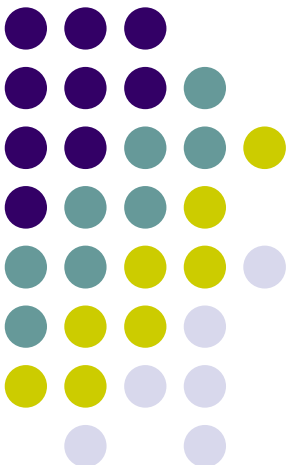
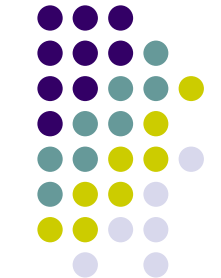
Υλοποίηση Δικτυακών Υποδομών και Υπηρεσιών: Access Lists

Πανεπιστήμιο Πελοποννήσου
Τμήμα Επιστήμης & Τεχνολογίας Τηλεπικοινωνιών

Ευάγγελος Α. Κοσμάτος

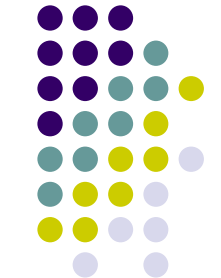


ACL Summary



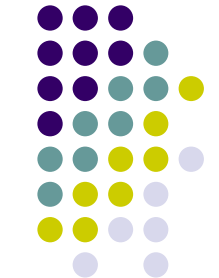
- Access lists perform several functions within a Cisco router, including:
  - Implement security / access procedures
  - Act as a protocol "firewall"
- Extended access lists allow filtering on address, protocol, and applications.
- Access lists are used to limit broadcast traffic.

ACCESS LISTS



- Filter the packet flows that flow in or out router interfaces.
- Help protect expanding network resources without impeding the flow of legitimate communication.
- Differentiate packet traffic into categories that permit or deny other features.

ACCESS LISTS

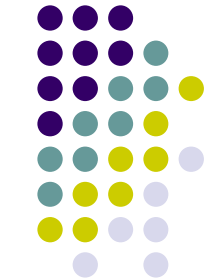


- You can also use access lists to:
  - Identify packets for priority or custom queuing
  - Restrict or reduce the contents of routing updates
- Access lists also process packets for other security features to:
  - Provide IP traffic dynamic access control with enhanced user authentication using the lock-and-key feature
  - Identify packets for encryption
  - Identify Telnet access allowed to the router virtual terminals

What are Access Lists?

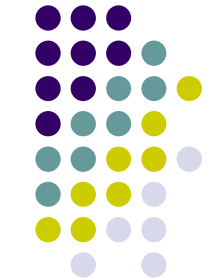
- Statements that specify conditions that an administrator sets so the router will handle the traffic covered by the access list in an out-of-the ordinary manner.
- Give added control for processing the specific packets in a unique way.
- Two main types of access lists are:
  - Standard
  - Extended

Standard Access Lists



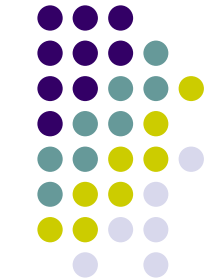
- Standard access lists for IP check the source address of packets that could be routed.
- The result permits or denies output for an entire protocol suite, based on the network/subnet/host address.

Standard Access Lists



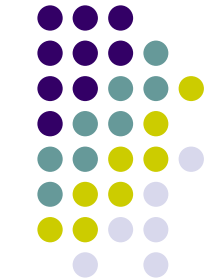
- Packets coming in E0 are checked for address and protocol. If permitted, the packets are output through S0, which is grouped to the access list.
- If the packets are denied by the standard access list, all these packets for the given category are dropped.

Extended Access Lists



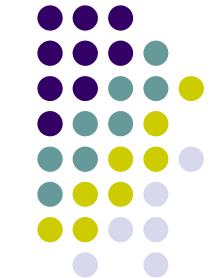
- Check for both source and destination packet addresses.
- Also can check for specific protocols, port numbers, and other parameters.
- Also permits or denies with more granularity.

