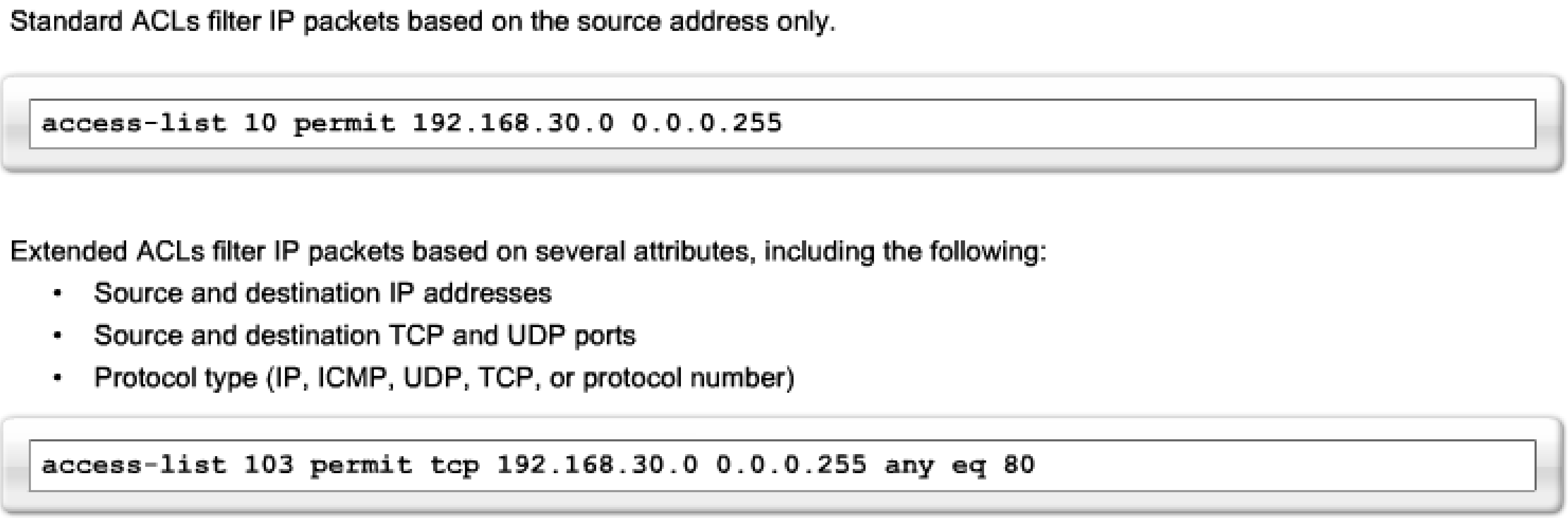
Types of ACLs



Standard ACLs filter IP packets based on the source address only.

```
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
```

The two main tasks involved in using ACLs are as follows:

Step 1. Create an access list by specifying an access list number or name and access conditions.

Step 2. Apply the ACL to interfaces or terminal lines.

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:

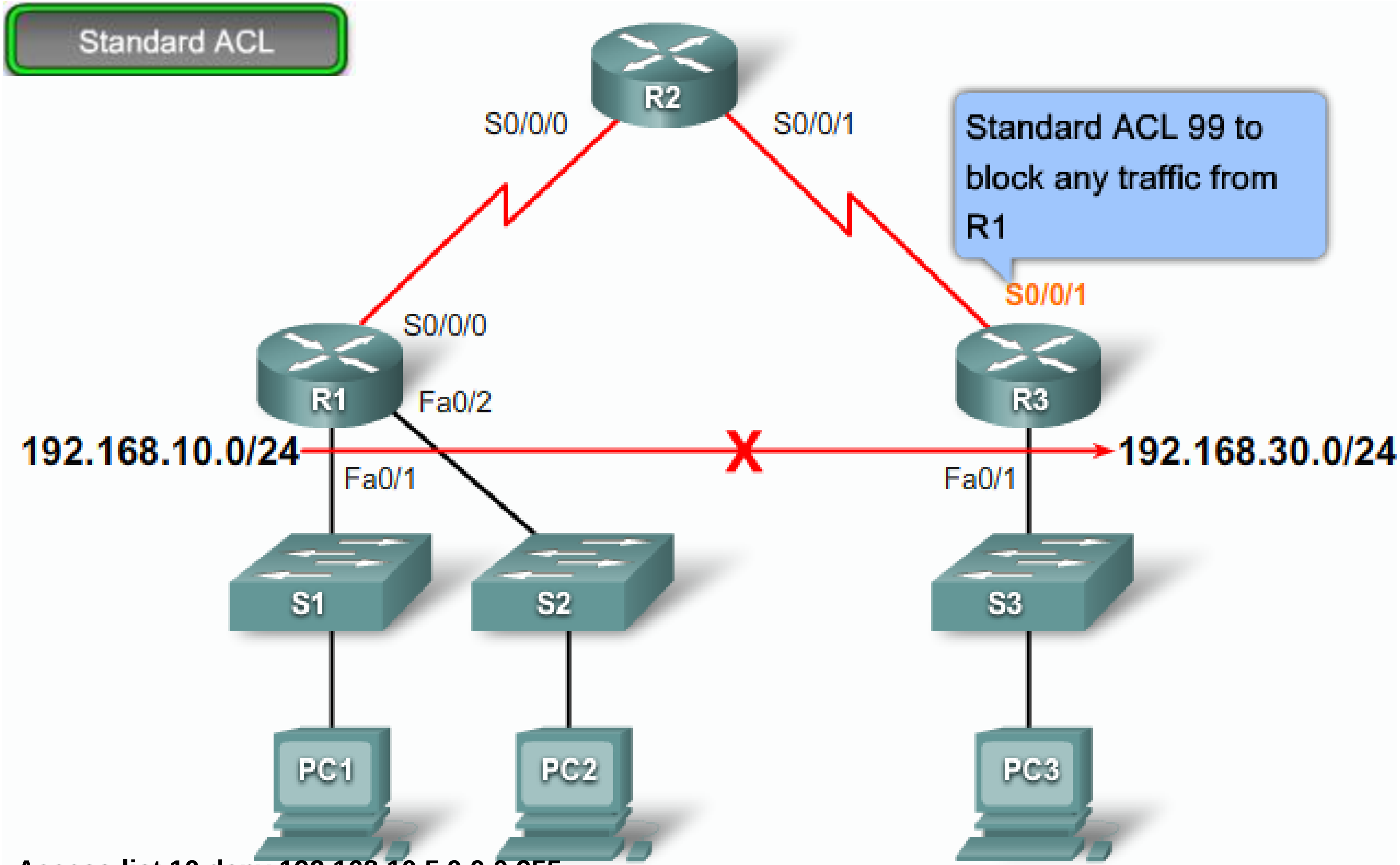
You assign a name by providing the name of the ACL:

- 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.

Where to place ACLs?

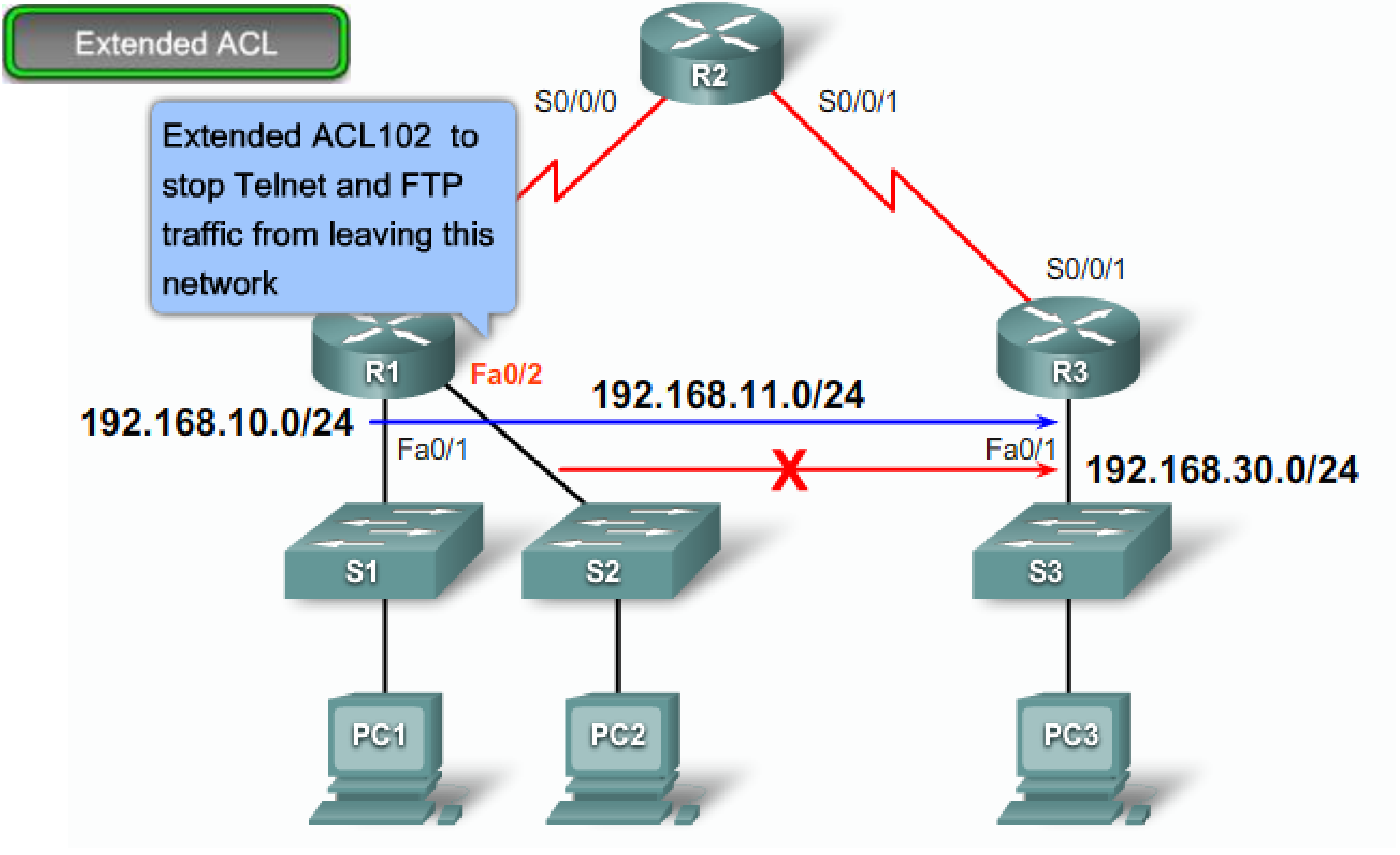
- Locate extended ACLs as close as possible to the source of the traffic denied. This way, undesirable traffic is filtered without crossing the network infrastructure.
- Because standard ACLs do not specify destination addresses, place them as close to the destination as possible.

