ACL – Access Control Lists

CCNA Exploration Semester 4 - Chapter 5

Dnes

- Zabezpečenie sietí cez ACL
- Konfigurácia štandardných ACL
- Konfigurácia rozšírených ACL
- Implementácia a overenie funkčnosti ACL

Access Control Lists

- Cisco ACL
  - Triediace a kontrolné zoznamy
  - Najznámejšie nasadenie ako pravidlá riadenia IP prevádzky (FW)
    - Paketový filter
  - Použité aj všade tam kde je potrebná nejaká klasifikácia alebo identifikácia toku, napr. NAT, QoS klasifikácia, filtrovanie výpisov a pod.
  - Logovanie

Úlohy ACL

- Obmedzenie nechcenej prevádzky
  - Filter na nejaký obsah, napr. video
- Riadenie prevádzky
  - Povolenie určitého typu prevádzky, služby a zakázanie iného
    - Povoľ SMTP a zakáž telnet
- Riadenie IP toku
  - Napr. príjem a zasielanie updates, riadenie smerovania
- Poskytnutie základnej bezpečnosti
  - Riadenie kto môže kam pristupovať

ACL paketový filter

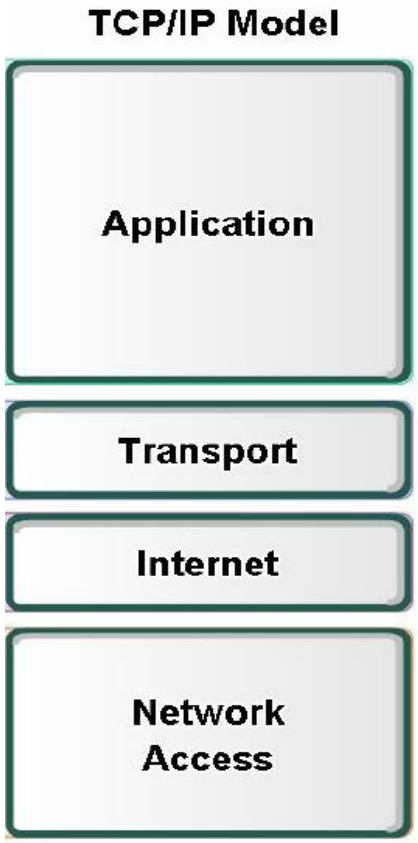
- Zoznam testovacích podmienok, ktoré sa aplikujú na IP prevádzku prechádzajúcu rozhraniami smerovača
- Podmienky určujú
  - Povoľ (**Permit**) danú prevádzku ak spĺňa podmienku
  - Zakáž (**Deny**) danú prevádzku ak spĺňa podmienku
- Defaultne smerovače nemajú implementované ACL filtre

Kontrola prístupu v IP sieťach – riešenie podmienok do ACL

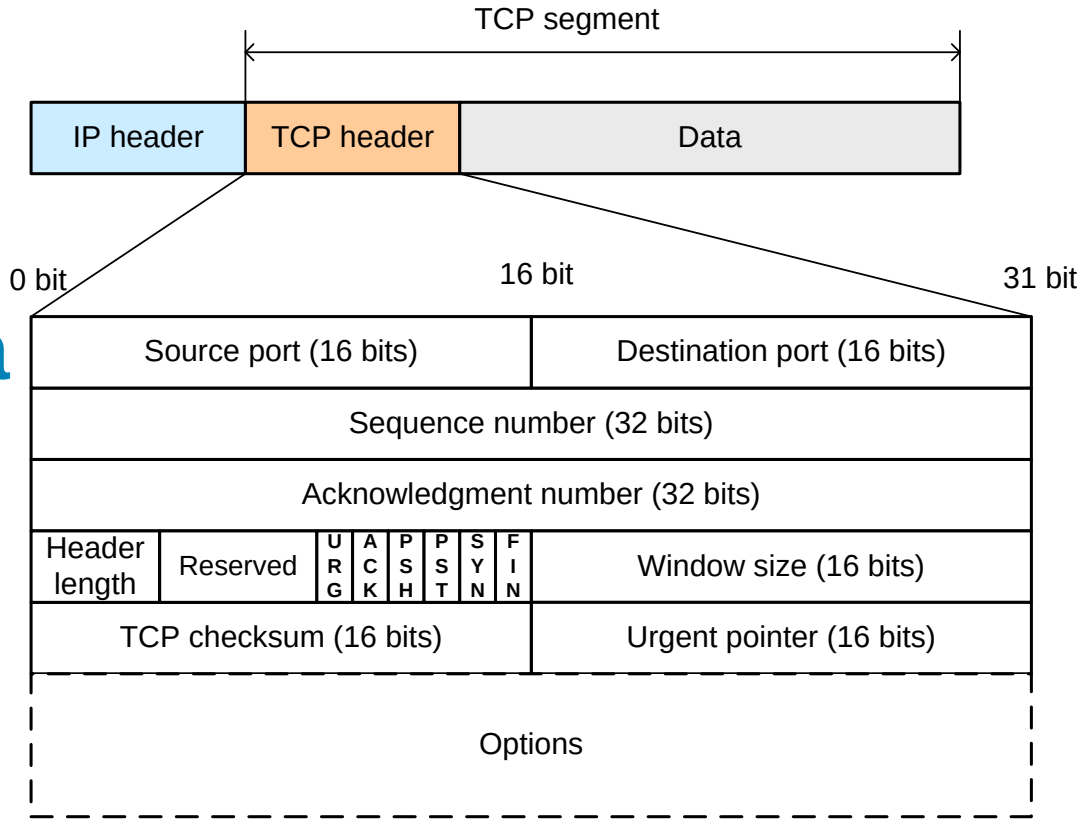
- Aké máme možnosti na riešenie riadenia prístupu v IP sieťach?
  - Rozlíšenie smeru toku dát
    - Odkiaľ (Zdroj/Source/Sender)
    - Kam (Cieľ/Destination/Receiver)
  - Kto
    - Skupina alebo jednotlivo (odosielateľ/-telia – príjemca/-ovia)
      - Ako rozlíšiť? (Maska)
  - Parametre
    - IP adresa (S, R)
    - Typ protokolu (IP, ICMP, TCP, UDP)
    - Služba (Číslo portu (S, R))
      - TCP, UDP

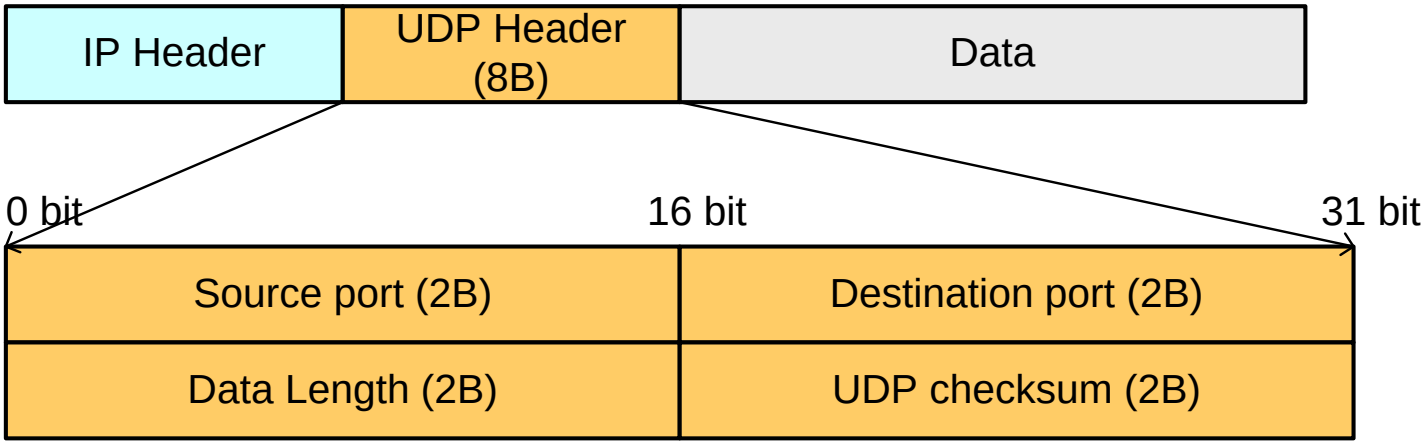
Pri ACL ber do úvahy vlastnosti TCP/IP architektúry

- TCP/IP model je vrstvový!!