Extended Access Lists



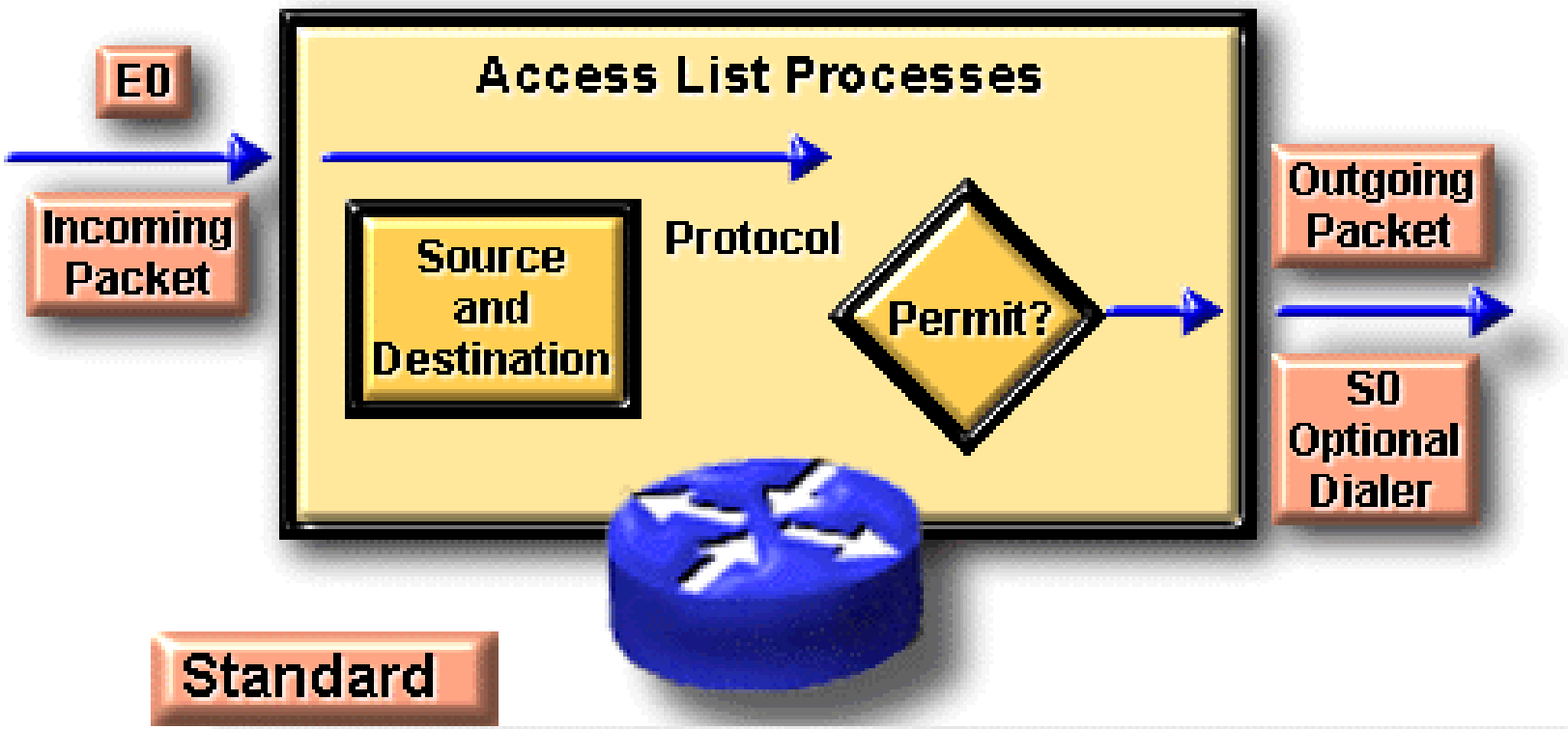
- Check for specific protocols, port numbers, and other parameters.
  - This allows administrators more flexibility to describe what checking the access list will do. Packets can be permitted or denied output based on where the packet originated and on its destination.

Extended Access Lists



- Also permits or denies with more granularity.
  - For example, it can allow electronic mail traffic from E0 to specific S0 destinations, while denying remote logins or file transfers

What Are Access Lists?



