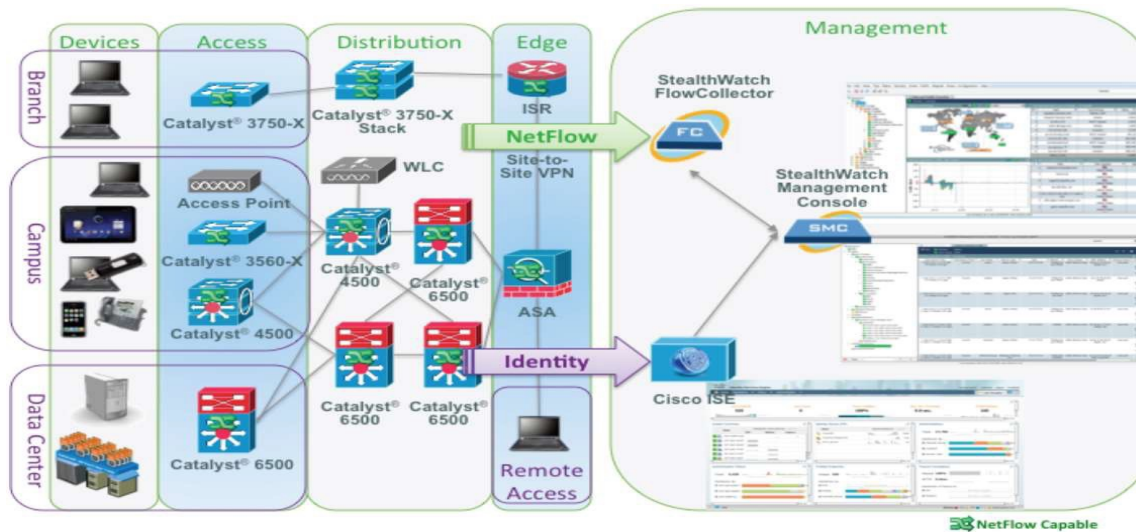


CIS 3250

Access Control Lists



Wildcard Mask

- A **wildcard mask** is a string of bits telling a router which part of an IP address to look at.
- The one bits in a wildcard mask indicate which bits are “don’t care” bits and can assume any value.
 - Used by routing protocols to define the networks or subnets to be advertised
 - Used by Access Control Lists (ACLs) to define addresses or address ranges
- They allow an ACL to filter on individual addresses, subnets, or ranges of addresses.

How does a Wildcard Mask Work?

- They are 32 bits, just like IPv4 addresses.
- A 0 in the mask means the corresponding bit in the associated IP address needs to match.
- A 1 in the mask means the corresponding bit in the associated IP address does not need to match or can take any value ("don't care").

Target

- The wildcard mask can target:
- A single host or IP address (e.g., 0.0.0.0)
- An entire network
- A subnet
- A range of addresses
- All even or all odd networks

Example 1 – Match All bits

192								168								1				1															
1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	0	0	0		0	0	0	1	.	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0	.	0	0	0	0		0	0	0	0	.	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	0	0	0		0	0	0	1	.	0	0	0	0	0	0	0	1

- A mask of 0.0.0.0 targets a single host: 192.168.1.1 (in this case)

Example 2 – Match Zero Bits

192									168									1					1												
1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	0	0	0		0	0	0	1	.	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0

- A mask of 255.255.255.255 targets all hosts (the associated IP address is irrelevant)

Example 3 – Match Some Bits

192								168								1				1															
1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	0	0	0		0	0	0	1	.	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0	.	1	1	1	1	1	1	1	1	
1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	0	0	0	0	0	0	1	.	0	0	0	0	0	0	0	0	

- A mask of 0.0.0.255 targets a range of IP address 192.168.1.0 to 192.168.1.255 (i.e., the 192.168.1.0/24 network)

Range of Addresses

- What if you wanted to filter on a range of addresses?
- 192.168.16. 0 – 192.168.31.255

192.168.16.0 – 192.168.31.255

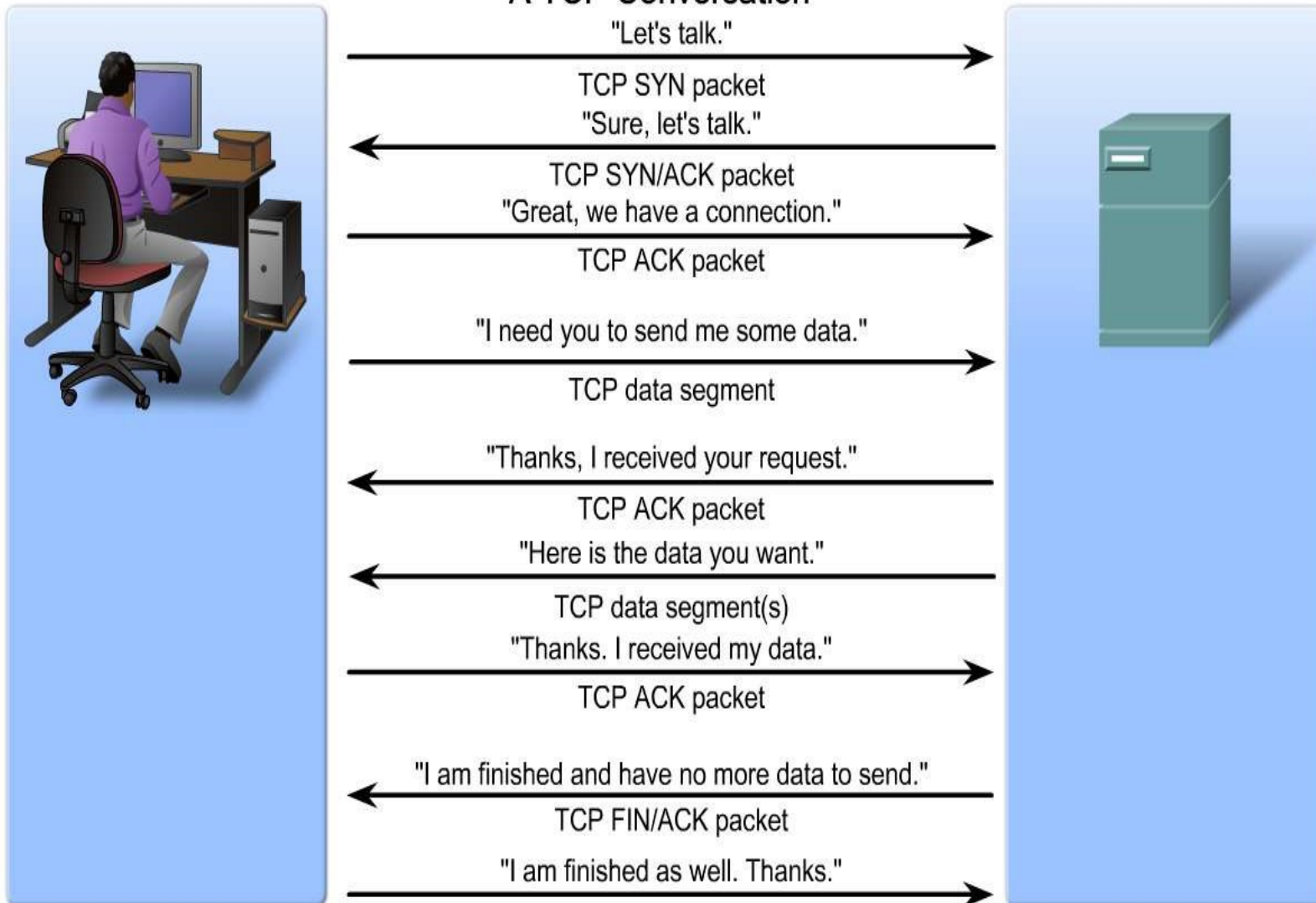
192								168								16				0														
1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	0	0	1	0	0	0	0	.	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0	.	0	0	0	0	1	1	1	1	.	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	0	0	1	1	1	1	.	1	1	1	1	1	1	1	1	

- Targets a range of addresses
- 192.168.16.0 is the first match
- 192.168.31.255 is the last match

Access Control Lists - ACLs

- ACLs can be used to filter traffic based on network or host addresses
- ACLs can also filter based on the TCP port being used

TCP Conversation



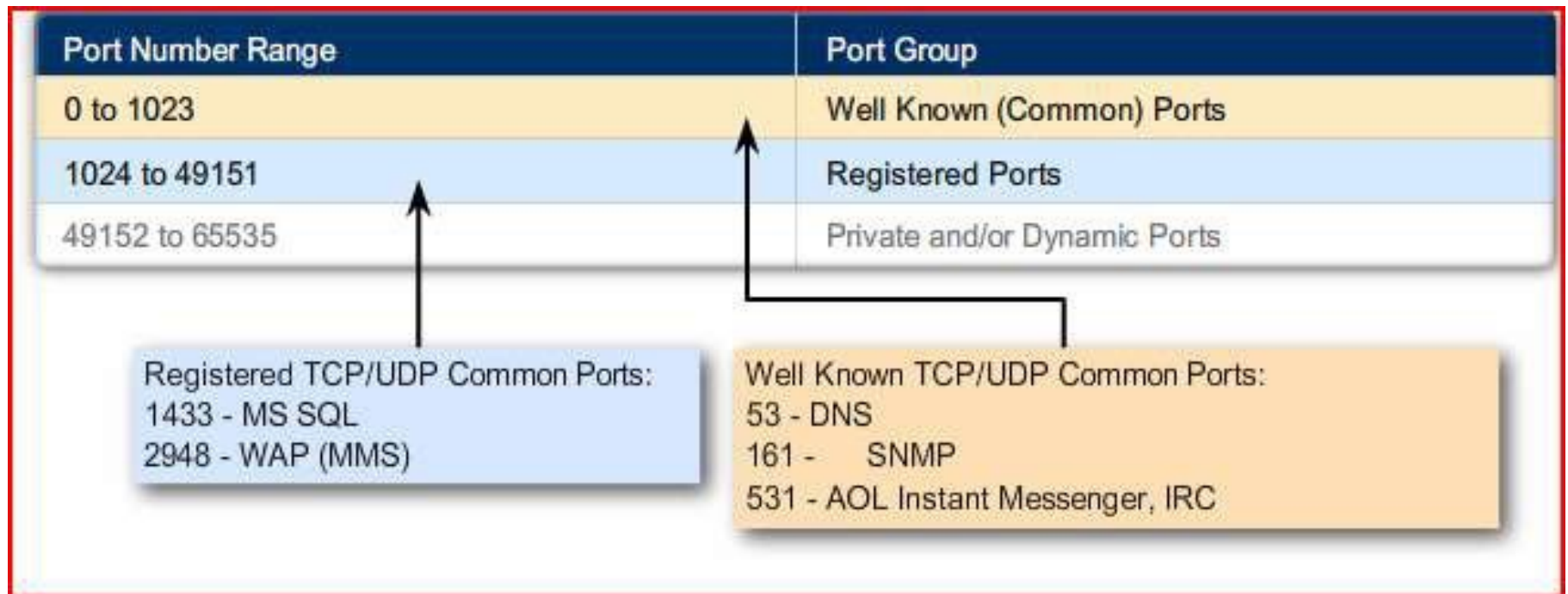
TCP Port Numbers

Port Number Range	Port Group
0 to 1023	Well Known (Common) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

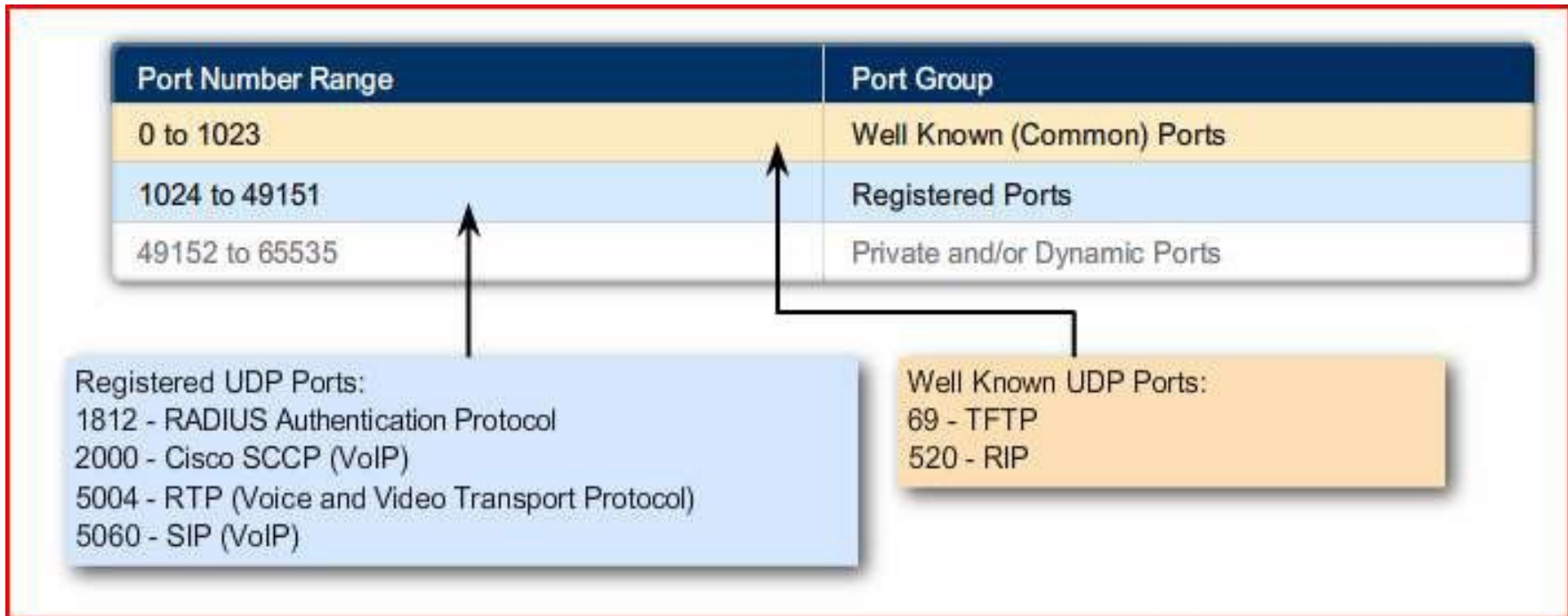
Registered TCP Ports: 1863 - MSN Messenger 8008 - Alternate HTTP 8080 - Alternate HTTP	Well Known TCP Ports 21 - FTP 23 - Telnet 25 - SMTP 80 - HTTP 110 - POP3 194 - Internet Relay Chat (IRC) 443 - Secure HTTP (HTTPS)
---	---

The diagram illustrates the classification of TCP port numbers. It features a table with three rows: '0 to 1023' (Well Known (Common) Ports), '1024 to 49151' (Registered Ports), and '49152 to 65535' (Private and/or Dynamic Ports). Below the table, two callout boxes provide examples. The 'Registered TCP Ports' box (light blue) lists 1863 (MSN Messenger), 8008 (Alternate HTTP), and 8080 (Alternate HTTP). The 'Well Known TCP Ports' box (light orange) lists 21 (FTP), 23 (Telnet), 25 (SMTP), 80 (HTTP), 110 (POP3), 194 (Internet Relay Chat (IRC)), and 443 (Secure HTTP (HTTPS)). Arrows point from the callout boxes to their respective rows in the table.

