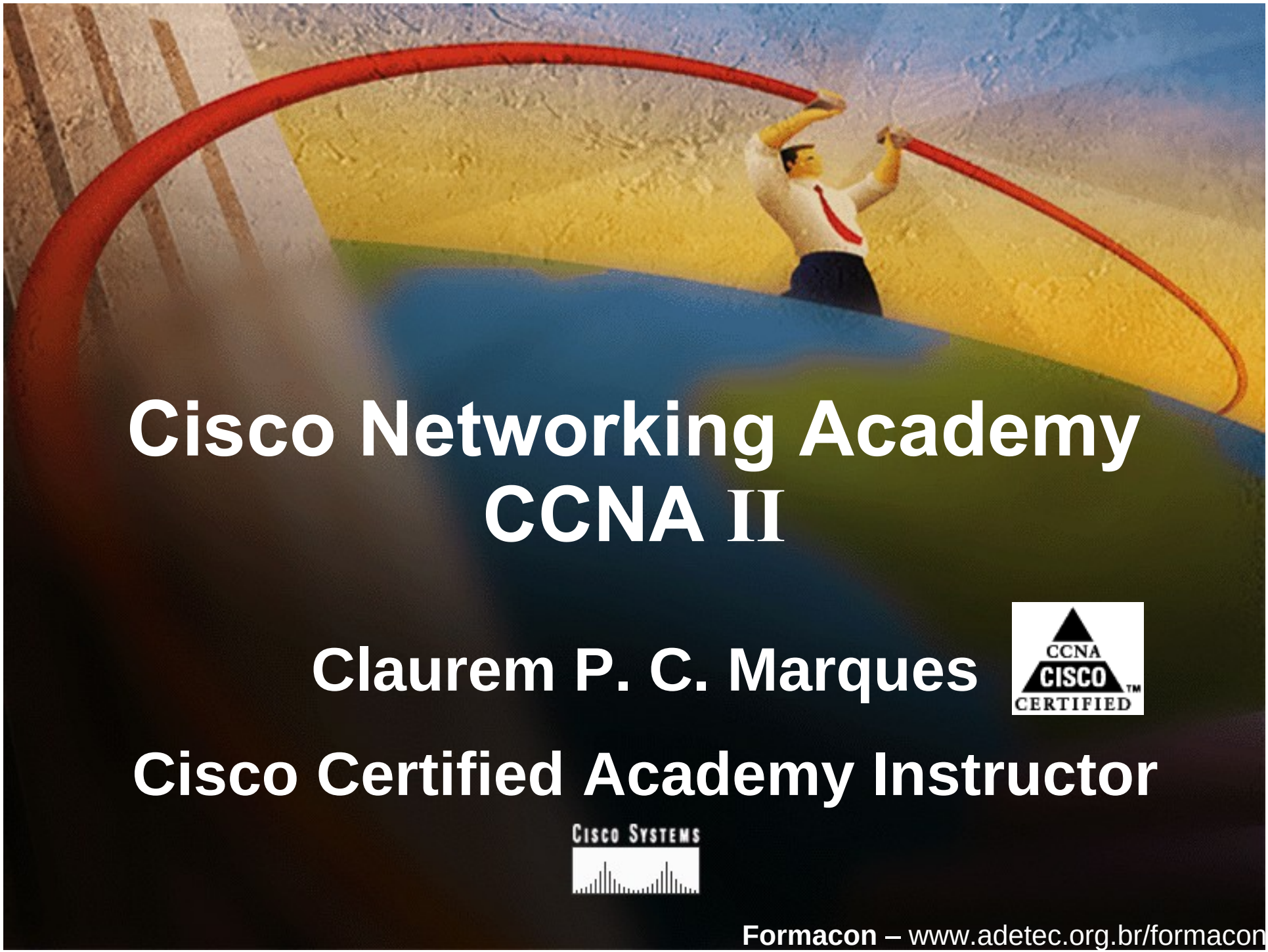


Cisco Networking Academy
CCNA II

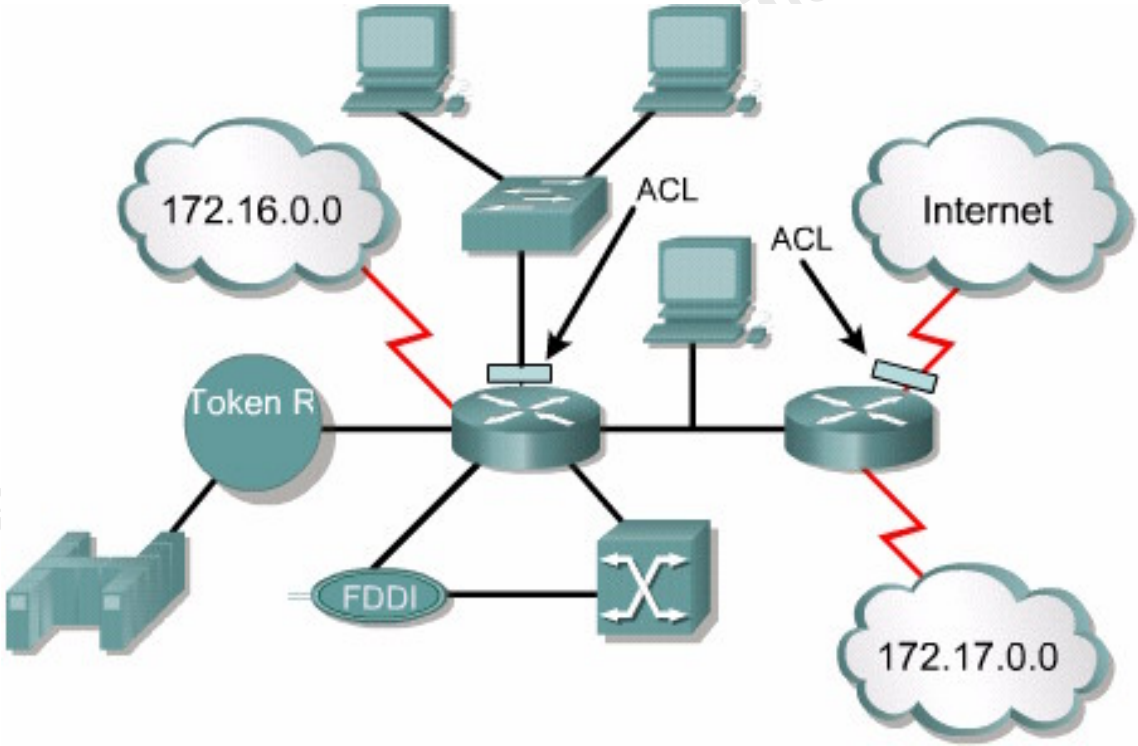
Claurem P. C. Marques

Cisco Certified Academy Instructor

Capítulo 11 – Listas de Controle de Acesso (ACLs)