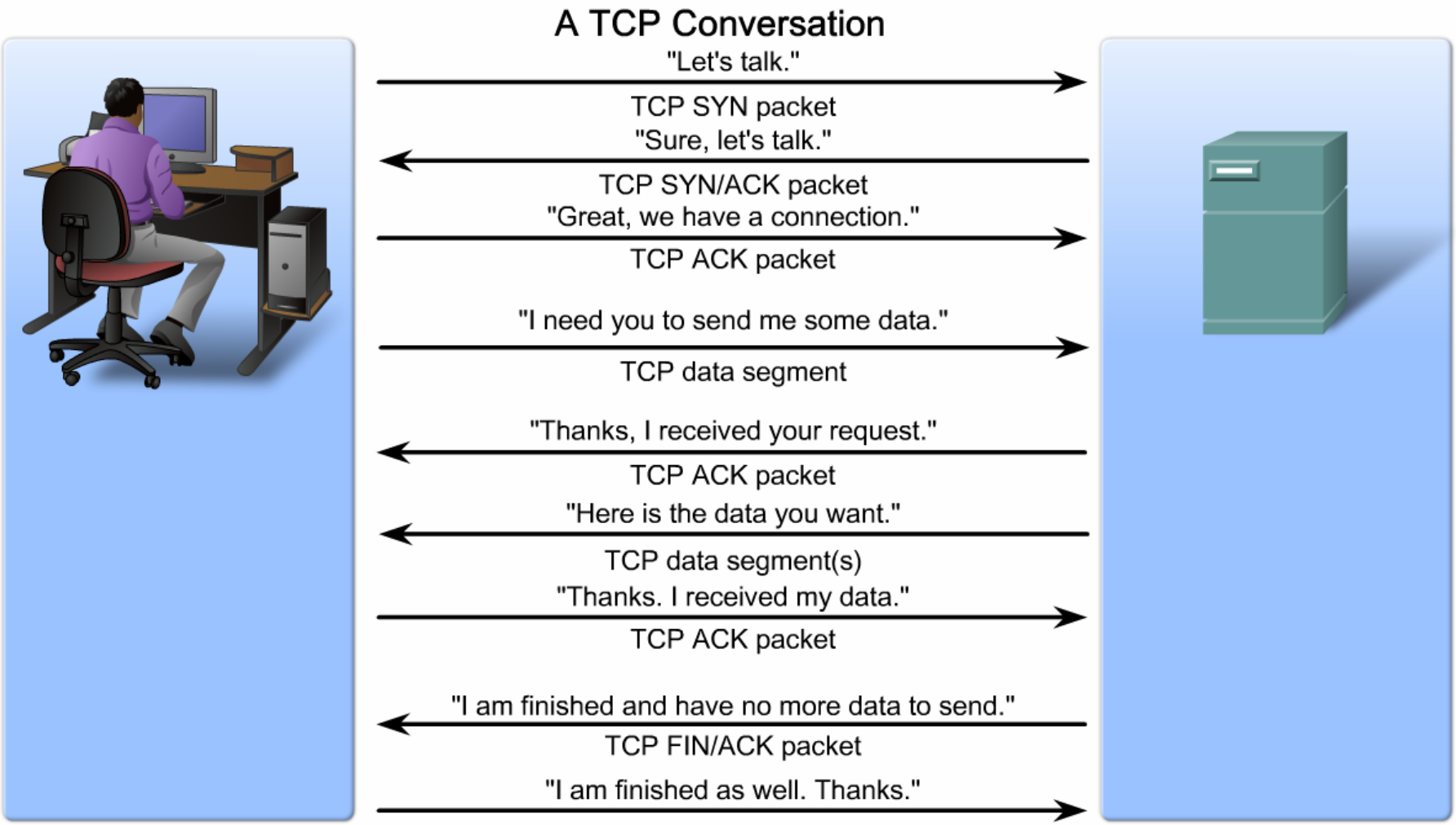


Formát TCP a UDP segmentu/datagramu

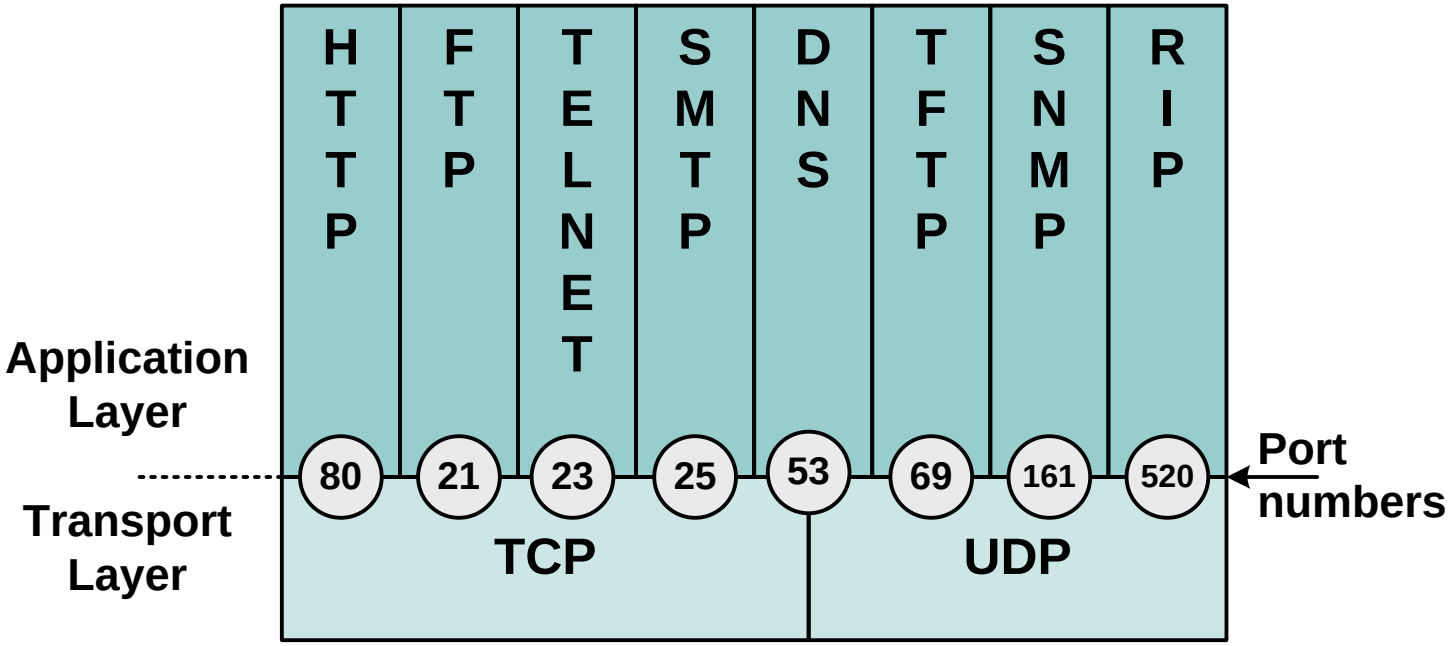




Pri ACL brať do úvahy vlastnosti TCP: TWH, ACK a ukončenie

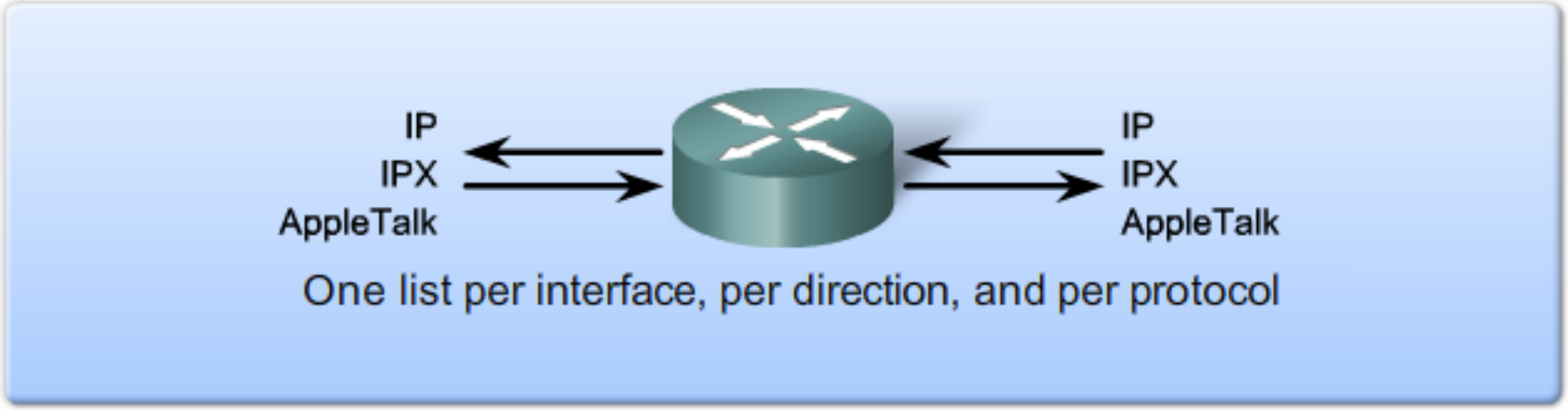


Čísla portov (rozlíšenie apl. toku)



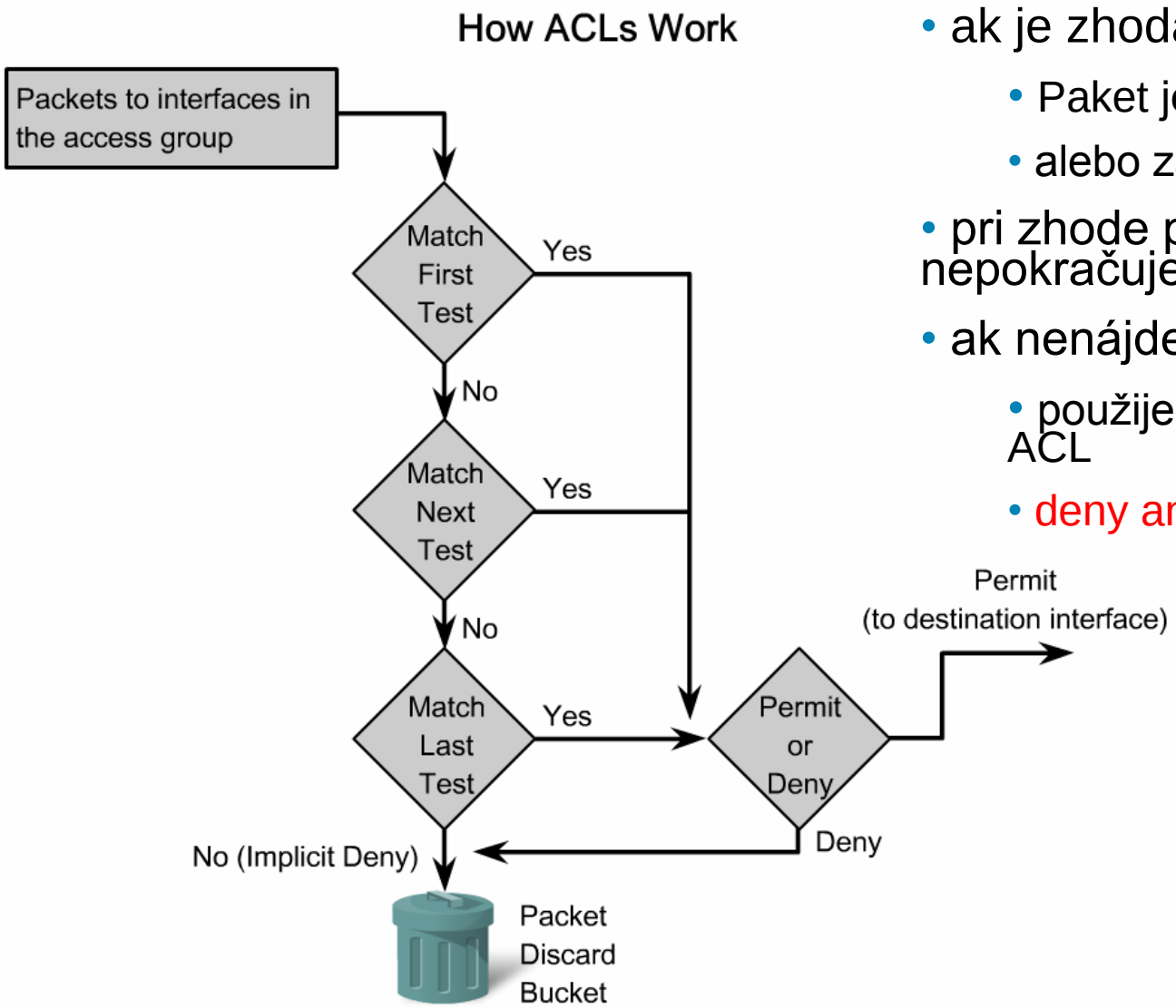
- FTP: 20 (data), 21
- Telnet: 23
- SMTP: 25
- WINS replication: 42
- DNS: 53 (UDP)
- BOOTP, DHCP: 67 (server), 68 (klient)
- TFTP: 69
- HTTP: 80
- Kerberos: 88 (UDP i TCP)
- POP3: 110
- NNTP: 119
- NTP: 123
- RPC Locator: 135 (TCP, UDP)
- IMAPv2: 143
- SNMP: 161
- IMAPv3: 220
- HTTPS: 443

Nasadenie ACL



- Jeden ACL per protokol
  - ACL je definovaný pre každý podporovaný protokol zvlášť
- Jeden ACL per smer
  - ACL riadi tok iba v jednom smere, nie v oboch
  - Komplikovanejšie riešenia ACL vyžadujú implementáciu ACL na viac rozhraniach (filter In a OUT smer)
- Jeden ACL per interface

