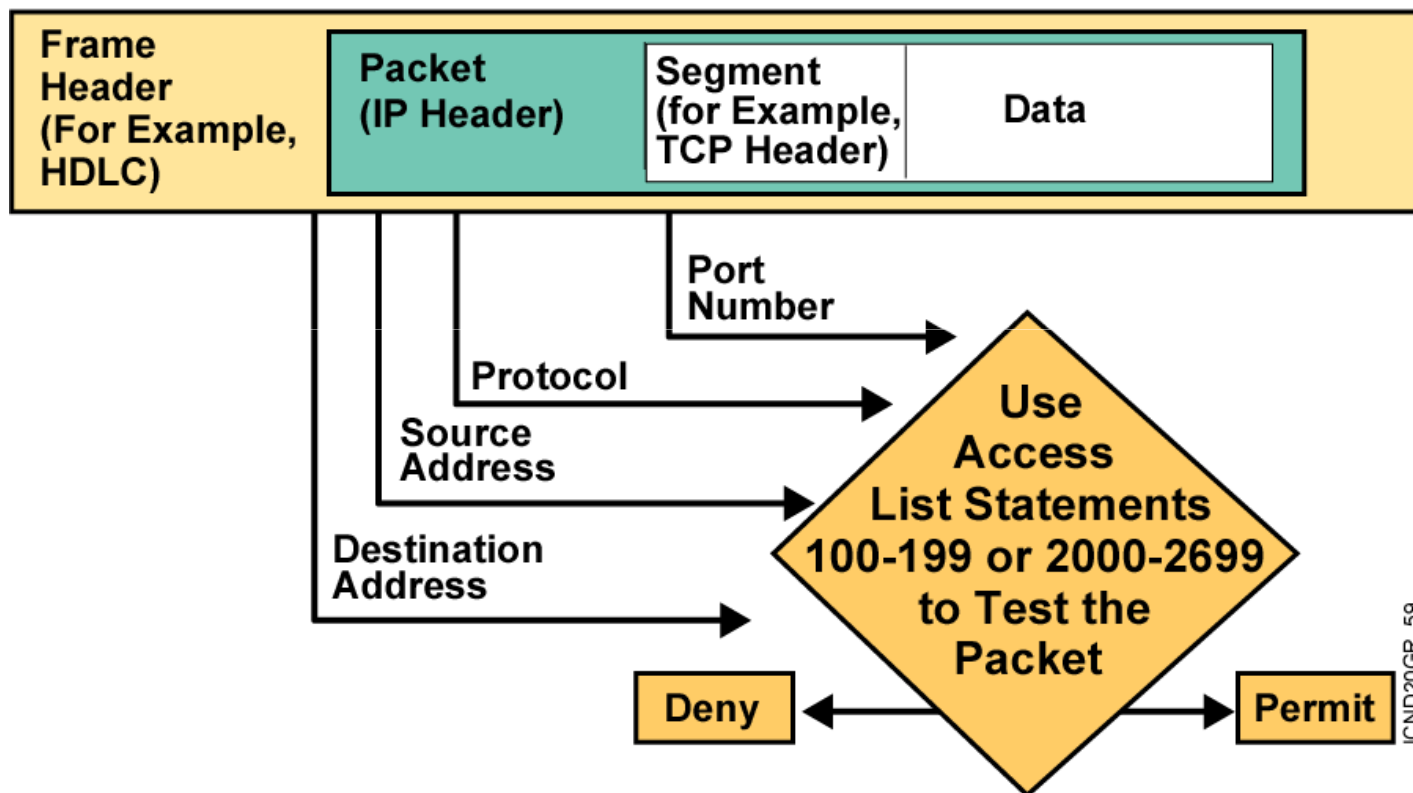
- **Standard IP list (1-99)**는 **IP Packet**에 **Source Address**를 조건으로 갖는다.
- **Extended IP list (100-199)**는 **Source and Destination Address** 와 특정 **TCP/IP Protocol Suite Protocol**과 **Destination Port**를 조건으로 갖는다.
- **Standard IP list (1300-1999) (Expanded range)**
- **Extended IP list (2000-2699) (Expanded range)**
- **Named IP list**는 **Standard**와 **Extended** 이름으로 선언하여 각 조건을 검사한다.

Testing Packets with Standard Access Lists



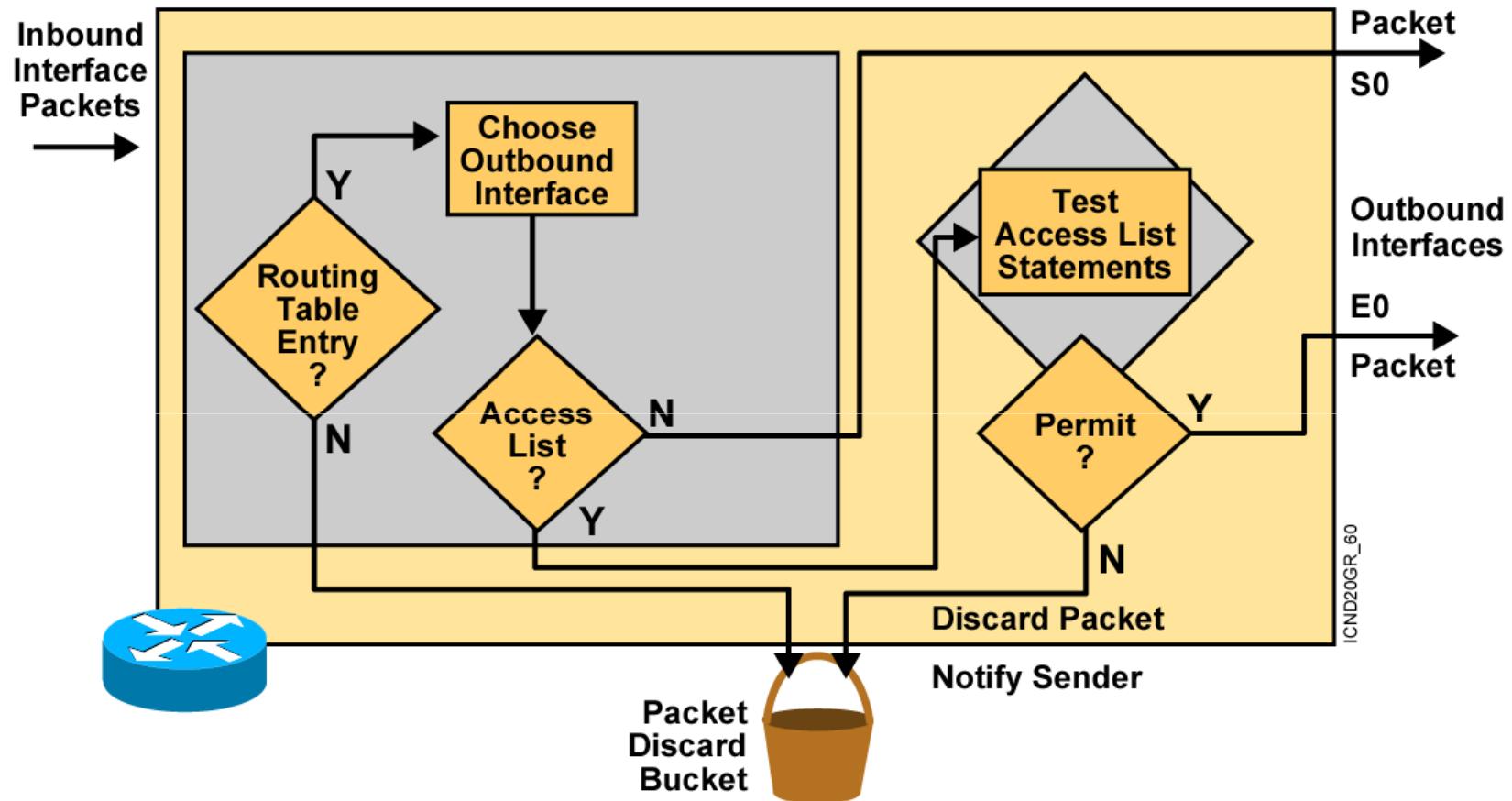
Testing Packets with Extended Access Lists