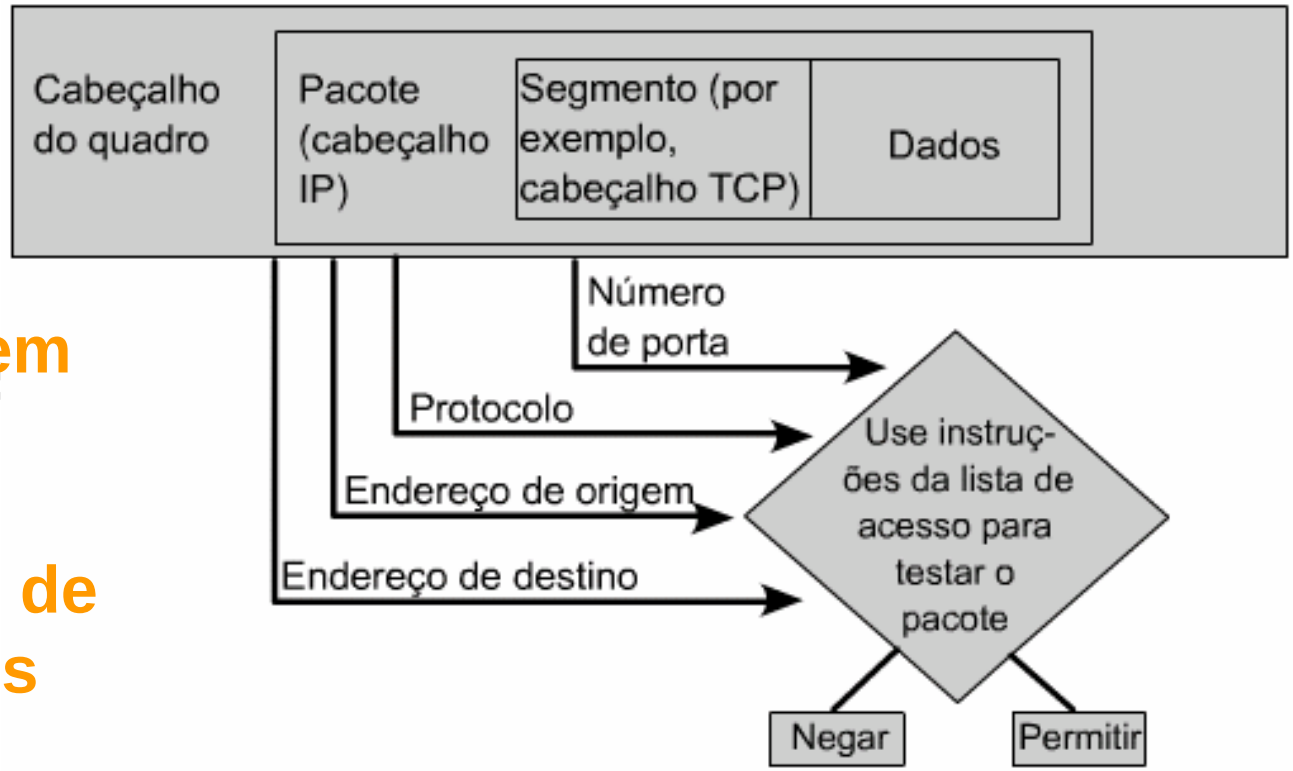
O que são ACLs

- **ACLs são listas de condições aplicadas ao tráfego que viaja através da rede**
- **Estas listas tem o objetivo de informar o roteador sobre tipos tráfego que ele deve aceitas ou recusar**



Verificação dos Cabeçalhos dos Pacotes e das Camadas Superiores

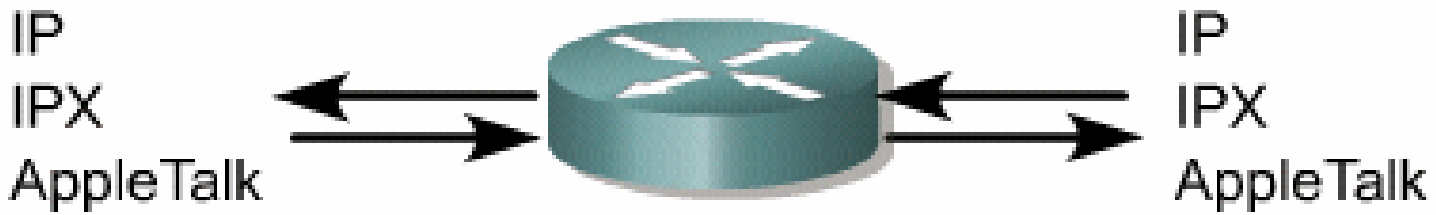
- **ACLs podem ser criadas para todos os protocolos de rede roteados, dessa maneira elas podem controlar o acesso a uma rede ou sub-rede.**



Endereços de origem e destino;
Protocolos;
Números de portas de camadas superiores

Agrupamento de ACL em um Router

- **As ACLs devem ser definidas por protocolo, por direção ou por porta**
- **Para controlar o fluxo de tráfego em uma interface, deve-se definir uma ACL para cada protocolo ativado na interface**

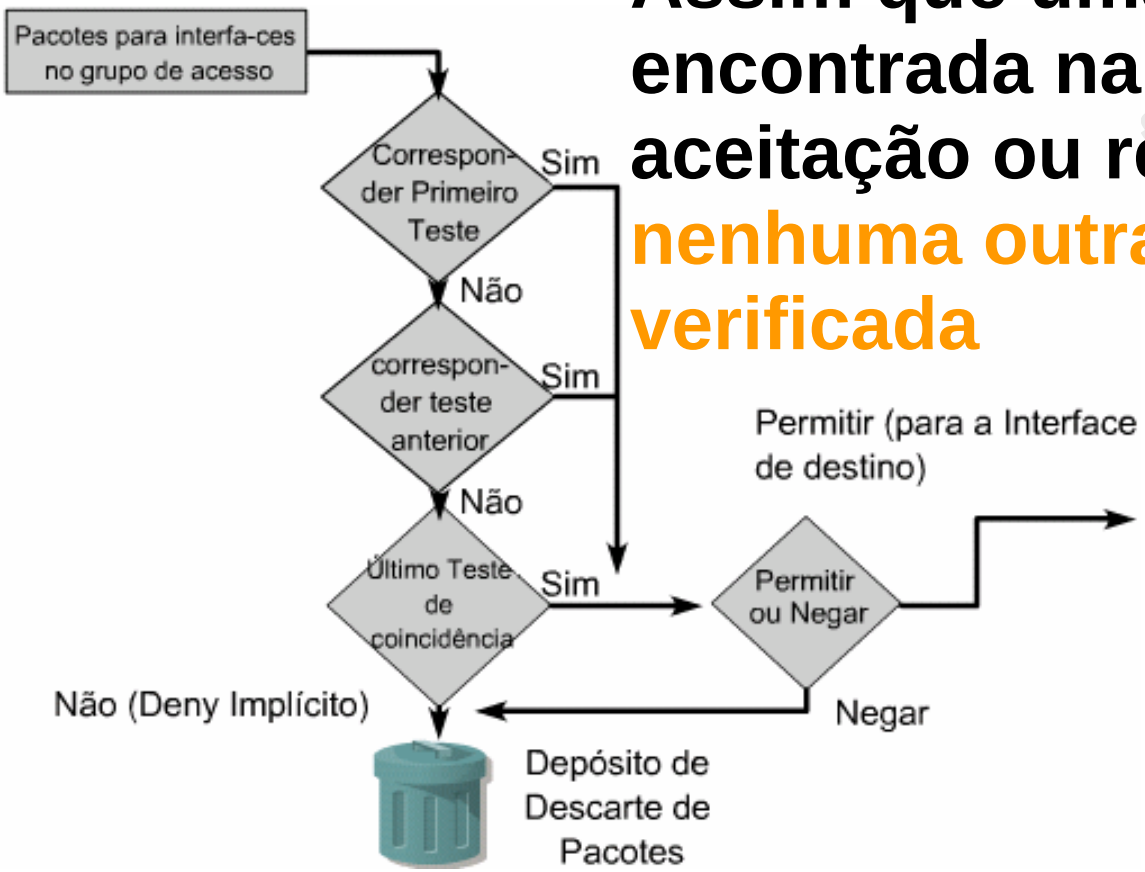


Uma lista por porta, por direção, por protocolo

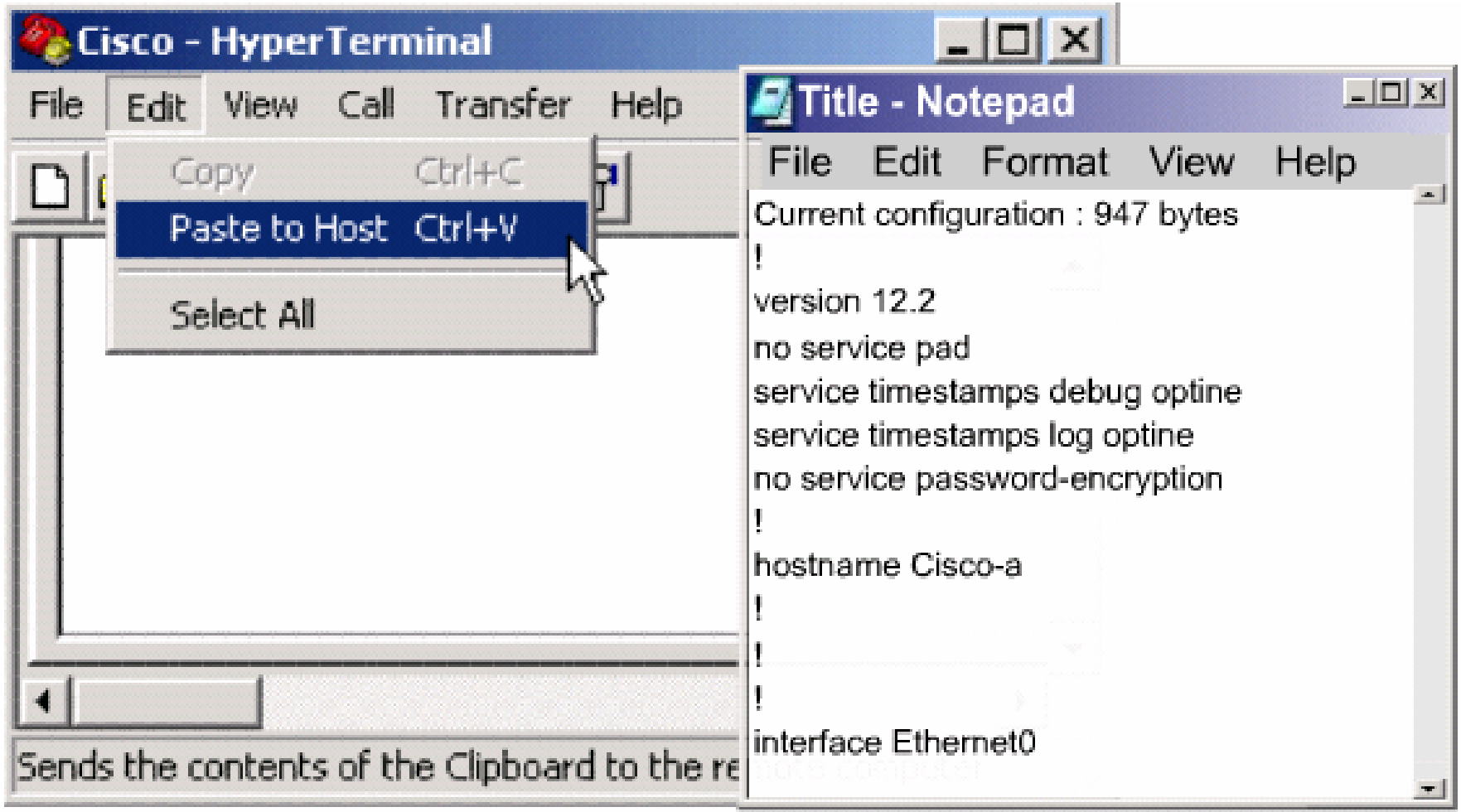
Com duas interfaces e três protocolos sendo executados, esse roteador poderia ter um total de 12 ACLs separadas aplicadas.