TCP/UDP Common Ports



UDP Port Numbers



More Info: http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

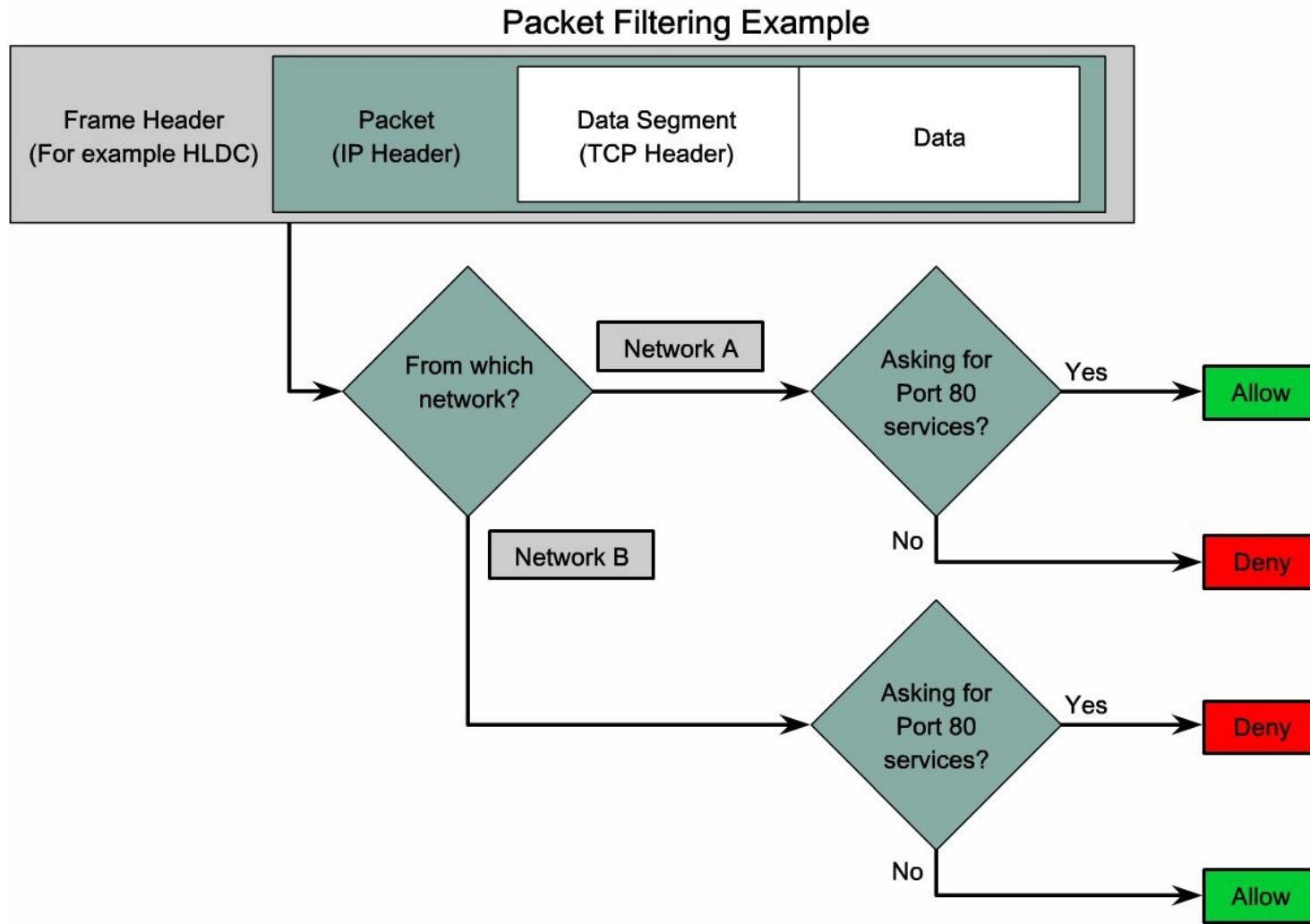
Packet Filtering

- Packet filtering, sometimes called static packet filtering, controls access to a network by analyzing the **incoming** and **outgoing** packets and passing or halting them based on stated criteria.
- A router acts as a **packet filter** when it forwards packets according to **filtering rules**.
- When a packet arrives at the packet-filtering router, the router extracts certain information from the packet header and **decides, according to the filter rules,** whether the packet can pass through or be discarded.
- Packet filtering works at the network and transport layers of the Open Systems Interconnection (OSI) model.

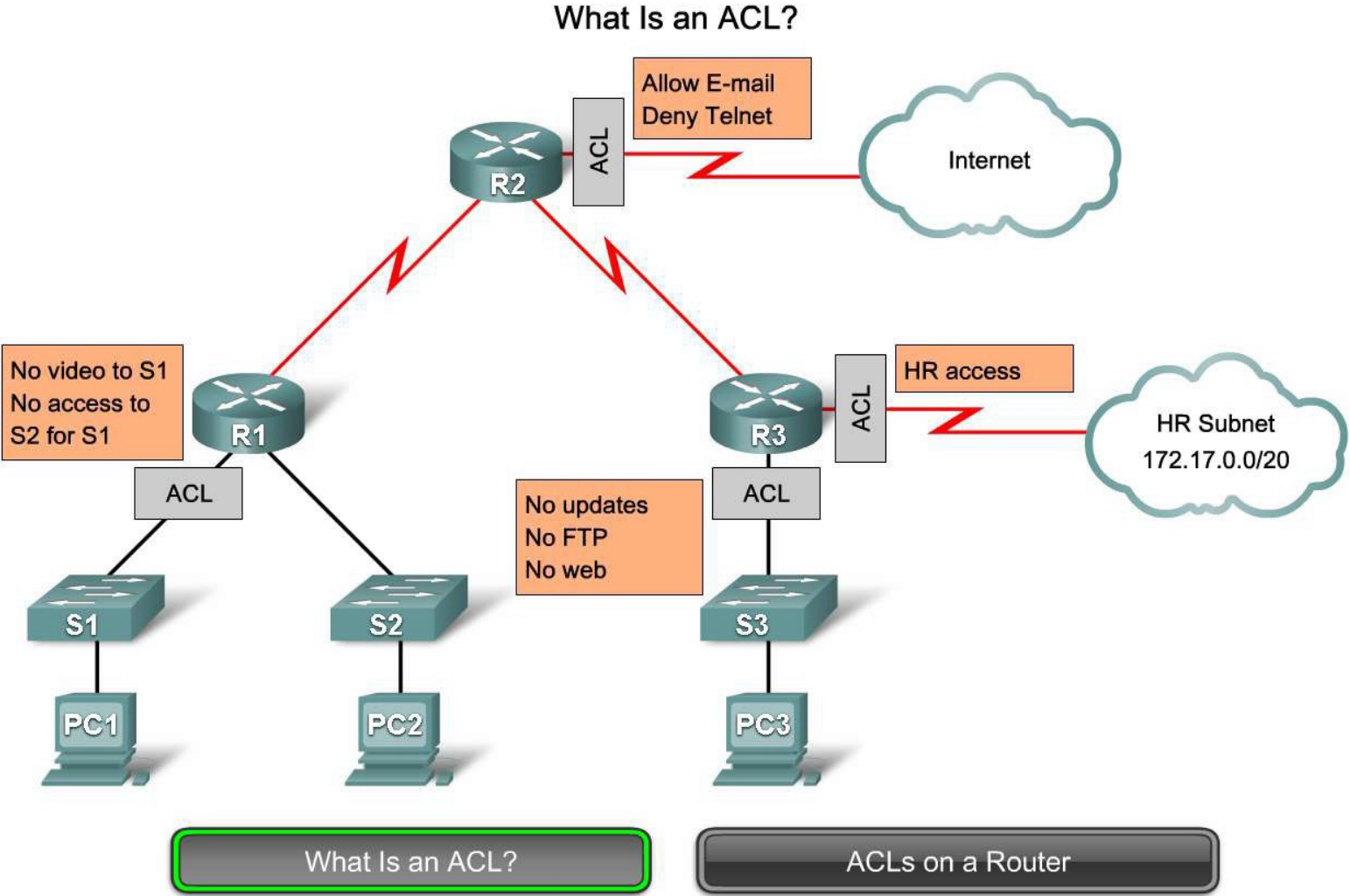
Access Control List - ACL

- An ACL is a **sequential list** of permit or deny statements that apply to IP addresses or upper-layer protocols. The ACL can extract information from the packet header, test it against its rules, and make "allow" or "deny" decisions based on the following.
 - Source IP address
 - Destination IP address
 - ICMP message type
- The ACL can also extract upper-layer information and test it against its rules. Upper-layer information includes the following.
 - TCP/UDP source port
 - TCP/UDP destination port

Allowing or Blocking Traffic



What is an ACL?



What is an ACL?

- An ACL is a **router configuration script** that controls whether a router permits or denies packets to pass based on criteria found in the packet header.
- ACLs are also used for selecting types of traffic to be analyzed, forwarded, or processed in other ways.
- As each packet comes through an interface with an ACL associated, **the ACL is checked from top to bottom, one line at a time**, looking for criteria that match the incoming packet.
- The ACL enforces one or more corporate security policies by applying a permit or deny rule to determine the packet's fate. ACLs can be configured to control access to a network, subnet, or host.

ACL Guidelines

- Use ACLs in firewall routers positioned between your internal and external networks.
- Use ACLs on a router positioned between two parts of your network to control traffic entering or exiting a specific part of your internal network.
- Configure ACLs on **border routers** - routers situated at the edges of your networks. This provides a basic buffer from the outside network or between a less controlled area of your network and a more sensitive area of your network.
- Configure ACLs for each network protocol configured on the border router interfaces. You can configure ACLs on an interface to filter **inbound traffic**, **outbound traffic**, or both.

ACLs vs Firewalls

- Firewalls are more complex than ACLs. They typically include packet filtering functions but also filter at higher levels in the protocol stack.
- For example, a firewall might allow only certain HTTP methods to pass. However, this requires dissecting the HTTP protocol.
- High-level filtering is more complex and entails more overhead.
- ACLs, in contrast, only operate at the network or transport layers and don't need any protocol context.
- ACLs are simpler, faster, with less overhead and can reasonably be done by routers

Which To Use?

- You can put a firewall at the edge of your network and let it do packet filter as well as high level filtering. This puts all your security configuration in one place (single point of failure, but also easier to manage and verify).
- You can let the routers do packet filtering, and use a firewall, if necessary, for only higher level operations.
- You can use packet filtering on routers throughout your network for more fine-grained control.
- You can do *both*: packet filtering in a firewall *and* in your routers. This is **defense in depth**.

ACL Traffic Filter Rules



With two interfaces and three protocols running, this router could have a total of 12 separate ACLs applied.

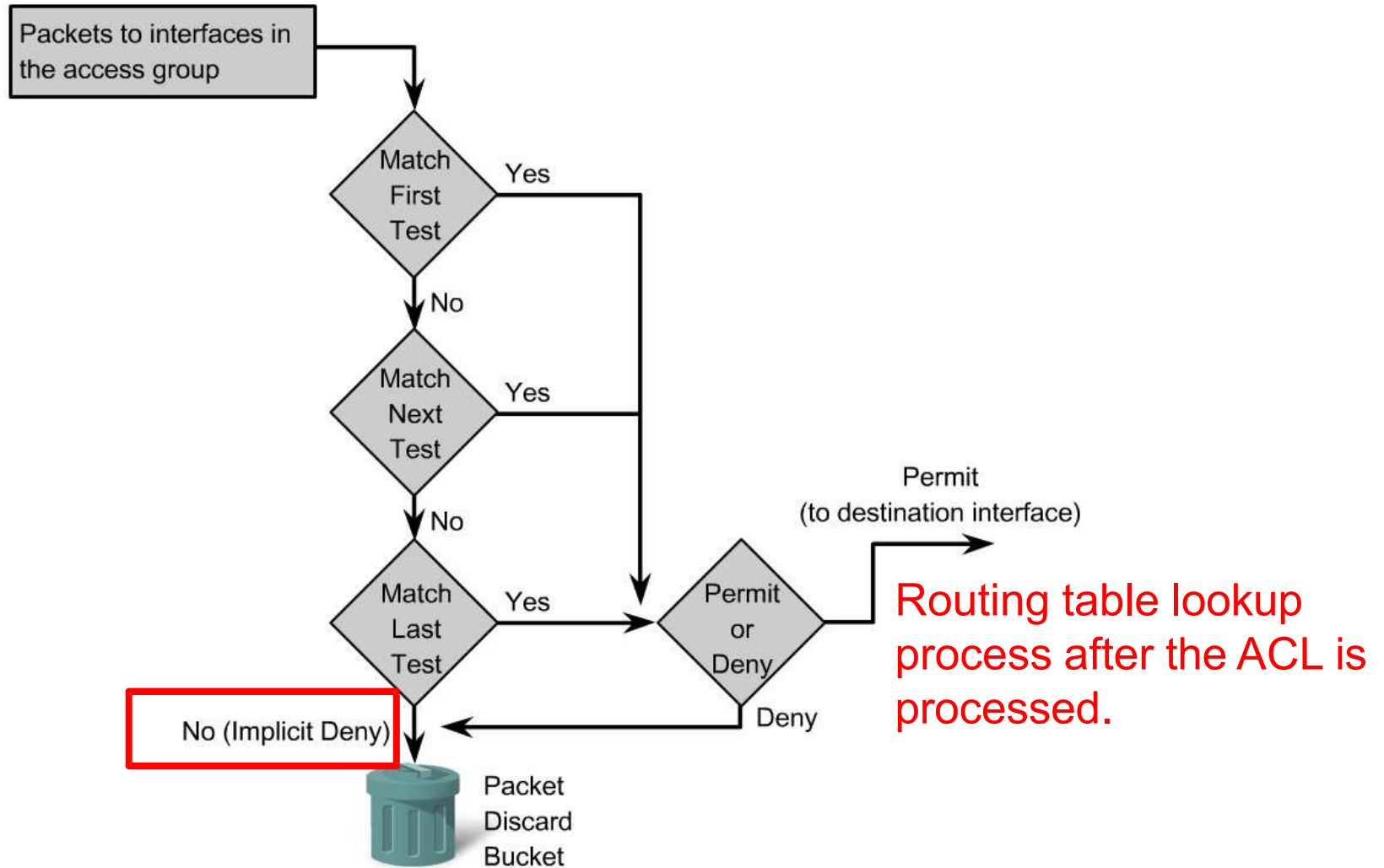
The three Ps for using ACLs

You can only have one ACL per protocol, per interface, and per direction:

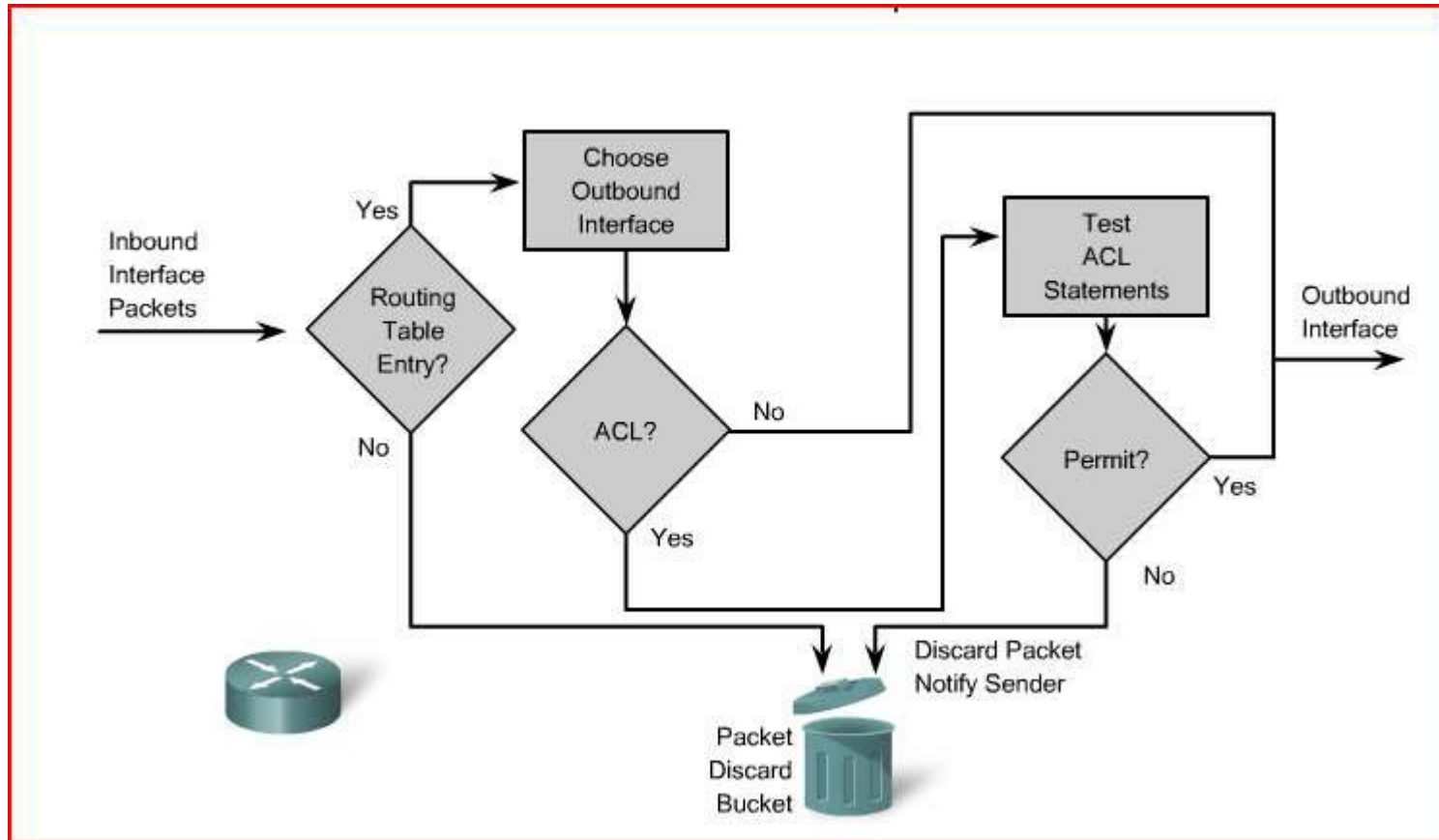
- One ACL per protocol (e.g., IP or IPX)
- One ACL per interface (e.g., FastEthernet0/0)
- One ACL per direction (i.e., IN or OUT)

ACL Operation – Inbound ACLs

How ACLs Work



Outbound ACL Example



Types of ACLs

- There are two types of ACLs
 - **Standard:** filter packets based on source IP *only*. These are much more limited.
 - **Extended:** filter packets based on multiple criteria like source and destination IP, source and destination ports, and more. These are much more flexible.
 - The distinction is probably Cisco-specific (older devices only supported standard ACLs).

Standard ACL Example

- Standard ACLs filter IP packets based on the source IP address only!

```
access-list 10 permit 192.168.30.0 0.0.0.255
```

- Each ACL has either a number or a name (the above is number 10).
- Each ACL consists of a list of access control entries (the above has one entry).
- Each ACE is “permit” or “deny”.
- Standard ACLs have entries that only mention source IP address (range).

Extended ACL Example

```
access-list 10 permit 192.168.30.0 0.0.0.255
```

Extended ACLs filter IP packets based on several attributes, including the following:

- Source and destination IP addresses
- Source and destination TCP and UDP ports
- Protocol type (IP, ICMP, UDP, TCP, or protocol number)

```
access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80
```

Numbering and Naming ACLs

Numbered ACL:

You assign a number based on which protocol you want filtered:

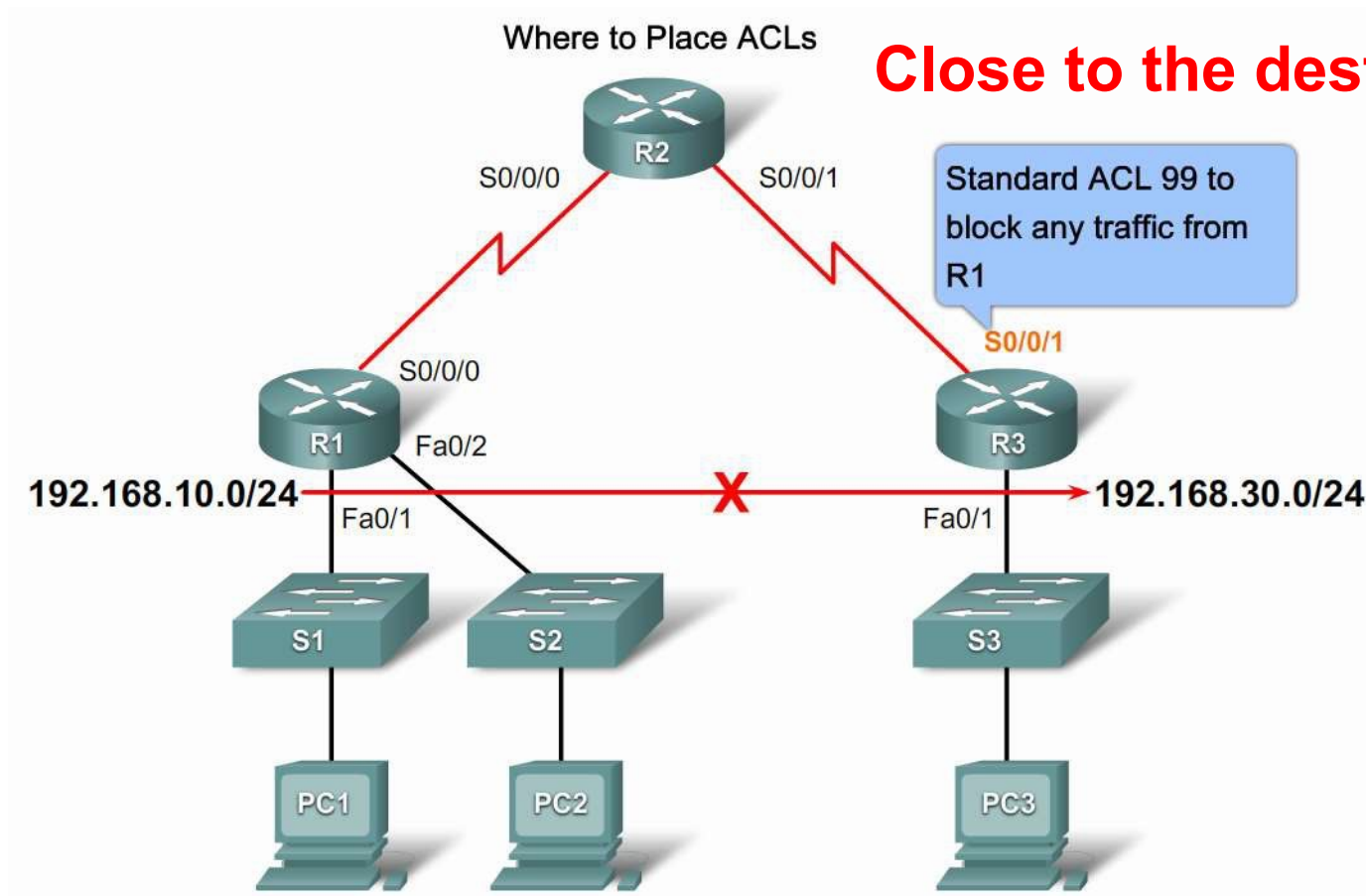
- (1 to 99) and (1300 to 1999): Standard IP ACL
- (100 to 199) and (2000 to 2699): Extended IP ACL

Named ACL:

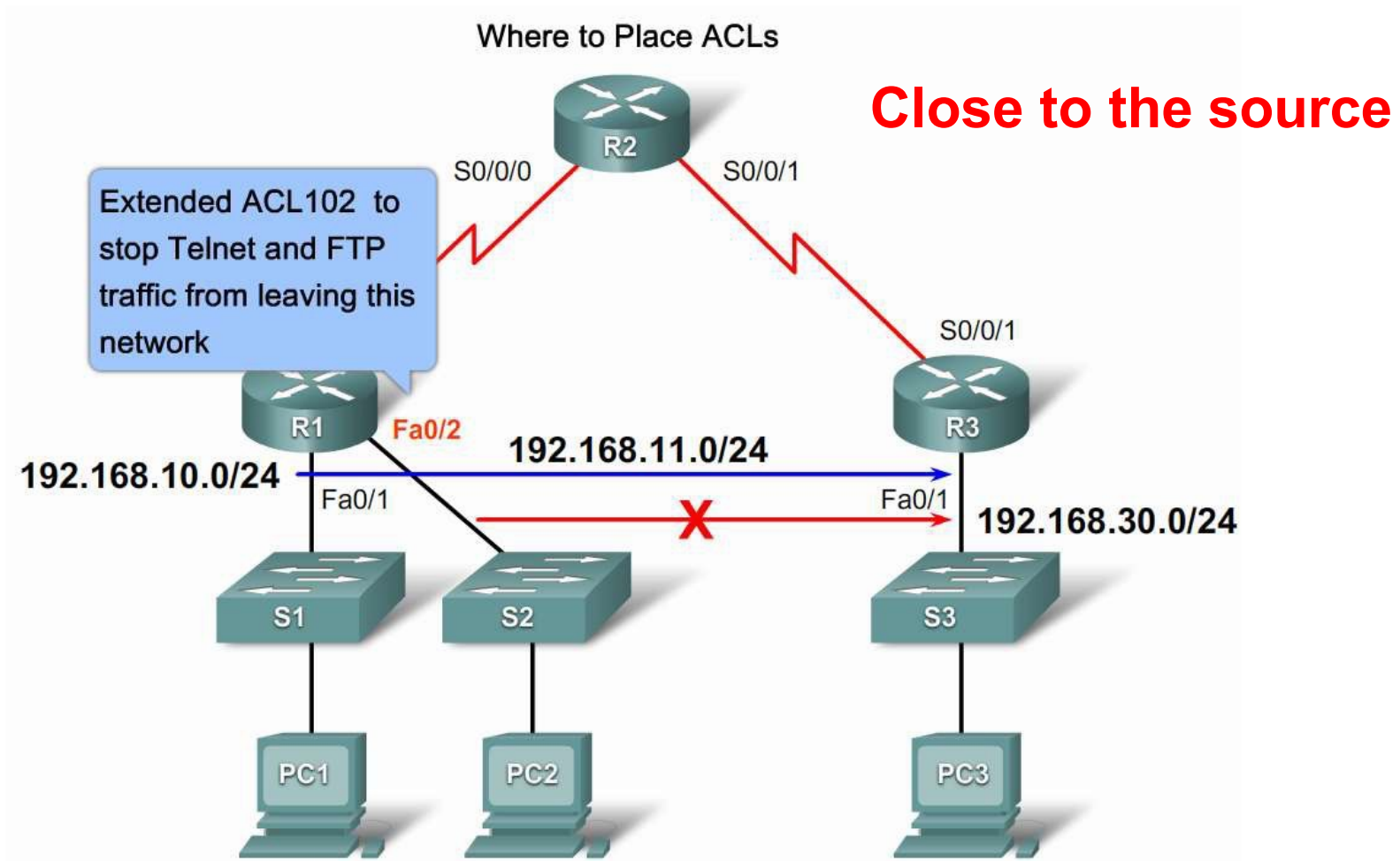
You assign a name by providing the name of the ACL:

- Names can contain alphanumeric characters.
- It is suggested that the name be written in CAPITAL LETTERS.
- Names cannot contain spaces or punctuation and must begin with a letter.
- You can add or delete entries within the ACL.

ACL Placement – Standard ACL



Extended ACL Placement – Extended ACL



ACL Best Practices

Guideline	Benefit
Base your ACLs on the security policy of the organization.	This will ensure you implement organizational security guidelines.
Prepare a description of what you want your ACLs to do.	This will help you avoid inadvertently creating potential access problems.
Use a text editor to create, edit and save ACLs.	This will help you create a library of reusable ACLs.
Test your ACLs on a development network before implementing them on a production network.	This will help you avoid costly errors.