An Example from a TCP/IP Packet



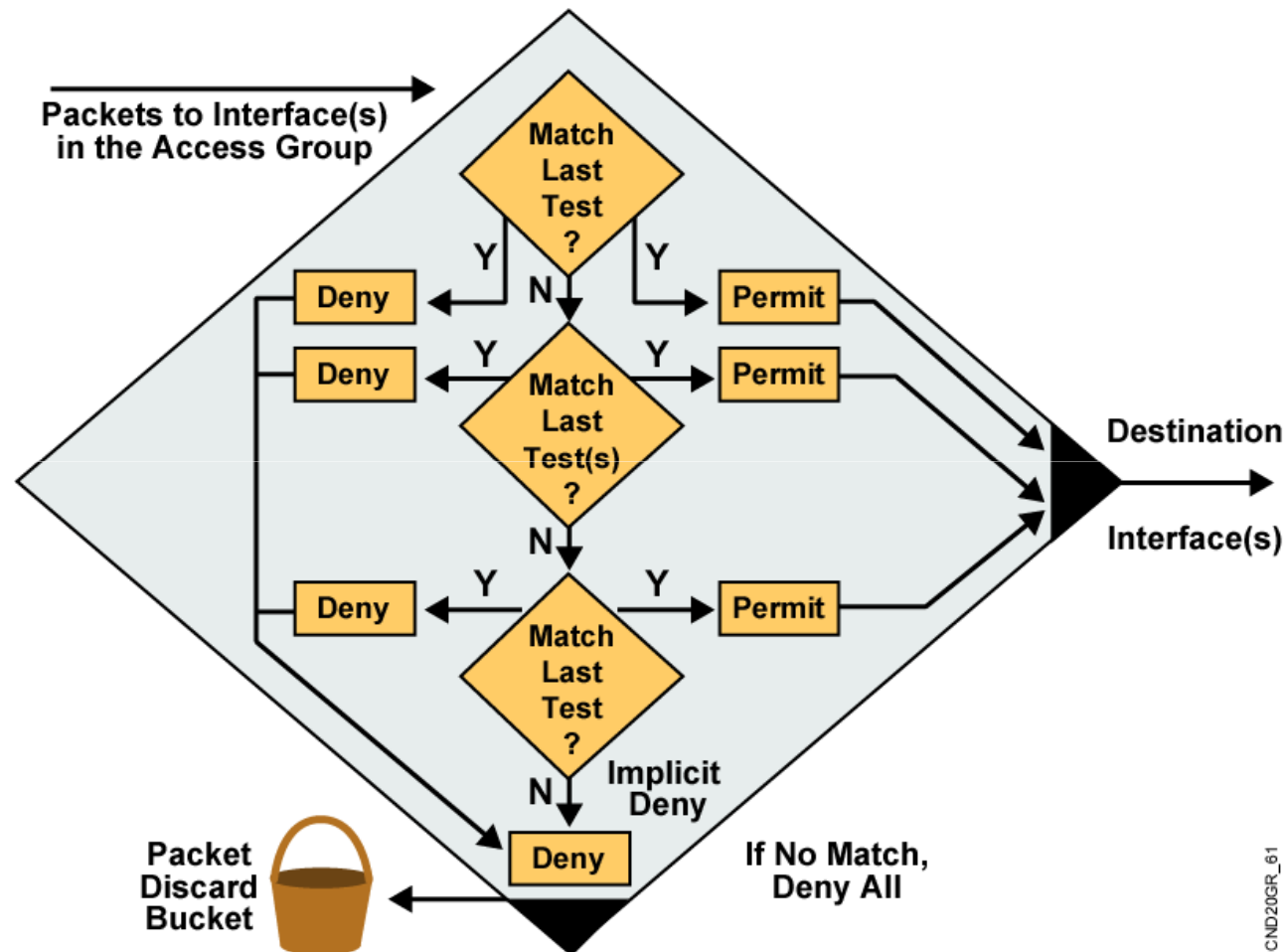
ICND20GR_59

Outbound ACL Operation



- **Access list**에 매치되지 않는 모든 **Packet**은 암시적으로 거부된다.

A List of Tests: Deny or Permit



Wildcard Bits: How to Check the Corresponding Address Bits

Octet Bit Position and Address Value for Bit										Examples
128	64	32	16	8	4	2	1			
0	0	0	0	0	0	0	0	=		Check All Address Bits (Match All)
0	0	1	1	1	1	1	1	=		Ignore Last 6 Address Bits
0	0	0	0	1	1	1	1	=		Ignore Last 4 Address Bits
1	1	1	1	1	1	0	0	=		Check Last 2 Address Bits
1	1	1	1	1	1	1	1	=		Do Not Check Address (Ignore Bits in Octet)

- **Wildcard mask bit 0**은 대응 **bit** 값을 검사하라는 것을 의미한다.
- **Wildcard mask bit 1**은 대응 **bit** 값을 검사하지 말고 무시하라는 것을 의미한다.