Funcionamento das ACLs

- **A ordem em que as instruções da ACL são posicionadas é importante**
- **Assim que uma correspondência é encontrada na lista, a ação de aceitação ou rejeição é realizada e nenhuma outra instrução da ACL é verificada**



Gerenciamento de Configuração



Um bom hábito a ser estabelecido seria a manutenção dos arquivos de configuração do seu roteador com um editor de texto. Depois, use a função colar para o host no HyperTerminal para inseri-la no roteador.

Criando ACLs

Definir a ACL usando o seguinte comando:

```
Router(config)# número da lista de acesso
 {permit | deny} {condições de teste}
```

Depois, você precisa aplicar as ACLs em uma interface usando o comando access-group, como mostrado neste exemplo:

```
Router(config-if)# {protocolo} access-group número da
lista de acesso
```

- Ao atribuir uma ACL à uma interface, é necessário a definição entre ACL de entrada e ACL de saída (se nada for especificado saída é padrão).
- As ACLs de saída são geralmente mais eficientes do que as de entrada e por isso são preferidas.

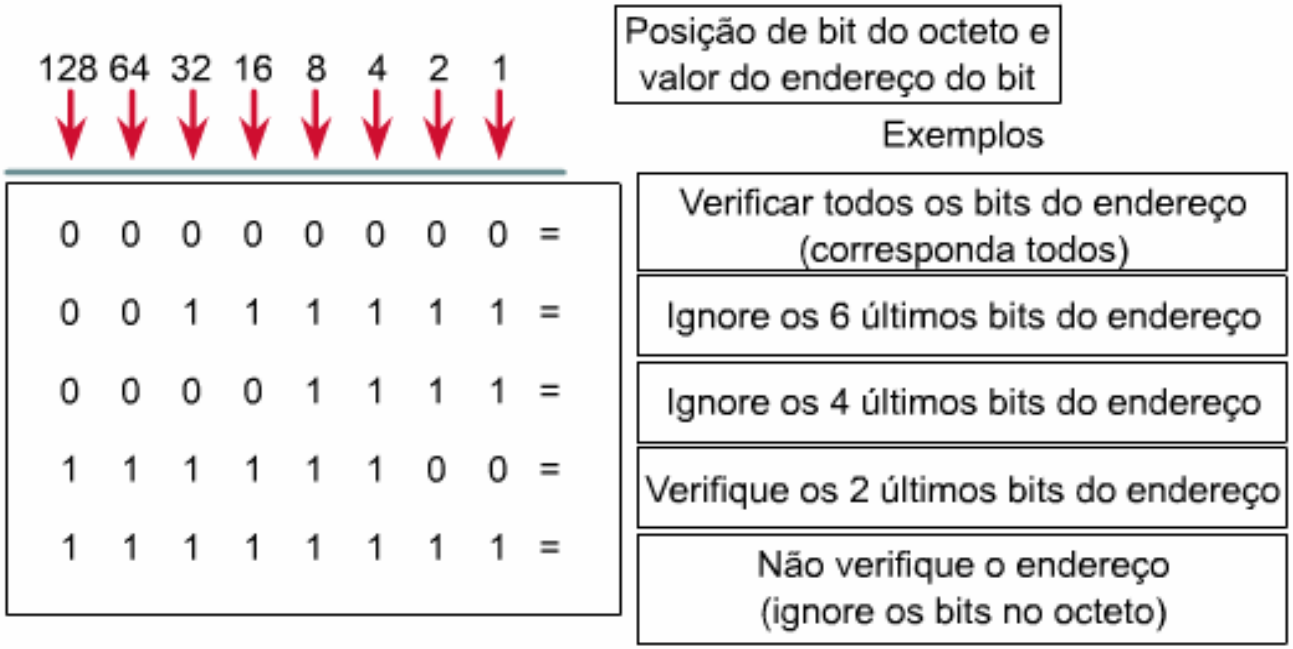
Números para ACLs

Protocolo	Intervalo
IP	1-99
IP estendido	100-199
AppleTalk	600-699
IPX	800-899
IPX estendido	900-999
Protocolo de anúncio de serviços IPX	1000-1099

- **Quando configurar ACLs em um roteador, você deverá identificar cada ACL com exclusividade, atribuindo um número à ACL do protocolo. Quando você usar um número para identificar uma ACL, o número deverá estar dentro de um intervalo específico que seja válido para o protocolo.**

Máscara Curinga

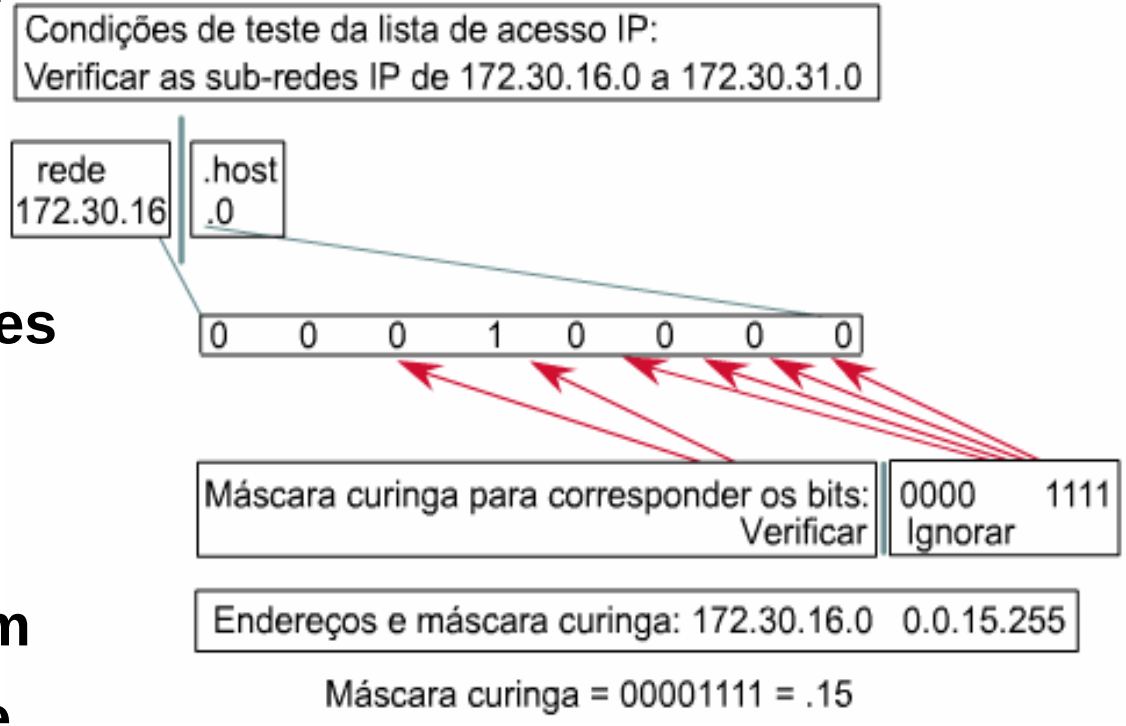
- **Embora ambas tenham 32 bits, as máscaras-curinga e as máscaras de sub-rede IP operam de forma distinta.**