prevent traffic originating in the 192.168.10.0/24 network from getting to the 192.168.30.0/24 network.



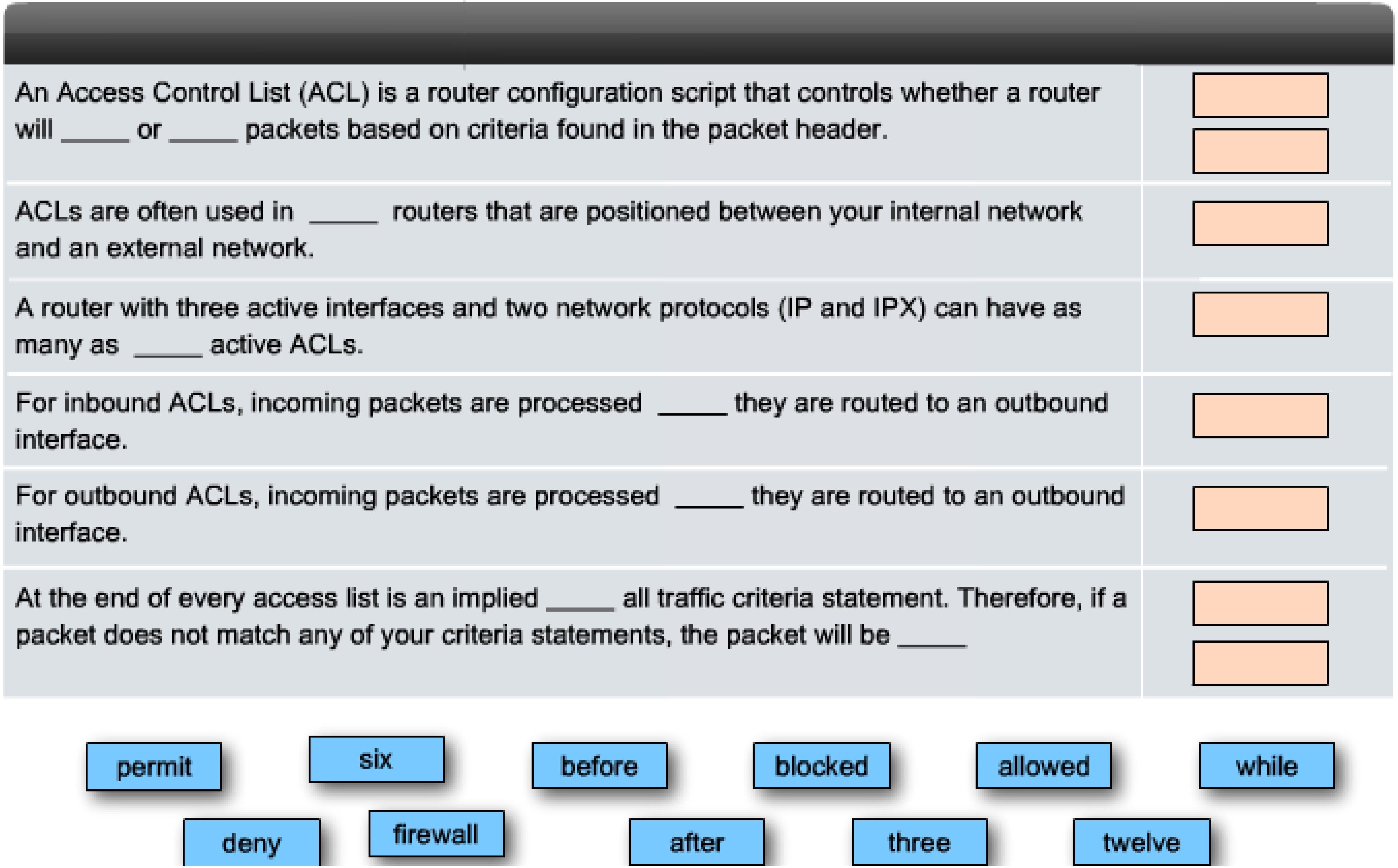
Access-list 10 deny 192.168.10.5 0.0.0.255
Access-list 10 permit any

the administrator of the 192.168.10.0/24 and 192.168.11.0/24 networks (referred to as Ten and Eleven, respectively, in this example) wants to deny Telnet and FTP traffic from Eleven to the 192.168.30.0/24 network (Thirty, in this example). At the same time, other traffic must be permitted to leave Ten.



Extended ACL

Extended ACL102 to stop Telnet and FTP traffic from leaving this network



An Access Control List (ACL) is a router configuration script that controls whether a router will _____ or _____ packets based on criteria found in the packet header.

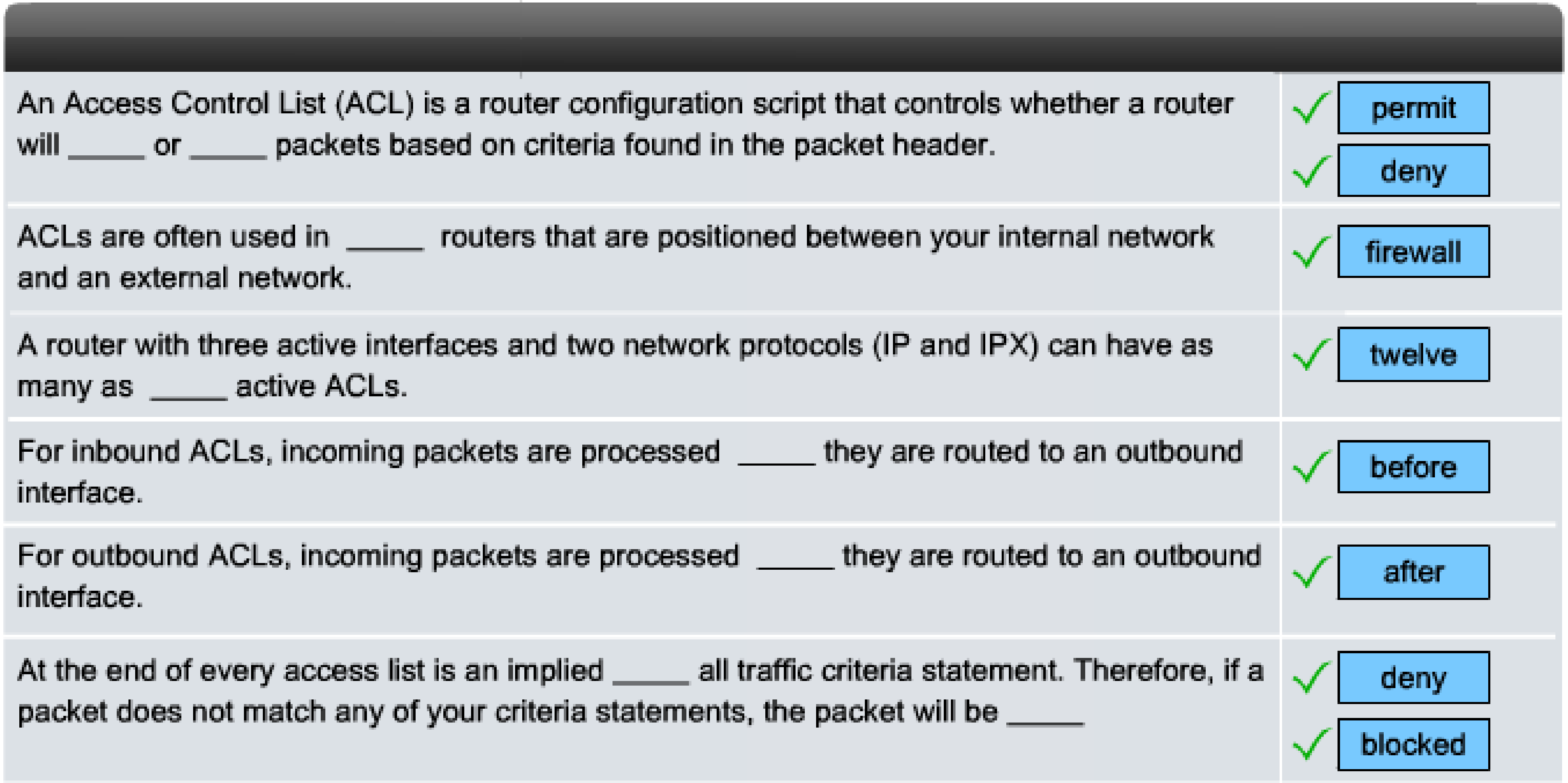
ACLs are often used in _____ routers that are positioned between your internal network and an external network.