Wildcard Bits to Match a Specific IP Host Address

Test 조건 : 모든 **Address bit** 검사 (모두 일치)

1개의 IP Host Address, 예를 들어 :

172.30.16.29




Wildcard Mask: 0.0.0.0
(Checks All Bits)

- 위 예 **172.30.16.29 0.0.0.0** 은 모든 **Address**를 검사해서 매치되는 주소 즉 **172.30.16.29 IP**를 갖는 호스트를 지정한다.
- 하나의 **IP**를 알리기 위해 **IP Address** 앞에 약어 **host**를 사용할 수 있다. 예를 들면 "**172.30.16.29 0.0.0.0**" 대신 "**host 172.30.16.29**" 를 사용할 수 도 있다.



Wildcard Bits to Match Any IP Address

Test 조건 : 모든 Address bit 무시 (Match any)
모든 IP Address :

0.0.0.0

Wildcard Mask: 255.255.255.255
(Ignore All)

ICND20GR_65

- 모든 **Address**를 받아들이려면 **IP Address**는 **0.0.0.0**을 입력하고 **Wildcard mask**는 모든 값을 무시(검사 없이 허용)할 거면 **255.255.255.255**를 지정한다.
- 관리자는 모든 주소를 지정할 목적으로 **0.0.0.0 255.255.255.255** 을 명시하는 대신 **any**라는 문자를 사용할 수 있다.



Wildcard Bits to Match IP Subnets

- **172.30.16.0/24**에서 **172.30.31.0/24** 까지의 **IP Subnet** 검사하기
 - **Address and Wildcard mask :**
172.30.16.0 0.0.15.255

		Network		.Host					
		172.30.16.0							
Wildcard Mask:	0	0	0	1	0	0	0	0	
	0	0	0	0	1	1	1	1	
	<---- Match ---->			<----- Don't Care ----->					
	0	0	0	1	0	0	0	0	= 16
	0	0	0	1	0	0	0	1	= 17
	0	0	0	1	0	0	1	0	= 18
				:					:
	0	0	0	1	1	1	1	1	= 31



Configuring IP Access Lists



Access List Command Overview

Step 1: Access-list 명령어로 IP Traffic Filter list에 Entry를 만든다.

```
Router(config)#access-list access-list-number {permit | deny}  
{test_conditions}
```

Step 2: ip access-group 명령어로 기존 Access-list를 Interface에 적용한다.

```
Router(config-if)#{protocol} access-group access-list-number  
{in | out}
```



Standard IP Access List Configuration

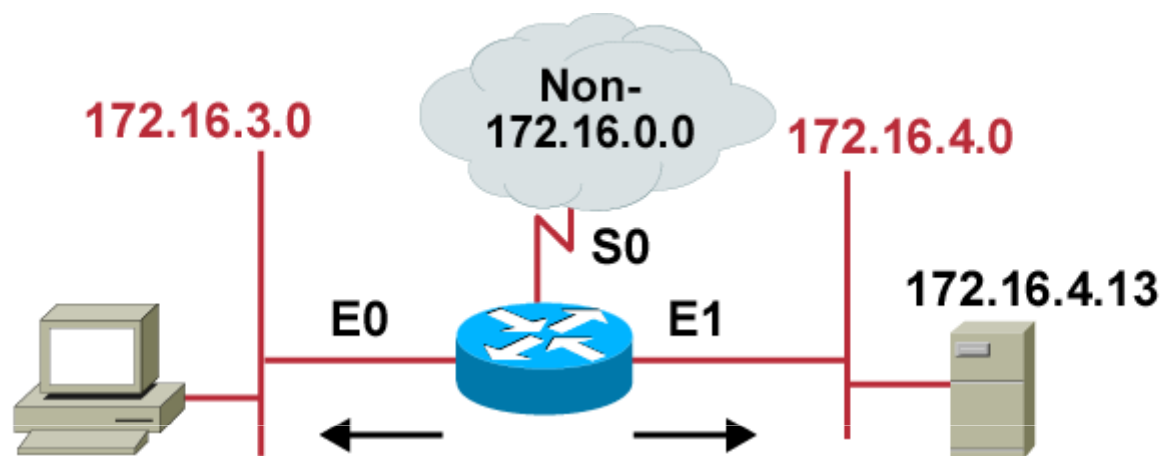
```
Router(config)#access-list access-list-number  
{permit | deny | remark} source [mask]
```

- **Access-list-number** : Entry가 속할 **list** 번호 설정 **1 ~ 99** , **1300 ~ 1999** 사이의 번호가 들어간다.
- **Permit | deny | remark** 는 해당 **Entry**에 매치되면 취할 **Action**을 정의
- **Source** 는 송신지 **IP Address**를 정의한다.
- **mask**는 **Wildcard mask**를 사용하여 **Address** 필드의 어느 비트들이 일치되어야 하는지 설정한다.

```
Router(config-if)#ip access-group access-list-number {in | out}
```

- **List**를 적용할 **Interface**에 설정한다.
- **Inbound** 또는 **Outbound** 시 검사하도록 설정한다.
- **Default = outbound**
- **Interface**에서 “**no ip access-group *access-list-number***” 명령을 사용하여 적용된 **Access-list**를 제거한다.