- 0 = verificar o valor do bit correspondente
- 1 = não verificar (ignorar) esse valor do bit correspondente

Exemplo de uso da Máscara Curinga

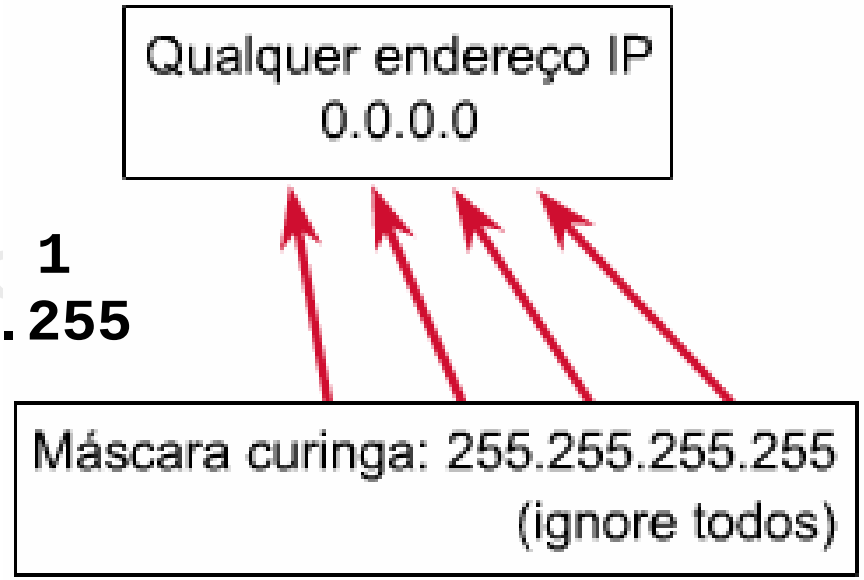
- **O endereço 172.30.16.0 com a máscara-curinga 0.0.15.255 corresponde às sub-redes de 172.30.16.0 a 172.30.31.0.**
- **A máscara-curinga não faz correspondência com nenhuma outra sub-rede.**



10101100.00011110.0001 0000. 00000000	Endereço IP Sub-rede
11111111.11111111.1111 1111. 00000000	Máscara de Sub-rede
00000000.00000000.0000 1111. 11111111	Máscara Curinga
vvvvvvvv.vvvvvvvv.vvvvnnnn. nnnnnnnn	Resultado

O comando any

- **O "any" é um atalho do IOS para 0.0.0.0 255.255.255.255 em uma instrução de lista de acesso.**



- **A invés de usar:**

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

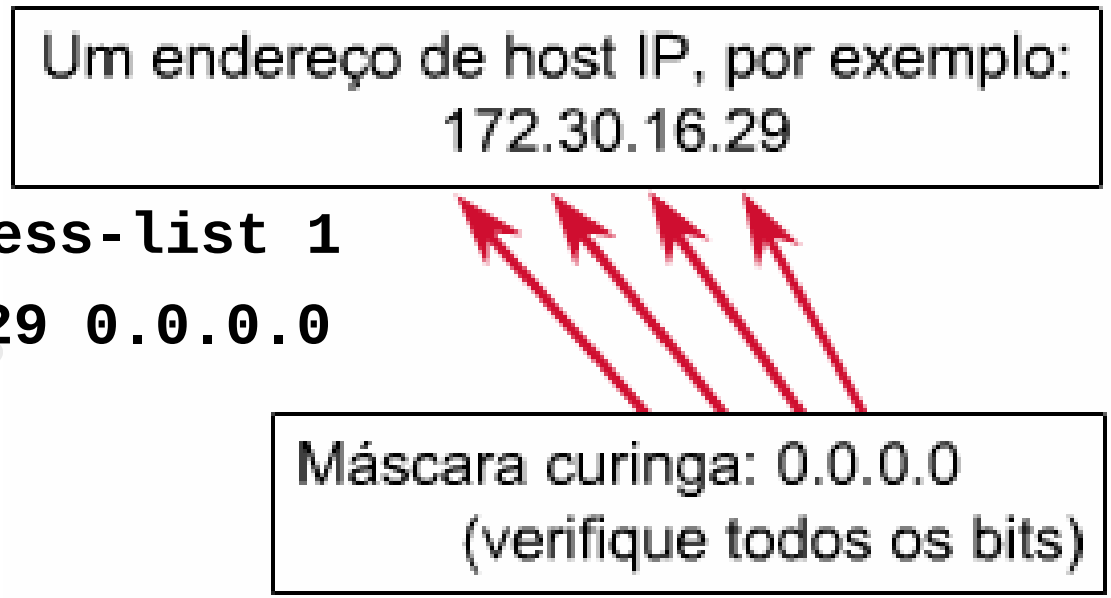
- **Pode ser usado:**

```
Router(config) # access-list 1
  permit any
```

O comando `host`

- **Um outro atalho do IOS é o comando "host", que substitui 0.0.0.0 por uma máscara-curinga, o que significa que todos os bits devem ser verificados e devem coincidir para que a instrução da lista de acesso seja verdadeira.**
  - **A invés de usar:**

```
Router(config)# access-list 1
  permit 172.30.16.29 0.0.0.0
```



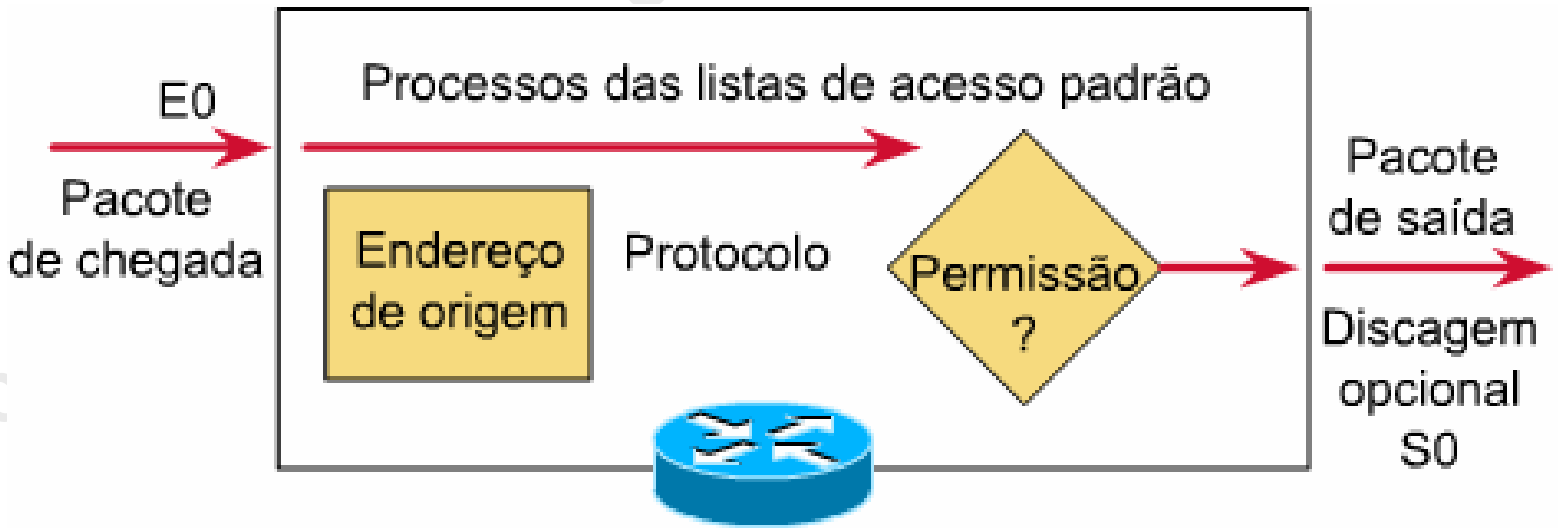
  - **Pode ser usado:**

```
Router(config)# access-list 1
  permit host 172.30.16.29
```

ACLs Padrão

- **As ACLs padrão, embora mais fáceis de serem criadas, proporcionam menos controle sobre o tráfego na rede.**
- **Você usa as ACLs padrão quando deseja bloquear todo tráfego de uma rede, permitir todo tráfego de uma rede específica ou negar conjuntos de protocolos.**

OBS.: Normalmente são definidas próximo ao destino



Criando uma ACL Padrão

- **A sintaxe completa do comando é:**

```
Router(config)# access-list número da ACL {deny |
  permit} origem [máscara curinga origem] [log]
```

- **Use a forma no desse comando para retirar uma ACL padrão. Esta é a sintaxe:**

```
Router(config)# no access-list número da ACL
```