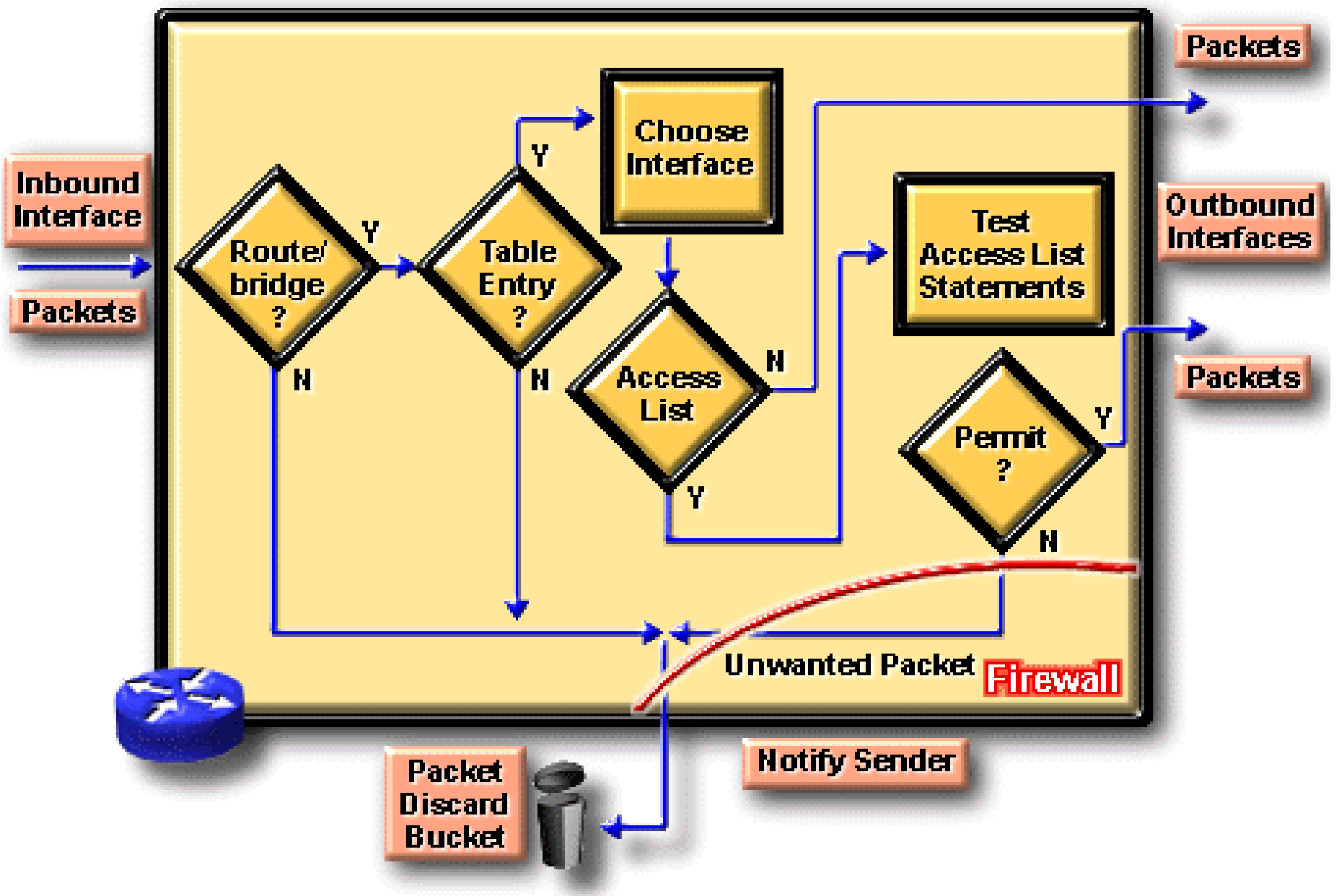
Standard

- Simpler address specifications
- Generally permits or denies entire protocol suite

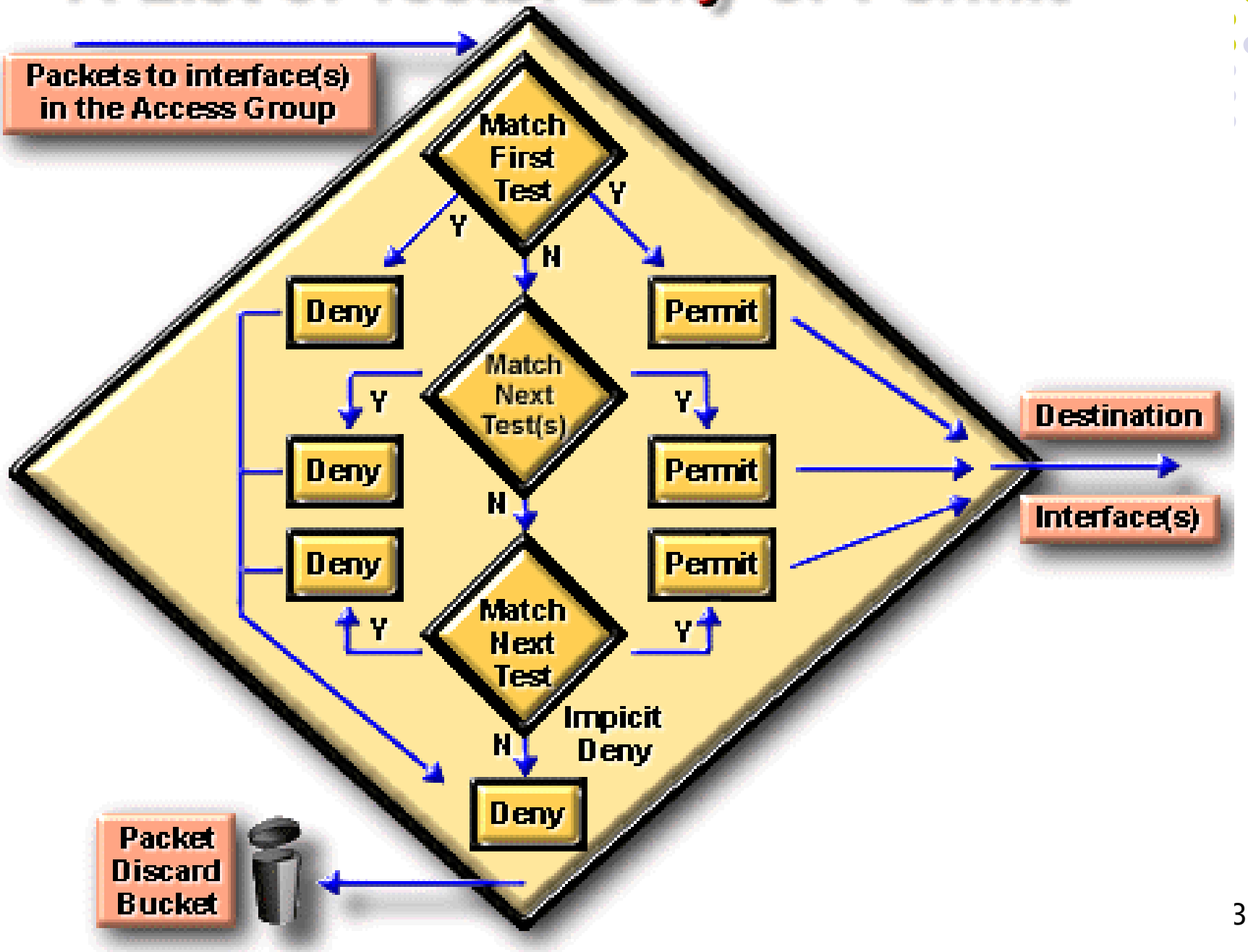
Extended

- More complex address specifications
- Generally permits or denies specific protocols

How Access Lists Work



A List of Tests: Deny or Permit



A List of Tests: Deny or Permit

- Access list statements operate in sequential, logical order.
- Evaluate packets from the top down.
- If a packet header and access list statement match, the packet skips the rest of the statements.
- If a condition match is true, the packet is permitted or denied. There can be only one access list per protocol per interface.

Deny Any Statement

- For logical completeness, an access list must have conditions that test true for all packets using the access list.
- A final implied statement (DENY ANY) covers all packets for which conditions did not test true.
- This final test condition matches all other packets. It results in a deny.
- Instead of proceeding in or out an interface, all these remaining packets are dropped.