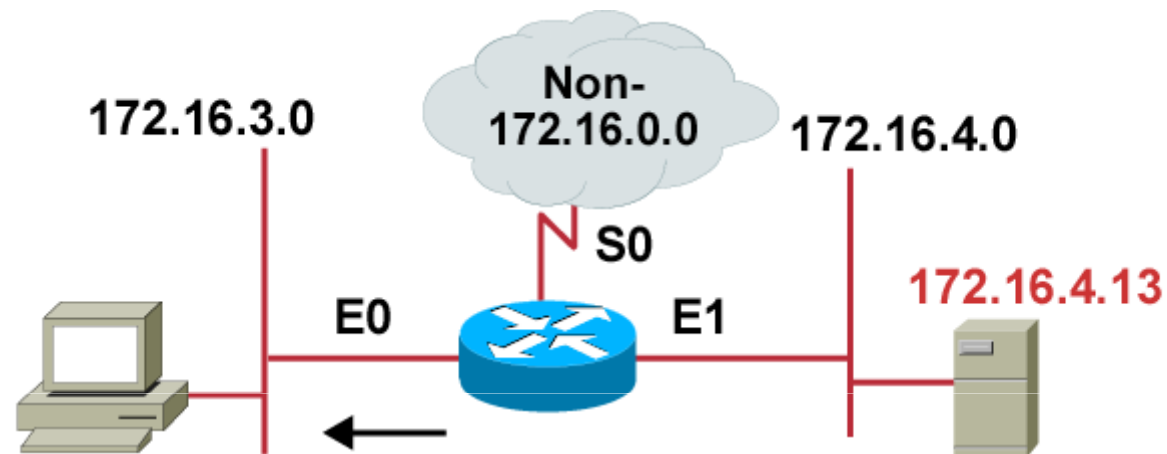
Standard IP Access List Example 1



```
Router(config)#access-list 1 permit 172.16.0.0 0.0.255.255  
(implicit deny all - not visible in the list)  
(access-list 1 deny 0.0.0.0 255.255.255.255)
```

```
Router(config)#interface ethernet 0  
Router(config)#ip access-group 1 out  
Router(config)#interface ethernet 1  
Router(config)#ip access-group 1 out
```

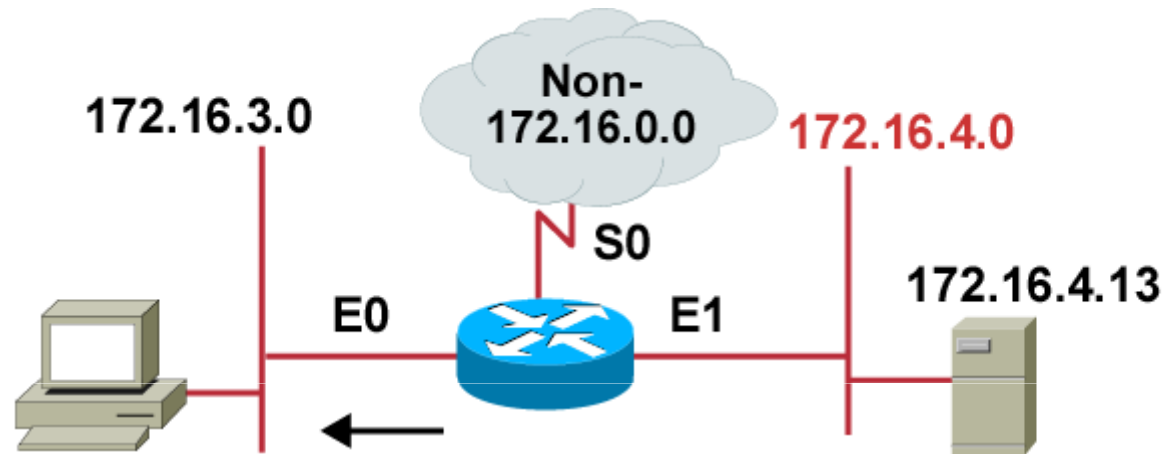
Standard IP Access List Example 2



```
Router(config)#access-list 1 deny 172.16.4.13 0.0.0.0
Router(config)#access-list 1 permit 0.0.0.0 255.255.255.255
(implicit deny all)
(access-list 1 deny 0.0.0.0 255.255.255.255)

Router(config)#interface ethernet 0
Router(config)#ip access-group 1 out
```

Standard IP Access List Example 3



```
Router(config)#access-list 1 deny 172.16.4.0 0.0.0.255
Router(config)#access-list 1 permit any
(implicit deny all)
(access-list 1 deny 0.0.0.0 255.255.255.255)

Router(config)#interface ethernet 0
Router(config)#ip access-group 1 out
```



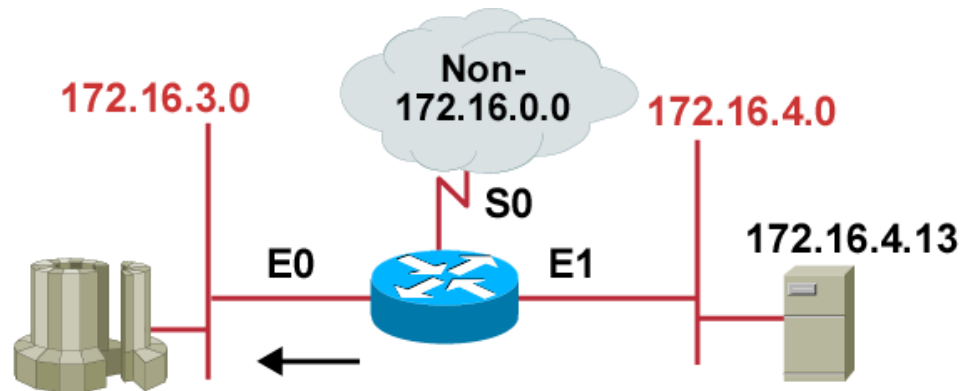
Extended IP Access List Configuration

```
Router(config)#access-list access-list-number  
{permit | deny} protocol source source-wildcard [operator port]  
destination destination-wildcard [operator port] [established] [log]
```

- **Access-list-number** : Entry가 속할 list 번호 설정 100 ~ 199 , 2000 ~ 2699 사이의 번호가 들어간다.
- **Permit | deny | remark** 는 해당 Entry에 매치되면 취할 Action을 정의
- **Source**와 **Destination**은 송수신지 **IP Address**를 정의한다.
- **mask**는 **Wildcard mask**를 사용하여 **Address** 필드의 어느 비트들이 일치되어야 하는지 설정한다.
- **Operator port**는 lt (less than), gt (greater than), eq (equal to), neq (not equal to)와 **Protocol Port** 번호를 명시한다.
- **established**는 **Inbound TCP**에 대해서만 사용된다.
- **log**는 **Console**로 log Message를 보낸다.

```
Router(config-if)#ip access-group access-list-number  
{in | out}
```