- **Exemplos:**

```
Access-list 33 permit 172.16.0.0 0.0.255.255 log
  (permite todo tráfego de 172.16.0.0)

Access-list 44 deny 172.16.13.7 0.0.0.0 log (nega
  todo tráfego do host 172.16.13.7)

Access-list 55 deny 172.16.64.0 any log (nega
  todo tráfego da rede 172.16.64.0)
```

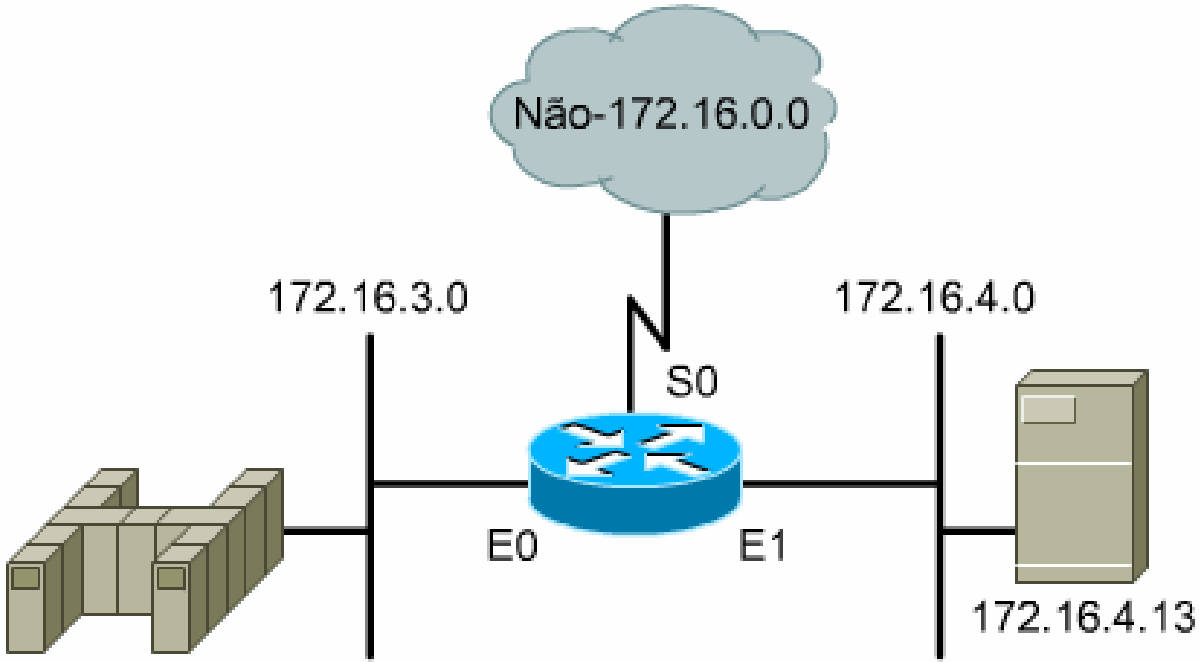
Verificando e agrupando as ACLs


```
Router# show access-lists
  access-list 1 permit 192.5.34.0 0.0.0.255
  access-list 1 permit 128.88.0.0 0.0.255.255
  access-list 1 permit 36.0.0.0 0.255.255.255

  access-list 2 permit 36.48.0.3 0.0.0.0
  access-list 2 permit 37.50.1.1 37.50.1.2


Router(config)# interface ethernet 0
Router(config-if)# ip access-group 1 in
Router(config-if)# interface serial 0
Router(config-if)# ip access-group 2
```

Exemplo 1: ACL Padrão

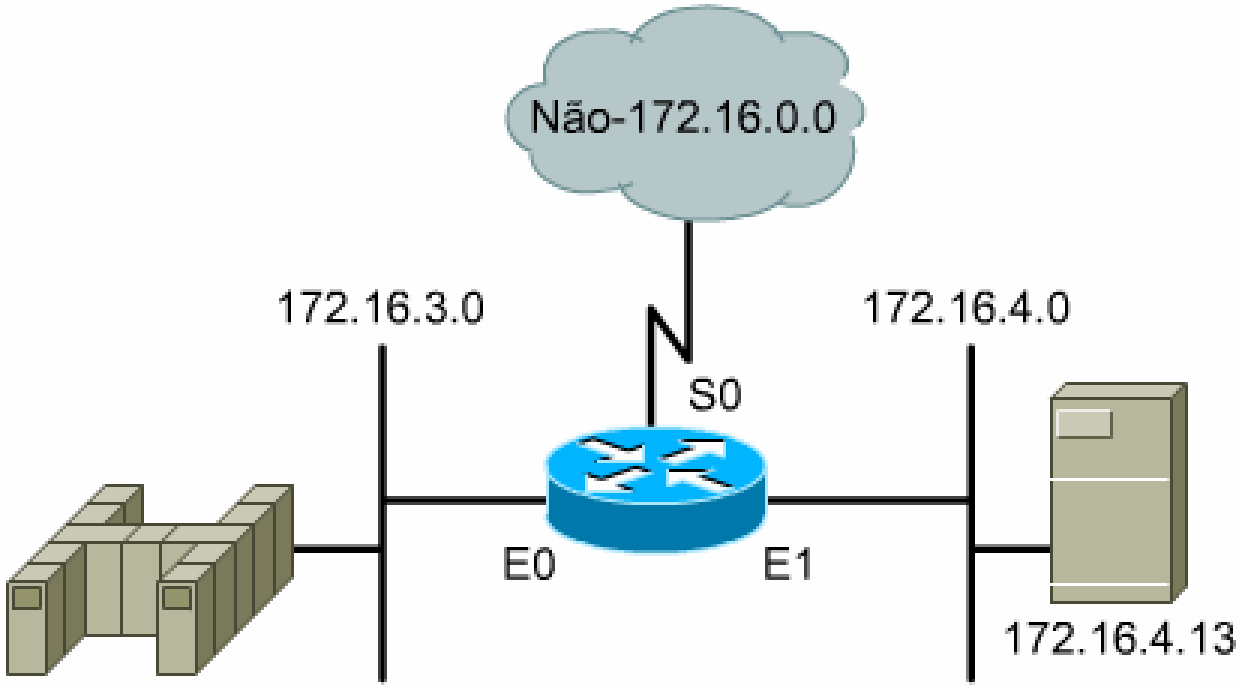


Saída do comando

```
access-list 1 permit 172.16.0.0  0.0.255.255
(deny any implícito - não visível na lista)
(access-list 1 deny 0.0.0.0  255.255.255.255)

interface ethernet 0
ip access-group 1 out
interface ethernet 1
ip access-group 1 out
```

Exemplo 2: ACL Padrão

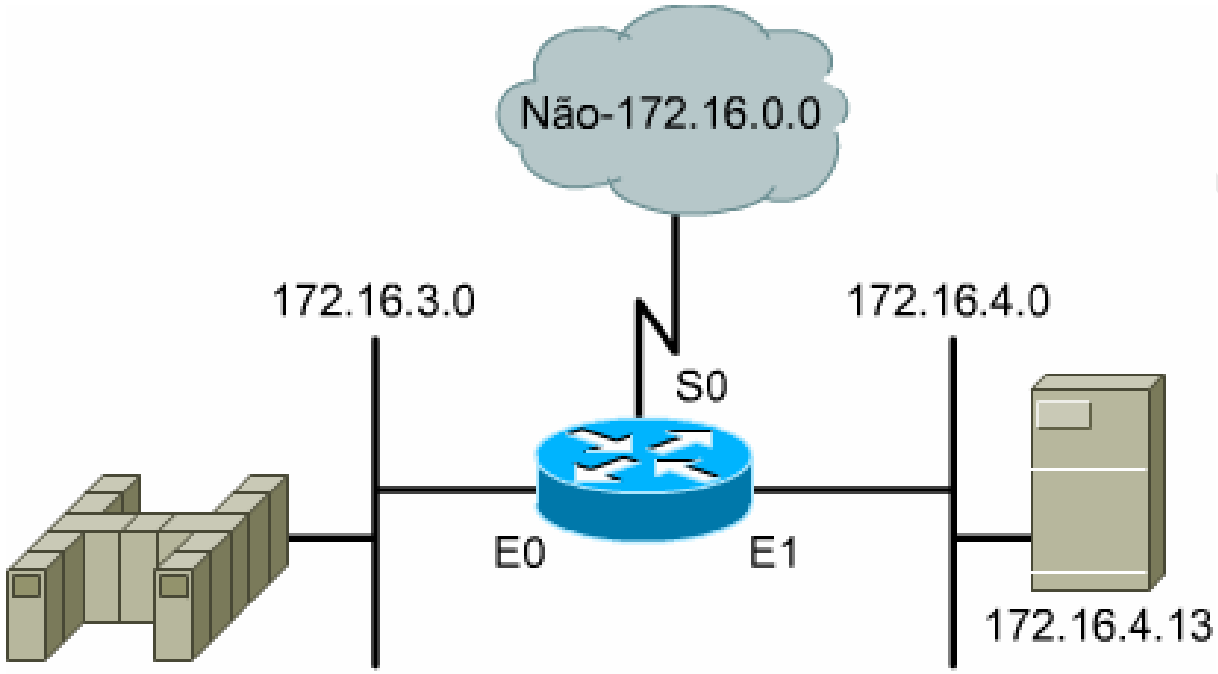


Saída do comando

```
access-list 1 deny 172.16.4.13  0.0.0.0
access-list 1 permit 0.0.0.0  255.255.255.255
(deny any implícito)
(access-list 1 deny 0.0.0.0  255.255.255.255)

interface ethernet 0
ip access-group 1 out
```