Access List Command Overview

- In practice, access list commands can be lengthy character strings.
- Access lists can be complicated to enter or interpret.
- However, you can simplify understanding the general access list configuration commands by reducing the commands to two general elements

Access List Command Overview

Step 1: Set parameters for this access list test statement (which can be one of several statements)

```
Router(config)#
access-list access-list-number {permit I deny} {test conditions}
```

Step 2: Enable an interface to become part of the group that uses the specified access list

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

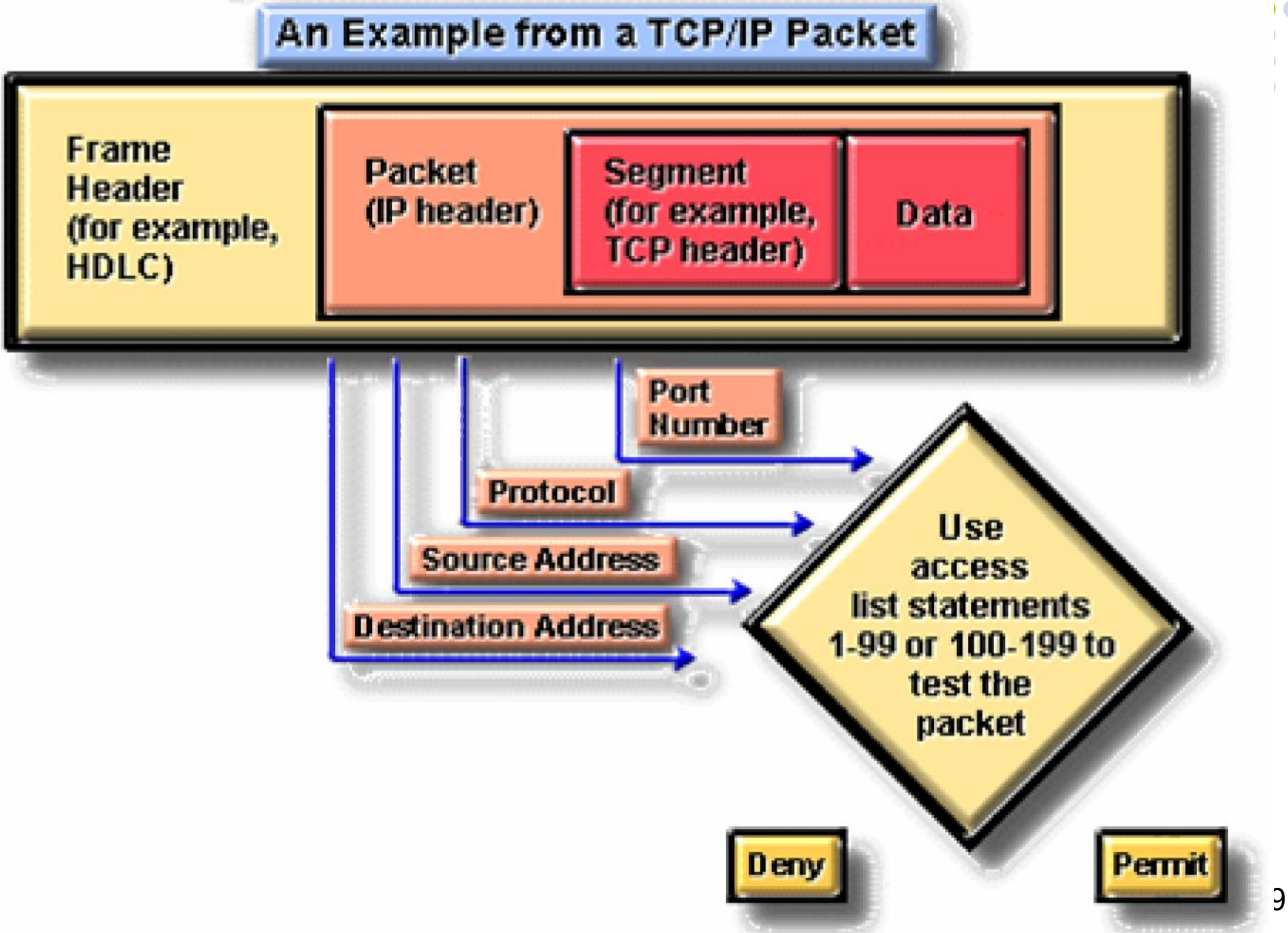
- Access lists are numbered (for IP, numbered or named)

How to Identify Access Lists

Access List Type	Number Range/Identifier
IP Standard Extended	1-99 100-199 Named (Cisco IOS 11.2 and later)
IPX Standard SAP filters	800-899 1000-1099
Apple Talk	600-699

- Number identifies the protocol and type
- Other number ranges for most protocols

Testing Packets with Access Lists



Key Concepts for IP Access Lists

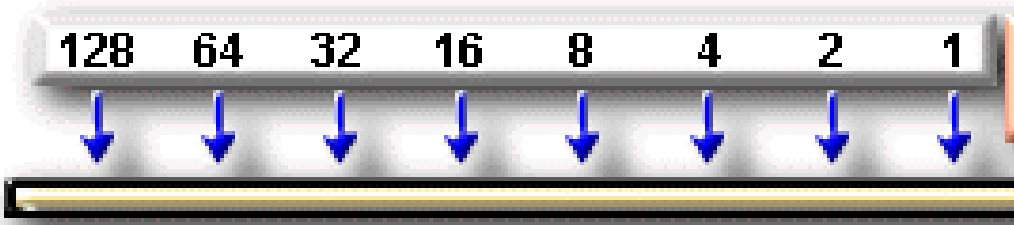
Standard lists (1 to 99) test conditions of all IP packets from source addresses