A router with three active interfaces and two network protocols (IP and IPX) can have as many as _____ active ACLs.

For inbound ACLs, incoming packets are processed _____ they are routed to an outbound interface.

For outbound ACLs, incoming packets are processed _____ they are routed to an outbound interface.

At the end of every access list is an implied _____ all traffic criteria statement. Therefore, if a packet does not match any of your criteria statements, the packet will be _____



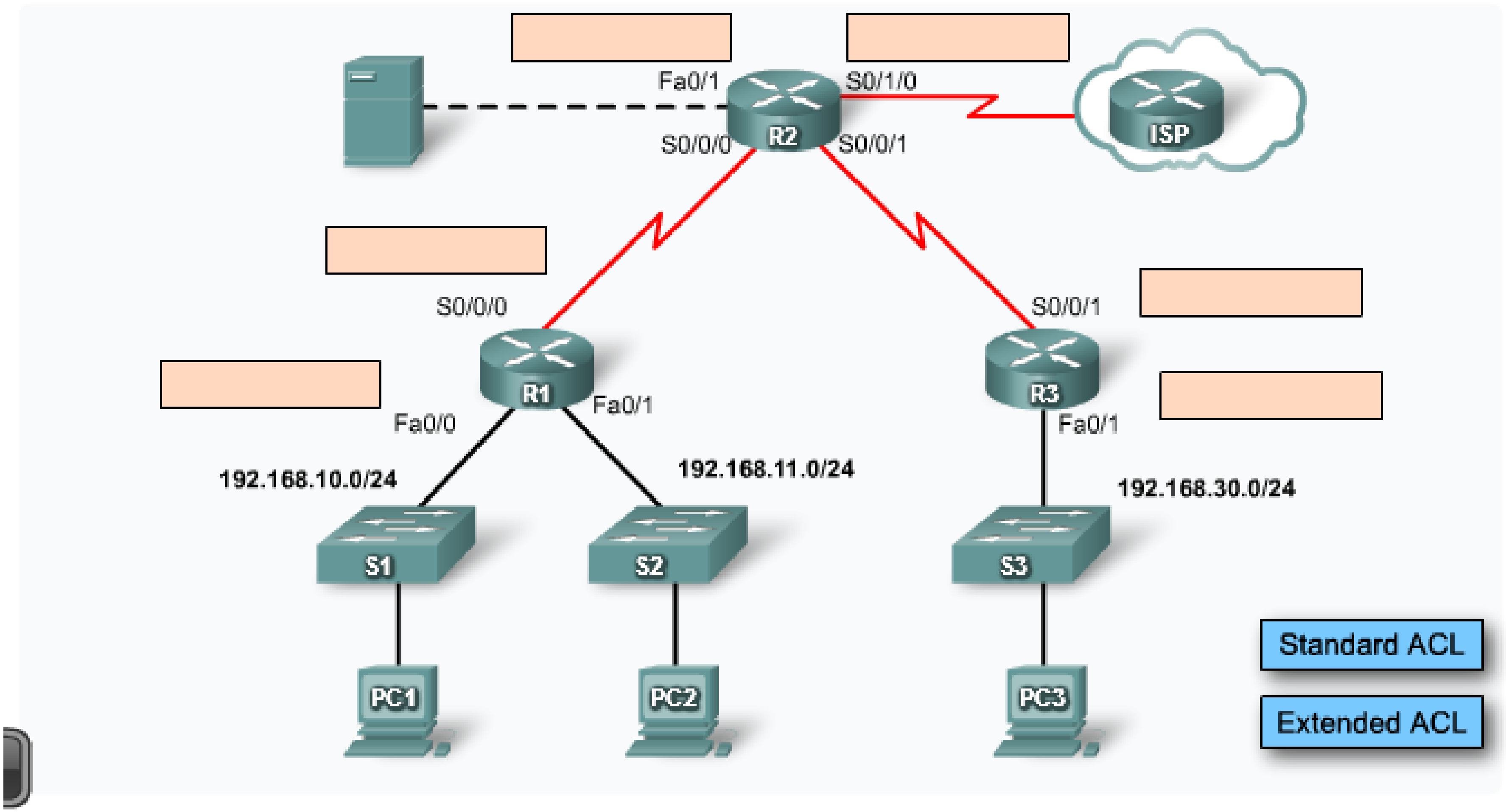
An Access Control List (ACL) is a router configuration script that controls whether a router will _____ or _____ packets based on criteria found in the packet header.	✓ permit ✓ deny
ACLs are often used in _____ routers that are positioned between your internal network and an external network.	✓ firewall
A router with three active interfaces and two network protocols (IP and IPX) can have as many as _____ active ACLs.	✓ twelve
For inbound ACLs, incoming packets are processed _____ they are routed to an outbound interface.	✓ before
For outbound ACLs, incoming packets are processed _____ they are routed to an outbound interface.	✓ after
At the end of every access list is an implied _____ all traffic criteria statement. Therefore, if a packet does not match any of your criteria statements, the packet will be _____	✓ deny ✓ blocked

	Standard	Extended
Can filter traffic based on source IP address		
Can filter traffic based on destination IP address		
Can filter traffic based on protocol type		
Uses numbers 1 - 99		
Uses number 100 - 199		
Uses number 1300 - 1999		
Can use a name instead of a number		

	Standard	Extended
Can filter traffic based on source IP address	✓	✓
Can filter traffic based on destination IP address		✓
Can filter traffic based on protocol type		✓
Uses numbers 1 - 99	✓	
Uses number 100 - 199		✓
Uses number 1300 - 1999	✓	
Can use a name instead of a number	✓	✓

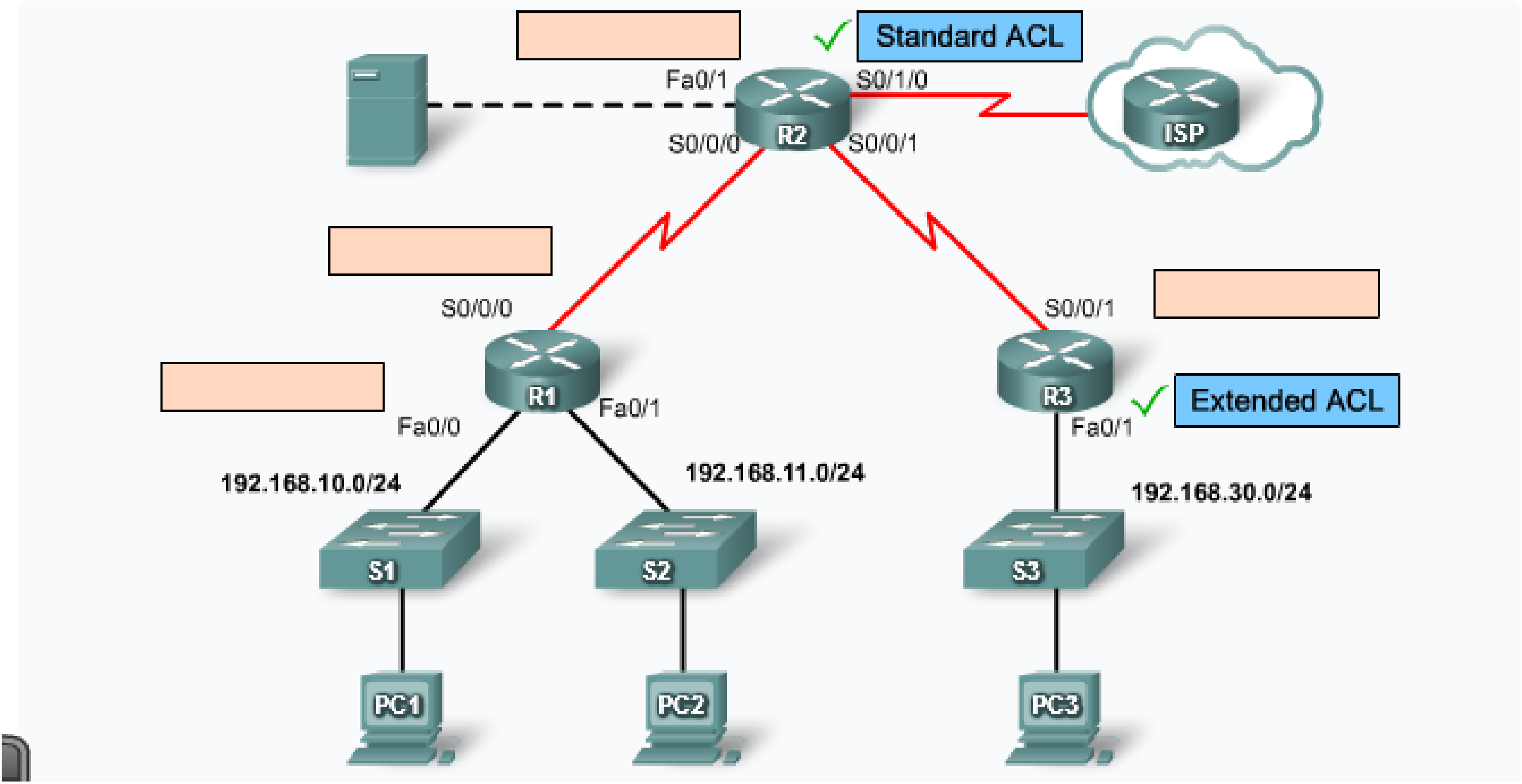
Network Policy #1: Use a standard ACL to stop the 192.168.10.0/24 network from accessing the Internet through ISP.