Extended Access List Example 1

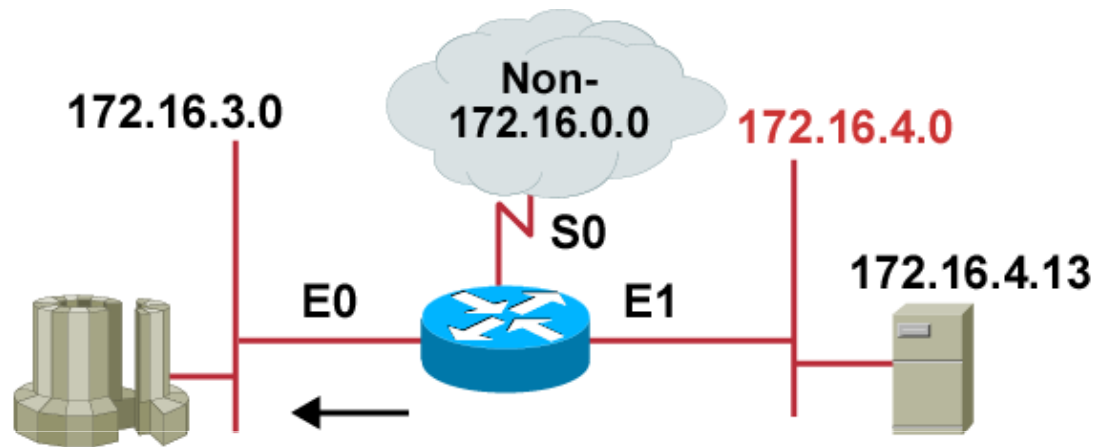


```
Router(config)#access-list 101 deny tcp 172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 21
Router(config)#access-list 101 deny tcp 172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 20
Router(config)#access-list 101 permit ip any any
(implicit deny all)
(access-list 101 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255)

Router(config)#interface ethernet 0
Router(config)#ip access-group 101 out
```

- **deny list**는 **172.16.4.0 Subnet**에서 **182.16.3.0 subnet**으로 가는 **FTP Traffic**을 거부한다.
- **Permit**은 다른 모든 **IP Traffic**이 **E0 Interface**로 나가는 것을 허용한다.

Extended Access List Example 2



```
Router(config)#access-list 101 deny tcp 172.16.4.0 0.0.0.255 any eq 23
Router(config)#access-list 101 permit ip any any
(implicit deny all)

Router(config)#interface ethernet 0
Router(config)#ip access-group 101 out
```

- **deny**는 **172.16.4.0 Subnet**에서 **e0 Interface**로 나가는 **Telnet Traffic**을 거부한다.
- **Permit**은 다른 모든 **IP Traffic**이 **E0 Interface**로 나가는 것을 허용한다.



Named IP Access list?

- **Named IP Access list** 고려사항 :
 - **Named IP Access list**는 **IOS 11.2** 이전 **Version**에서는 호환되지 않는다.
 - 여러 개의 액세스 리스트에 같은 이름을 사용할 수 없다.
- **Named IP Access list** 생성 단계 :
 1. **Named IP Access-list Mode**로 이동한다.
Router(config)# **ip access-list {stanard | extended}** name
 2. **Test** 조건을 입력한다.
Router(config-{std|ext}-nacl)# **{permit|deny}** {test conditions}
Router(config-{std|ext}-nacl)# **no {permit|deny}** {test conditions}
 3. 해당 **Access-list**를 **Interface**에 적용하기
Router(config-if)# **ip access-group** name {in | out}

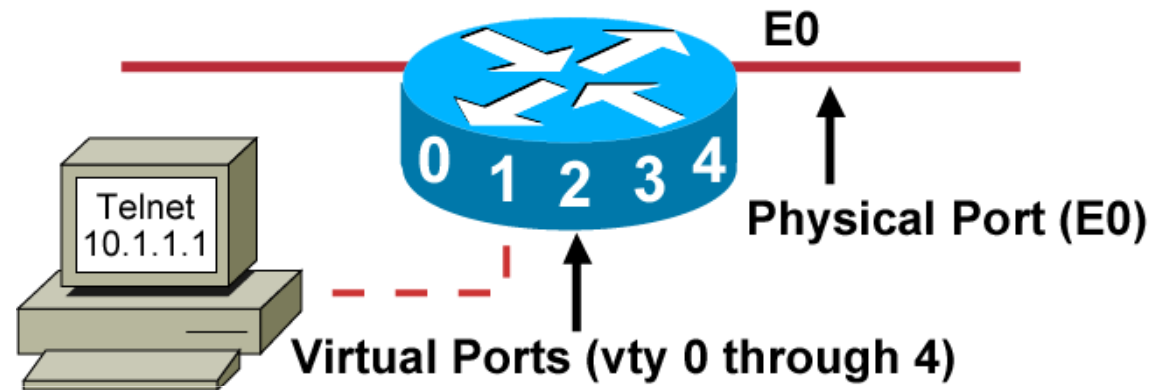


Using Named IP Access Lists

```
Router(config)#ip access-list extended screen
Router(config-ext-nacl)# deny tcp 172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 23
Router(config-ext-nacl)# permit ip any any
Router(config)# interface ethernet 0
Router(config-if)# ip access-group screen out
```

- **deny**는 **172.16.4.0 Subnet**에서 **e0 Interface**로 나가는 **Telnet Traffic**을 거부하는 **Named Access list**이다.
- **Permit**은 다른 모든 **IP Traffic**이 **E0 Interface**로 나가는 것을 허용한다.

How to Control vty Access



- **VTY**는 라우터에 **Telnet** 접속을 위해 할당된 가상포트이다.
- **Interface**를 경유해서 지나가는 트래픽이 아니기 때문에 **Interface**에서 제어할 수 없으므로 **line vty 0 4**에서 제어한다.



vty Commands

- 접속 제어할 포트 번호를 활성화 한다.

```
Router(config)#line vty { vty# | vty-range }
```

- 적용할 **Access-list**를 적용한다.

```
Router(config-line)#access-class access-list-number {in | out}
```



vty Access Example

Controlling Inbound Access

```
access-list 12 permit 192.168.1.0 0.0.0.255  
(implicit deny all)  
!  
line vty 0 4  
access-class 12 in
```

- **192.168.1.0/24 Subnet**에 해당하는 **IP Address**를 갖는 호스트만 접속을 허용한다.