Extended lists (100 to 199) can test conditions of

- Source and destination addresses
- Specific TCP/IP-suite protocols
- Destination ports

Wildcard bits indicate how to check the corresponding address bits (0=check, 1=ignore)

How to Use Wildcard Mask Bits

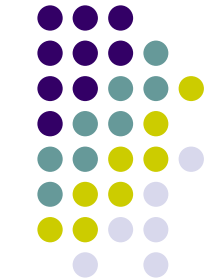


Octet bit position and address value for bit

									Examples
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)

- 0 means check corresponding bit value
- 1 means ignore value of corresponding bit

Wildcard Mask Bits



- IP access lists use wildcard masking.
- Wildcard Masking for IP address bits uses the number 1 and the number 0 to identify how to treat the corresponding IP address bits.
  - A wildcard mask bit 0 means “check the corresponding bit value.”
  - A wildcard mask bit 1 means “do not check (ignore) that corresponding bit value.”

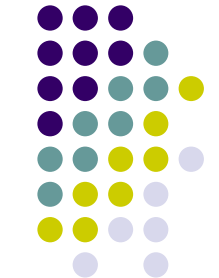
Wildcard Masks

- By carefully setting wildcard masks,
  - an administrator can select single or
  - several IP addresses for permit or deny tests.
  - Refer to the example in the graphic.

Wildcard Masks

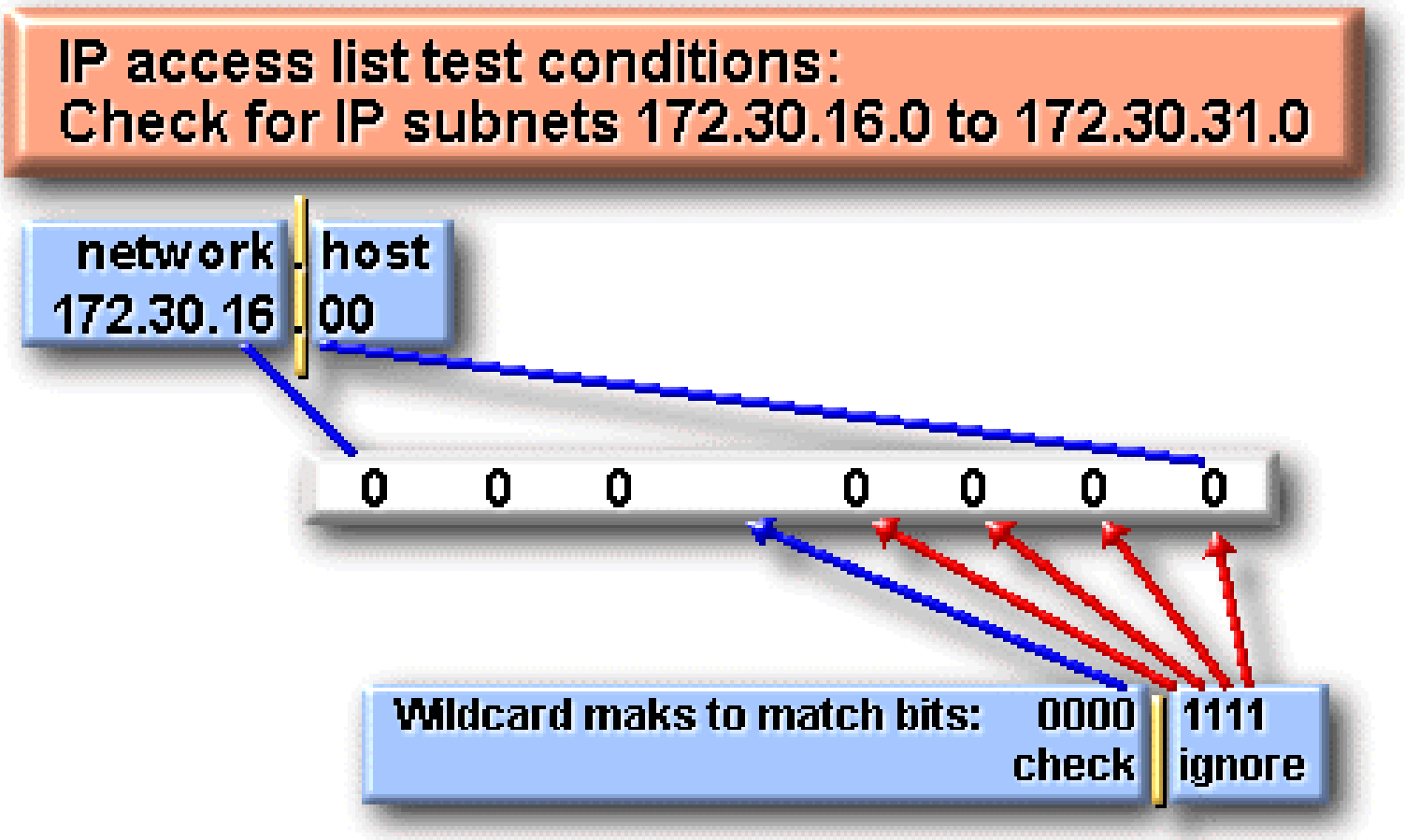
- Wildcard masking for access lists operates differently from an IP subnet mask.
- A **zero** in a bit position of the access list mask indicates that the corresponding bit in the address must be checked;
- A **one** in a bit position of the access list mask indicates the corresponding bit in the address is not “interesting” and can be ignored.