Network Policy #2: Use an extended ACL to stop the 192.168.30.0/24 network from accessing the Web/TFTP Server.



Network Policy #1: Use a standard ACL to stop the 192.168.10.0/24 network from accessing the Internet through ISP.

Network Policy #2: Use an extended ACL to stop the 192.168.30.0/24 network from accessing the Web/TFTP Server.



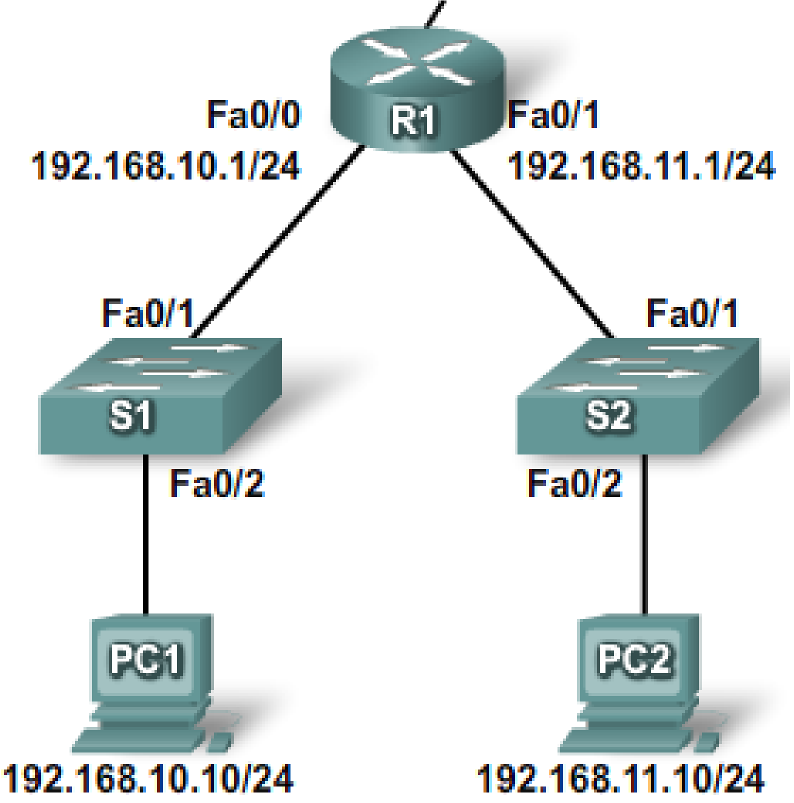
CONFIGURING STANDARD ACLS

Quick review

- when traffic comes into the router, it is compared to ACL statements based on the order that the entries occur in the router.
- The router continues to process the ACL statements until it has a match.
- For this reason, you should have the most frequently used ACL entry at the top of the list.
- If no matches are found when the router reaches the end of the list, the traffic is denied because there is an implied deny for traffic.

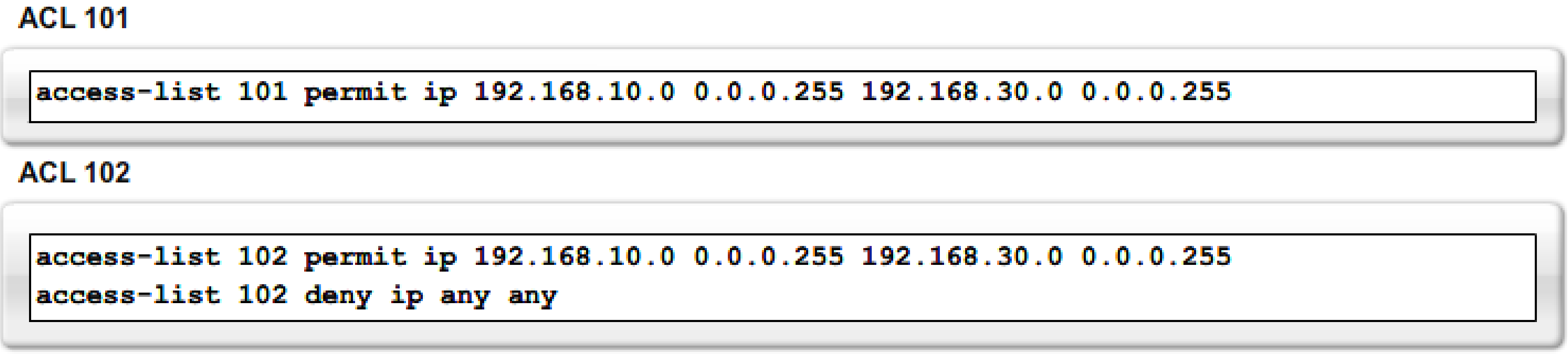
Quick review

- A single-entry ACL with only one deny entry has the effect of denying all traffic.
- You must have at least one permit statement in an ACL or all traffic is blocked.



the two ACLs (101 and 102) in the figure have the same effect.

Network 192.168.10.0 would be permitted to access network 192.168.30 while 192.168.11.0 would not be allowed.


ACL 101

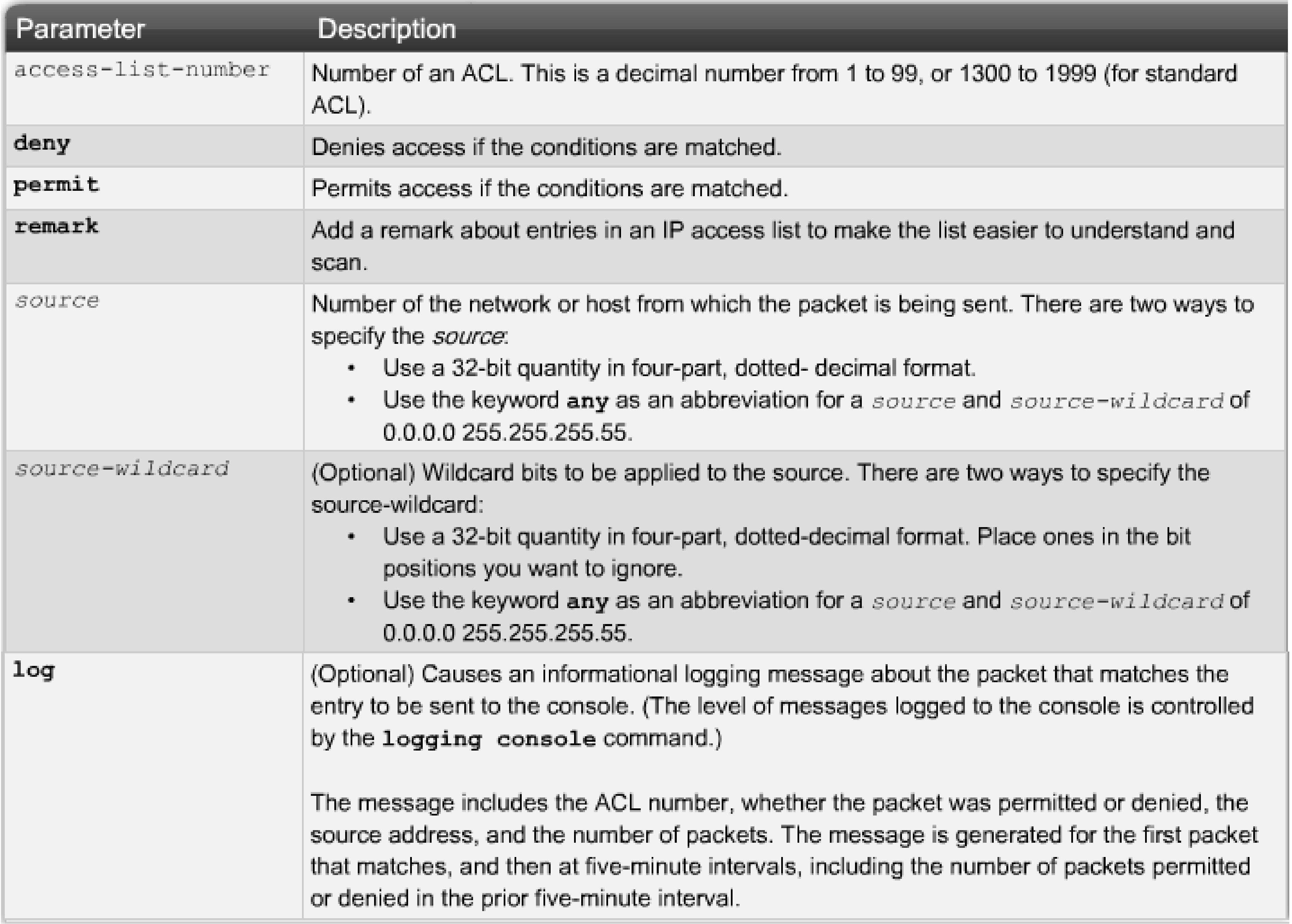
```
access-list 101 permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255
```