Entering Criteria Statements

ACL 101

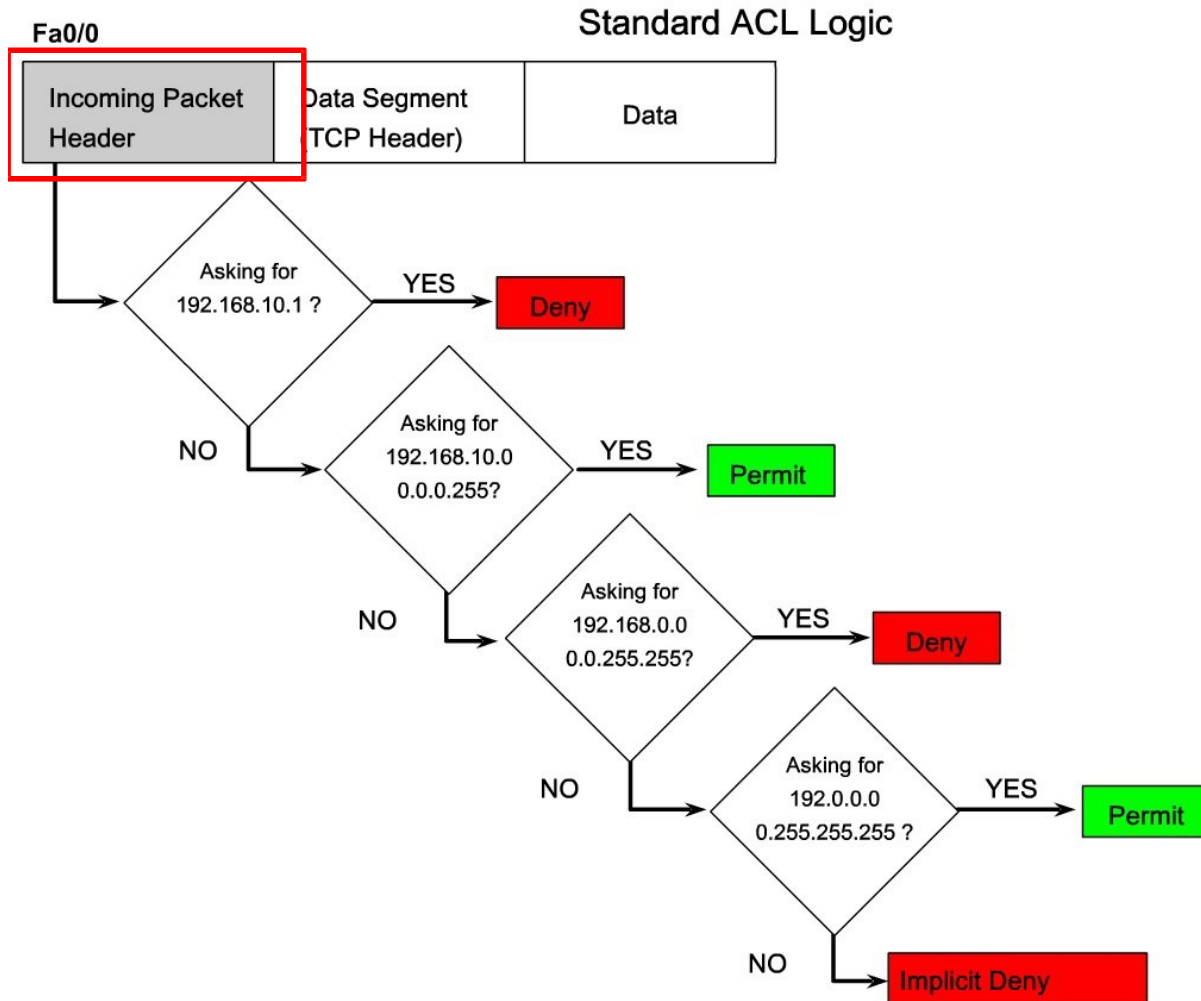
```
access-list 101 permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255
```

ACL 102

```
access-list 102 permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255  
access-list 102 deny ip any any
```

Use Notepad or another text editor to create and edit ACLs

Standard ACL Logic



```
access-list 2 deny  
192.168.10.1
```

```
access-list 2 permit  
192.168.10.0 0.0.0.255
```

```
access-list 2 deny  
192.168.0.0 0.0.255.255
```

```
access-list 2 permit  
192.0.0.0 0.255.255.255
```

If packets are permitted, they are routed through the router to an output interface. If packets are not permitted, they are dropped.

Command Syntax

Standard ACL `access-list` Command Syntax

Parameter	Description
<code>access-list-number</code>	Number of an ACL. This is a decimal number from 1 to 99, or 1300 to 1999 (for standard ACL).
<code>deny</code>	Denies access if the conditions are matched.
<code>permit</code>	Permits access if the conditions are matched.
<code>remark</code>	Add a remark about entries in an IP access list to make the list easier to understand and scan.
<code>source</code>	Number of the network or host from which the packet is being sent. There are two ways to specify the <i>source</i> : <ul style="list-style-type: none">• Use a 32-bit quantity in four-part, dotted- decimal format.• Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.55.
<code>source-wildcard</code>	(Optional) Wildcard bits to be applied to the source. There are two ways to specify the source-wildcard: <ul style="list-style-type: none">• Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.• Use the keyword any as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.55.
<code>log</code>	<p>(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the <code>logging console</code> command.)</p> <p>The message includes the ACL number, whether the packet was permitted or denied, the source address, and the number of packets. The message is generated for the first packet that matches, and then at five-minute intervals, including the number of packets permitted or denied in the prior five-minute interval.</p>

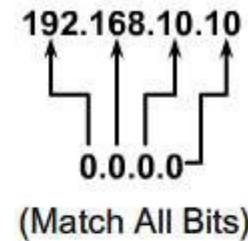
Wildcard Bit Mask Keywords

Example 1:

- 192.168.10.10 0.0.0.0 matches all of the address bits
- Abbreviate this wildcard mask using the IP address preceded by the keyword **host** (host 192.168.10.10)

host

Wildcard Mask:

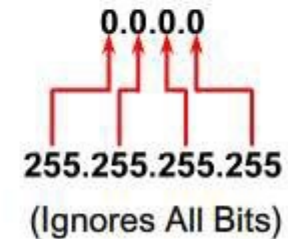


Example 2:

- 0.0.0.0 255.255.255.255 ignores all address bits
- Abbreviate expression with the keyword **any**

any

Wildcard Mask:



The Any and Host Keywords

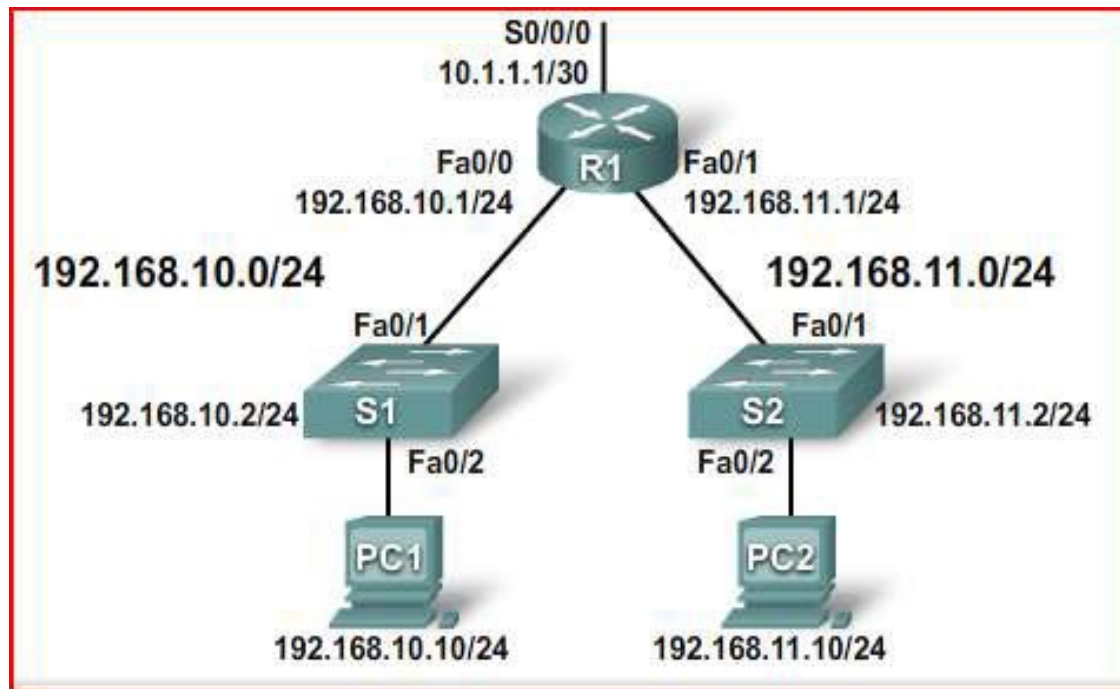
Example 1:

```
R1(config)#access-list 1 permit 0.0.0.0 255.255.255.255  
R1(config)#access-list 1 permit any
```

Example 2:

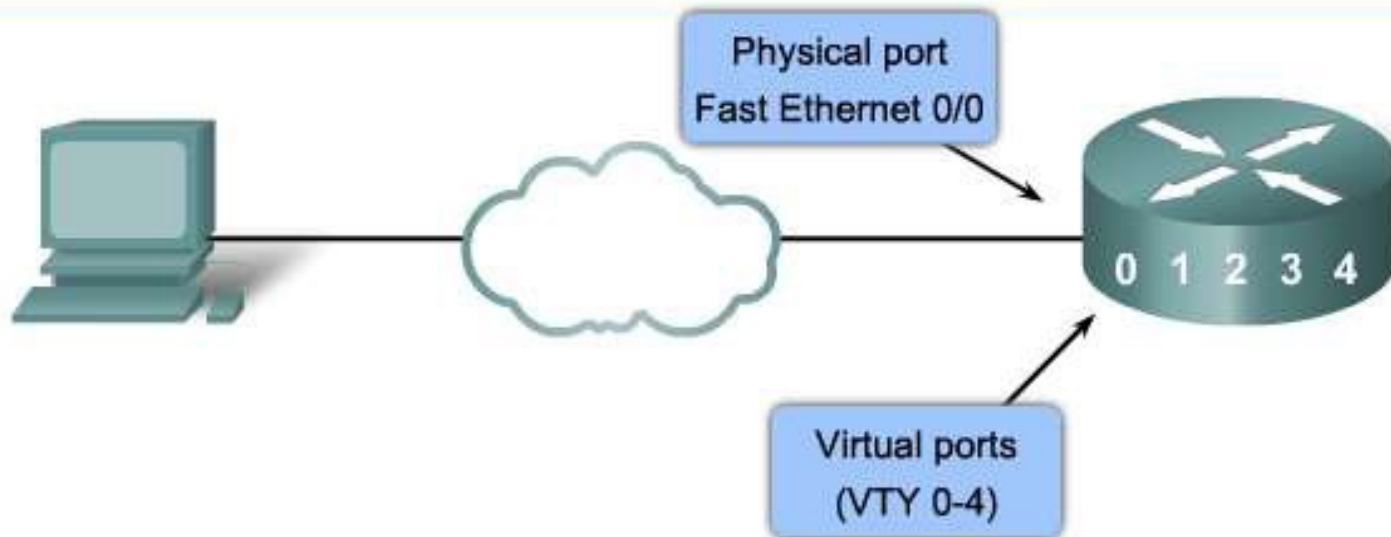
```
R1(config)#access-list 1 permit 192.168.10.10 0.0.0.0  
R1(config)#access-list 1 permit host 192.168.10.10
```

Deny a Specific Subnet



```
R1(config)# access-list 1 deny 192.168.10.0 0.0.0.255
R1(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)# interface S0/0/0
R1(config-if)# ip access-group 1 out
```


Control VTY Access

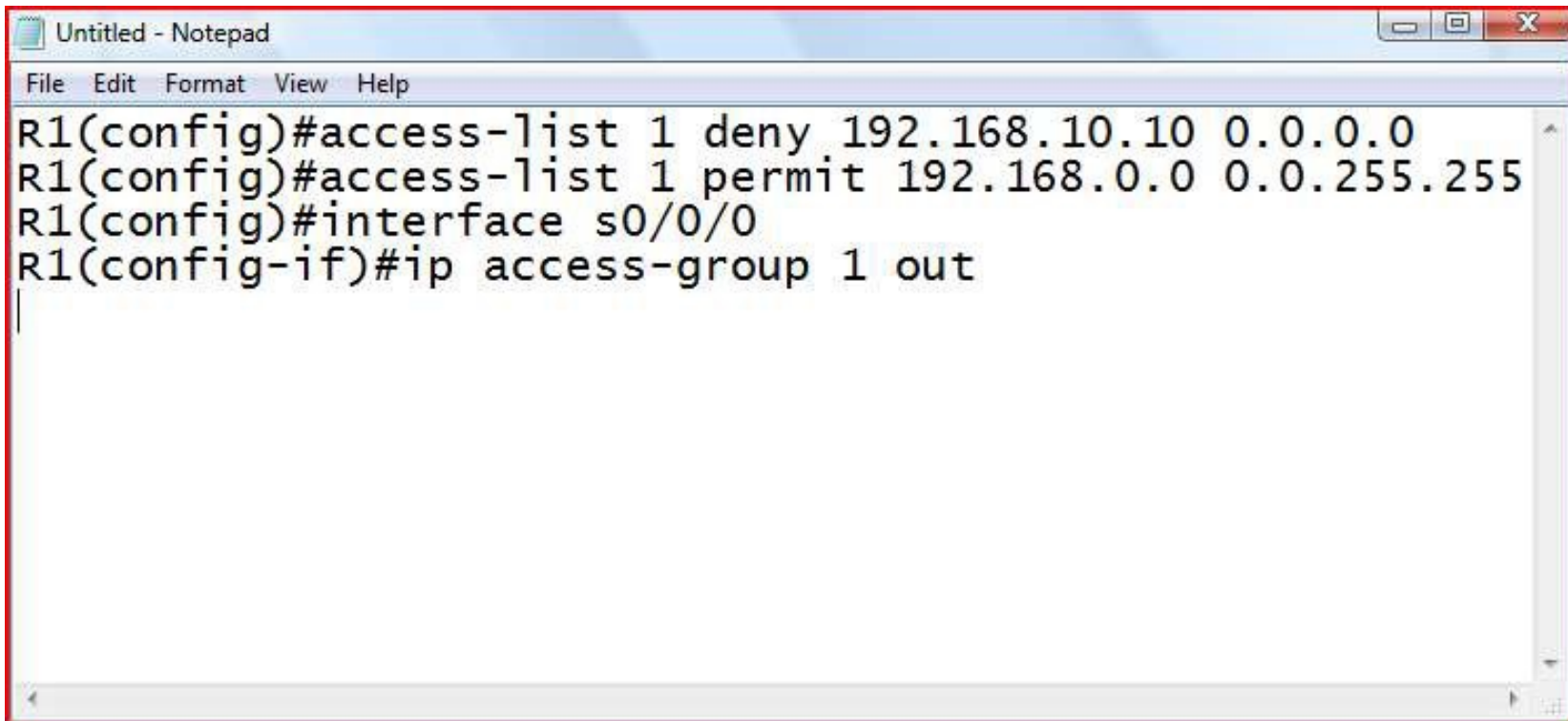


```
R1(config)#access-list 21 permit 192.168.10.0 0.0.0.255
R1(config)#access-list 21 permit 192.168.11.0 0.0.0.255
R1(config)#access-list 21 deny any

R1(config)#line vty 0 4
R1(config-line)#login
R1(config-line)#password secret
R1(config-line)#access-class 21 in
```