Wildcard Masking



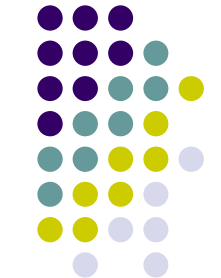
- An administrator wants to test an IP address for sub-nets that will be permitted or denied.
- Assume the IP address is Class B (first two octets are the network number) with eight bits of sub-netting (the third octet is for sub-nets).
- The administrator wants to use IP wildcard masking bits to match sub-nets 172.30.16.0 to 172.30.31.0

How to Use Wildcard Mask Bits (cont.)



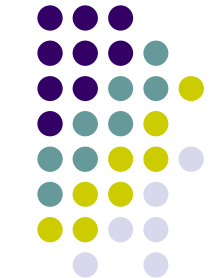
- Address and wildcard mask: 172.30.16.0 0.0.15.255

Wildcard Masking



- In this example,
- the address 172.30.16.0
- with the wildcard mask 0.0.15.255
- matches subnets 172.30.16.0 to 172.30.31.0

Standard ACL (1-99)



- **Access-list list# {permit/deny} source IP [wildcard mask]**
- **interface [router port]**
- **ip access-group [list#] in|out (out is the default)**
- **no ip access-group [list#] in|out (disable the access list)**
- If a match is made, the action defined in this access list statement is performed.
- If no match is made with an entry in the access list, the deny action is performed (implicit deny)

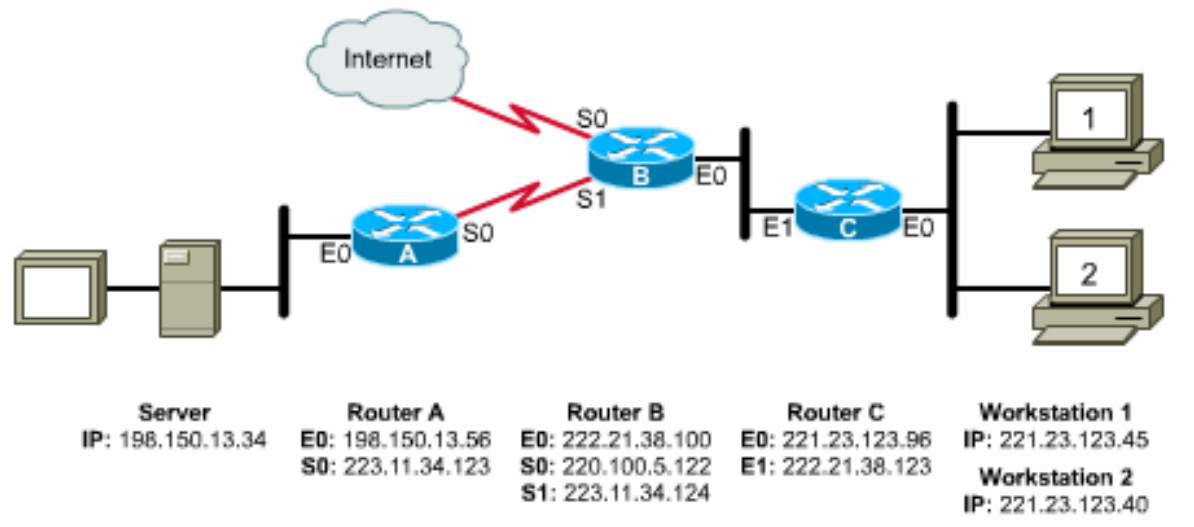
- Should be put close to the destination address because you can not specify the destination address.

Extended ACL (100-199)

- **Access-list list# {permit/deny} protocol source [source mask] destination [destination mask] operator [port]**
- Should be put close to the source
- **e.g.: access-list 112 permit tcp 195.143.2.0 0.0.0.255 195.143.3.2 0.0.0.0 eq smtp**
- **or: access-list 112 permit tcp 195.143.2.0 0.0.0.255 host 195.143.3.2 eq smtp**

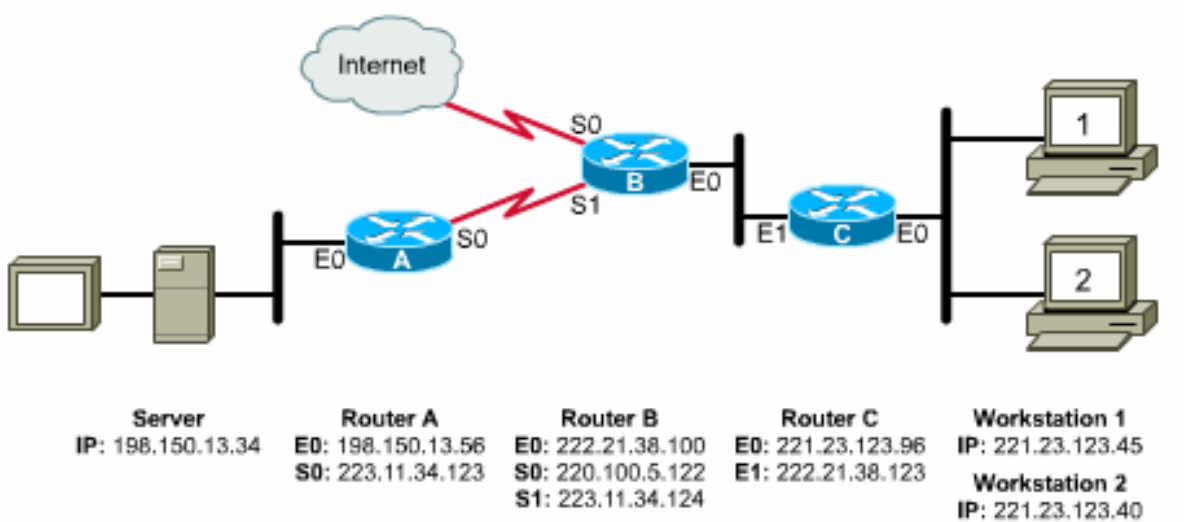
Correct Placement of Extended ACLs

- Since extended ACLs have destination information, you want to place it as close to the source as possible.
- Place an extended ACL on the first router interface the packet enters and specify inbound in the **`access-group`** command.