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
```

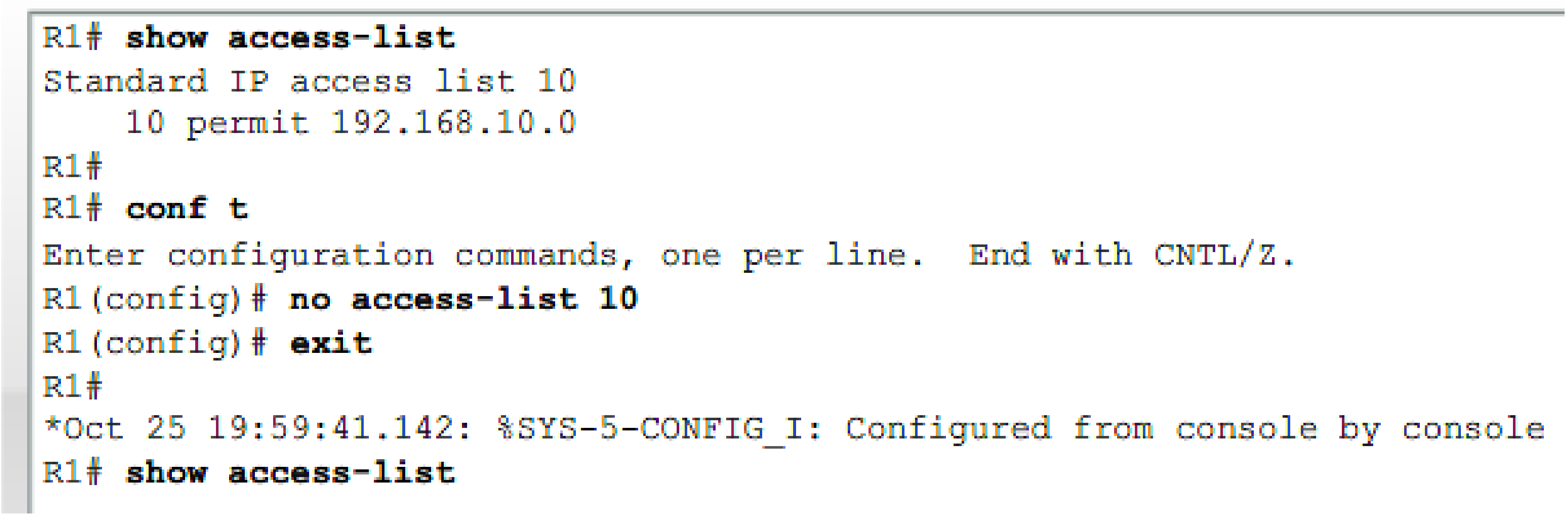
Configuring Standard ACLs

- To configure numbered standard ACLs on a Cisco router, you must first create the standard ACL and then activate the ACL on an interface.
- Router(config)#**access-list** *access-list-number* **deny/permit remark** *source_IP_address* [source-wildcard] [log]
- For example, to create a numbered ACL designated 10 that would permit network 192.168.10.0 /24, you would enter:
  - R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255



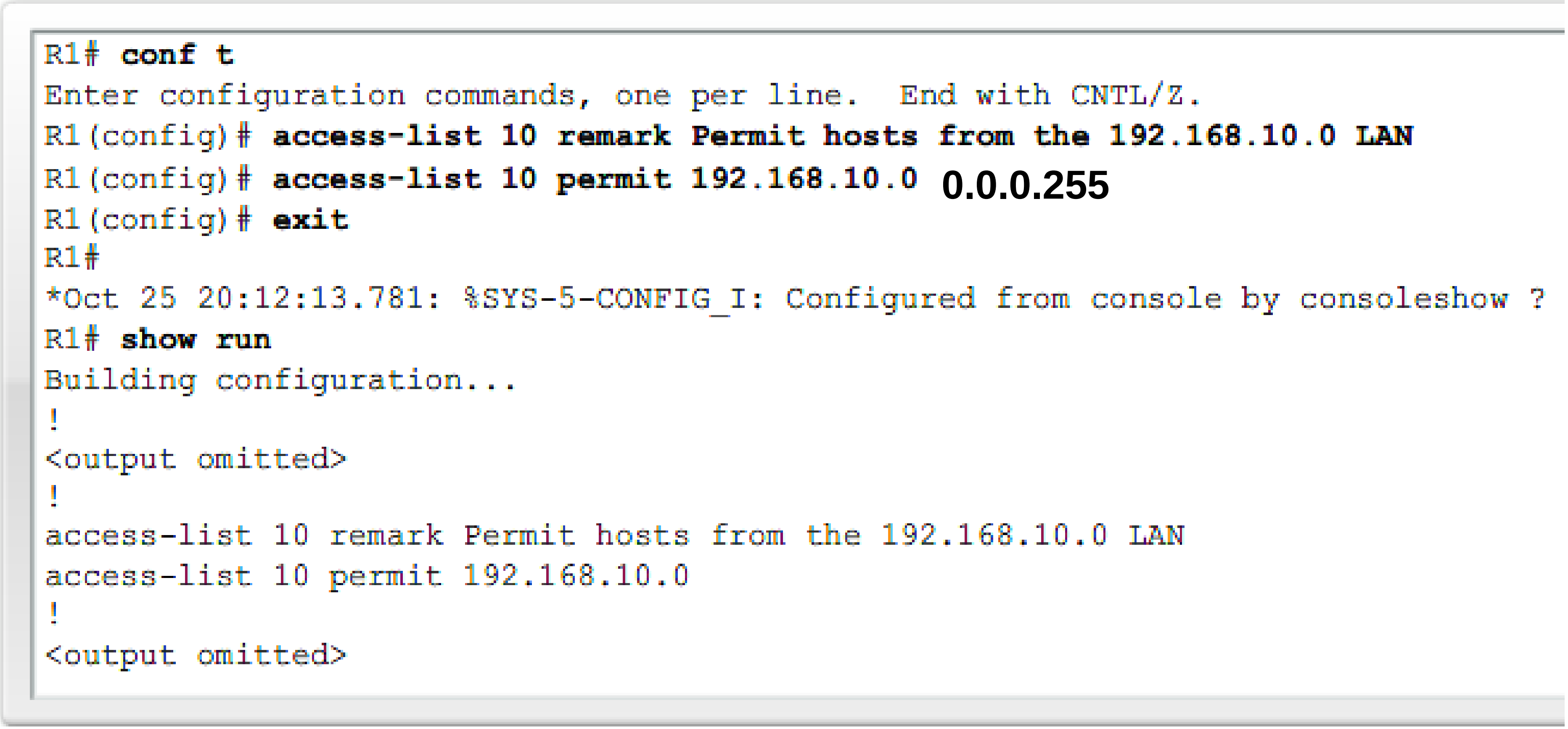
Parameter	Description
`access-list-number`	Number of an ACL. This is a decimal number from 1 to 99, or 1300 to 1999 (for standard ACL).
`deny`	Denies access if the conditions are matched.
`permit`	Permits access if the conditions are matched.
`remark`	Add a remark about entries in an IP access list to make the list easier to understand and scan.
`source`	Number of the network or host from which the packet is being sent. There are two ways to specify the *source*: • Use a 32-bit quantity in four-part, dotted- decimal format. • Use the keyword **`any`** as an abbreviation for a *`source`* and *`source-wildcard`* of 0.0.0.0 255.255.255.55.
`source-wildcard`	(Optional) Wildcard bits to be applied to the source. There are two ways to specify the source-wildcard: • 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 *`source`* and *`source-wildcard`* of 0.0.0.0 255.255.255.55.
`log`	(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 **`logging console`** command.) 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.

Removing an ACL