Monitoring Access List Statements

```
Router#show {protocol} access-list {access-list number}
```

```
Router#show access-lists {access-list number}
```

```
Router#show access-lists
```

Standard IP access list 1

```
permit 10.2.2.1
```

```
permit 10.3.3.1
```

```
permit 10.4.4.1
```

```
permit 10.5.5.1
```

Extended IP access list 101

```
permit tcp host 10.22.22.1 any eq telnet
```

```
permit tcp host 10.33.33.1 any eq ftp
```

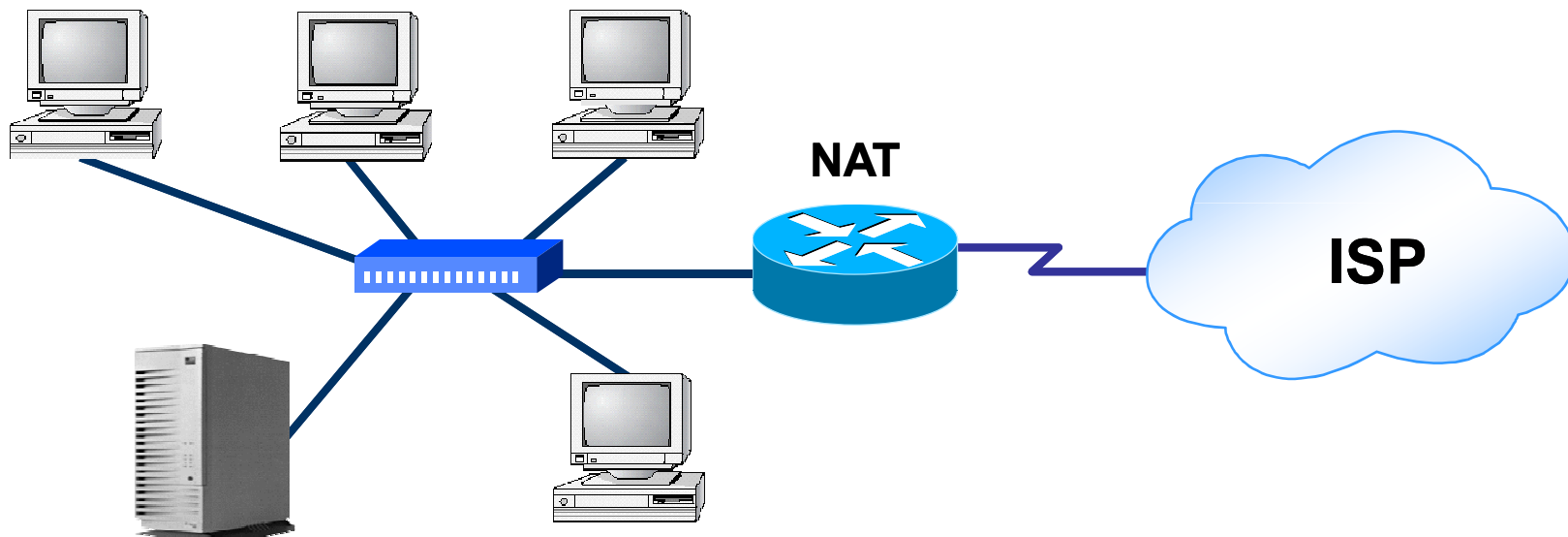
```
permit tcp host 10.44.44.1 any eq ftp-data
```



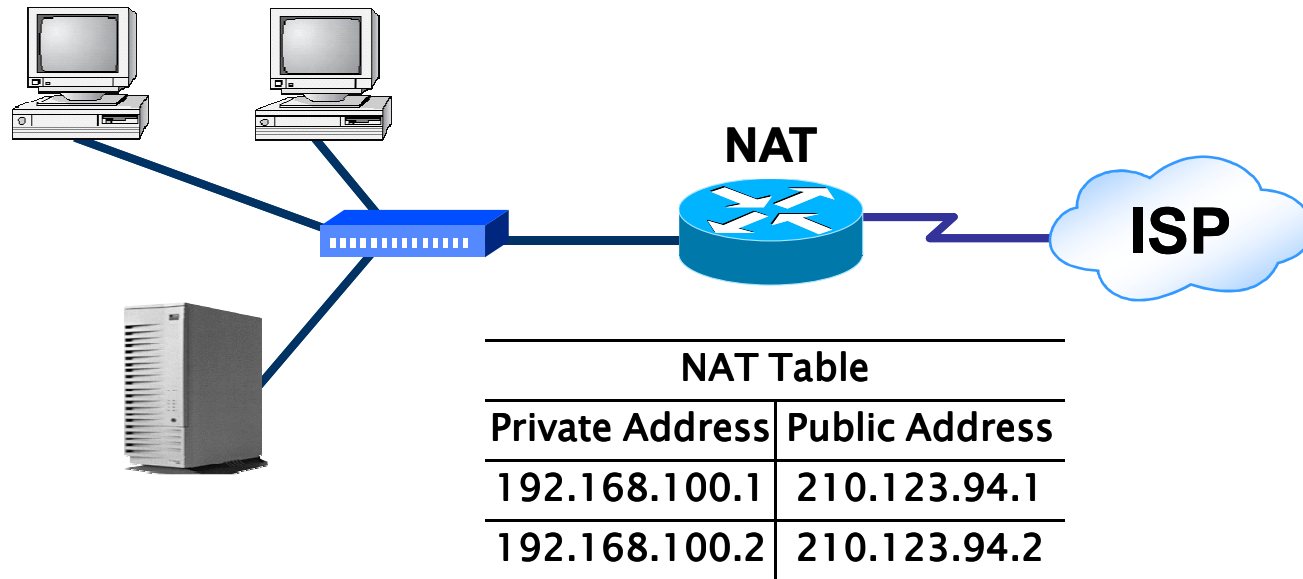
Network Address Translation

NAT(Network Address Translation)

NAT는 RFC1631에 정의된 것으로 IP Header 에 한 주소를 다른 주소로 바꾸는 기술이다. NAT는 사설주소를 사용하는 호스트들이 인터넷 서비스를 이용할 수 있도록 하기 위해서 사용한다.



Dynamic & Static NAT



동적 NAT는 호스트의 요구하는 Traffic을 받으면 IP 주소내에 사설 IP를 라우터에 설정된 주소풀에 있는 공인 IP로 변환 한 후 외부로 전달 한다. 외부에서 응답신호가 라우터로 돌아오면 NAT 라우터는 NAT Table에 있는 이전 정보로 목적지로 들어온 주소를 사설 IP로 변환 해서 내부망으로 전달 한다.

정적 NAT는 외부주소로 들어온 요청을 내부서버에 전달 될 수 있도록 목적지 주소를 변환 하는 기능이다. 이 방법으로 사설망에 서버를 구현하고 외부 주소로 들어오는 연결을 내부 서버로 전달 할 수 있다.