Ako ACL pracuje

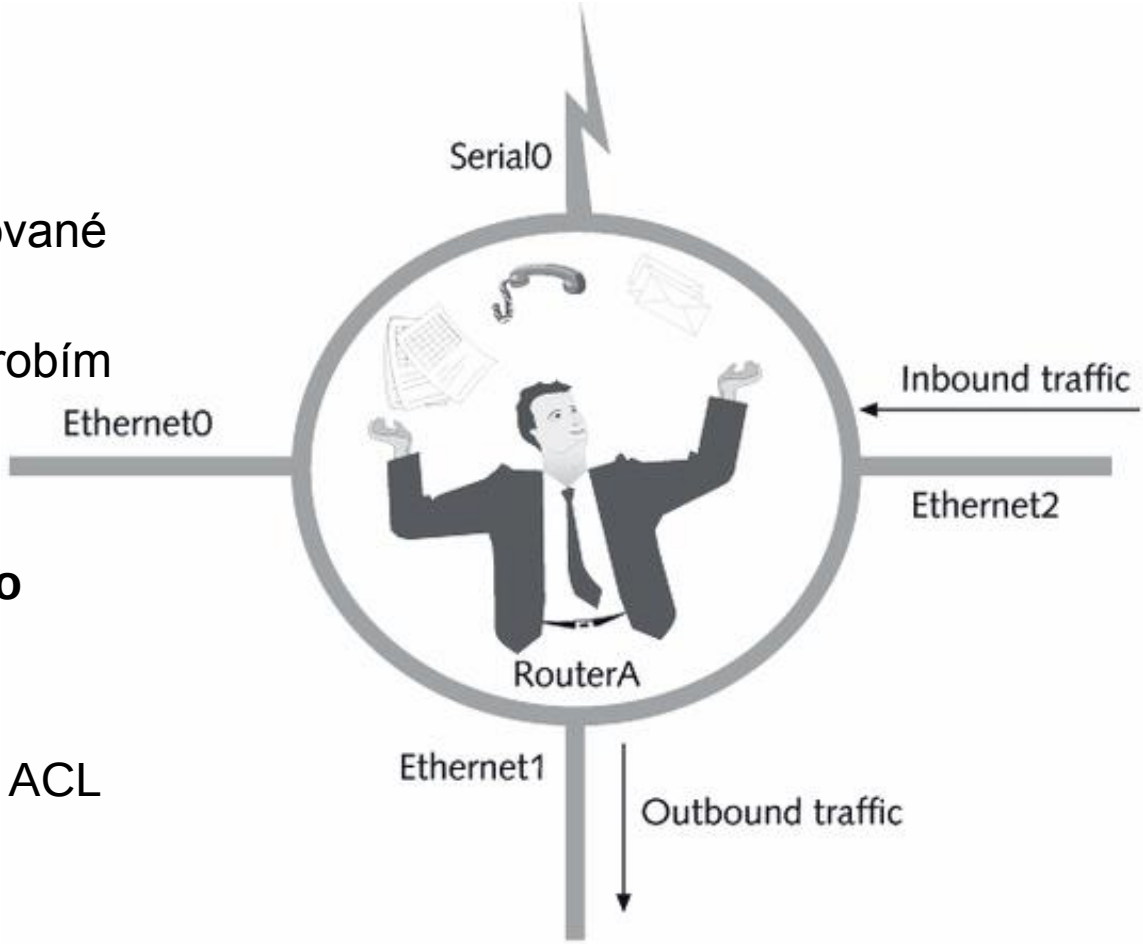


ACL je zoznam podmienok

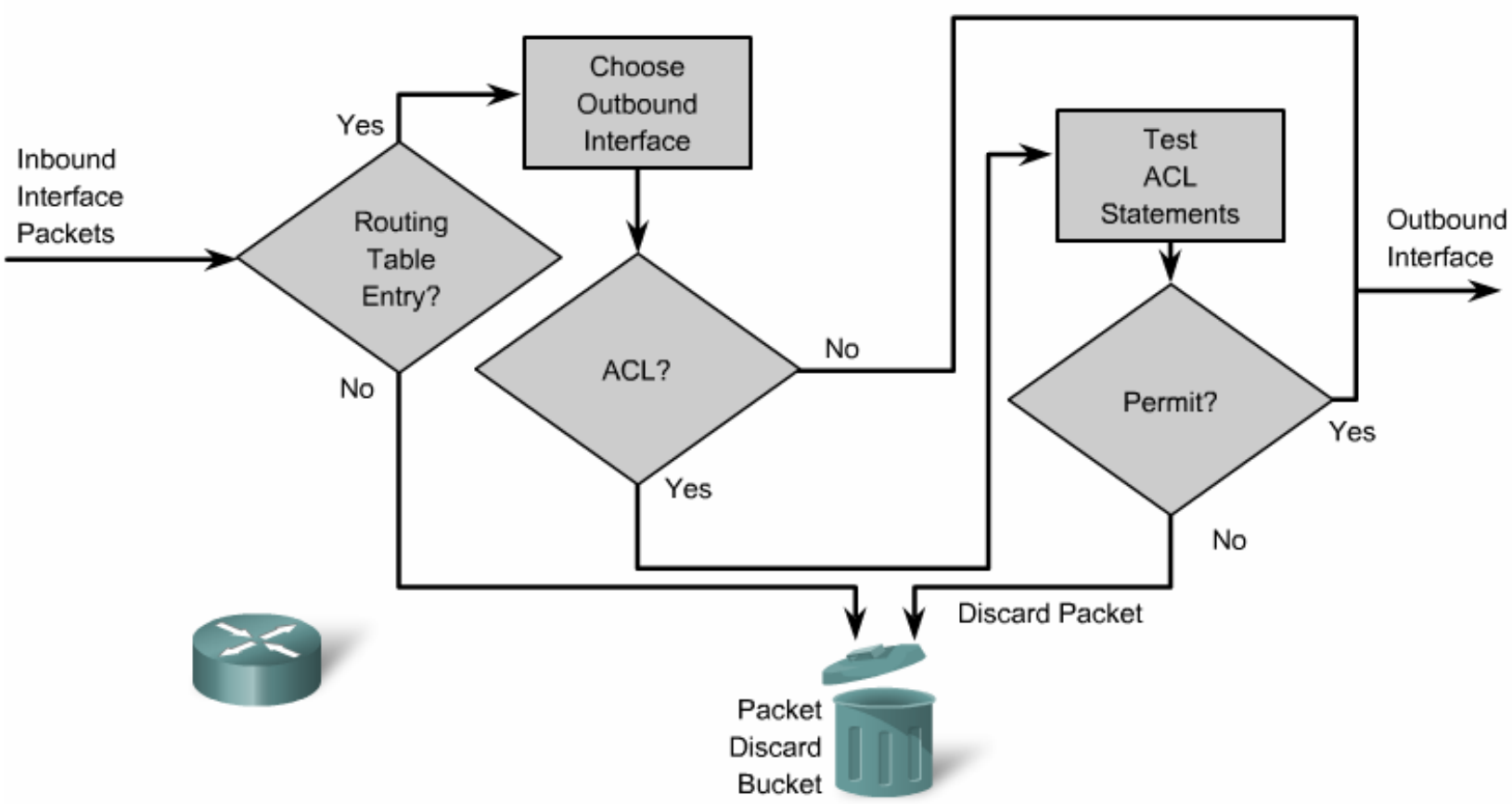
- prehľadávaný sekvenčne
- ak je zhoda na podmienku
  - Paket je povolený (permit)
  - alebo zahodený (deny)
- pri zhode podmienky už ďalej nepokračujem
- ak nenájdem ani jednu podmienku
  - použijem default akciu na konci ACL
  - deny any

Nasadenie ACL

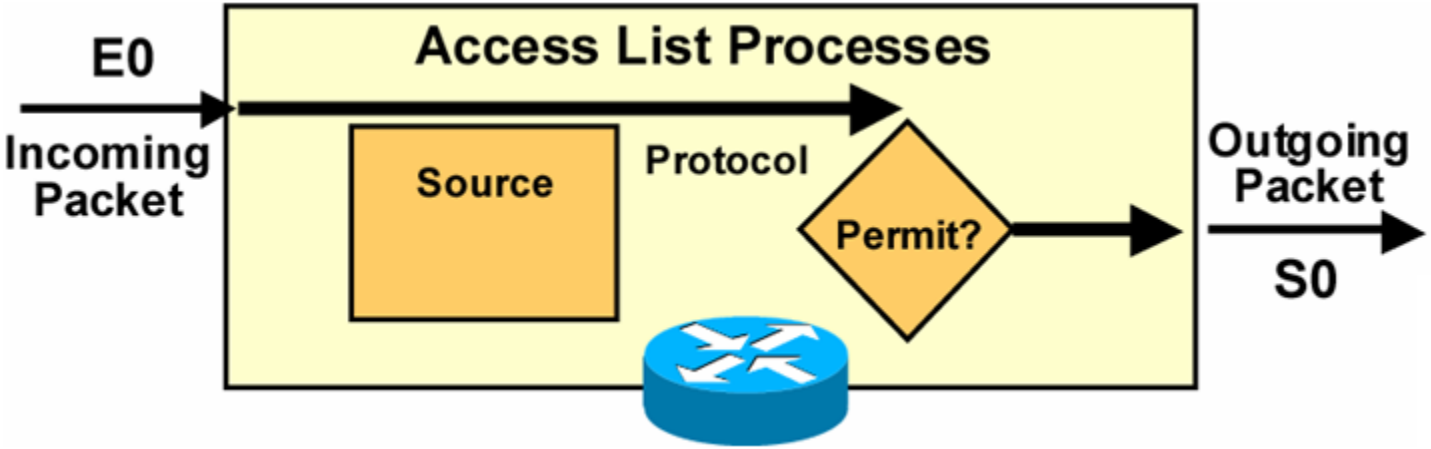
- **Inbound ACLs**
  - Smer paketov je **do (in) smerovača**
    - Vstupujú cez rozhranie
  - Vstupujúce pakety sú spracované skôr ako sú smerované
  - šetrím výkon smerovača, nerobím routing pre discard pakety
- **Outbound ACLs**
  - Smer paketov je **von (out) zo smerovača**
    - Vystupujú cez rozhranie
  - Skôr ako je paket postupený ACL je vykonané smerovanie