```
R1# show access-list
Standard IP access list 10
    10 permit 192.168.10.0
R1#
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# no access-list 10
R1(config)# exit
R1#
*Oct 25 19:59:41.142: %SYS-5-CONFIG_I: Configured from console by console
R1# show access-list
```

Document an ACL



```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# access-list 10 remark Permit hosts from the 192.168.10.0 LAN
R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255
R1(config)# exit
R1#
*Oct 25 20:12:13.781: %SYS-5-CONFIG_I: Configured from console by consoleshow ?
R1# show run
Building configuration...
!
<output omitted>
!
access-list 10 remark Permit hosts from the 192.168.10.0 LAN
access-list 10 permit 192.168.10.0
!
<output omitted>
```

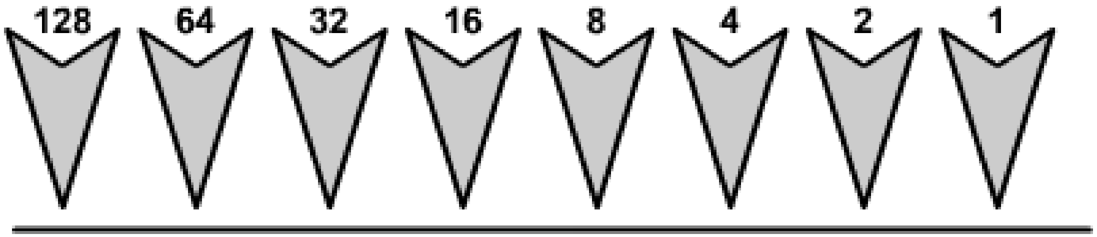
ACL Wildcard Masking

A wildcard mask is a string of binary digits telling the router which parts of the subnet number to look at.

Wildcard masks use binary 1s and 0s to filter individual or groups of IP addresses to permit or deny access to resources based on an IP address.

By carefully setting wildcard masks, you can permit or deny a single or several IP addresses

Wildcard mask bit 0 - Match the corresponding bit value in the address
Wildcard mask bit 1 - Ignore the corresponding bit value in the address



Octet Bit Position and Address Value for Bit

128	64	32	16	8	4	2	1		Examples
0	0	0	0	0	0	0	0	=	Match 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	=	Ignore First 6 Address Bits
1	1	1	1	1	1	1	1	=	Do Not Check Address (Ignore Bits in Octet)

0 means to match the value of the corresponding address bit
1 means to ignore the value of the corresponding address bit

	Decimal Address	Binary Address
IP address to be processed	192.168.10.0	11000000.10101000.00001010.00000000
Wildcard mask	0.0.255.255	00000000.00000000.11111111.11111111
Resulting IP address	192.168.0.0	11000000.10101000.00000000.00000000

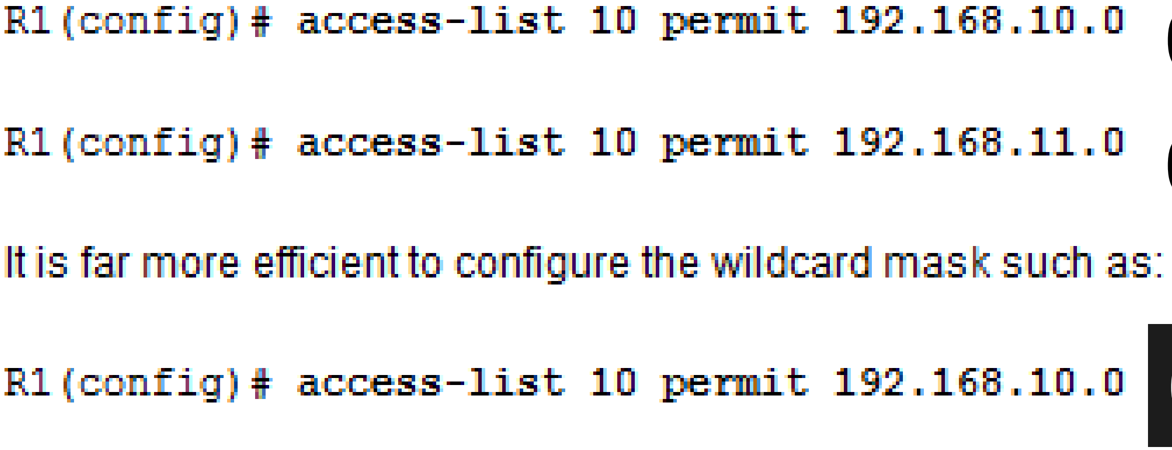
Wildcard Mask Examples

	Decimal	Binary
IP Address	192.168.1.1	11000000.10101000.00000001 .00000001
Wildcard Mask	0.0.0.0.	00000000.00000000.00000000.00000000
Result	192.168.1.1	11000000.10101000.00000001 .00000001

	Decimal	Binary
IP Address	192.168.1.1	11000000.10101000.00000001 .00000001
Wildcard Mask	255.255.255.255	11111111.11111111.11111111.11111111
Result	0.0.0.0	00000000.00000000.00000000.00000000

	Decimal	Binary
IP Address	192.168.1.1	11000000.10101000.00000001 .00000001
Wildcard Mask	0.0.0.255	00000000.00000000.00000000.11111111
Result	192.168.1.0	11000000.10101000.00000001.00000000

	Decimal	Binary
IP Address	192.168.1.0	11000000.10101000.00000001 .00000000
Wildcard Mask	0.0.254.255	00000000.00000000.11111110.11111111
Result	192.168.1.0	11000000.10101000.00000001.00000000
	All odd numbered subnets in the 192.168.0.0 major network	



```
R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255

R1(config)# access-list 10 permit 192.168.11.0 0.0.0.255
```

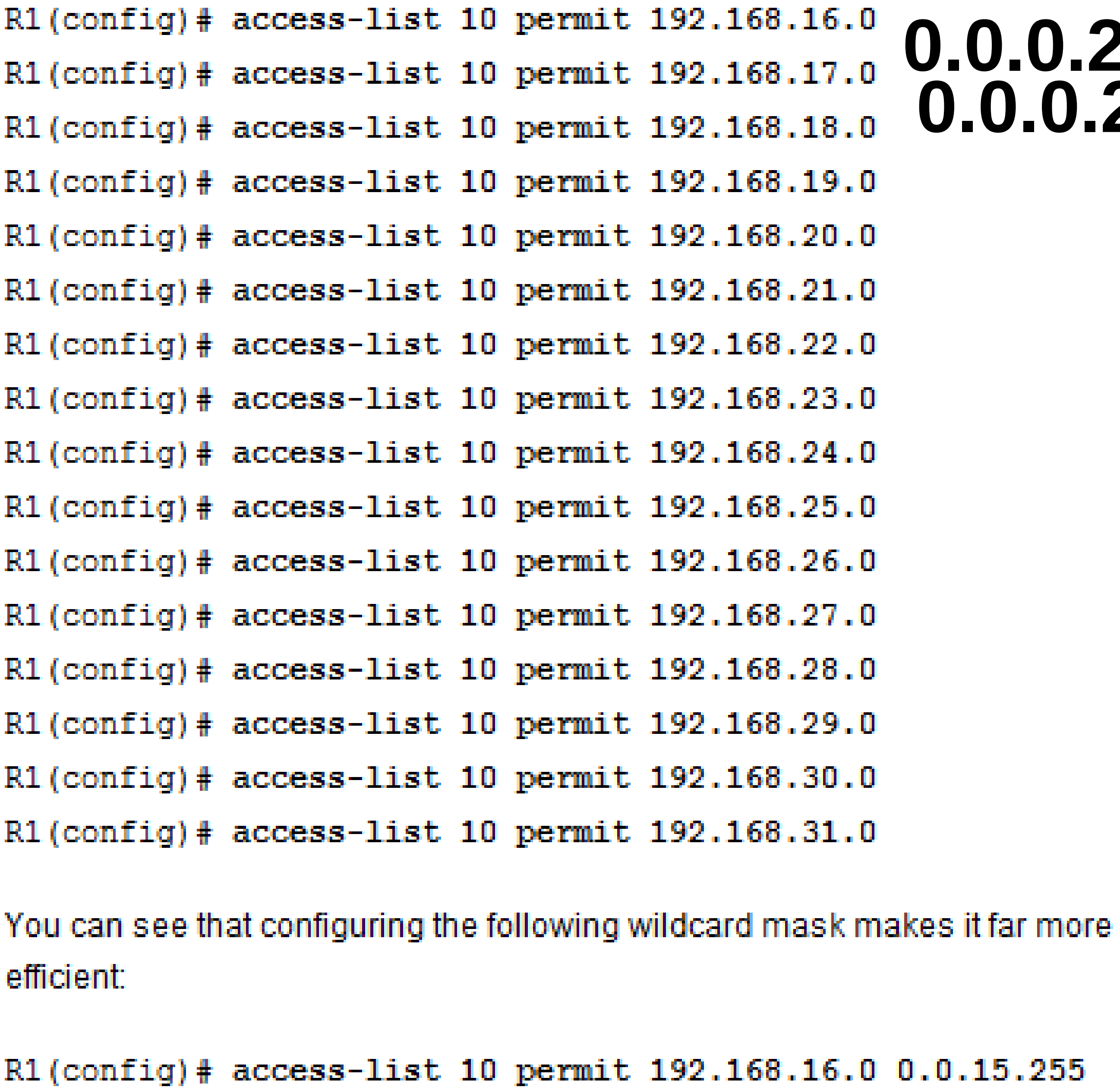
It is far more efficient to configure the wildcard mask such as:

```
R1(config)# access-list 10 permit 192.168.10.0 0.0.1.255
```

Ip: 192.168.(00001010).0

192.168.(00001011).0

Wm: 0.0.(00000001).255



```
R1(config)# access-list 10 permit 192.168.16.0
R1(config)# access-list 10 permit 192.168.17.0
R1(config)# access-list 10 permit 192.168.18.0
R1(config)# access-list 10 permit 192.168.19.0
R1(config)# access-list 10 permit 192.168.20.0
R1(config)# access-list 10 permit 192.168.21.0
R1(config)# access-list 10 permit 192.168.22.0
R1(config)# access-list 10 permit 192.168.23.0
R1(config)# access-list 10 permit 192.168.24.0
R1(config)# access-list 10 permit 192.168.25.0
R1(config)# access-list 10 permit 192.168.26.0
R1(config)# access-list 10 permit 192.168.27.0
R1(config)# access-list 10 permit 192.168.28.0
R1(config)# access-list 10 permit 192.168.29.0
R1(config)# access-list 10 permit 192.168.30.0
R1(config)# access-list 10 permit 192.168.31.0
```

You can see that configuring the following wildcard mask makes it far more efficient:

```
R1(config)# access-list 10 permit 192.168.16.0 0.0.15.255
```

0.0.0.255

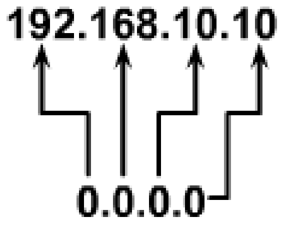
0.0.0.255

Wildcard Bit Mask Abbreviations

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)

Wildcard Mask:



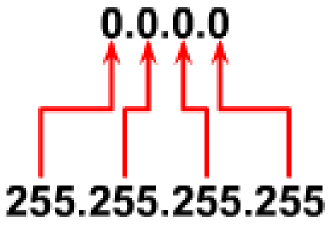
(Match All Bits)

```
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
```

Example 2:

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

Wildcard Mask:



(Ignores All Bits)

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

Procedure for Configuring Standard ACLs

Step 1 Use the **access-list** global configuration command to create an entry in a standard IPv4 ACL.

```
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
```

Enter the global **no access-list** command to remove the entire ACL. The example statement matches any address that starts with 192.168.10.x. Use the **remark** option to add a description to your ACL.

Step 2 Use the interface configuration command to select an interface to which to apply the ACL

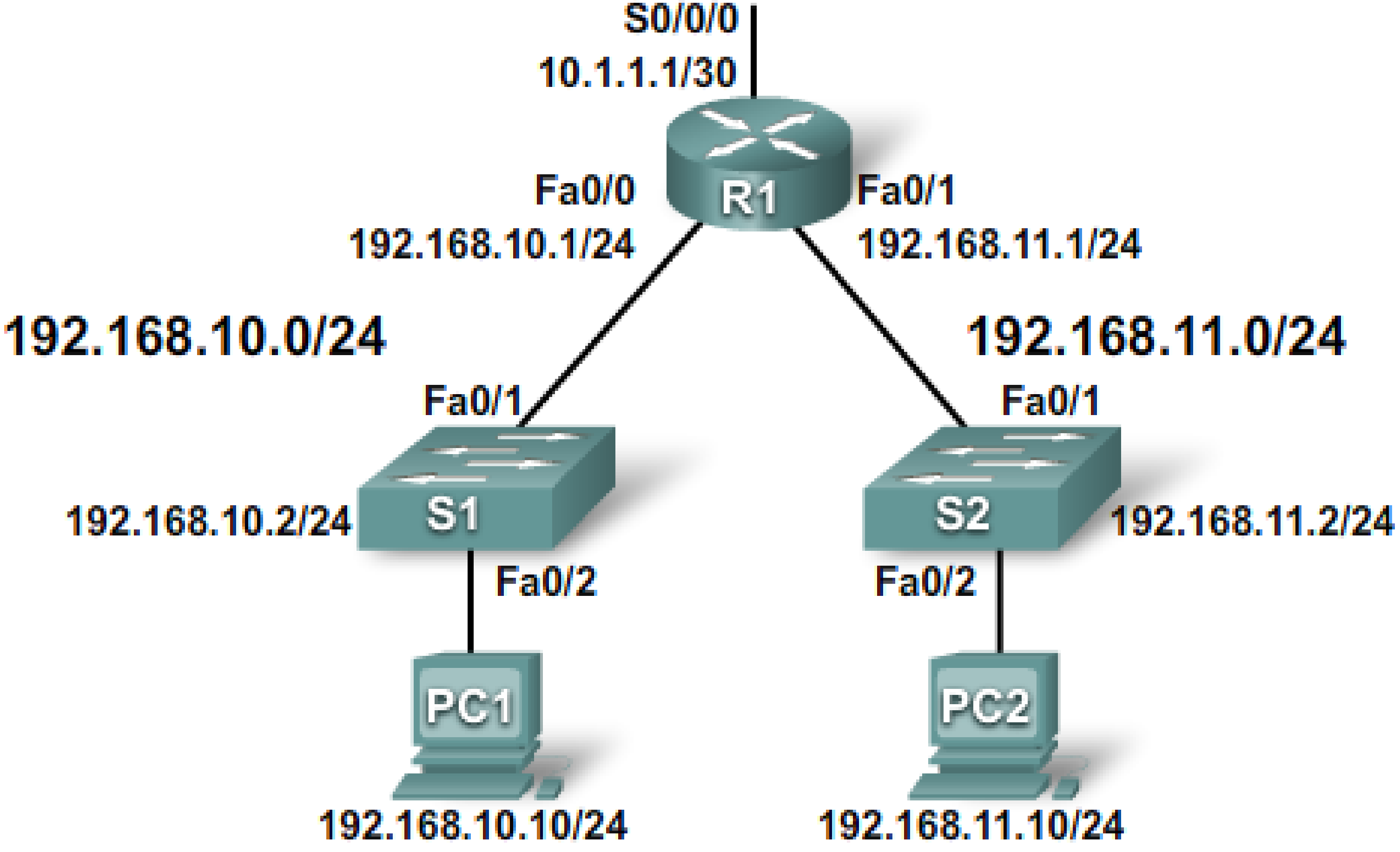
```
R1(config)# interface FastEthernet 0/0
```

Step 3 Use the **ip access-group** interface configuration command to activate the existing ACL on an interface.