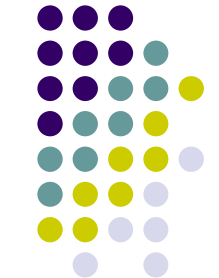


Correct Placement of Extended ACLs

- In the graphic below, we want to deny network 221.23.123.0 from accessing the server 198.150.13.34.
- What router and interface should the access list be applied to?
  - Write the access list on Router C, apply it to the E0, and specify in
  - This will keep the network free of traffic from 221.23.123.0 destined for 198.150.13.34 but still allow 221.23.123.0 access to the Internet



Example



- Configure an access list that blocks network 210.93.105.0 from exiting serial port s0 on some router. Allow all other to pass.
- access-list 4 deny 210.93.105.0 0.0.0.255

  access-list 4 permit any

  interface s0

  ip access-group 4

Example (continued)

- Same example but would like to block only the **first half IP** of the network.
- access-list 4 deny 210.93.105.0 0.0.0.127

  access-list 4 permit any

  interface s0

  ip access-group 4

Example (continued)

- Same example but would like to block only the **second half IP** of the network.
- access-list 4 deny 210.93.105.128 0.0.0.127

  access-list 4 permit any

  interface s0

  ip access-group 4

Example (continued)

- Same example but would like to block only the **even numbered IP** of the network.
- access-list 4 deny 210.93.105.0 0.0.0.254

  access-list 4 permit any

  interface s0

  ip access-group 4

Example (continued)

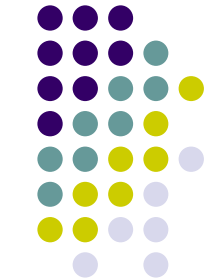
- Same example but would like to block only the **odd numbered IP** of the network.
- access-list 4 deny 210.93.105.1 0.0.0.254

  access-list 4 permit any

  interface s0

  ip access-group 4

Time Savers: the any command

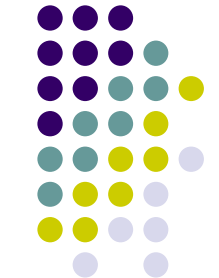


- Since ACLs have an implicit “deny any” statement at the end, you must write statements to permit others through.
  - **`Lab-A(config)#access-list 1 deny 192.5.5.0 0.0.0.127`**
  - **`Lab-A(config)#access-list 1 permit 0.0.0.0 255.255.255.255`**
- Since the last statement is commonly used to override the “deny any,” Cisco gives you an option--the **`any`** command:
  - **`Lab-A(config)#access-list 1 permit any`**

Time Savers: the `host` command

- Many times, a network administrator will need to write an ACL to permit a particular host (or deny a host). The statement can be written in two ways. Either...
  - **`Lab-A(config)#access-list 1 permit 192.5.5.10 0.0.0.0`**
- or...
  - **`Lab-A(config)#access-list 1 permit host 192.5.5.10`**

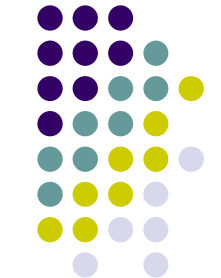
Ext. ACL Misc



- Port accounting
- access-list 106 permit udp any any
  - eq Match only packets on a given port number
  - fragments Check non-initial fragments
  - gt Match only packets with a greater port number
  - log Log matches against this entry
  - log-input Log matches against this entry, incl. input interface
  - lt Match only packets with a lower port number
  - neq Match only packets not on a given port number
  - precedence Match packets with given precedence value
  - range Match only packets in the range of port numbers
  - tos Match packets with given TOS value

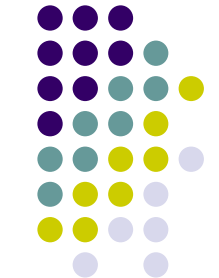
```
Router(config)#access-list access-list-number {permit/deny}{test-conditions}
```

Ext. ACL Misc. cnt.



- TCP header fields
- access-list 106 permit udp any any
  - ack Match on the ACK bit
  - eq Match only packets on a given port number
  - established Match established connections
  - fin Match on the FIN bit
  - fragments Check non-initial fragments
  - gt Match only packets with a greater port number
  - log Log matches against this entry
  - log-input Log matches against this entry, incl. input interface
  - lt Match only packets with a lower port number
  - neq Match only packets not on a given port number
  - precedence Match packets with given precedence value
  - psh Match on the PSH bit
  - range Match only packets in the range of port numbers
  - rst Match on the RST bit
  - syn Match on the SYN bit
  - tos Match packets with given TOS value
  - urg Match on the URG bit

Naming ACLs



- One nice feature in the Cisco IOS is the ability to name ACLs. This is especially helpful if you need more than 99 standard ACLs on the same router.
- Once you name an ACL, the prompt changes and you no longer have to enter the **`access-list`** and ***`access-list-number`*** parameters.
- In the example below, the ACL is named **`over_and`** as a hint to how it should be placed on the interface--**`out`**

```
Lab-A(config)# ip access-list standard over_and
Lab-A(config-std-nacl)#deny host 192.5.5.10
.........
Lab-A(config-if)#ip access-group over_and out
```