Outbound ACL činnosť

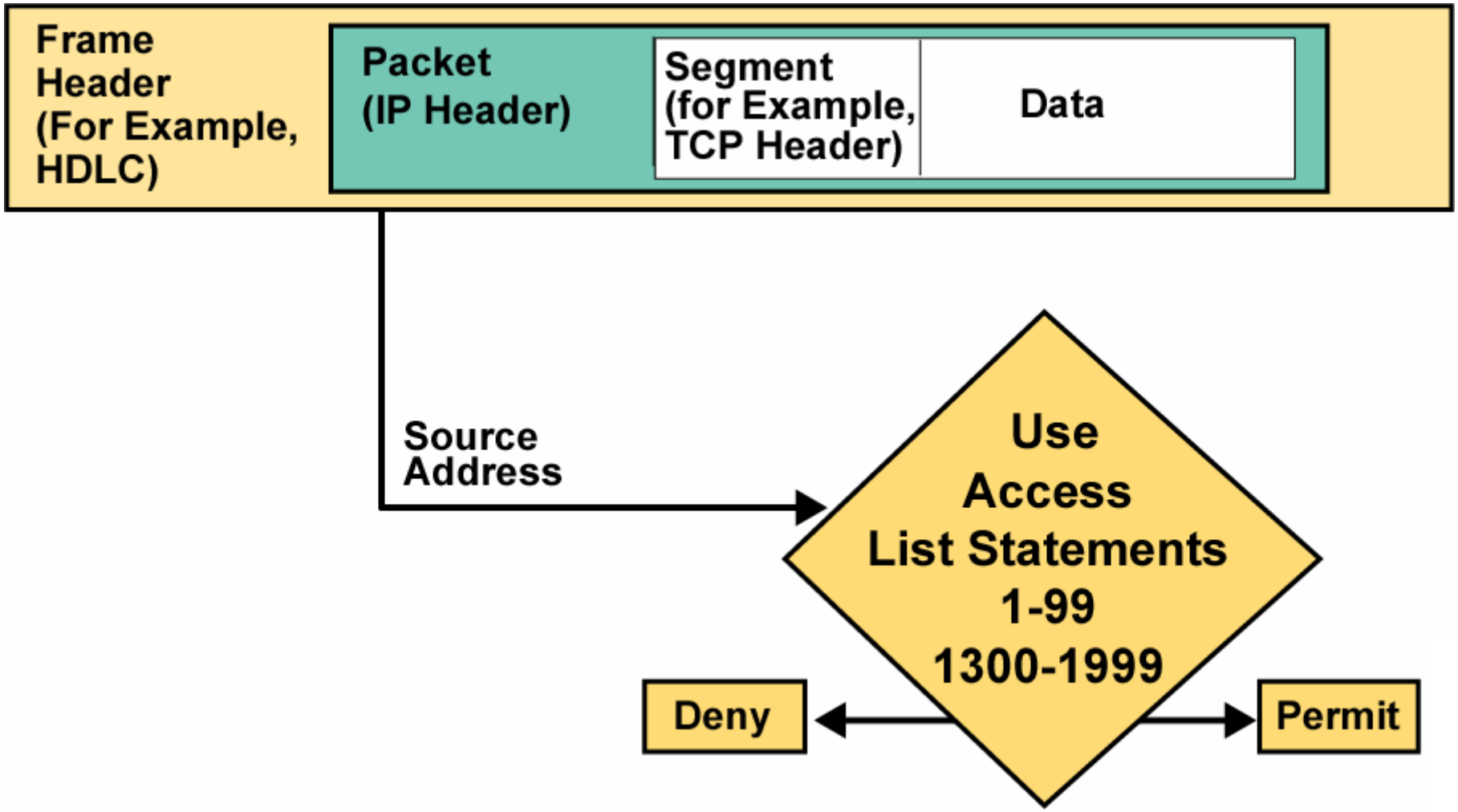


Typy ACL

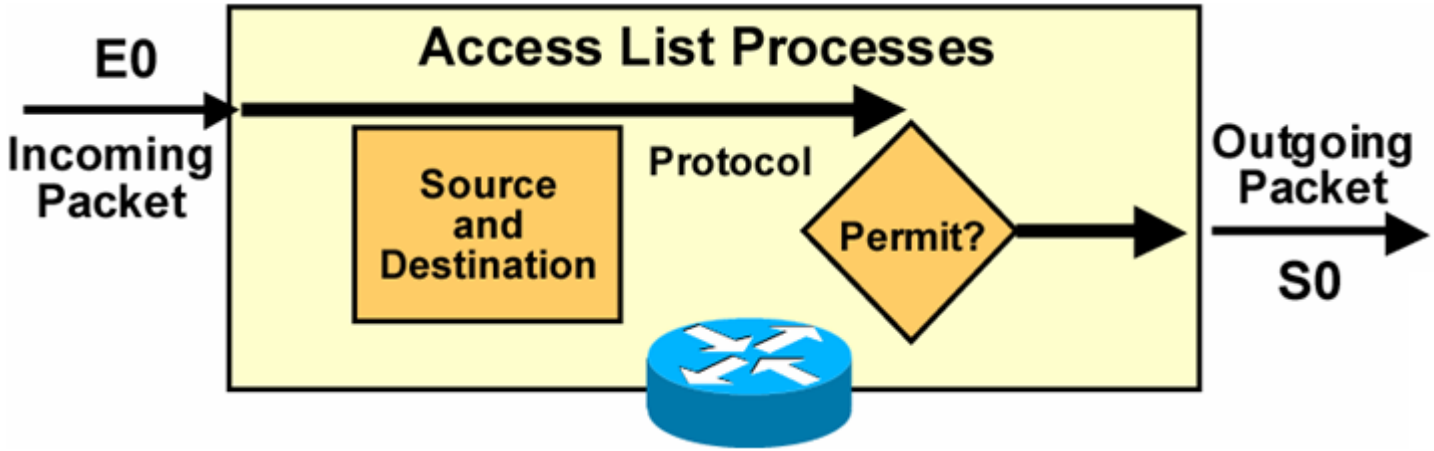


- **Štandardné (standard) ACL**
  - Všeobecne povoľujem a zakazujem celý protokolový stack
    - Napr. celé IP a pod.
  - Na podmienku sa kontroluje len zdrojová adresa

Standard ACL

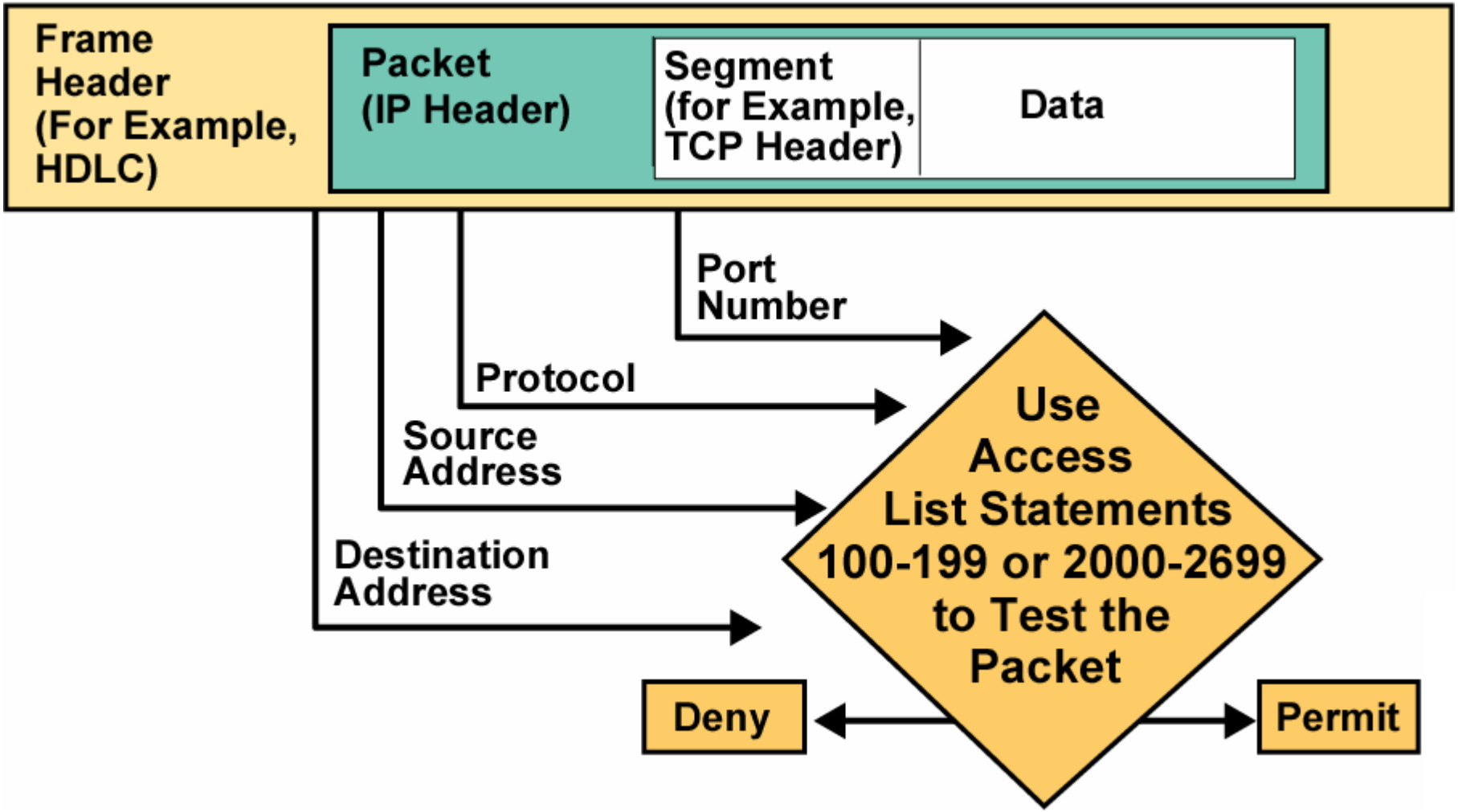


Typy ACL



- **Rozšírené (extended) ACL**
  - Voči podmienke sa kontroluje zdrojová aj cieľová adresa
  - Zdrojový a cieľový port
  - Povoľujem a zakazujem konkrétny protokol alebo komunikáciu definovanú portom

Extended ACL



Číslované a pomenované ACL

- Číslovanie a pomenovanie
  - Za účelom zjednodušenia identifikácie ACL
- **Standard IP ACL**
  - 1 – 99
  - 1300 - 1999
- **Extended IP ACL**
  - 100 – 199
  - 2000 – 2699
- Pri standard a extended ACL neviem mazať podmienky, pridávať viem len na koniec zoznamu podmienok
- **Pomenované ACL**
  - Acl je identifikované menom, alfanumerickým
  - Meno nesmie obsahovať medzery
  - Výhoda:
    - **Môžem mazať a pridávať podmienky**

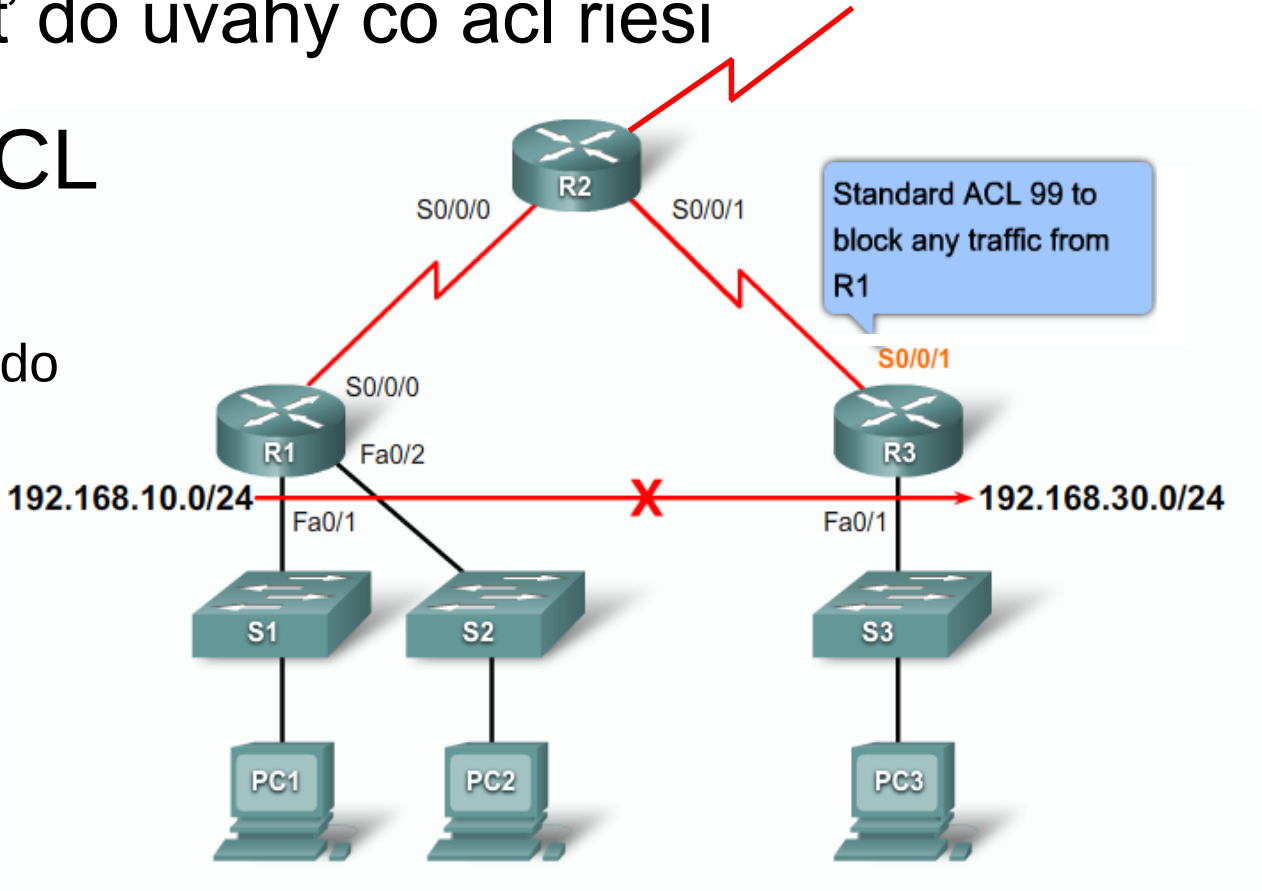
Umiestnenie ACL – standard ACL

- Tak aby mal najlepší dopad na funkčnosť
  - Treba brať do úvahy čo acl rieši
- Standard ACL