Access-class to VTY line

Creating ACLs in Notepad



```
Untitled - Notepad
File Edit Format View Help
R1(config)#access-list 1 deny 192.168.10.10 0.0.0.0
R1(config)#access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)#interface s0/0/0
R1(config-if)#ip access-group 1 out
```


Editing Numbered ACL

Step 1

```
R1#show running-config | include access-list  
access-list 20 permit 192.168.10.100  
access-list 20 deny 192.168.10.0 0.0.0.255
```

Step 2

```
access-list 20 permit 192.168.10.11  
access-list 20 deny 192.168.10.0 0.0.0.255
```

Step 3

```
R1#conf t  
Enter configuration commands, one per line. End with  
CTRL/Z.  
R1(config)#no access-list 20  
R1(config)#access-list 20 permit 192.168.10.100  
R1(config)#access-list 20 deny 192.168.10.0 0.0.0.255
```

Named ACL Examples

Example 1:

```
Router(config)# access-list 1 remark Permit only Jones workstation through  
Router(config)# access-list 1 permit 192.168.10.13  
Router(config)# access-list 1 remark Do not allow Smith through  
Router(config)# access-list 1 deny 1 192.168.10.14
```

Example 2:

```
Router(config)# ip access-list extended TELNETTING  
Router(config-ext-nacl)# remark Do not allow Jones workstation to Telnet  
Router(config-ext-nacl)# deny tcp host 192.168.10.13 any eq telnet
```

Named ACL Syntax

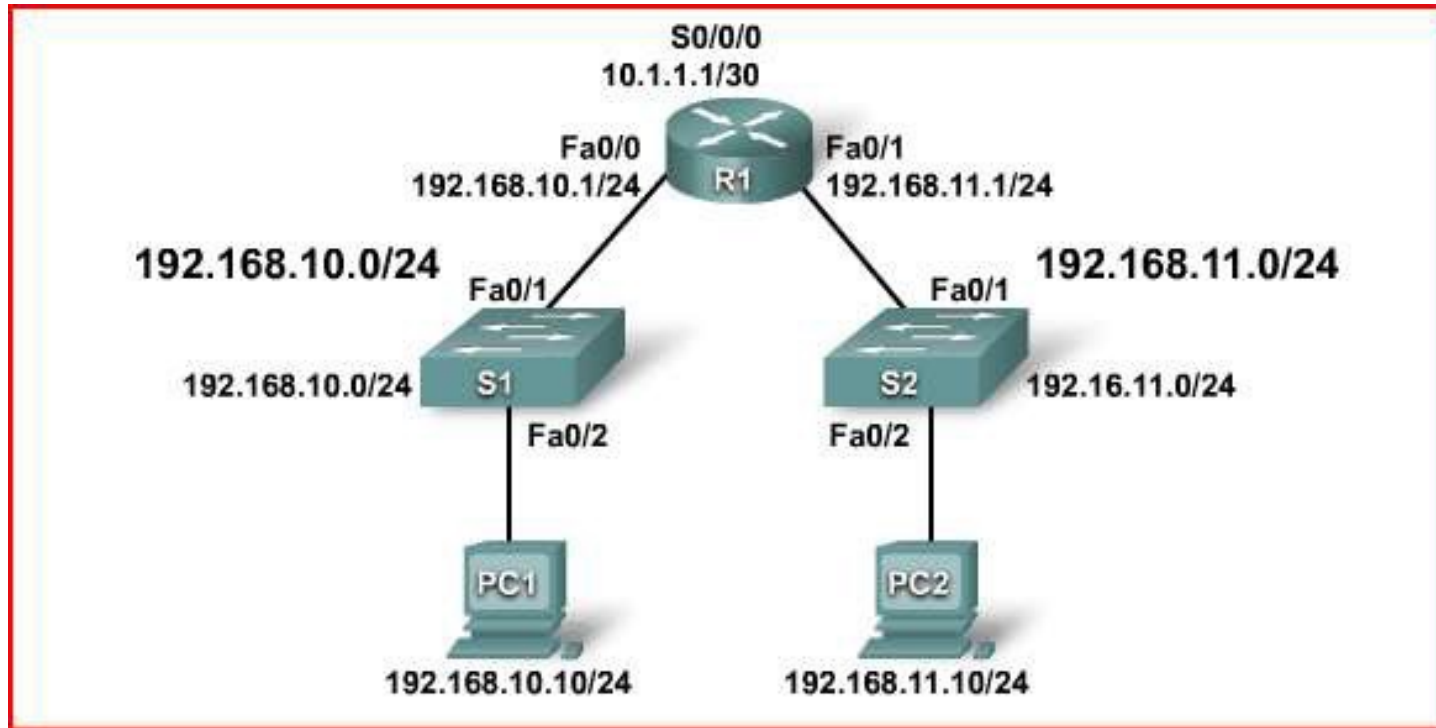
```
Router(config)# ip access-list [standard | extended] name
```

- Alphanumeric name string must be unique and cannot begin with a number

```
Router(config-std-nacl)# [permit | deny | remark] {source [source-wildcard]} [log]
```

```
Router(config-if)#ip access-group name [in | out]
```

Named ACL Example



```
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# deny host 192.168.11.10
R1(config-std-nacl)# permit any
R1(config-std-nacl)# interface Fa0/0
R1(config-if)# ip access-group NO_ACCESS out
```

Monitoring and Verifying

```
R1# show access-lists {access-list-number|name}
```

```
R1# show access-lists
```

```
Standard IP access list SALES
```

```
10 deny 10.1.1.0 0.0.0.255
```

```
20 permit 10.3.3.1
```

```
30 permit 10.4.4.1
```

```
40 permit 10.5.5.1
```

```
Extended IP access list ENG
```

```
10 permit tcp host 192.168.10.2 any eq telnet (25 matches)
```

```
20 permit tcp host 192.168.10.2 any eq ftp
```

```
30 permit tcp host 192.168.10.2 any eq ftp-data
```

Adding Lines to Named ACLs

```
R1# show access-lists
Standard IP access list WEBSERVER
 10 permit 192.168.10.11
 20 deny   192.168.10.0, wildcard bits 0.0.0.255
 30 deny   192.168.11.0, wildcard bits 0.0.0.255
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip access-list standard WEBSERVER
R1(config-std-nacl)# 15 permit host 192.168.11.10
R1(config-std-nacl)# end
R1#
*Nov  1 19:20:57.591: %SYS-5-CONFIG_I: Configured from console by console
R1# sho access-lists
Standard IP access list WEBSERVER
 10 permit 192.168.10.11
 15 permit 192.168.11.10
 20 deny   192.168.10.0, wildcard bits 0.0.0.255
 30 deny   192.168.11.0, wildcard bits 0.0.0.255
R1#
```

Extended ACLs

- Extended ACLs are used more often than standard ACLs because they provide a greater range of control.
- Like standard ACLs, extended ACLs check the source packet addresses, but they **also check the destination address, protocols, and port numbers (or services)**.
- This gives a greater range of criteria on which to base the ACL.
- For example, an extended ACL can simultaneously allow email traffic from a network to a specific destination while denying file transfers and web browsing.

Extended ACL Example

Using port numbers

```
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 23
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 21
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 22
```

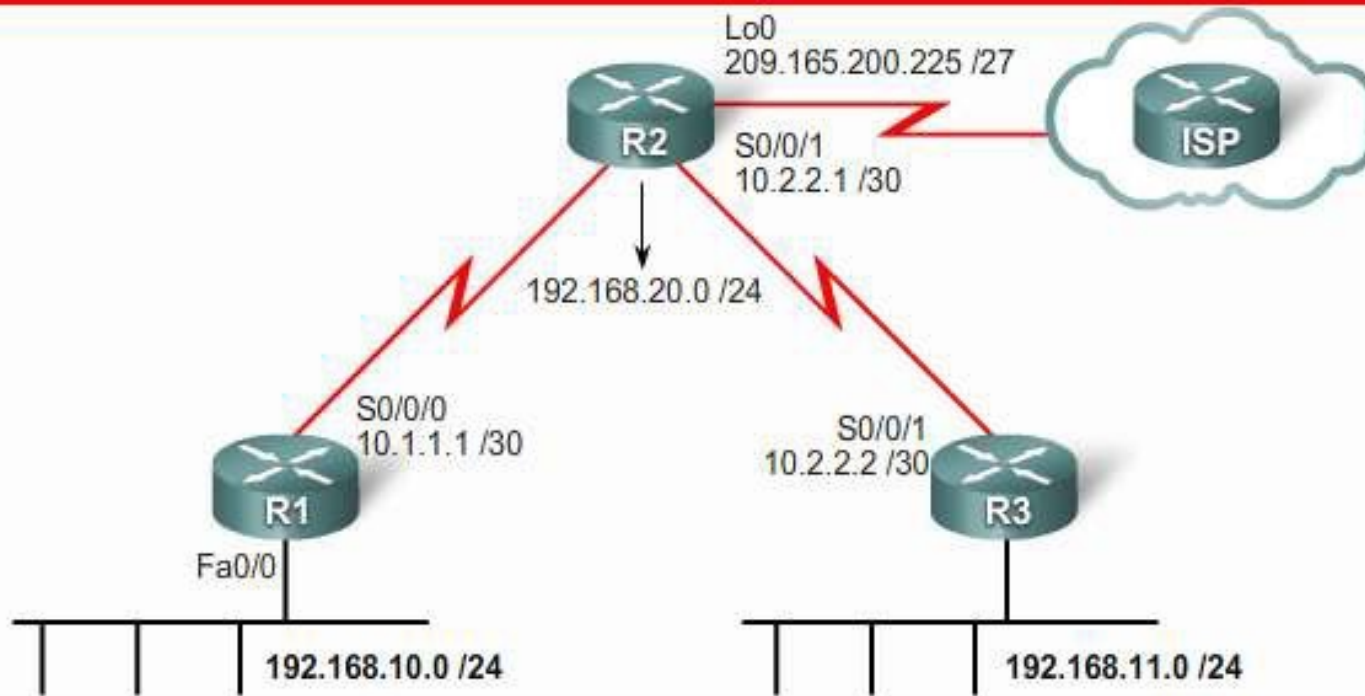
Using keywords

```
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq telnet
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp-data
```


Generating Port Numbers

```
R1(config)#access-list 101 permit tcp any eq ?  
  
<0-65535> Port number  
bgp Border Gateway Protocol (179)  
chargen Character generator (19)  
cmd Remote commands (rcmd, 514)  
daytime Daytime (13)  
discard Discard (9)  
domain Domain Name Service (53)  
drip Dynamic Routing Information Protocol (3949)  
echo Echo (7)  
exec Exec (rsh, 512)  
finger Finger (79)  
ftp File Transfer Protocol (21)  
ftp-data FTP data connections (20)  
gopher Gopher (70)  
hostname NIC hostname server (101)  
ident Ident Protocol (113)  
irc Internet Relay Chat (194)  
klogin Kerberos login (543)  
kshell Kerberos shell (544)  
login Login (rlogin, 513)  
lpd Printer service (515)  
nntp Network News  
Transport Protocol (119)  
pim-auto-rp PIM Auto-RP (496)  
pop2 Post Office Protocol v2 (109)  
pop3 Post Office Protocol v3 (110)
```