Verifying ACLs

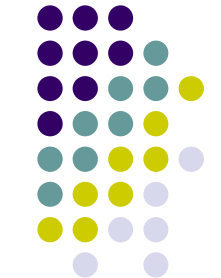
- Show commands:
  - **`show access-lists`**
    - shows all access-lists configured on the router
  - **`show access-lists {`*`name`*` | `*`number`*`}`**
    - shows the identified access list
  - **`show ip interface`**
    - shows the access-lists applied to the interface--both inbound and outbound.
  - **`show running-config`**
    - shows all access lists and what interfaces they are applied on

Enhanced Access Lists

Cisco routers support several enhanced types of access lists:

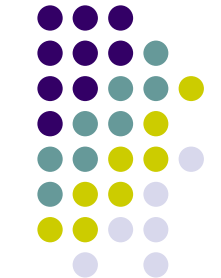
- Time-Based—Access lists whose statements become active based upon the time of day and/or day of the week.
- Reflexive—Create dynamic openings on the untrusted side of a router based on sessions originating from a trusted side of the router.
- Dynamic (Lock and Key)—Create dynamic entries.
- Context-Based Access Control (CBAC)—Allows for secure handling of multi-channel connections based on upper layer information.

Extended ACL



- Time based
  - (conf)# time-range bar
    (conf-time-range)# periodic daily 10:00 to 13:00
    (conf-time-range)# ip access-list tin
    (config-ext-nacl)# deny tcp any any eq www time-range bar
    (config-ext-nacl)# permit ipv6 any any

Useful Ports TCP



- FTP data: 20
- FTP control: 21
- Telnet: 23
- SMTP: 25
- WWW: 80
- POP3: 110

Extended ACLs command syntax format (1)

IP

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
{deny | permit} protocol source source-wildcard
destination destination-wildcard [precedence precedence]
[tos tos] [log | log-input] [time-range time-range-name]
```

ICMP

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
{deny | permit} icmp source source-wildcard
destination destination-wildcard
[icmp-type | [[icmp-type icmp-code] | [icmp-message]]
[precedence precedence] [tos tos] [log | log-input]
[time-range time-range-name]
```

Extended ACLs command syntax format (2)

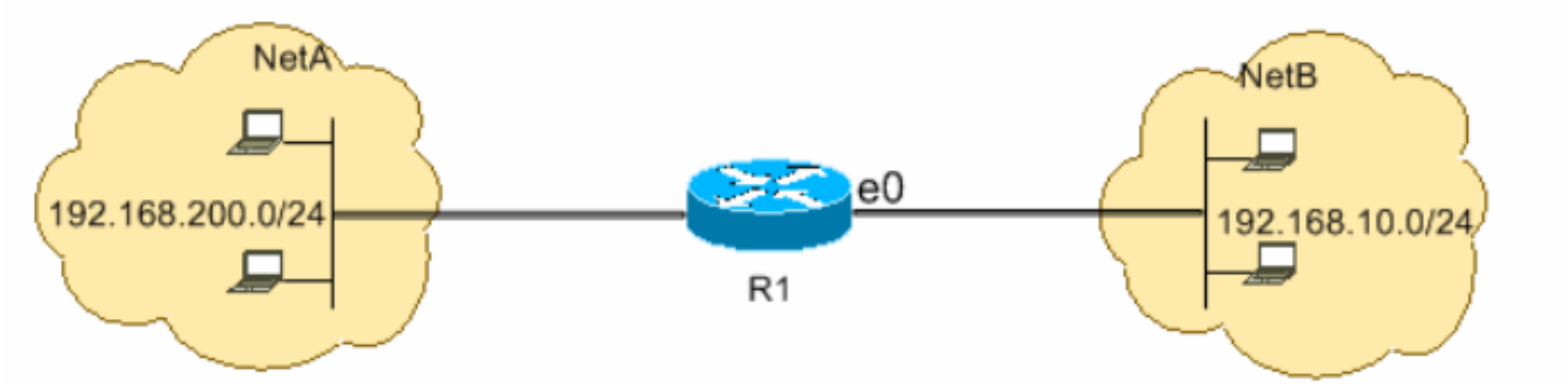
TCP

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
{deny | permit} tcp source source-wildcard [operator [port]]
destination destination-wildcard [operator [port]] [established]
[precedence precedence] [tos tos] [log | log-input]
[time-range time-range-name]
```

UDP

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
{deny | permit} udp source source-wildcard [operator [port]]
destination destination-wildcard [operator [port]]
[precedence precedence] [tos tos] [log | log-input]
[time-range time-range-name]
```

Ex1:Allow Access to a Range of Contiguous IP Addresses




```
hostname R1
!
interface ethernet0
ip access-group 101 in
!
access-list 101 permit ip 192.168.10.0  0.0.0.255
  192.168.200.0  0.0.0.255
```

Ex2: Allow HTTP, Telnet, Mail, POP3, FTP

```
hostname R1
!
interface ethernet0
ip access-group 102 in
!
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq telnet
access-list 102 permit tcp any any eq smtp
access-list 102 permit tcp any any pop3
access-list 102 permit tcp any any eq 21
access-list 102 permit tcp any any eq 20
```

Ex3: Allow Pings (ICMP)

```
hostname R1
!
interface ethernet0
ip access-group 102 in
!
access-list 102 permit icmp any any echo-reply
```

Ex4: Deny FTP Traffic (TCP, Port 21)

```
hostname R1
!
interface ethernet0
ip access-group 102 in
!
access-list 102 deny tcp any any eq ftp
access-list 102 deny tcp any any eq ftp-data
access-list 102 permit ip any any
```