napr. blok všetko

zo 192.168.10.0/24 do

192.168.30.0/24



Umiestnenie ACL – extended ACL

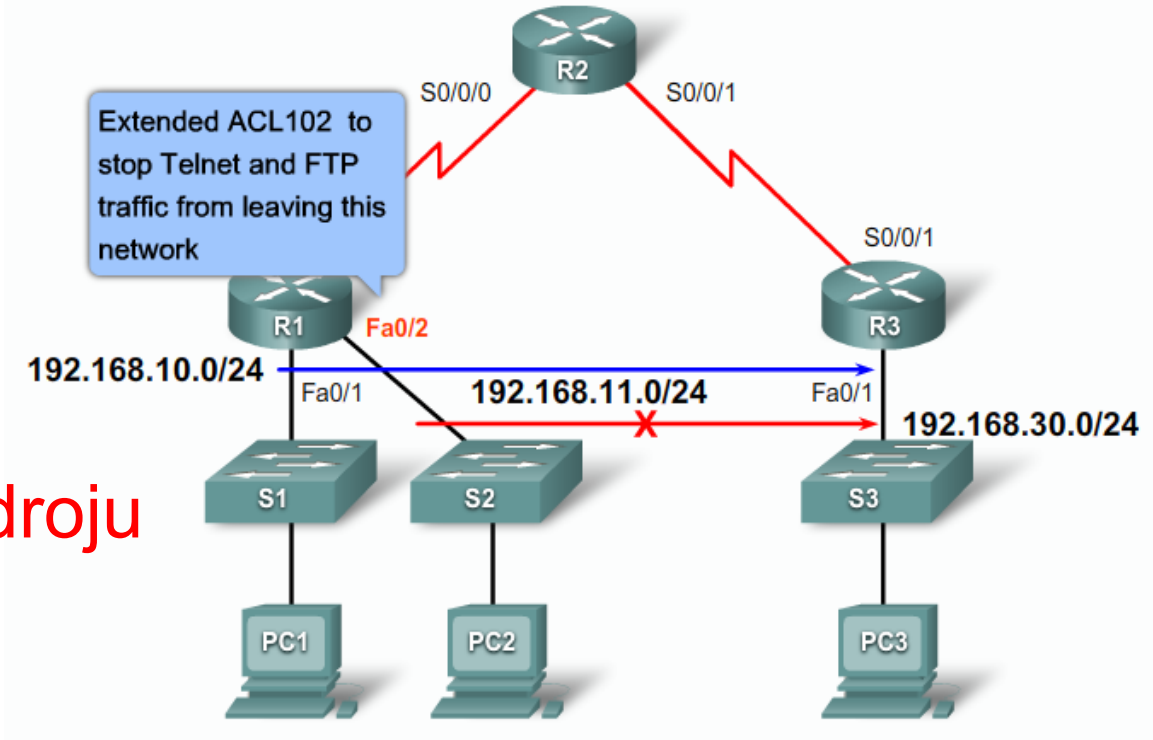
- Tak aby mal najlepší dopad na funkčnosť
  - Treba brať do úvahy čo acl rieši
- Extended ACL

napr. blok telnet a ftp

z 192.168.11.0/24 do

192.168.30.0/24

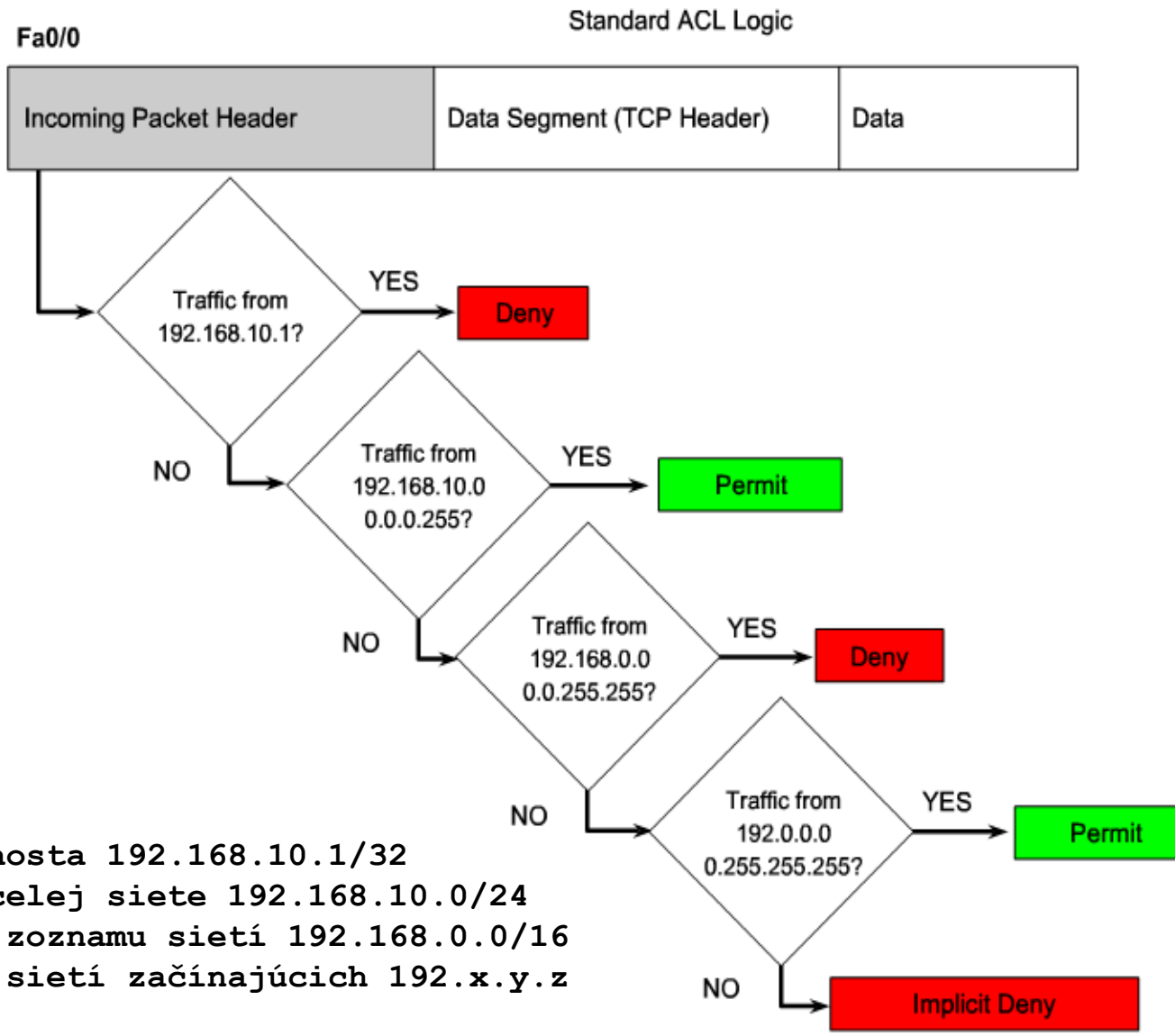
- Čo najbližšie k zdroju



Pri ACL ber do úvahy

- Smerovač aplikuje podmienky v poradí ako sú zadané (napísané)
  - Podmienky ACL sú aplikované sekvenčne
- Pakety sú porovnávané voči podmienkam až kým nenastane zhoda,
  - Zvyšok ACL sa už nekontroluje (first match)
- ACL zoznam defaultne vždy končí s implicitným **deny any**
  - Aj keď táto podmienka nemusí byť viditeľná priamo

Idea ACL




```
Podm.1: zakáž prevádzku z hosta 192.168.10.1/32
Podm.2: povoľ prevádzku z celej siete 192.168.10.0/24
Podm.3: zakáž prevádzku zo zoznamu sietí 192.168.0.0/16
Podm.4: povoľ prevádzku zo sietí začínajúcich 192.x.y.z
```

Pri ACL ber do úvahy

- ACL musí byť implementované na rozhranie aby nabralo na funkčnosti
  - V smere **In** (inbound) alebo **Out** (outbound)
- Na rozhranie môžem nasadiť len jeden acl per protokol a per smer
- Standard ACL
  - „Najbližšie k cieľu“
- Extended ACL
  - „Najbližšie k zdroju“

Odporúčania pre tvorbu ACL

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.

Štandardné ACL

Konfigurácia standard ACL

```
Router(config)# access-list ACCESS-LIST-# [deny | permit
| remark] TEST_PODMIENKA [WILDCARD] [log]
```

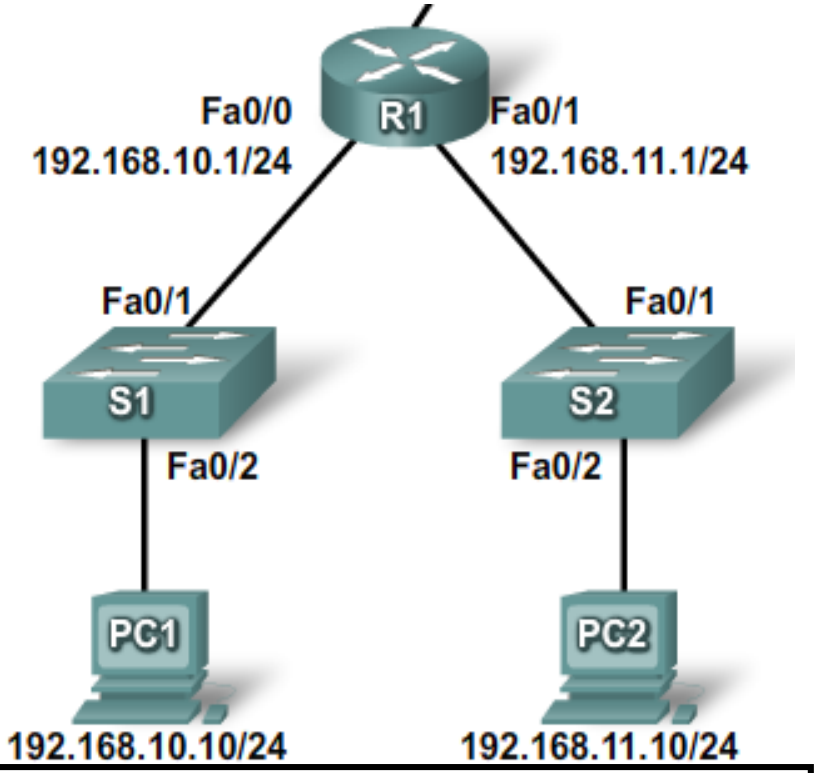
- ***ACCESS-LIST-#:*** Číslo acl
  - ACL môže mať veľa podmienok, ich príslušnosť k danému ACL je uvedená týmto číslom
- **Deny**: zakáž paket splňujúci podmienku
- **Permit**: povoľ paket splňujúci podmienku
- **Remark**: vlož poznámku o nasledujúcej položke
- ***TEST_PODMIENKA****:* Identifikátor podmienky vo forme IP adresy (bit pattern). Voči tejto podmienke sa budú porovnávať zdrojové IP adresy vstupujúcich paketov
- ***WILDCARD***: voliteľné. Špecifikácia, ktoré bity zdrojovej IP adresy zdroja sa budú porovnávať voči podmienke uvedenej v ***TEST_PODMIENKA****.*
- **Log:** loguje pakety, ktoré odpovedajú kritériu