Exemplo 3: ACL Padrão



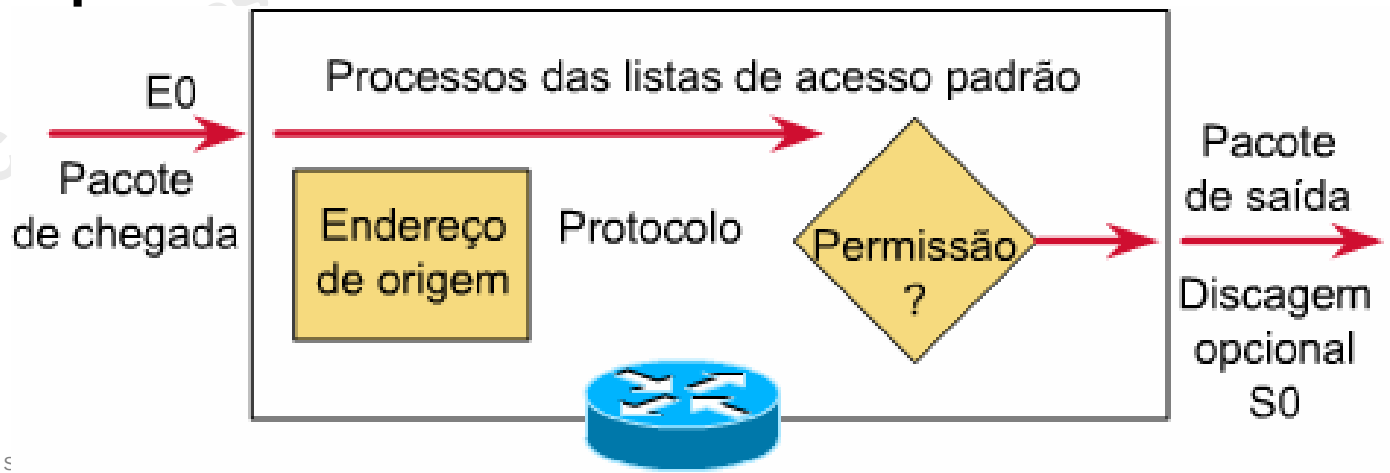
Saída do comando

```
access-list 1 deny 172.16.4.0  0.0.0.255
access-list 1 permit any
(deny any implícito)
access-list 1 deny any

interface ethernet 0
ip access-group 1 out
```

ACL Estendida

- **As ACLs estendidas são usadas mais freqüentemente para testar condições, porque proporcionam um intervalo maior de controle que as ACLs padrão**
- **Você usa uma ACL estendida quando deseja permitir tráfego da Web e negar o FTP (File Transfer Protocol) ou telnet de redes que não sejam da empresa.**
- **As ACLs estendidas verificam os endereços de origem e destino, protocolos específicos , números de portas e outros parâmetros. dos pacotes.**



OBS.: Normalmente são definidas próximo à origem

Criando uma ACL Estendida

- A sintaxe completa do comando é:

```
Router(config)# access-list número da ACL {permit |
    deny} protocolo End. de origem [máscara da origem]
    End. de destino [máscara do destino] operador
    [established]
```

- Operador
  - `it =` menor que
  - `gt =` maior que
  - `eq =` igual
  - `neq =` diferente

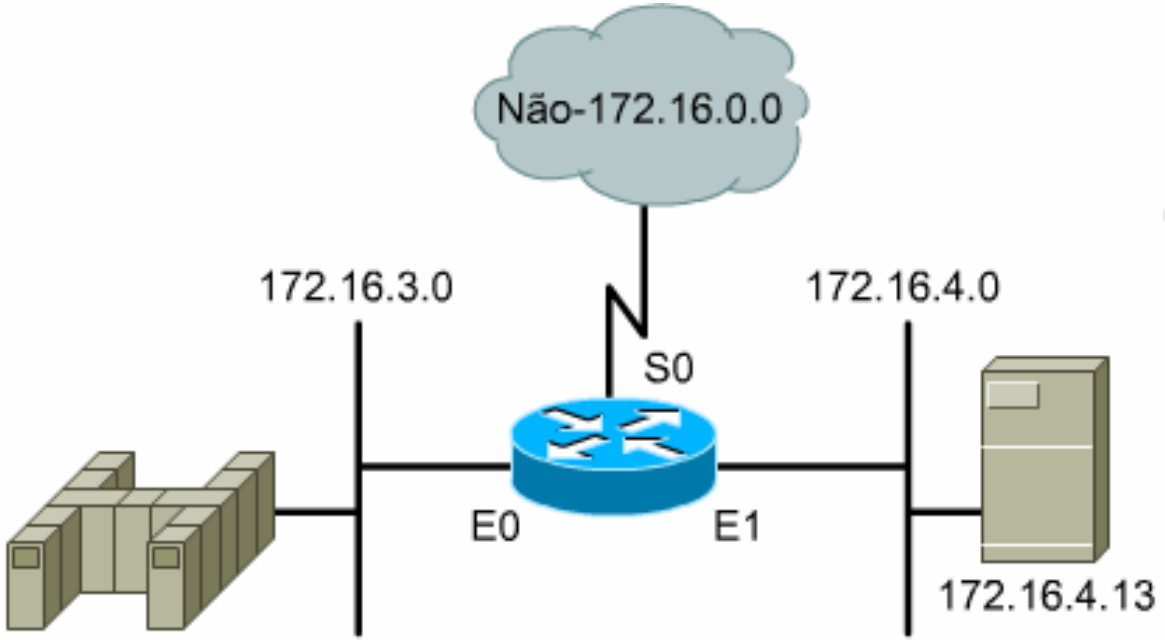
```
Router(config)# interface ethernet0
Router(config-if)# ip access-group 101 out
```

- Lembre-se de que somente uma ACL por interface, por direção, por protocolo é permitida.

Número de Portas

Decimal	Palavra-chave	Descrição	Protocolo
0		Reservado	
1-4		Não atribuído	
20	FTP-DADOS	FTP (dados)	TCP
21	FTP	FTP	TCP
23	TELNET	Conexão de terminal	TCP
25	SMTP	SMTP	TCP
42	NAMESERVER	Servidor do nome do host	UDP
53	DOMAIN	DNS	TCP/UDP
69	TFTP	TFTP	UDP
70		Gopher	TCP/IP
80	HTTP	WWW	TCP
133-159		Não atribuído	
160-223		Reservado	
162		FNP	UDP
224-241		Não atribuído	
242-251		Não atribuído	