Dynamic NAT Configuration

1. IP 변환에 사용할 전역 주소풀을 설정한다.

```
Router(config)#ip nat pool name start-ip end-ip {netmask Netmask  
| Prefix-length Prefix-length}
```

2. 내부에서 IP 변환을 허용할 주소를 Standard Access-list로 정의한다.

```
Router(config)#Access-list number permit source-address  
[Wildcard-mask]
```

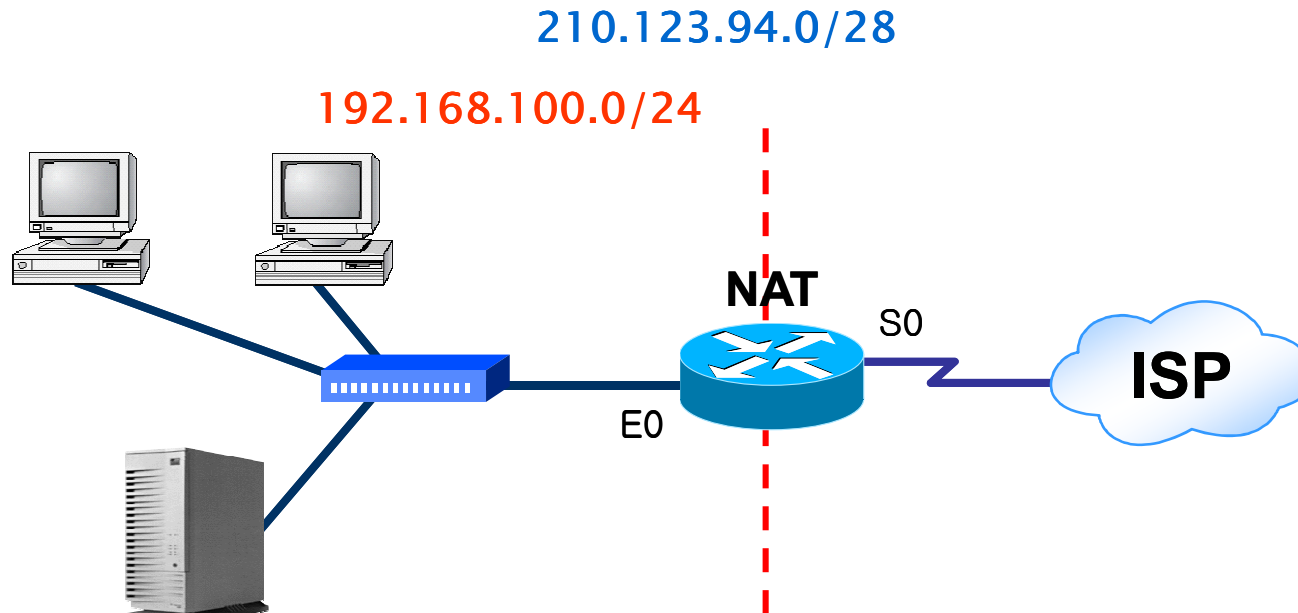
3. 동적 변환을 수립하기 위한 NAT 설정을 한다.

```
Router(config)#ip nat inside source list Access-list-number pool  
name [overload]
```

4. 각 인터페이스로 이동후 내부와 외부로 각각 설정한다.

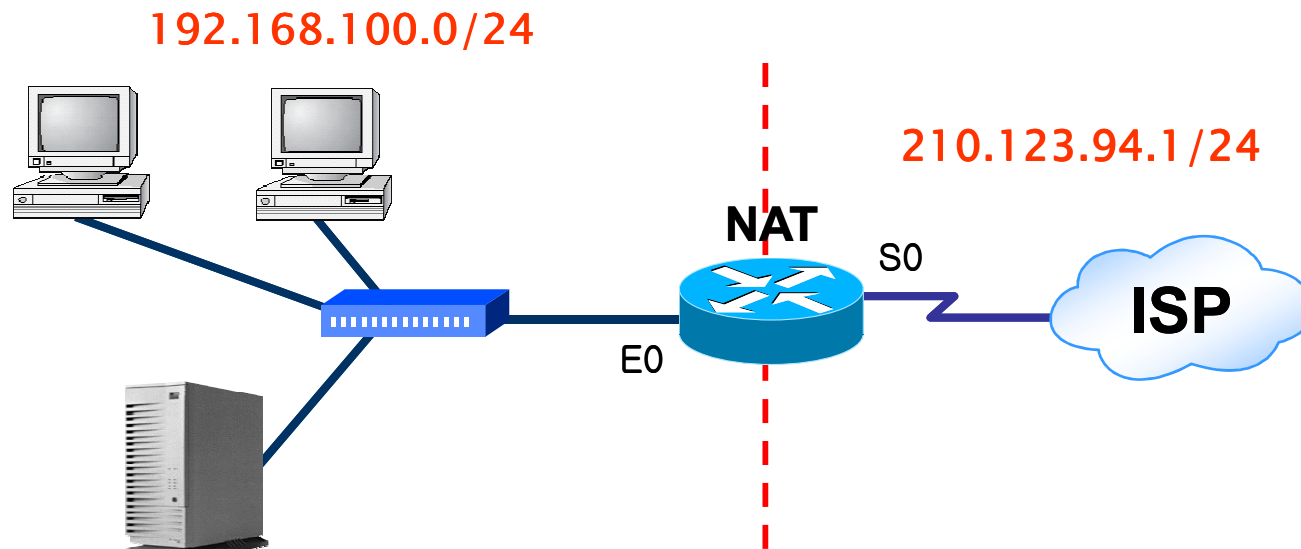
```
Router(config-if)#ip nat inside  
Router(config-if)#ip nat outside
```