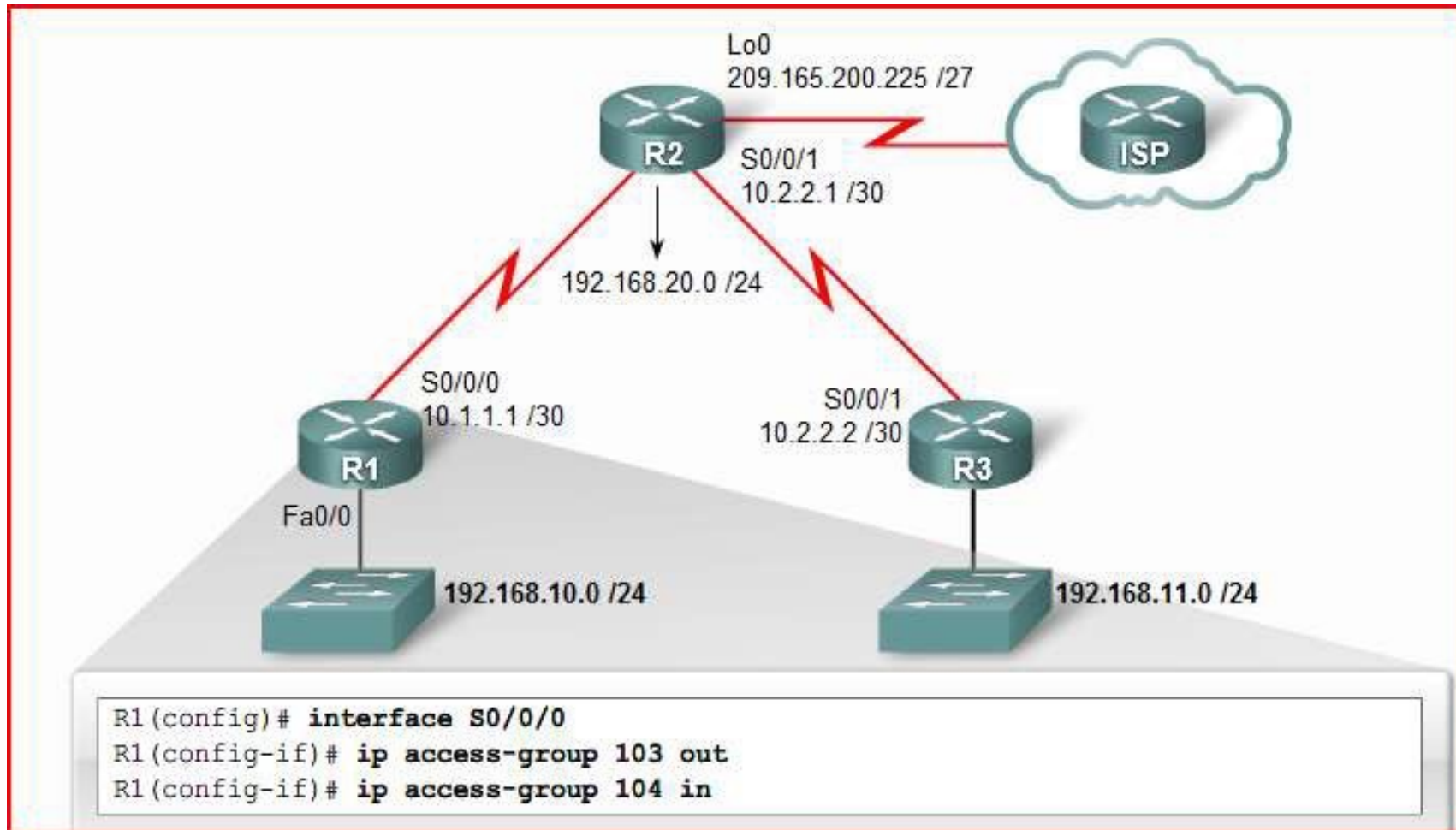
Configuring Extended ACLs



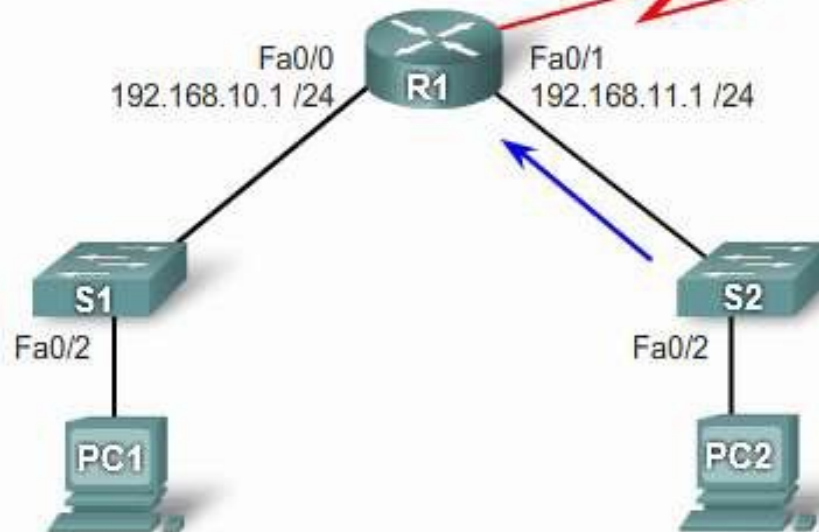
```
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)#access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
```

ACL103 allows requests to ports 80 and 443
ACL104 allows established HTTP and SHTTP replies

Applying ACL



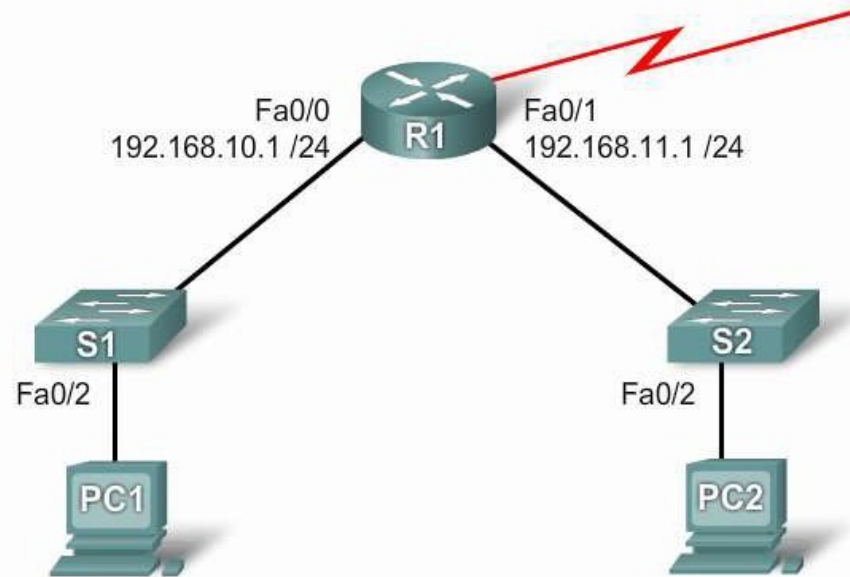
FTP Deny



```
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0  
0.0.0.255 eq 21  
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0  
0.0.0.255 eq 20  
R1(config)# access-list 101 permit ip any any  
R1(config)# interface Fa0/1  
R1(config-if)# ip access-group 101 in
```

Deny Telnet

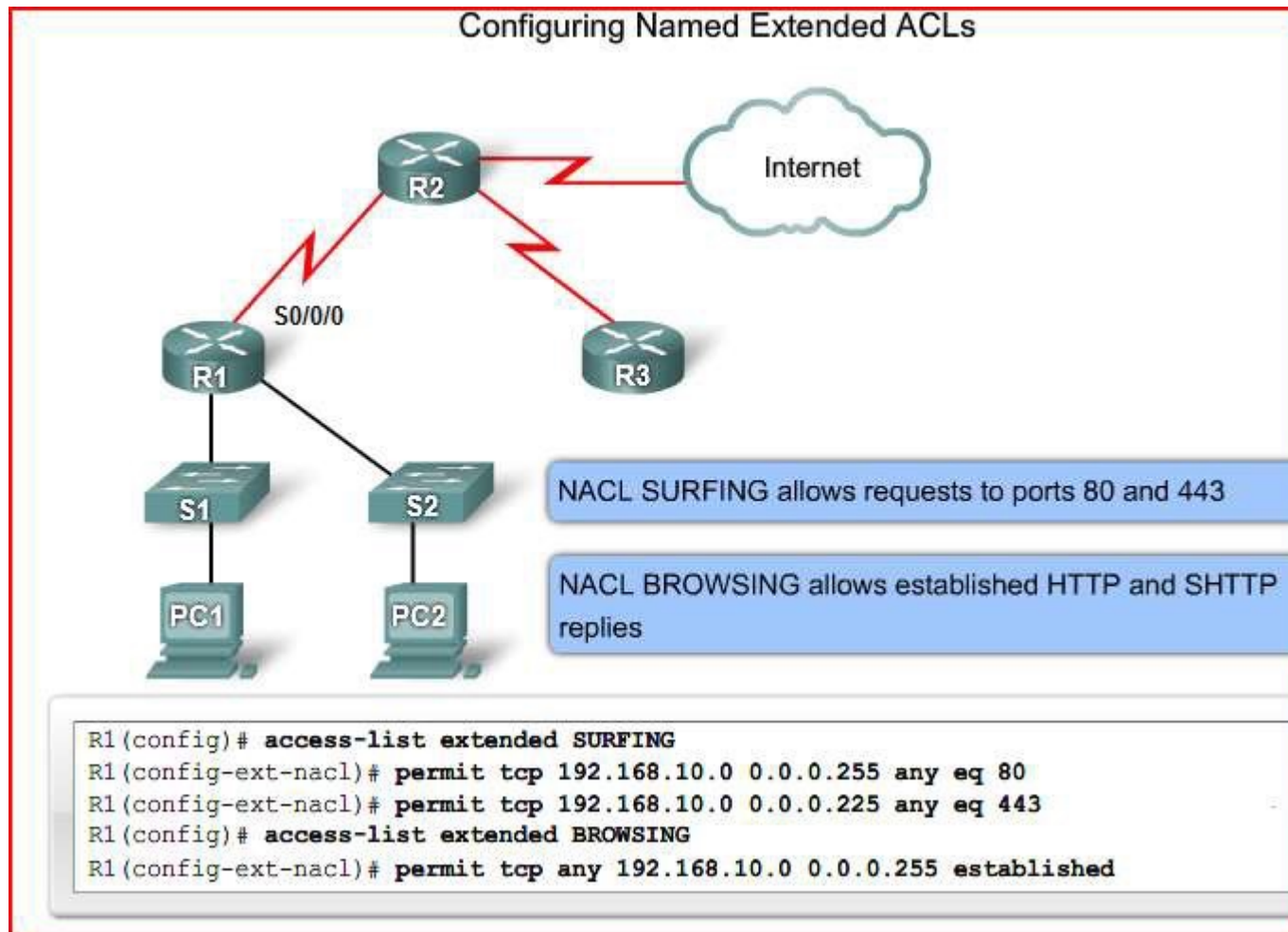
Extended ACL to Deny Only Telnet from Subnet



```
R1(config)#access-list 101 deny tcp 192.168.11.0 0.0.0.255 any eq 23
R1(config)#access-list 101 permit ip any any

R1(config)#interface Fa0/1
R1(config-if)#ip access-group 101 in
```

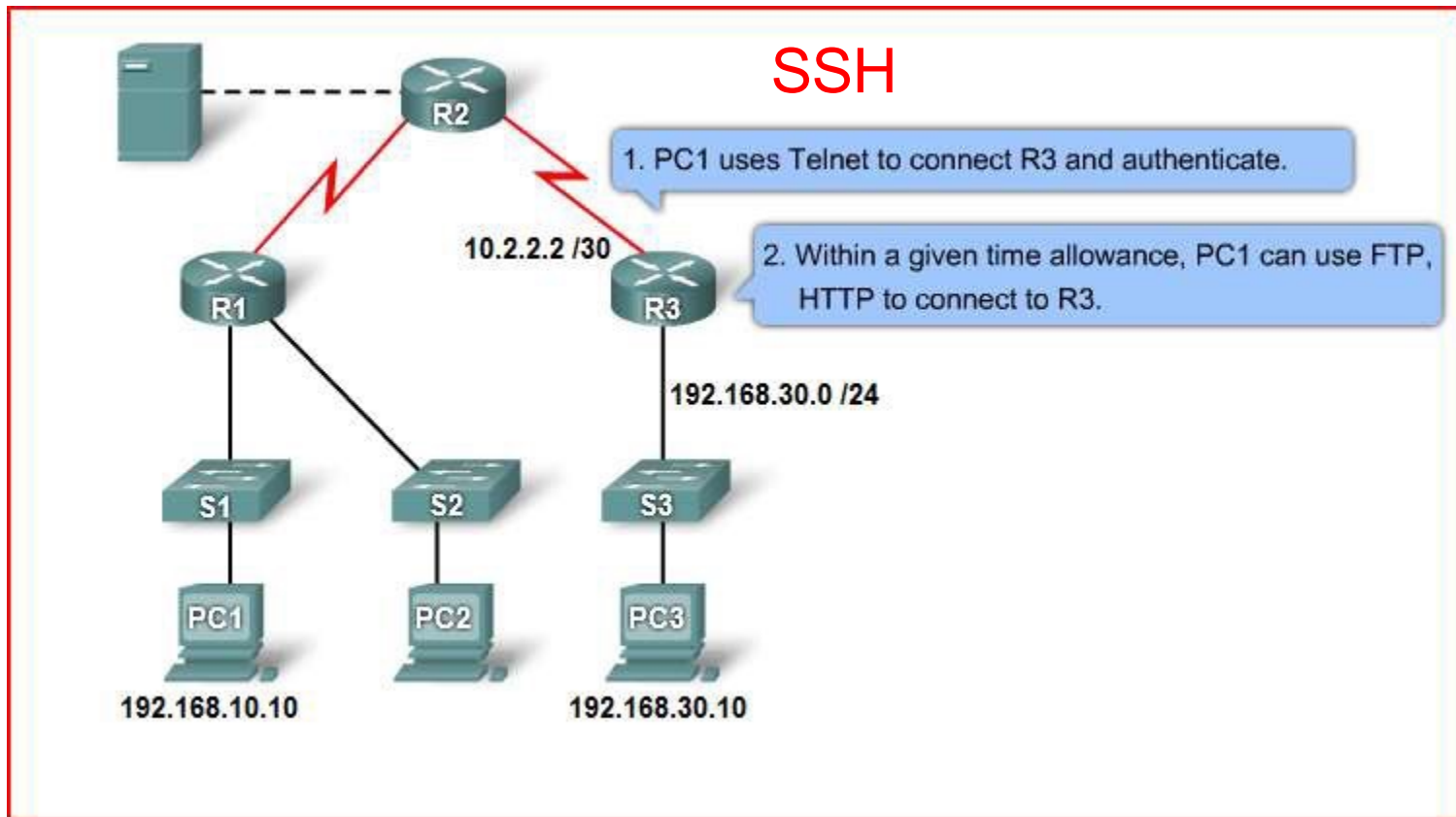
Named Extended ACLs



Complex ACLs

Complex ACL	Description
Dynamic ACLs (lock-and-key)	Users that want to traverse the router are blocked until they use Telnet to connect to the router and are authenticated
Reflexive ACLs	Allows outbound traffic and limits inbound traffic in response to sessions that originate inside the router
Time-based ACLs	Allows for access control based on the time of day and week