Exemplo 1: ACL Estendida

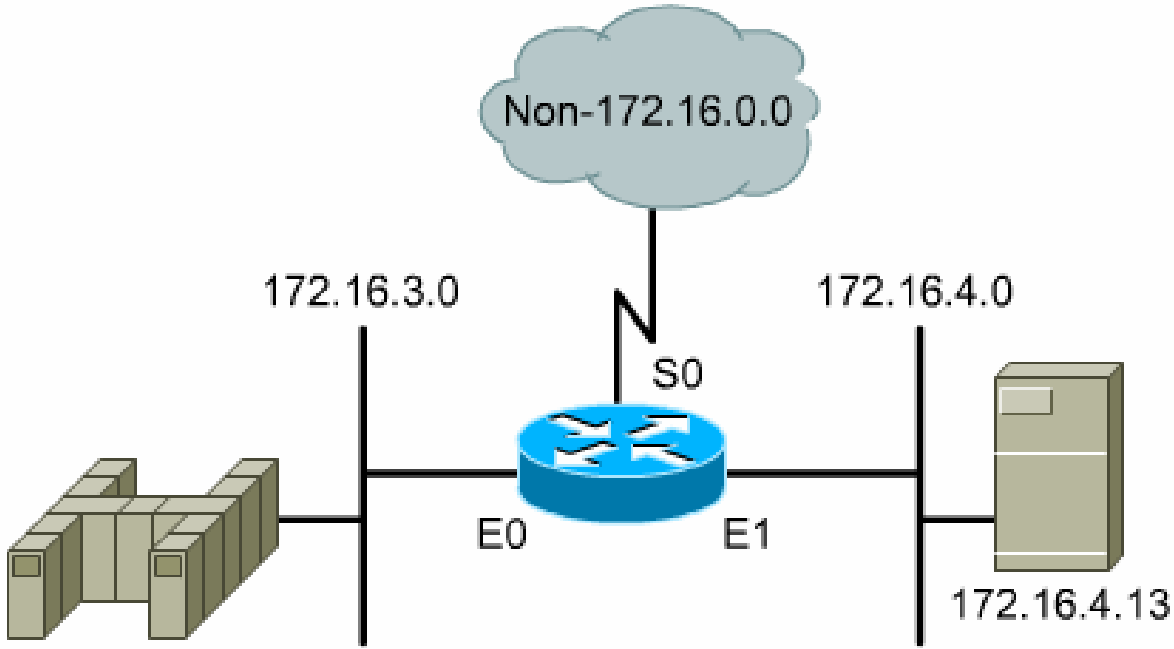


Saída do comando

```
access-list 101 deny tcp 172.16.4.0
    0.0.0.255 172.16.3.0  0.0.0.255  eq 21
access-list 101 permit ip 172.16.4.0
    0.0.0.255 0.0.0.0 255.255.255.255
(deny any implícito)
(access-list 101 deny ip 0.0.0.0
    255.255.255.255  0.0.0.0  255.255.255.255)

interface ethernet 0
ip access-group 101
```

Exemplo 2: ACL Estendida



Saída do comando

```
access-list 101 deny tcp 172.16.4.0  0.0.0.255 any eq 23
access-list 101 permit ip any any
(implicit deny any)
(access-list 101 deny ip 0.0.0.0  255.255.255.255
    0.0.0.0  255.255.255.255)

interface ethernet 0
ip access-group 101 out
```

ACLs com nomes

- **A sintaxe completa do comando é:**

Router(config)# {std- | ext-} *name da ACL* deny {[protocolo] *End. origem* [máscara-origem] *End. Destino* [máscara-destino] *operador* [established] | any}

- **Considere o seguinte antes de implementar as ACLs com nomes:**

➢ **As ACLs com nomes não são compatíveis com as versões Cisco IOS anteriores à versão 11.2.**

➢ **Você não pode usar o mesmo nome para várias ACLs. Além disso, ACLs de diferentes tipos não podem ter o mesmo nome.**

ACLs com nomes

Cisco.com

```
ip interface ethernet0/5
ip address 2.0.5.1.255.255.255.0
ip access-group Internetfilter out
ip access-group marketinggroup in
...
ip access-list standard Internetfilter
permit 1.2.3.4

deny any
ip access-list extended marketing_group
permit tcp any 171.69.0.0.0.255.255.255 eq telnet

deny tcp any any
deny udp any 171.69.0.0.0.255.255.255 lt 1024

deny ip any log
```

- **Quando você desejar identificar intuitivamente as ACLs usando um nome alfanumérico.**
- **Você tem mais de 99 ACLs simples e 100 estendidas para serem configuradas em um roteador para um determinado protocolo.**

O comando deny


```
ip access-list standard Internetfilter
deny 192.5.34.0.0.0.0.255
permit 128.88.0.0.0.0.255.255
permit 36.0.0.0.0.255.255.255
! (Observação: todos os demais acessos implicitamente recusados)
```

- **Sintaxe completa para este comando é:**

  `deny {origem [curinga da origem] | any}`

- **Use a forma `no` desse comando para remover uma condição de negação, usando a sintaxe a seguir:**

  `no deny {`*`origem`* `[curinga da origem] | any}`

O comando `permit`


```
ip access-list extended come-on
permit tcp any 171.69.0.0.0.255.255.255 eq telnet

deny tcp any any
deny udp any 171.69.0.0.0.255.255.255 lt 1024

deny ip any any
interface ethernet0/5
ip address 2.0.5.1 255.255.255.0
ip access-group over_out out
ip access-group come_on in
ip access-list standard over_and
permits 1.2.3.4

deny any
```

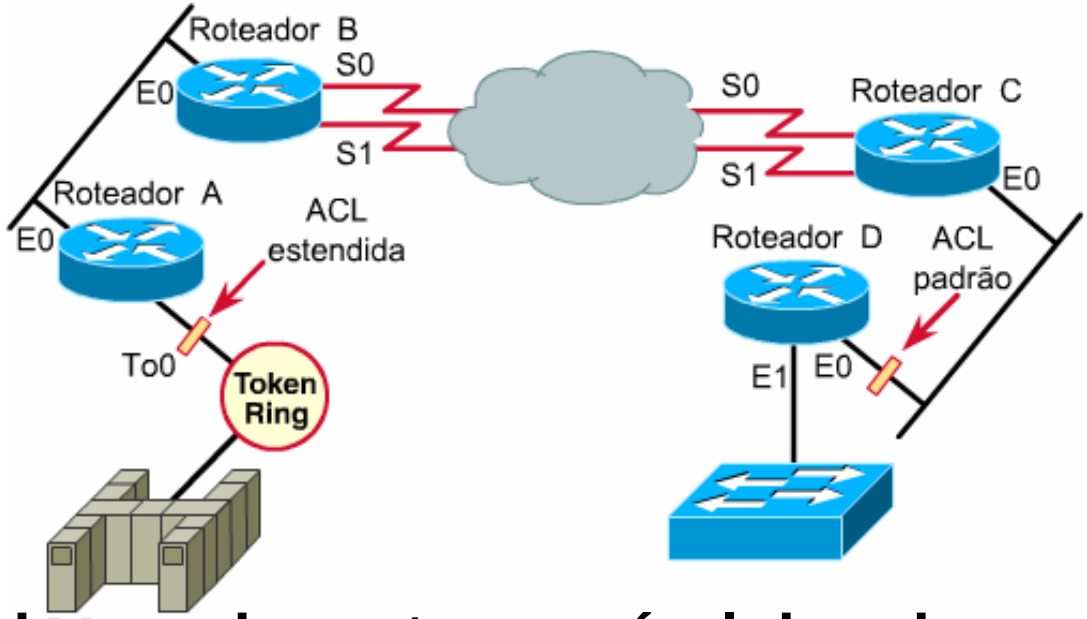
- **Sintaxe completa para este comando é:**

  `permit {origem [curinga da origem] | any}[log]`

- **Use a forma `no` desse comando para remover uma condição de negação, usando a sintaxe a seguir:**

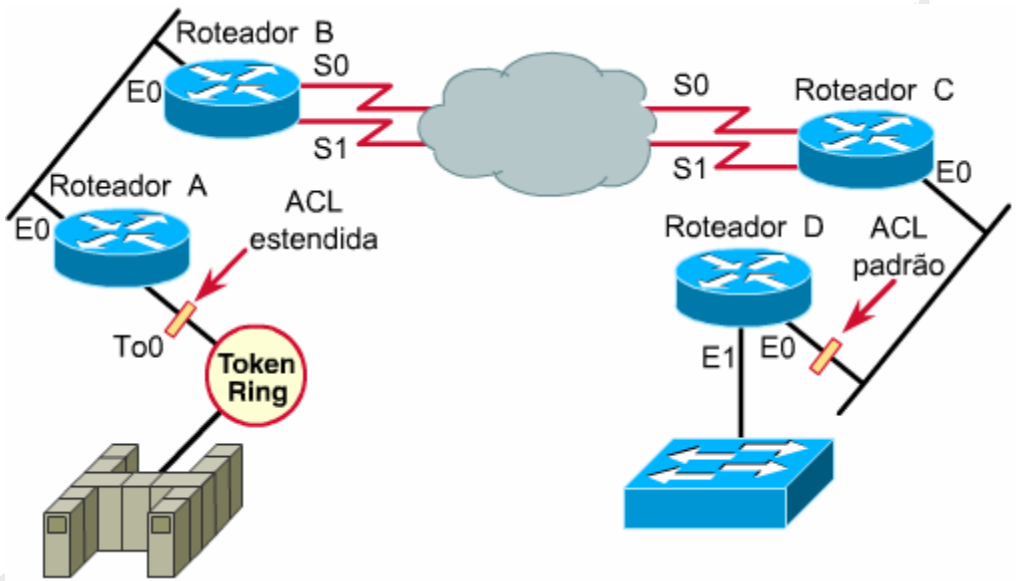
  `no permit {origem [curinga da origem] | any}[log]`