Príklad jednoduchého ACL

- ACL úloha:

Vytvor ACL, ktorý povolí IP prístup všetkým hostom zo siete 192.168.10.0/24 do siete 192.168.11.0/24, zakáže všetko ostatné

- Nasadenie?




```
Router(config)#access-list 2 permit 192.168.10.0
```

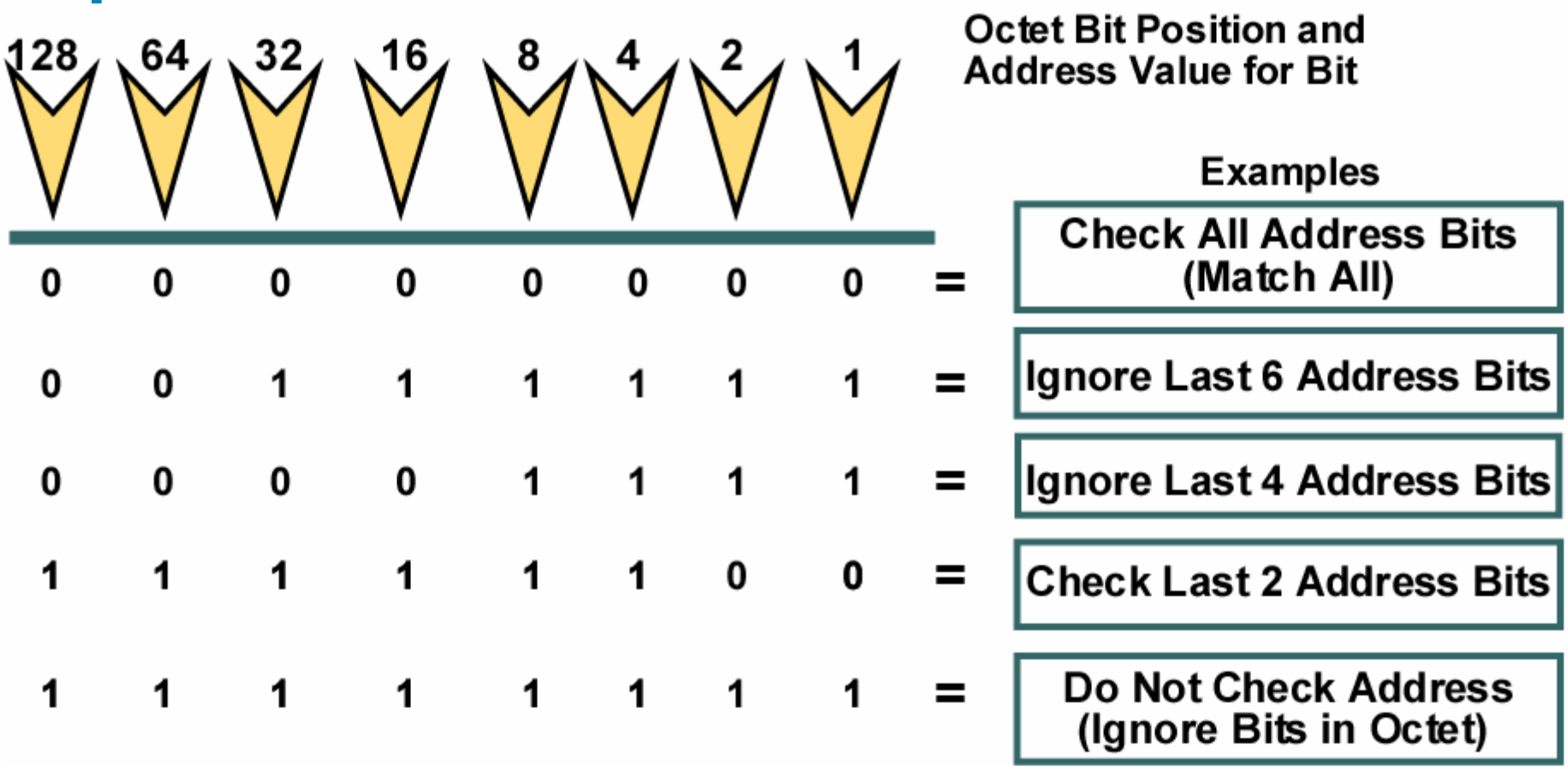
```
! Alebo to iste inak
Router(config)#access-list 2 remark Povol hosty z 192.168.10.0
Router(config)#access-list 2 permit 192.168.10.0 0.0.0.255
Router(config)#access-list 2 deny any
```

Defaultná podmienka

Wildcard Mask

- 32 bitov dlhá adresa, kt. určuje platnosť bitov podmienky
  - Dekadicky podelená na 4 čísla
    - **POZOR:** nie je to subnet maska!
- Definuje, ktoré bity IP adresy z paketu sa budú porovnávať s testovanou podmienkou ACL listu
  - Bity masky uvedené ako „**0**“
    - Odpovedajúce bity zdrojovej IP adresy z paketu sa **musia** porovnať s bitmi podmienky
  - Bity masky uvedené ako „**1**“
    - Odpovedajúce bity zdrojovej IP adresy z paketu sa **nemusia** porovnať s bitmi podmienky

Wildcard Mask – bity, ktoré treba porovnať



- *0* znamená kontroluj zhodu odpovedajúcich bitov IP adresy a podmienky
- *1* znamená ignoruj hodnotu odpovedajúcich bitov IP adresy a podmienky

Wildcard Mask - príklady

	Decimal	Binary
Testing condition	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
Testing condition	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
Testing condition	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
Testing condition	192.168.16.0	11000000.10101000.00010000.00000000
Wildcard Mask	0.0.15.255	00000000.00000000.00001111.11111111
Result Range	192.168.16.0 to 192.168.31.0	11000000.10101000.00010000.00000000 to 11000000.10101000.00011111.00000000

	Decimal	Binary
Testing condition	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	

Počítanie WM masky môže byť zjednodušené odčítaním masky siete od 255.255.255.255.

```
  255 . 255 . 255 . 255
- 255 . 255 . 255 . 240
  -----------------------
    0   .   0   .   0   . 15
```

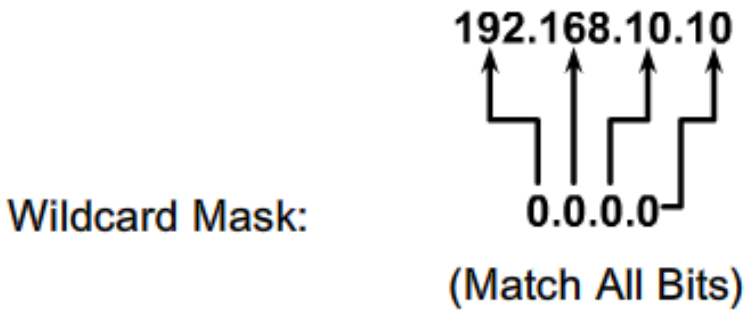
WM – kľúčové slová

Podmienka 192.168.10.10 0.0.0.0 vyžaduje kontrolu všetkých 32 bitov adresy voči podmienke.

ZJEDNODUŠENIE:

- Namiesto 192.168.10.10 0.0.0.0 použiť ako náhradu WM slovíčko host

host 192.168.10.10

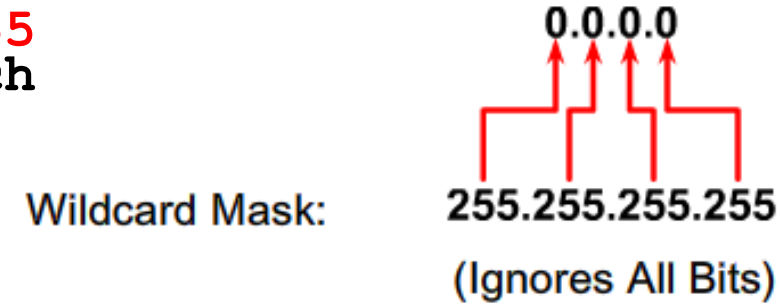


Podmienka 0.0.0.0 255.255.255.255 ignoruje porovnávanie na všetkých bitoch.

ZJEDNODUŠENIE:

- Namiesto 0.0.0.0 255.255.255.255 použiť ako náhradu slovíčko any

any



WM – kľúčové slová - použitie

```
Router(config)#access-list 2 permit|deny 192.168.10.132 0.0.0.0

! To iste s host
Router(config)#access-list 2 permit|deny host 192.168.10.132
```

```
Router(config)#access-list 3 permit|deny 0.0.0.0 255.255.255.255
or
Router(config)#access-list 3 permit|deny xxx.xxx.xxx.xxx 255.255.255.255

! To iste s any
Router(config)#access-list 3 permit|deny any
```

WM – vyčlenenie subnet

- Kontroluj zoznam sietí od 172.30.16.0/24 to 172.30.31.0/24.
  - ACL test condition (Address) and wildcard mask:
  - 172.30.16.0 0.0.15.255

Network .Host

172.30.16.0

	0	0	0	1	0	0	0	0		
Wildcard Mask:	0	0	0	0	1	1	1	1		
	|<----	Match	---->	|	<-----	Don't Care	----->	|		
	0	0	0	1	0	0	0	0	=	16
	0	0	0	1	0	0	0	1	=	17
	0	0	0	1	0	0	1	0	=	18
					:					:
	0	0	0	1	1	1	1	1	=	31

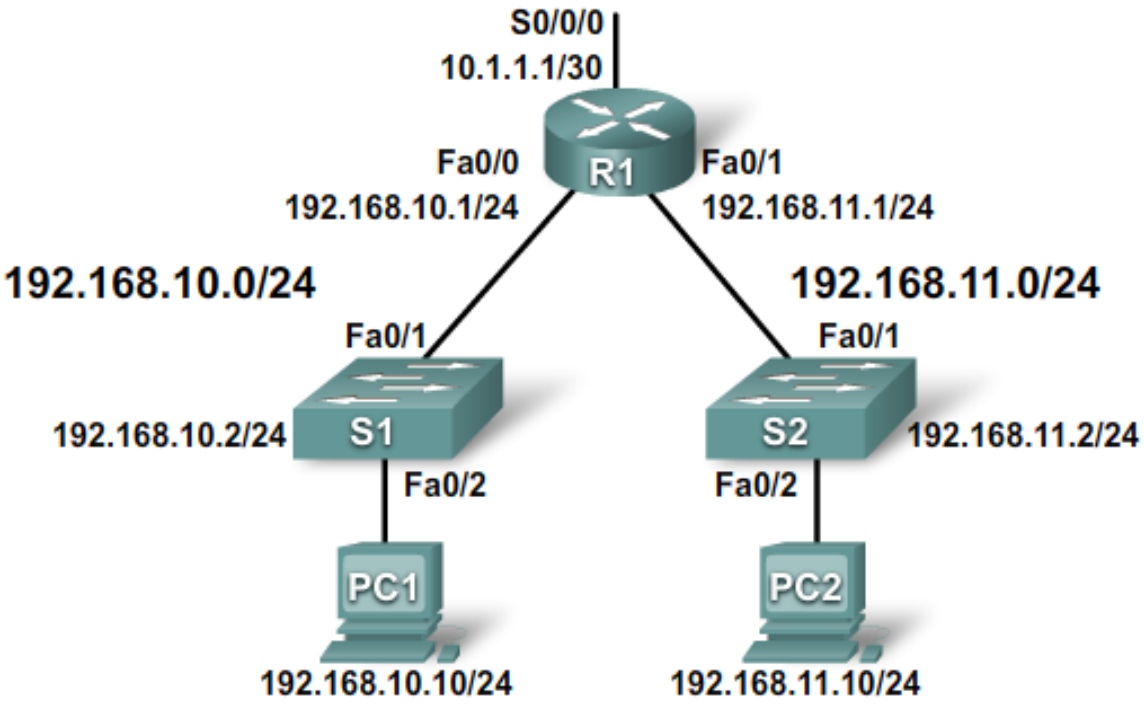
ICND20GR_66

Priradenie ACL na rozhranie

```
Router(config)# interface TYPE SPEC
Router(config-if)# ip access-group {ACCESS-LIST-# | ACCESS-LIST-NAME}
{in | out}
```