Dynamic ACL



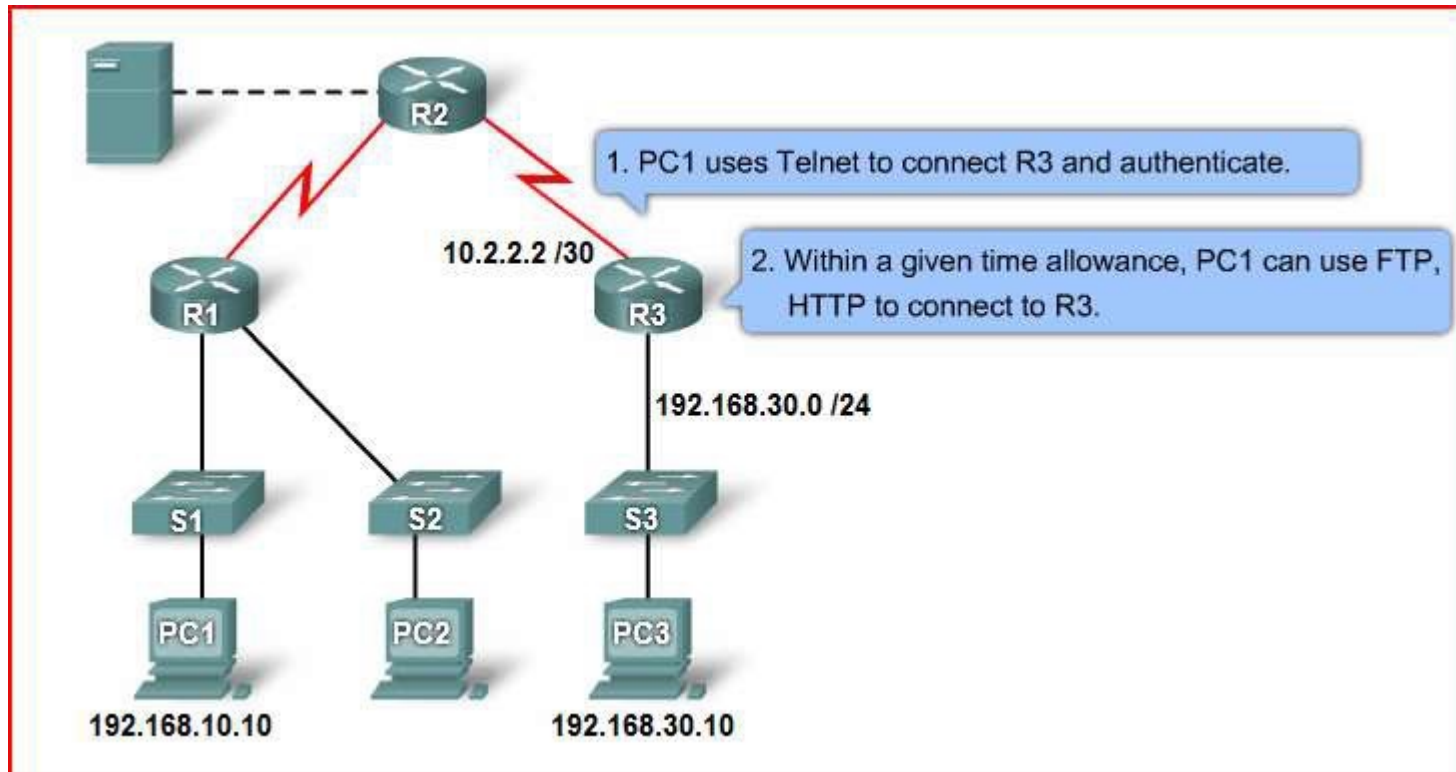
What are Dynamic ACLs?

- **Lock-and-key** is a traffic filtering security feature that uses dynamic ACLs, sometimes called lock-and-key ACLs.
- Available for IP traffic only. Dependent on Telnet or SSH connectivity, authentication (local or remote), and extended ACLs.
- Dynamic ACL configuration **starts with applying an extended ACL to block traffic through the router.**
- The extended ACL blocks users who want to traverse the router until they use Telnet to connect to the router and are authenticated.
- The Telnet connection is then dropped, and **a single-entry dynamic ACL** is added to the existing extended ACL. This permits traffic for a particular period; idle and absolute timeouts are possible.

When to use Dynamic ACLs?

- When **you want a specific remote user or group of remote users to access a host within your network**, connecting from their remote hosts via the Internet.
- Lock-and-key **authenticates** the user and then permits limited access through your firewall router for a host or subnet for a finite period.
- When you want a subset of hosts on a local network to access a host on a remote network that is protected by a firewall. With lock-and-key, you can enable access to the remote host only for the desired set of local hosts. **Lock-and-key requires the users to authenticate through a AAA, TACACS+ server, or another security server** before it allows their hosts to access the remote hosts.

Dynamic ACL

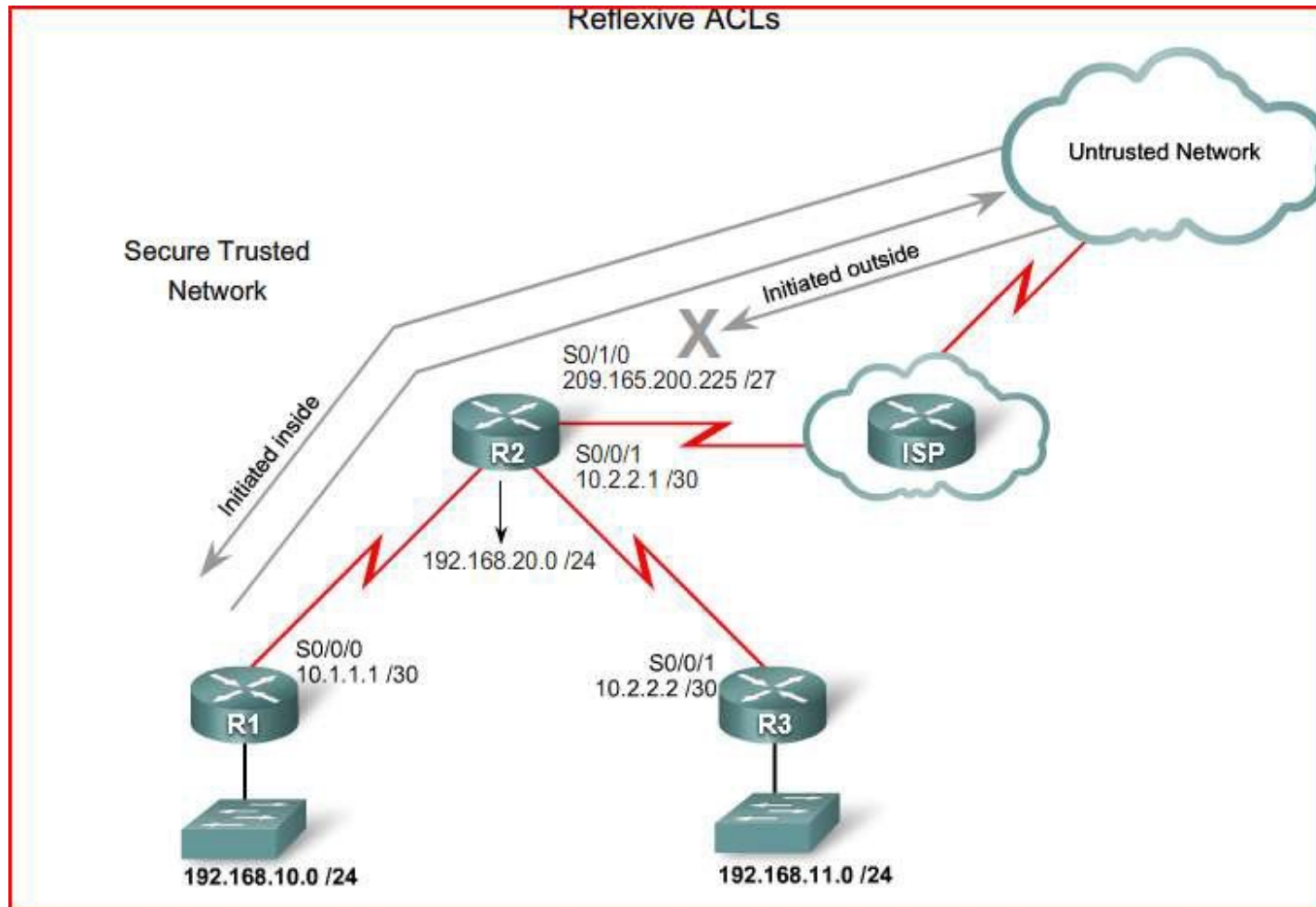


Dynamic ACL Example

Step 1	<pre>R3(config)#username Student password 0 cisco</pre>
Step 2	<pre>R3(config)# access-list 101 permit any host 10.2.2.2 eq telnet R3(config)#access-list 101 dynamic testlist timeout 15 permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255</pre>
Step 3	<pre>R3(config)#interface serial 0/0/1 R3(config-if)#ip access-group 101 in</pre>
Step 4	<pre>R3(config)#line vty 0 4 R3(config-line)#login local R3(config-line)# autocommand access-enable host timeout 5</pre>

Once the user is authenticated using Telnet, the autocommand command executes and the Telnet session terminates. The user can now access network 192.168.30.0. If there is up to 5 minutes of inactivity, the window will close.

Reflexive ACLs



What are Reflexive ACLs?

- Reflexive ACLs force the **reply traffic** from the destination of a known recent outbound packet **to go to the source of that outbound packet**.
- Network administrators use reflexive ACLs to allow IP traffic for **sessions originating from within their network** while denying IP traffic for sessions originating outside the network.
- These ACLs allow the router to manage session traffic dynamically. The router examines the outbound traffic and when it sees a new connection, it adds an entry to a **temporary ACL** to allow replies back in.
- Reflexive ACLs contain only **temporary entries**. These entries are automatically created when a new IP session begins, for example, with an outbound packet, and the entries are automatically removed when the session ends.

Reflexive ACL Step 1

Step 1

```
R2(config)#ip access-list extended OUTBOUNDFILTERS
R2(config-ext-nacl)# permit tcp 192.168.0.0 0.0.255.255 any
reflect TCPTRAFFIC
R2(config-ext-nacl)# permit icmp 192.168.0.0 0.0.255.255 any
reflect ICMPTRAFFIC
```

- Causes the router to keep track of traffic that was initiated on the inside.
- Dynamically creates two new reflexive lists for TCP and ICMP traffic

Reflexive ACL Step 2

Step 2

```
R2(config)#ip access-list extended INBOUNDFILTERS  
R2(config-ext-nacl)# evaluate TCPTRAFFIC  
R2(config-ext-nacl)# evaluate ICMPTRAFFIC
```

Step 3

```
R2(config)#interface S0/1/0  
R2(config-if)#ip access-group INBOUNDFILTERS in  
R2(config-if)#ip access-group OUTBOUNDFILTERS out
```

Creates an inbound policy that requires the router to check incoming traffic to see if it was initiated from inside and ties the reflexive ACL part of the OUTBOUNDFILTERS ACL, called TCPTRAFFIC, to the INBOUNDFILTERS ACL.

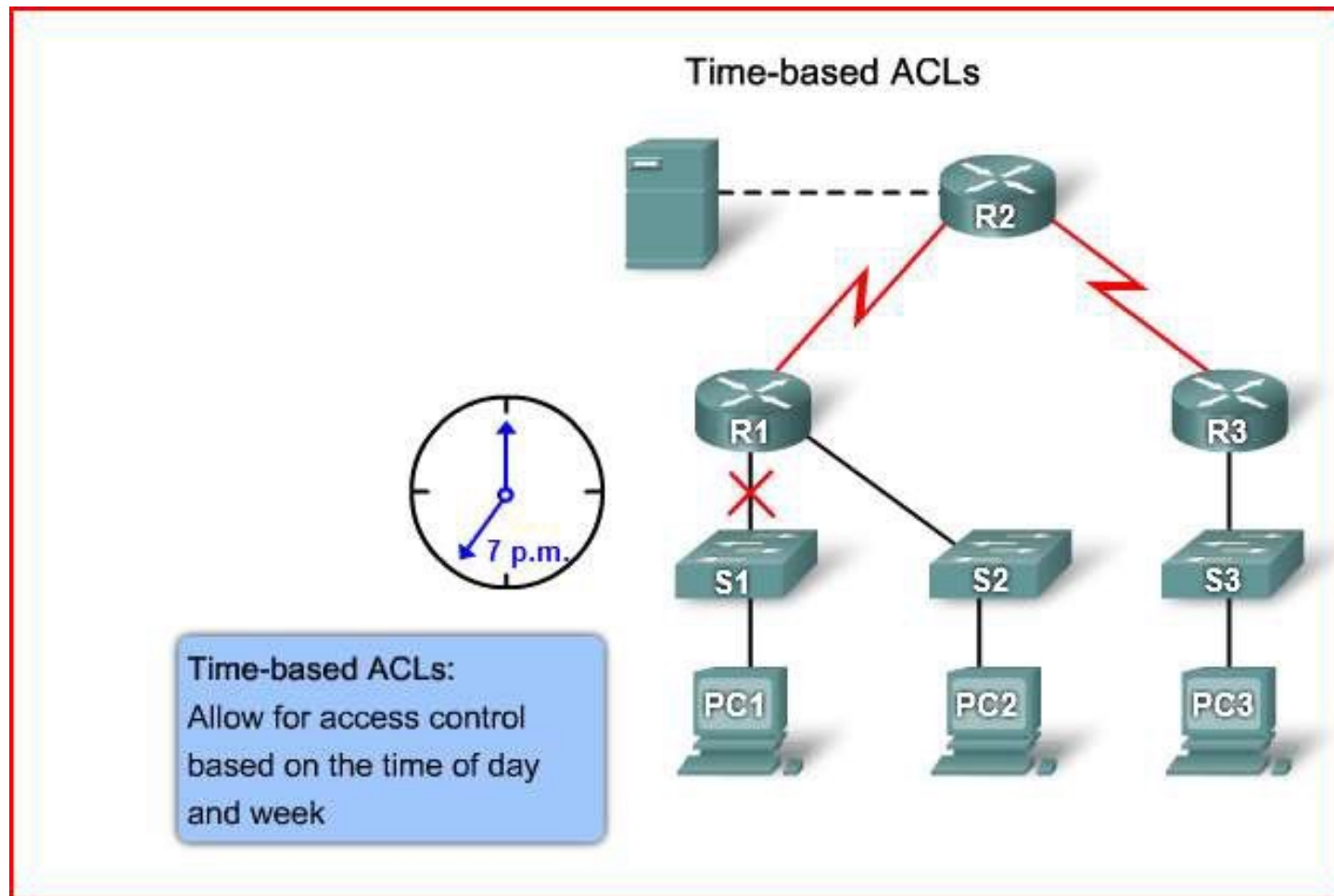
Reflexive ACL Step 3

Step 3

```
R2(config)#interface s0/1/0  
R2(config-if)#ip access-group INBOUNDFILTERS in  
R2(config-if)#ip access-group OUTBOUNDFILTERS out
```

Applies both an inbound and an outbound ACL to the interface.

Time-Based ACL



What are Time-Based ACL

- Time-based ACLs are similar to extended ACLs in function but allow **access control based on time**.
- To implement time-based ACLs, you create **a time range** that defines specific times of the day and week. You identify the time range with a name and then refer to it by a function. The time restrictions are imposed on the function itself.

Time-Based ACL Benefit

- Offers the network administrator more control over permitting or denying access to resources.
- Allows network administrators to control logging messages.
- ACL entries can log traffic at certain times of the day, but not constantly.
- Therefore, administrators can simply deny access without analyzing the many logs that are generated during peak hours.