```
R1(config-if)# ip access-group 1 out
```

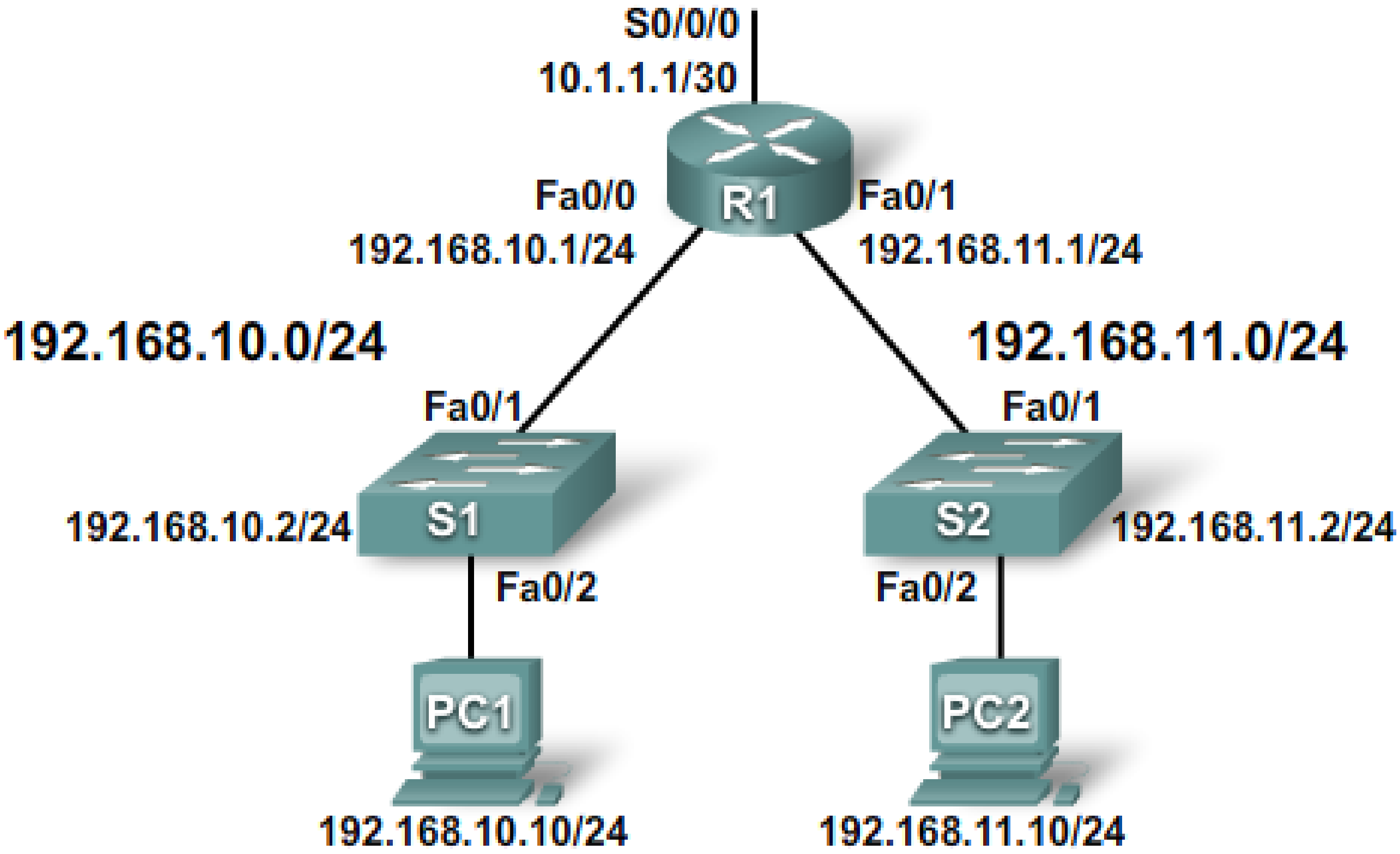
To remove an IP ACL from an interface, enter the **no ip access-group** command on the interface. This example activates the standard IPv4 ACL 1 on the interface as an outbound filter.

Standard ACL to Permit My Network Only



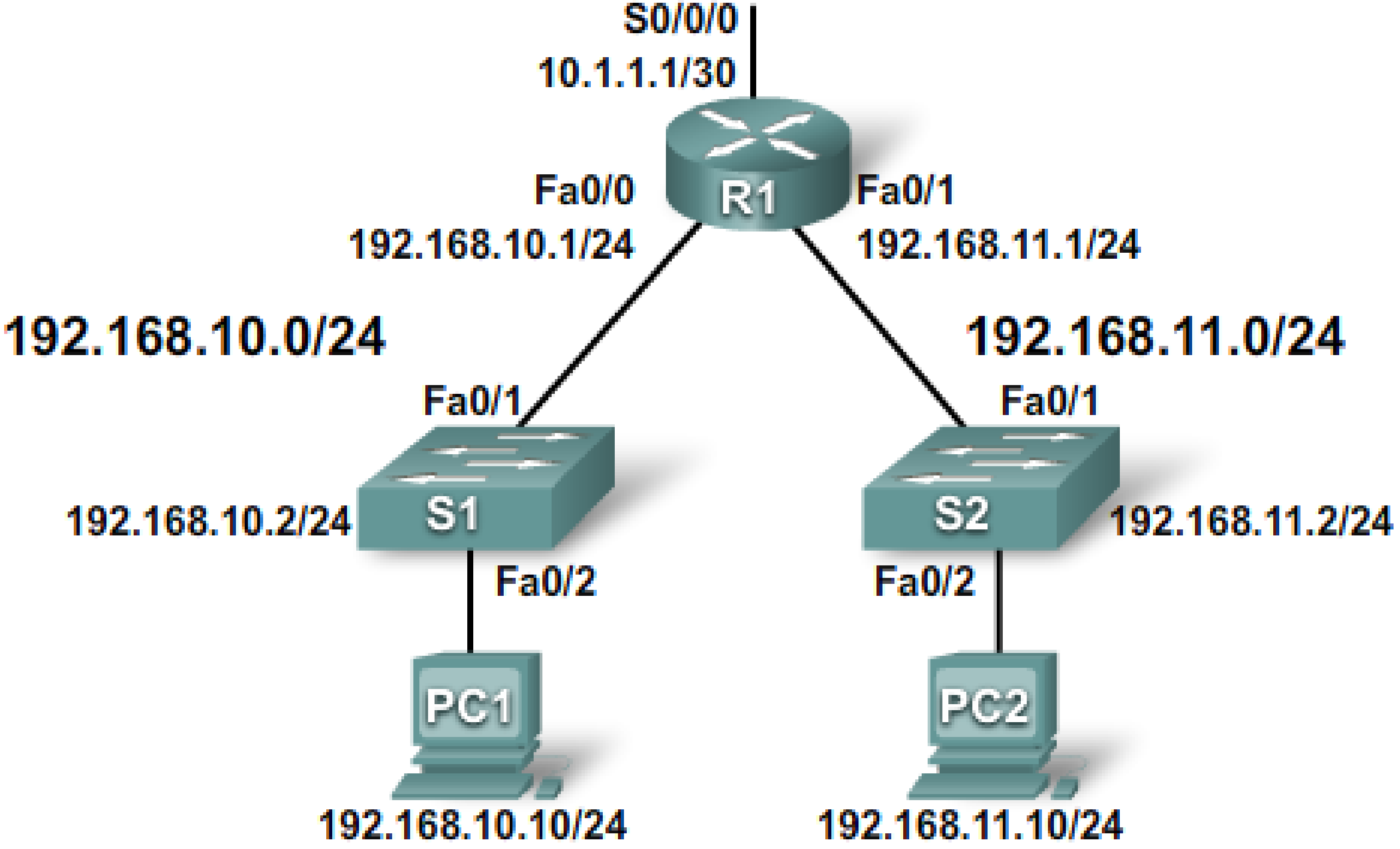
```
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)# interface S0/0/0
R1(config-if)# ip access-group 1 out
```

Standard ACL to Deny a Specific Host



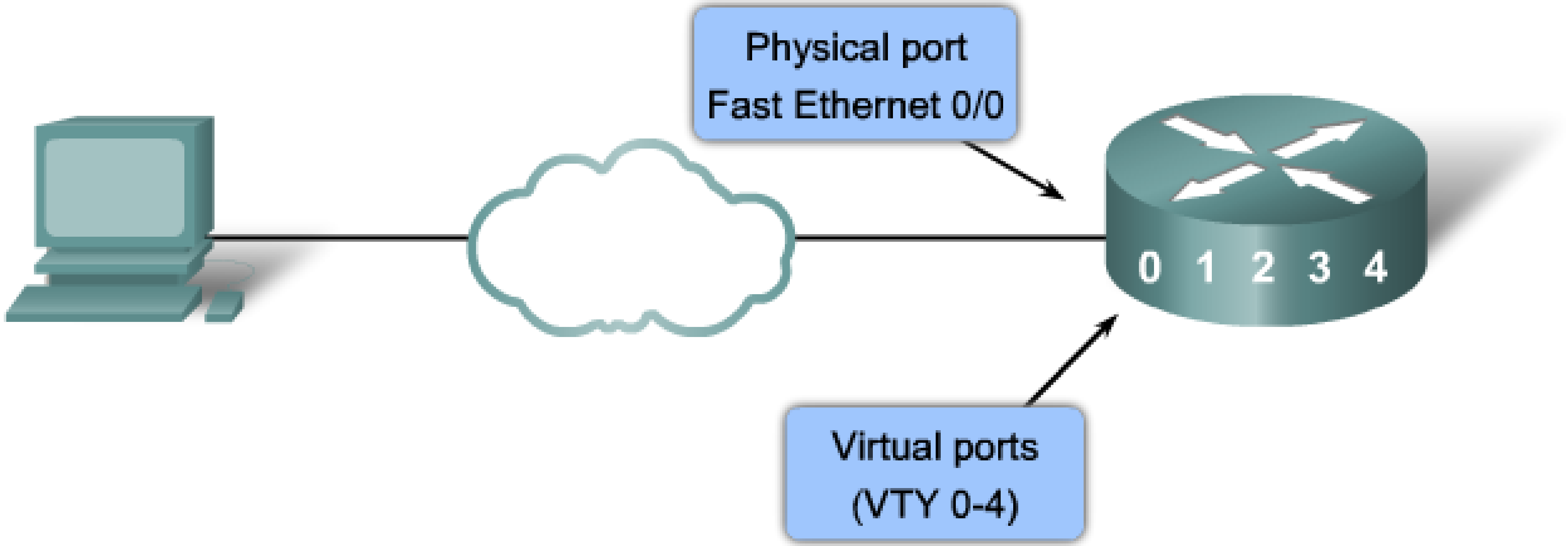
```
R1(config)#no access-list 1
R1(config)#access-list 1 deny 192.168.10.10 0.0.0.0
R1(config)#access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#interface S0/0/0
R1(config-if)#ip access-group 1 out
```

Standard ACL to Deny a Specific Subnet



```
R1(config)#no access-list 1
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
```

Standard ACLs to Control Virtual Terminal Access



Physical port
Fast Ethernet 0/0

Virtual ports
(VTY 0-4)

```
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
```