- ***ACCESS-LIST-# :*** Číslo acl, ktoré priraďujem na rozhranie
- ***ACCESS-LIST-NAME*** : alebo meno ACL, ktoré priraďujem
- ***IN*** | ***OUT***: v akom smere aplikujem ACL

Priradenie ACL – príklad 1

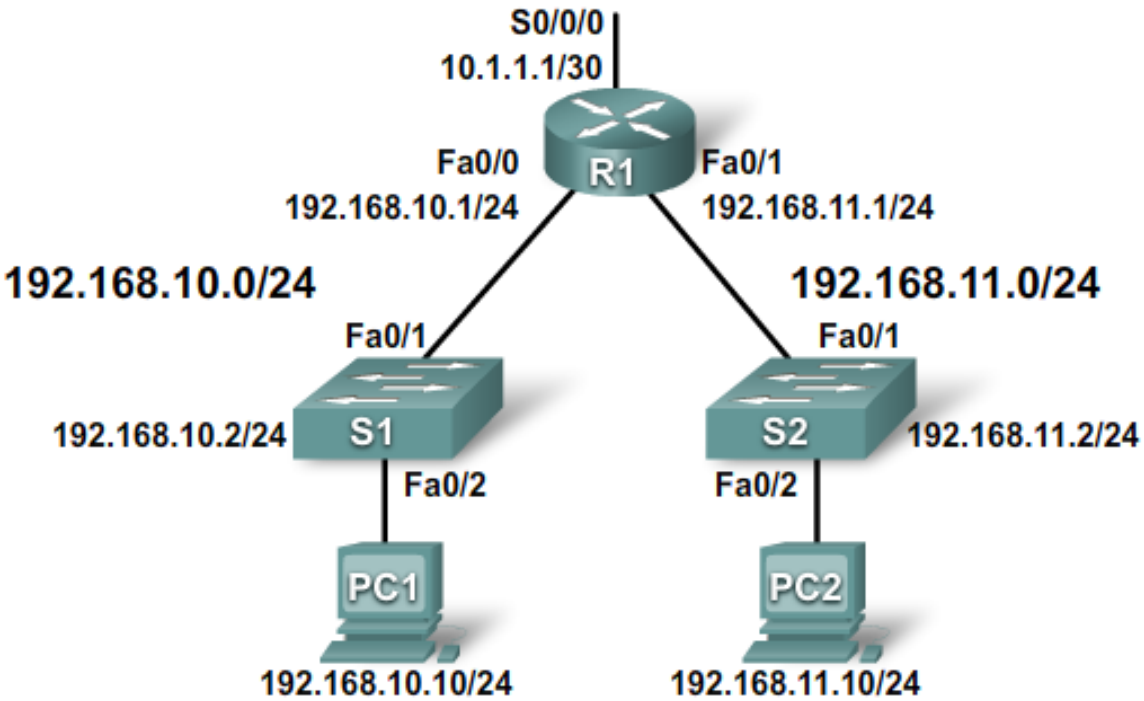



```
! Vytvorenie ACL
Router(config)#access-list 2 permit 192.168.10.0 0.0.0.255
! Priradenie ACL
Router(config)#interface fa 0/1
Router(config-if)#ip access-group 2 out
```

Čo robí ACL?

Pozor na default deny any na konci

Priradenie ACL – príklad 2

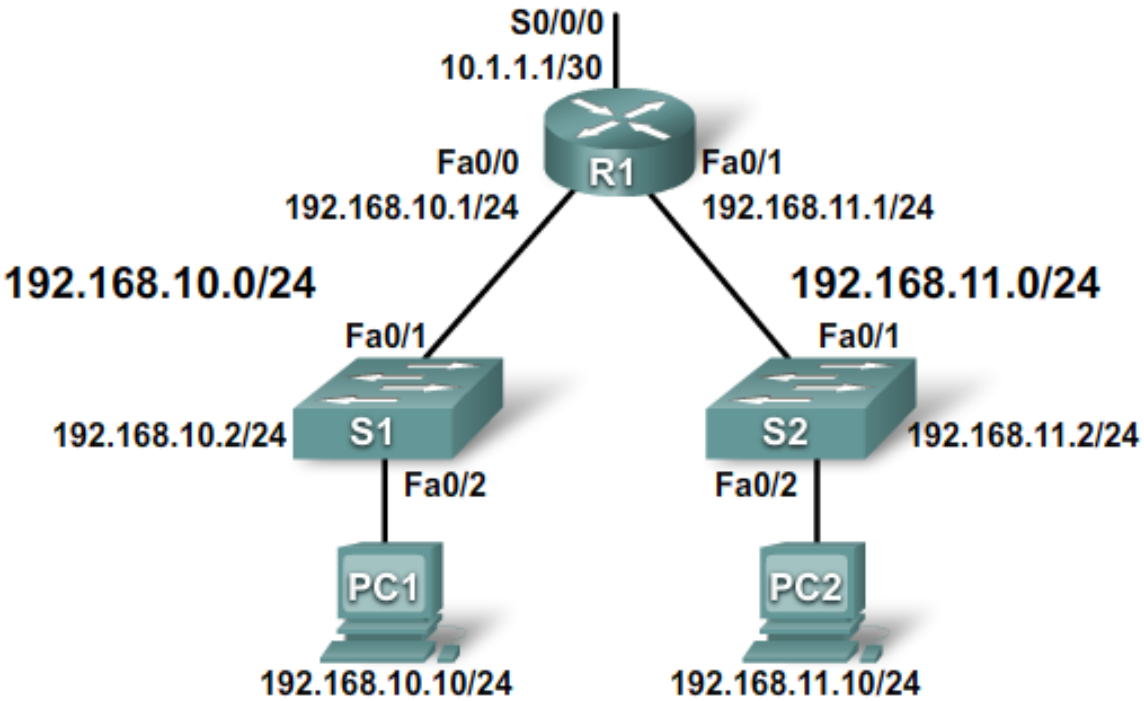



```
! Vytvorenie ACL
Router(config)#access-list 2 deny host 192.168.10.10
Router(config)#access-list 2 permit 192.168.10.0 0.0.0.255
! Priradenie ACL
Router(config)#interface s 0/0/0
Router(config-if)#ip access-group 2 out
```

Čo robí ACL?

Pozor na default deny any na konci

Priradenie ACL – príklad 2b

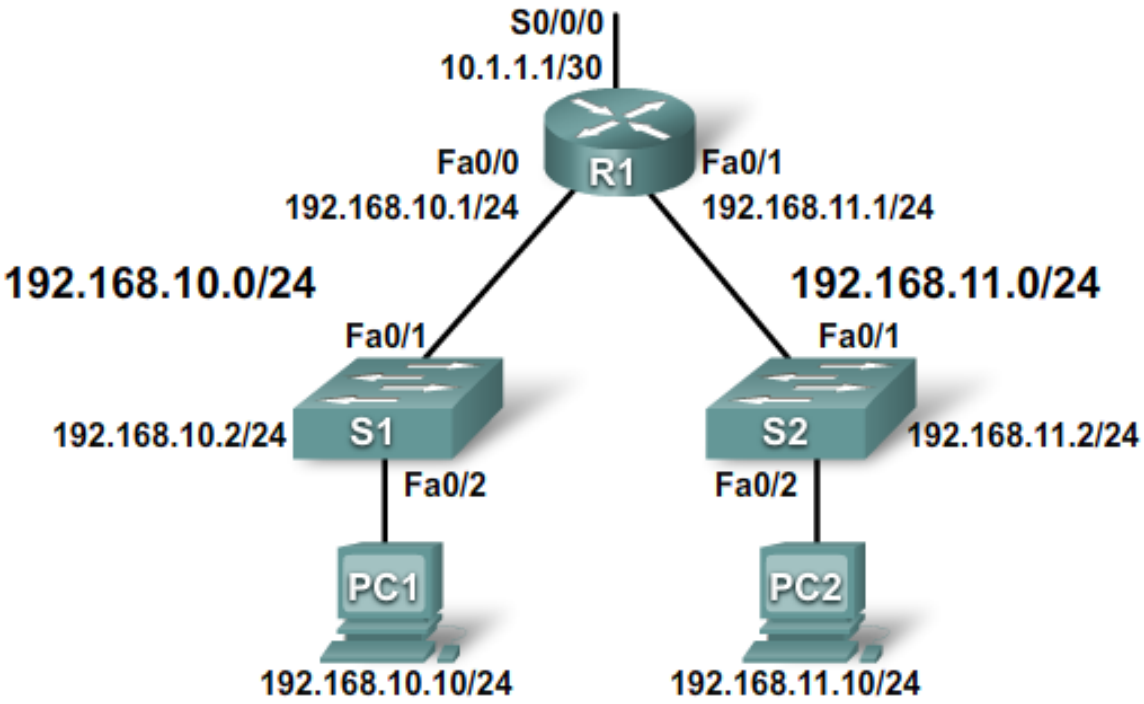



```
! Vytvorenie ACL
Router(config)#access-list 2 permit 192.168.10.0 0.0.0.255
Router(config)#access-list 2 deny host 192.168.10.10

! Priradenie ACL
Router(config)#interface s 0/0/0
Router(config-if)#ip access-group 2 out
```

Čo robí ACL? Čo sa stalo keď som vymenil poradie podmienok?