Time-Based ACL Example

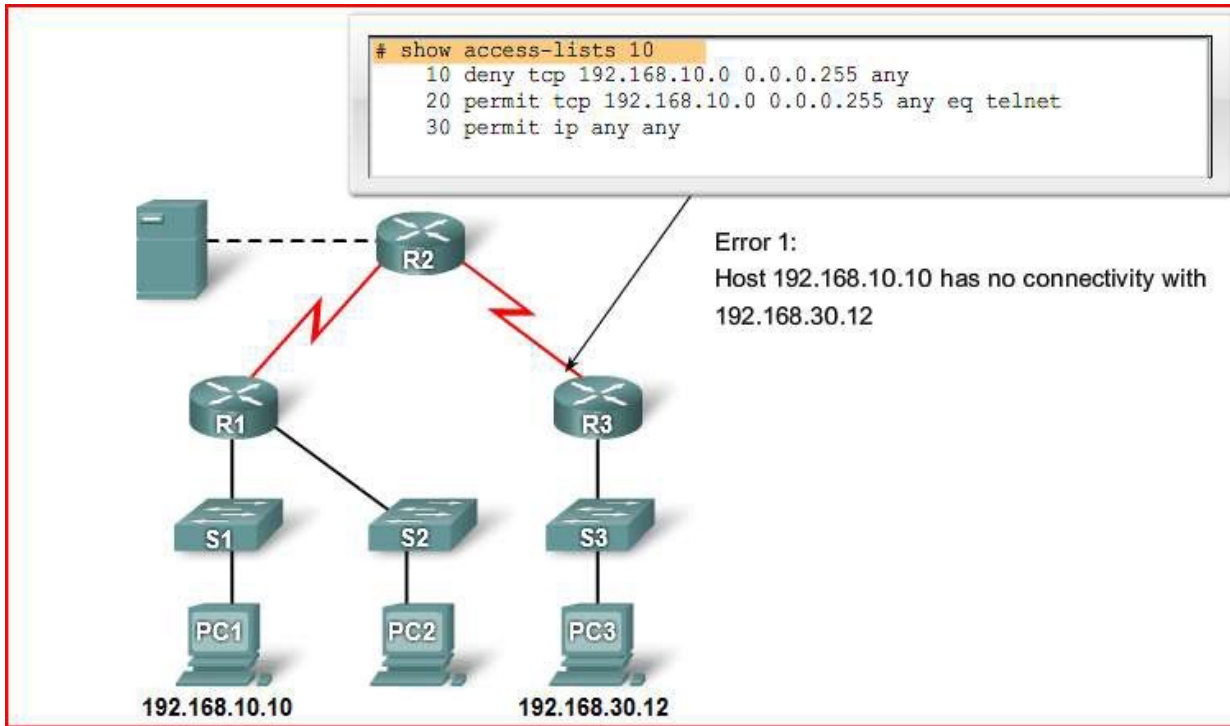
Step 1	<pre>R1(config)#time-range EVERYOTHERDAY R1(config-time-range)#periodic Monday Wednesday Friday 8:00 to 17:00</pre>
Step 2	<pre>R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq telnet time-range EVERYOTHERDAY</pre>
Step 3	<pre>R1(config)#interface s0/0/0 R1(config-if)#ip access-group 101 out</pre>

Step 1: Defines the time frame and names the ACL

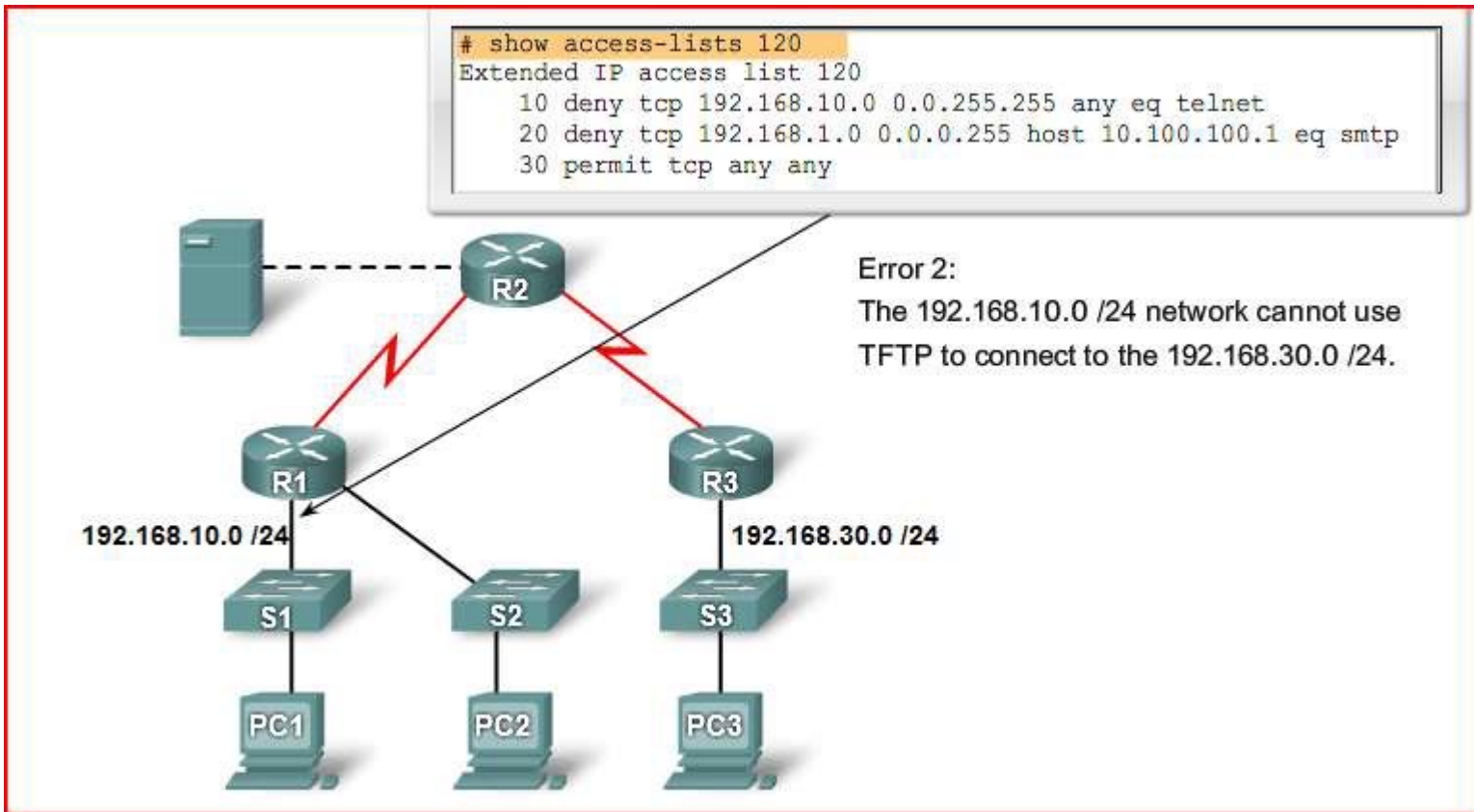
Step 2: Applies the time range to the ACL

Step 3: Applies the ACL to the interface

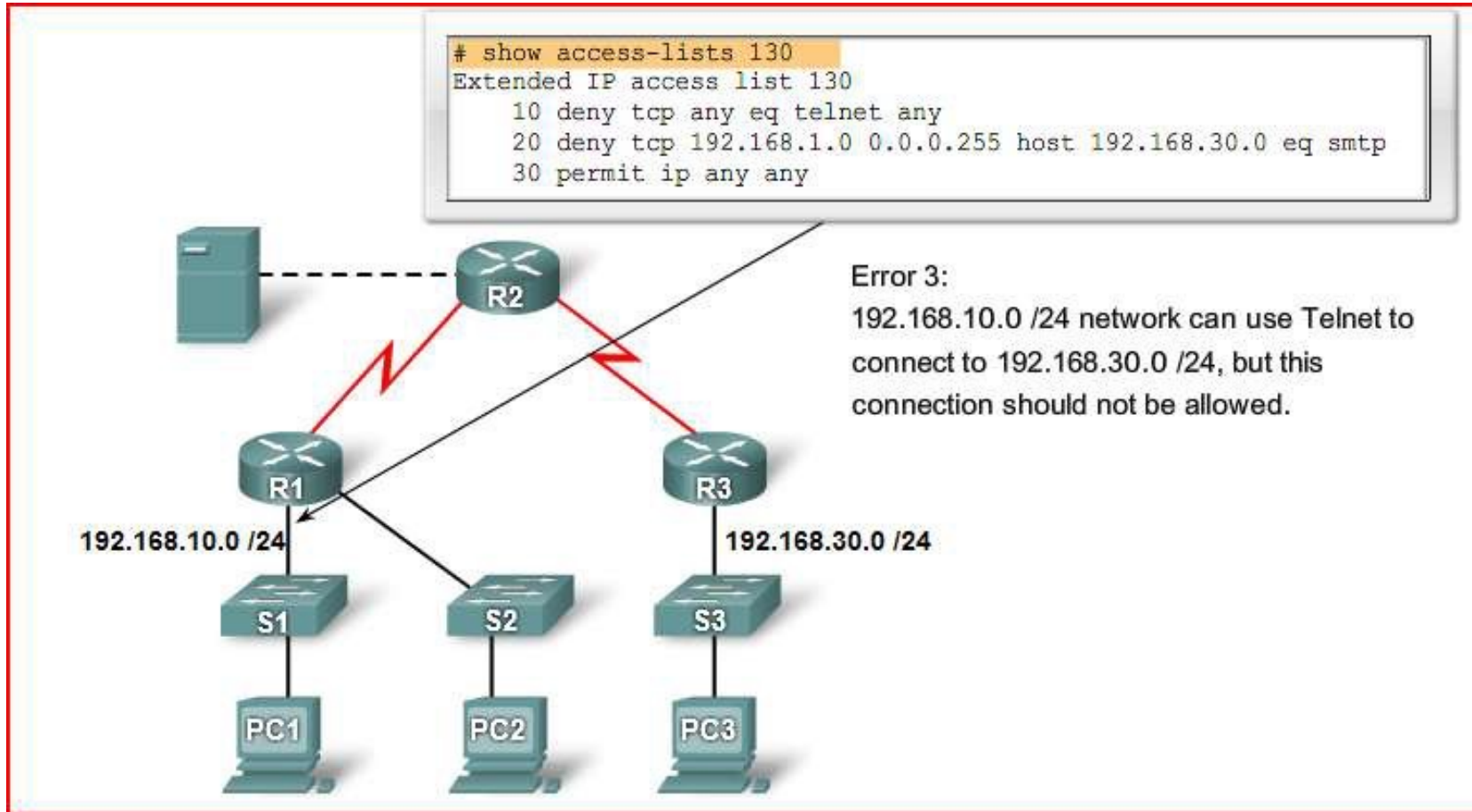
Troubleshooting



ACL Error Example 2

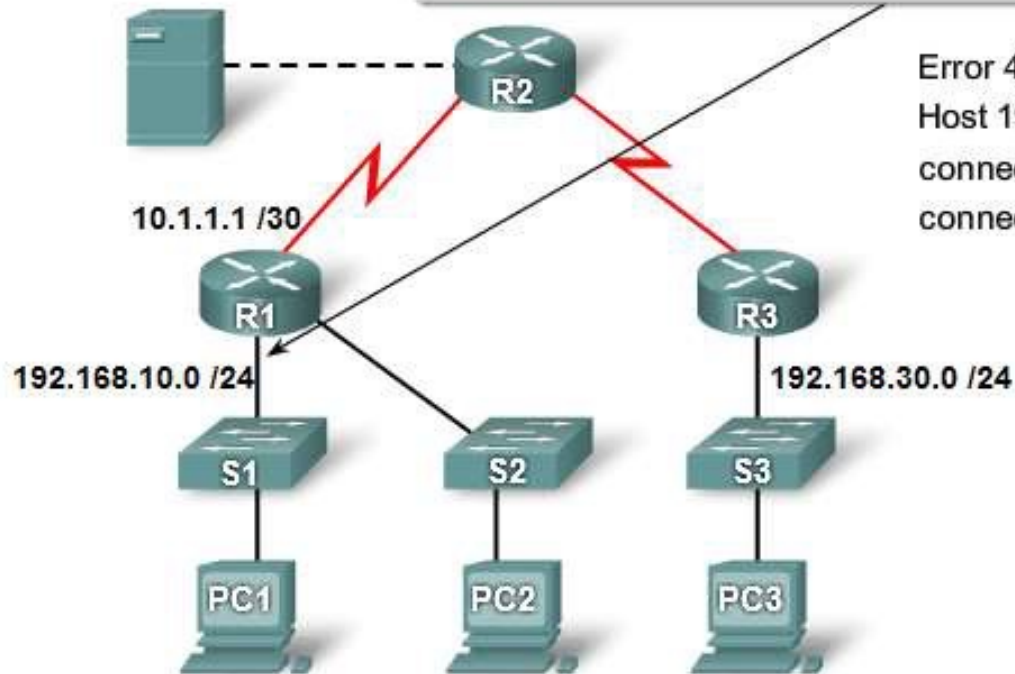


ACL Error Example 3



ACL Error Example 4

```
# show access-lists 140
Extended IP access list 140
10 deny tcp host 192.168.10.1 0.0.0.255 any eq telnet
20 deny tcp 192.168.1.0 0.0.0.255 host 10.100.100.1 eq smtp
30 permit ip any any
```



Error 4:
Host 192.168.10.10 can use Telnet to connect to 192.168.30.12, but this connection should not be allowed.

Summary

- An Access List (ACL) is:
 - A series of permit and deny statements that are used to filter traffic
- Standard ACL
 - Identified by numbers 1 - 99 and 1300 - 1999
 - Filter traffic based on source IP address
- Extended ACL
 - Identified by number 100 -199 & 2000 - 2699
 - Filter traffic based on
 - Source IP address
 - Destination IP address
 - Protocol
 - Port number

Summary

- Named ACL
 - Used with IOS 11.2 and above
 - Can be used for either standard or extended ACL
- ACL's use Wildcard Masks (WCM)
 - Described as the inverse of a subnet mask
 - Reason
 - 0 → check the bit
 - 1 → ignore the bit

Summary

- Implementing ACLs
 - 1st create the ACL
 - 2nd place the ACL on an interface
 - Standard ACL are placed nearest the destination
 - Extended ACL are placed nearest the source
- Use the following commands for verifying & troubleshooting an ACL
 - Show access-list
 - Show interfaces
 - Show run

Summary

- Complex ACL
 - Dynamic ACL
 - Reflexive ACL
 - Time based ACL