Posicionando ACLs



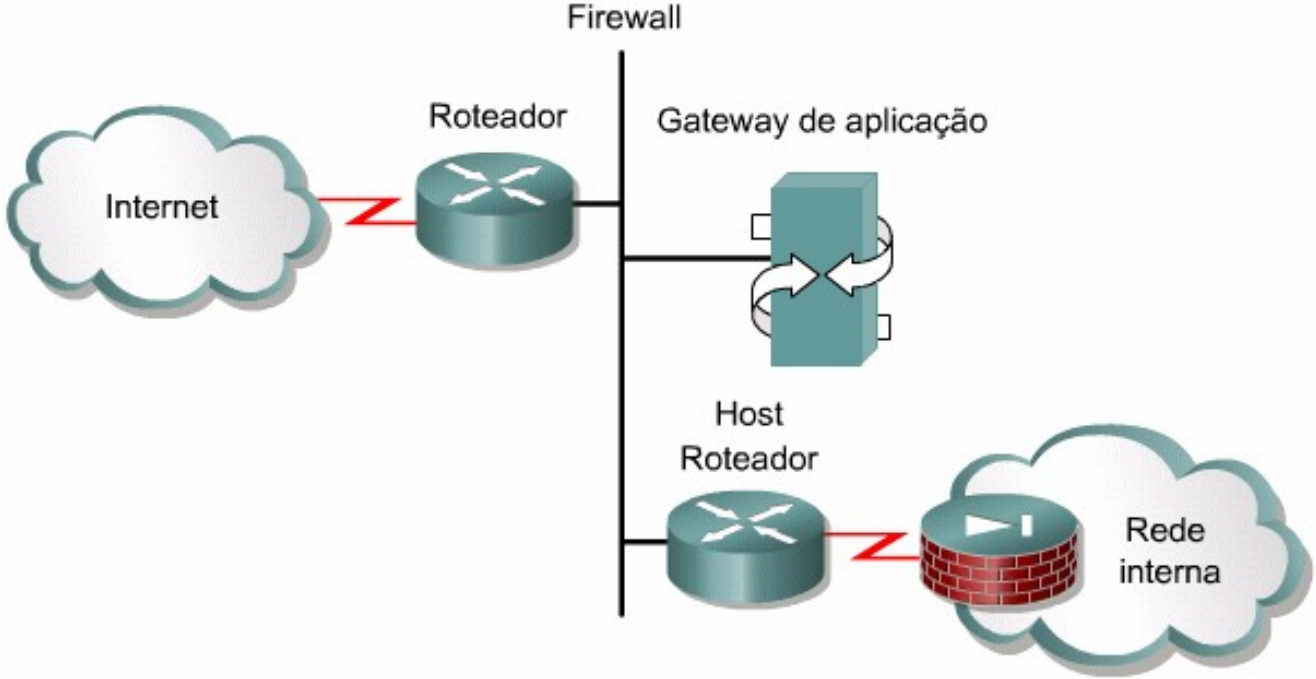
- **Coloque a ACL estendida o mais perto possível da origem do tráfego negado (a filtragem das ACLs estendidas usam os endereços de origem e/ou destino).**
- **No caso das ACLs padrão, elas só podem filtrar usando o endereço de origem (não os endereços de destino), por isso, elas devem ser colocadas o mais perto possível do destino.**

Usando ACLs em Roteadores Firewall



- **As ACLs devem ser usadas em roteadores de firewall, que são freqüentemente posicionados entre a rede interna e a rede externa, como a Internet.**
- **O roteador de firewall fornece um ponto de isolamento para que o resto da estrutura interna da rede não seja afetado.**

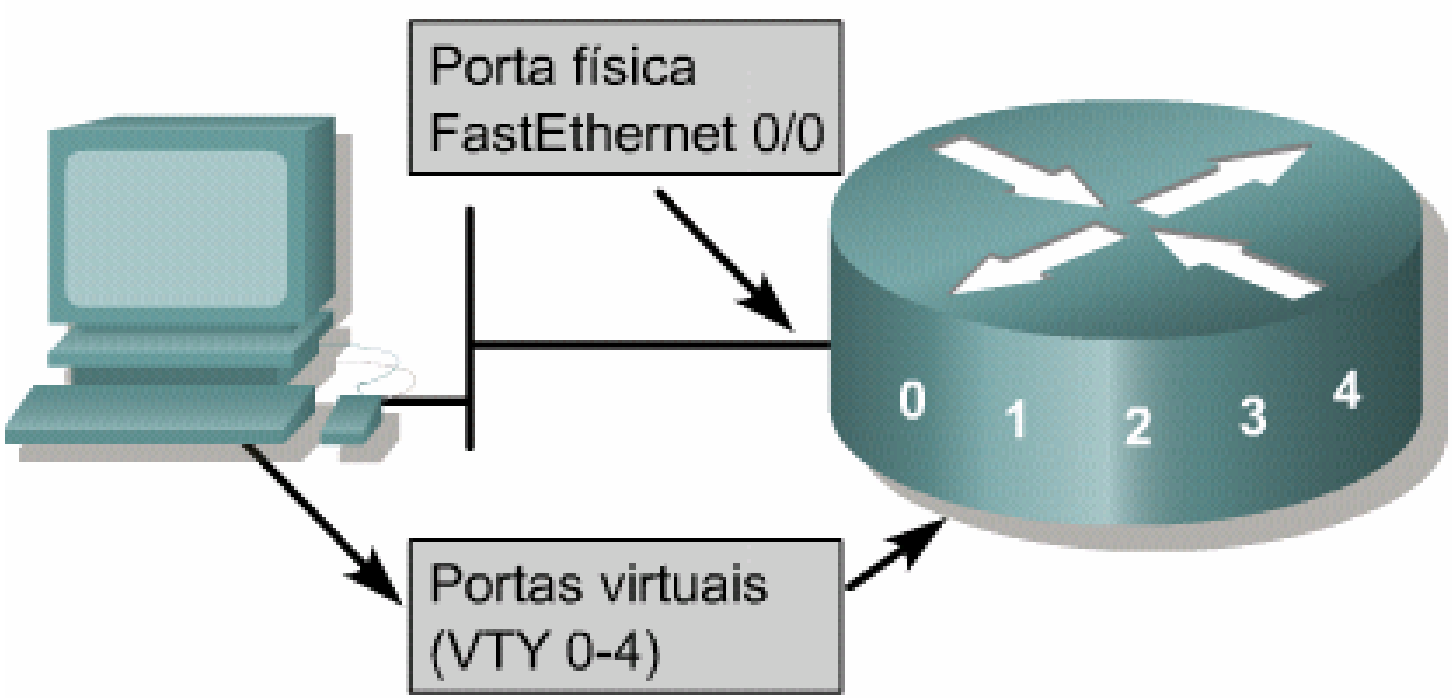
Arquitetura de Firewall



- **O roteador que é conectado à Internet (ou seja, o roteador externo) força todo o tráfego que chega a ir para o gateway do aplicativo. O roteador que é conectado à rede interna (ou seja, o roteador interno) aceita pacotes somente do gateway do aplicativo.**

Restringindo o acesso do terminal virtual

- **Lista de Acesso estendida para Telnet de saída não impede sessões Telnet iniciadas pelo roteador, por padrão**



Criando e Aplicando uma ACL na VTY

Cisco - Hyperterminal

```
Creating the standard list:

Rtl(config)#access-list 2 permit 172.16.1.0 0.0.0.255
Rtl(config)#access-list 2 permit 172.16.2.0 0.0.0.255
Rtl(config)#access-list 2 deny any

Applying the access list:

Rtl(config)#line vty 0 4
Rtl(config-line)#login
Rtl(config-line)#password secret
Rtl(config-line)#access-class 2 in
```

Verificando ACLs


```
Router(config)# show ip interface
Router(config)# show access-lists
```

Resumo

- As ACLs executam várias funções dentro de um roteador, incluindo procedimentos de segurança/acesso.
- As ACLs são usadas para controlar e gerenciar o tráfego.
- Em alguns protocolos, é possível aplicar até duas ACLs a uma interface: uma ACL de entrada e uma ACL de saída.
- Depois que um pacote coincide com uma instrução da ACL, ele pode ter seu acesso ao roteador negado ou permitido.
- Os bits da máscara curinga usam o número um (1) e o número zero (0) para identificar como lidar com os bits correspondentes do endereço IP.
- A criação e a aplicação das listas de acesso são verificadas por meio do uso de vários comandos show do IOS.

Resumo

- Os dois tipos principais de ACLs são: padrão e estendida.
- As ACLs com nome permitem a utilização de um nome para identificar a lista de acesso, em vez de um número.
- É possível configurar ACLs para todos os protocolos de rede roteados.
- As ACLs devem ser posicionadas onde permitirem o controle mais eficiente.
- Geralmente, as ACLs são usadas em roteadores de firewall.
- As listas de acesso também podem restringir o acesso via terminal virtual ao roteador.


CISCO SYSTEMS
EMPOWERING THE
INTERNET GENERATION℠