Dynamic NAT LAB



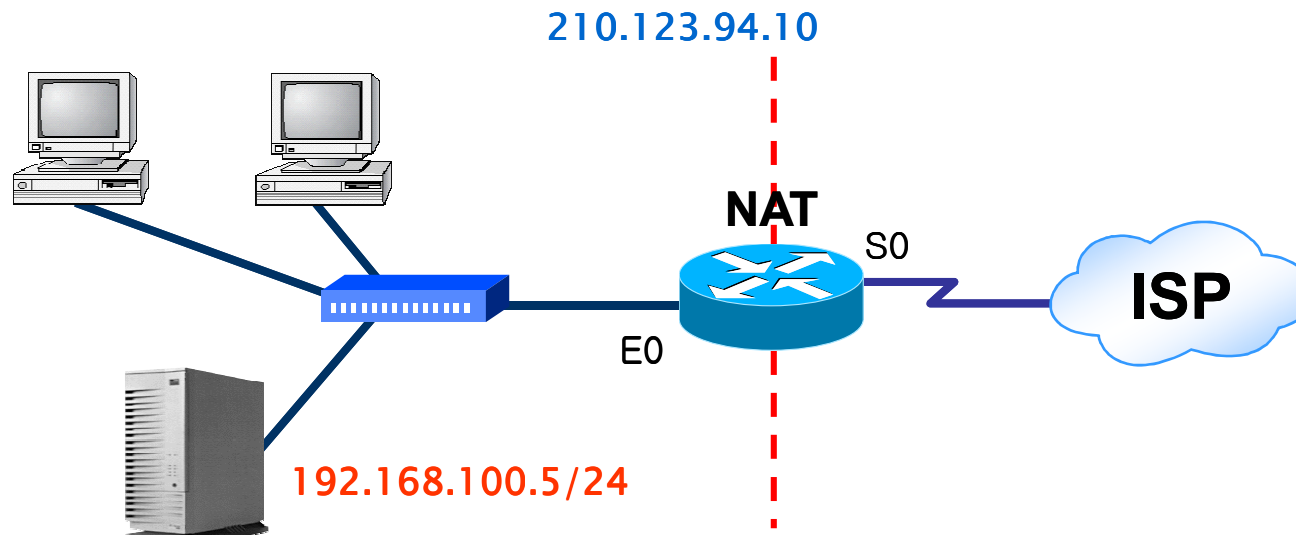
```
NAT(config)#ip nat pool Pub_IP 210.123.94.1 210.123.94.14 netmask 255.255.255.240
NAT(config)#access-list 50 permit 192.168.100.0 0.0.0.255
NAT(config)#ip nat inside source list 50 pool Pub_IP
NAT(config)#int e0
NAT(config-if)#ip nat inside
NAT(config-if)#int s0
NAT(config)#ip nat outside
```

NAT-PAT Example



```
NAT(config)#ip nat pool myhome 210.123.94.1 210.123.94.1 netmask 255.255.255.0
NAT(config)#access-list 50 permit 192.168.100.0 0.0.0.255
NAT(config)#ip nat inside source list 50 pool myhome overload
NAT(config)#int e0
NAT(config-if)#ip nat inside
NAT(config-if)#int s0
NAT(config)#ip nat outside
```

Static NAT Configuration



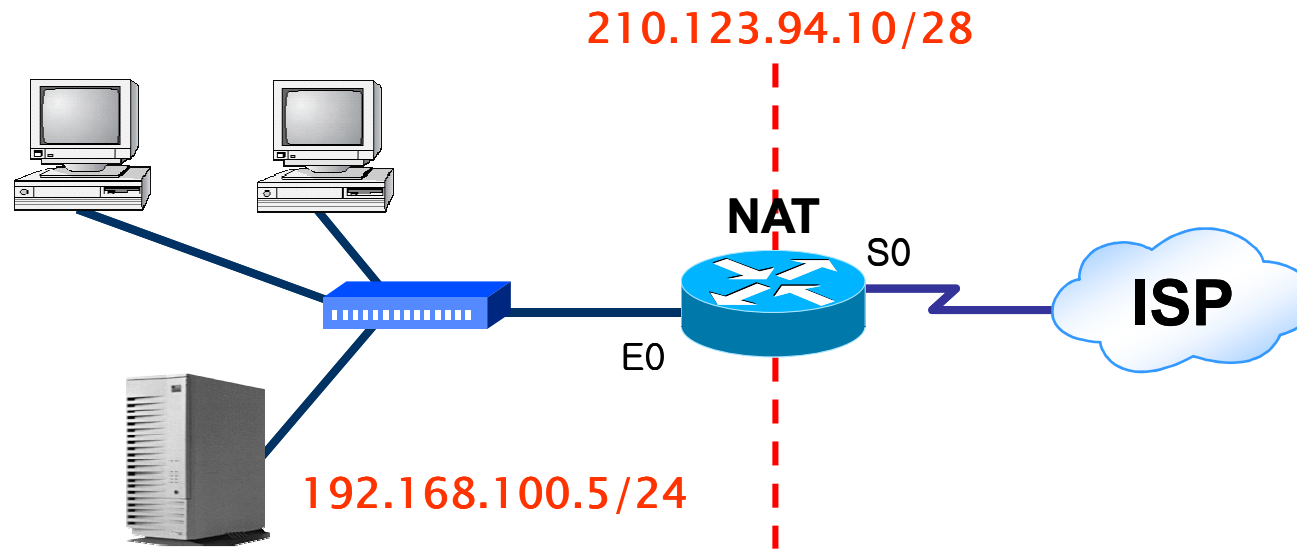
1. 정적 변환을 수립하기 위한 NAT 설정을 한다.

```
Router(config)#ip nat inside source Static local-ip global-ip
```

2. 각 인터페이스로 이동후 내부와 외부를 각각 설정한다.

```
Router(config-if)#ip nat inside  
Router(config-if)#ip nat outside
```

Static NAT Example



```
NAT(config)#ip nat inside source static 192.168.100.5 210.123.94.10
```

```
NAT(config)#int e0
```

```
NAT(config-if)#ip nat inside
```

```
NAT(config-if)#int s0
```

```
NAT(config)#ip nat outside
```



NAT Table 보기

R1#sh ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
udp	210.1.0.1:1438	10.10.10.1:1438	210.1.1.2:69	210.1.1.2:69
udp	210.1.0.1:1439	10.10.10.2:1438	210.1.1.2:69	210.1.1.2:69

R1#sh ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
	210.1.0.11	10.10.10.2		



NAT Monitoring

1. NAT Table에서 동적 변환 주소 엔트리를 제거하기

Router#clear ip nat translation *

2. 내부 및 외부 변환을 모두 포함하는 단순 동적 변환 주소 엔트리를 제거하기

Router#clear ip nat translation inside global-ip local-ip outside global-ip local-ip

3. 활성화된 변환 정보보기

Router#show ip nat translation [verbose]

4. 변환된 통계정보 보기

Router#show ip nat statistics

5. NAT 변환 상태 모니터링

Router#debug ip nat