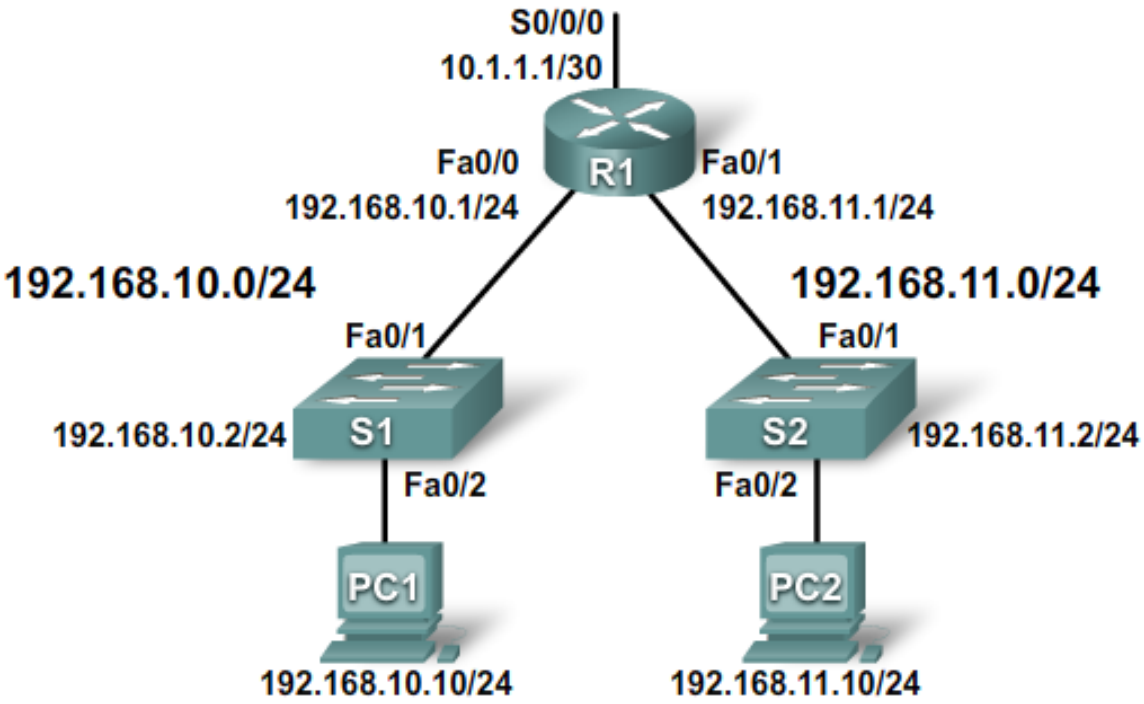
Priradenie ACL – príklad 3




```
! Vytvorenie ACL
Router(config)#access-list 2 deny host 192.168.10.10
Router(config)#access-list 2 permit 192.168.0.0 0.0.255.255
Router(config)#access-list 2 permit any
! Priradenie ACL
Router(config)#interface s 0/0/0
Router(config-if)#ip access-group 2 out
```

Čo robí ACL?

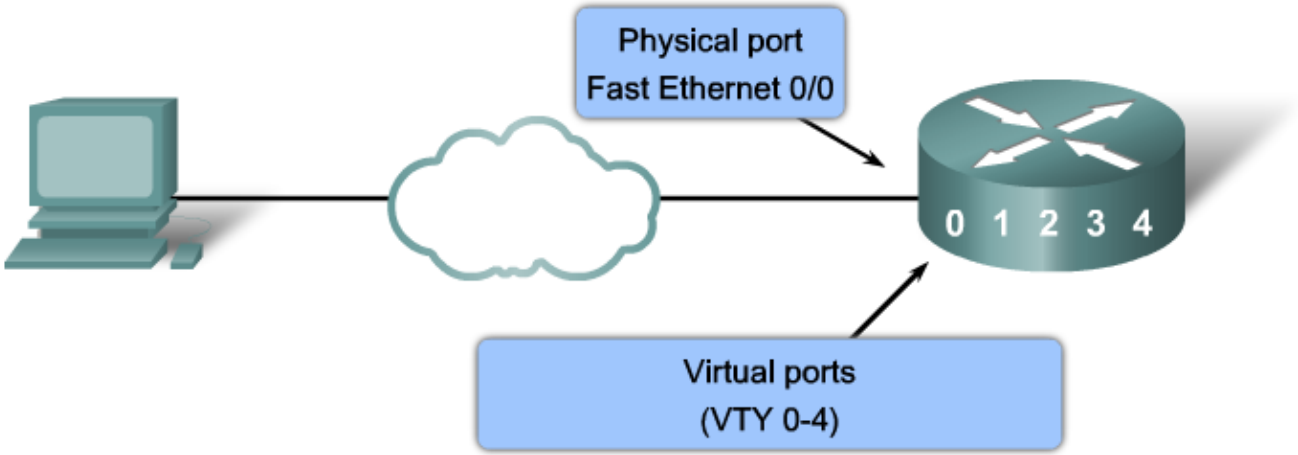
Priradenie ACL – príklad 3 – môžem riešiť inak?




```
! Vytvorenie ACL
Router(config)#access-list 2 deny host 192.168.10.10
Router(config)#access-list 2 permit any
! Priradenie ACL
Router(config)#interface s 0/0/0
Router(config-if)#ip access-group 2 out
```

Kontrola prístupu na VTY cez ACL

```
Router(config-line)# access-class ACCESS-LIST-NUMBER {in [vrf-also] | out}
```




```
! Vytvorenie ACL
Router(config)#access-list 21 permit host 192.168.10.10
Router(config)#access-list 21 permit host 158.193.152.108

! Priradenie ACL na vty line
Router(config)#line vty 0 4
Router(config-if)#access-class 21 in
```

Editovanie standard ACL

- Podmienky standard ACL sú pridávané v poradí ako sú zadávané adminom
- V starších IOS nie je možné neskôr doeditovať zmeny
  - Preto sa odporúča predpripraviť ACL v editore (napr. Notepad++)
  - Pri zmenách treba celý ACL zmazať a spraviť na novo

```
! Mam ACL
Router(config)#do sh run | include access-list
access-list 23 permit host 192.168.10.10
access-list 23 deny 192.168.10.0 0.0.0.255
```

```
! Chcem zmenit permit host z 10 na 11
access-list 23 permit host 192.168.10.11
access-list 23 deny 192.168.10.0 0.0.0.255
```

```
! Stary acl musim zrusit a vytvorit ho na novo
Router(config)#no access-list 23
Router(config)#access-list 23 remark Povol Durimu pristup
Router(config)#access-list 23 permit host 192.168.10.11
Router(config)#access-list 23 remark zakaz ostatnych
Router(config)#access-list 23 deny 192.168.10.0 0.0.0.255
```

Použitie poznámok v ACL

Konfigurácia pomenovaného standard ACL

- Výhoda pomenovaných ACL
  - Jednoduchšia identifikácia
  - V možnosti ich neskoršej editácie
    - Pridávanie podmienok aj na iné miesto ako na koniec ACL
    - Zmena podmienok

```
! Vytvorenie menneho ACL
! Meno je alfa numericky retazec, ktory nesmie zacinat cislom

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

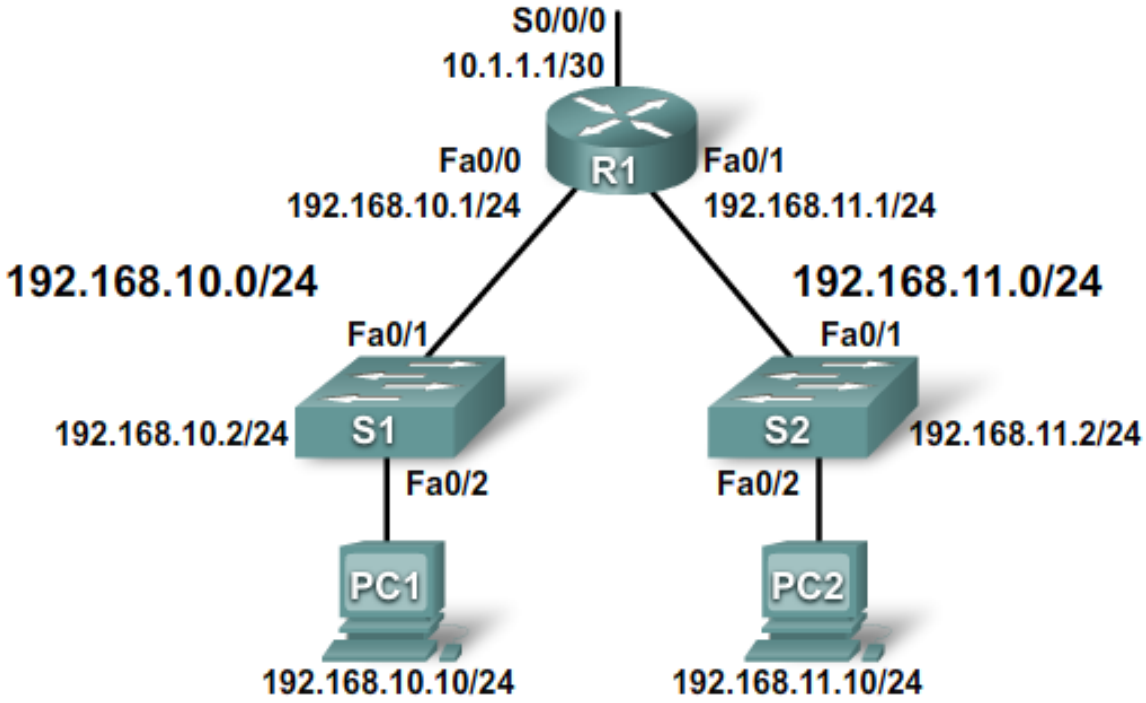
```
! Zadanie testovacich podmienok menneho ACL
! Poradie testovacich podmienkok je dane defaultne od 10 s krokom 10
! Zadanie "no" a cislo riadku acl vyhodi podmienku

Router(config-std-nacl)#[permit | deny | remark ] TEST_CONDITION WM [log]
```

```
! Priadenie menneho ACL na rozhranie

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

Konfigurácia pomenovaného standard ACL




```
! Vytvorenie pomenovaneho ACL
Router(config)#ip access-list standard MOJ-ACL
Router(config-std-nacl)# remark Povol Tomasovy pristup
Router(config-std-nacl)# permit host 192.168.10.10
Router(config-std-nacl)# remark Zakaz zvysok Tomasovej siete
Router(config-std-nacl)# deny 192.168.10.0 0.0.0.255

! Priradenie ACL
Router(config)#interface s 0/0/0
Router(config-if)#ip access-group MOJ-ACL out
```

Overenie ACL

```
Router#show access-list
```

```
Router#show ip access-list
```

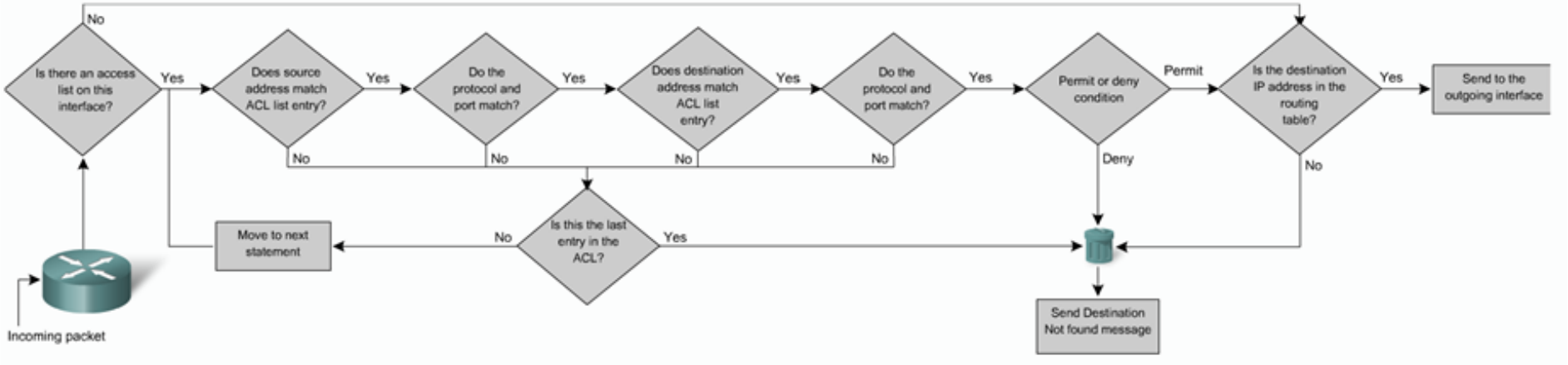
```
Router#show running-config
```

Post editácia pomenovaného standard ACL

Rozšírené ACL

Rozšírené (extended) ACL

- Rozšírené ACL testujú protokolovú sadu, zdrojovú IP, zdrojový port, cieľovú IP a cieľový port voči testovacej podmienke

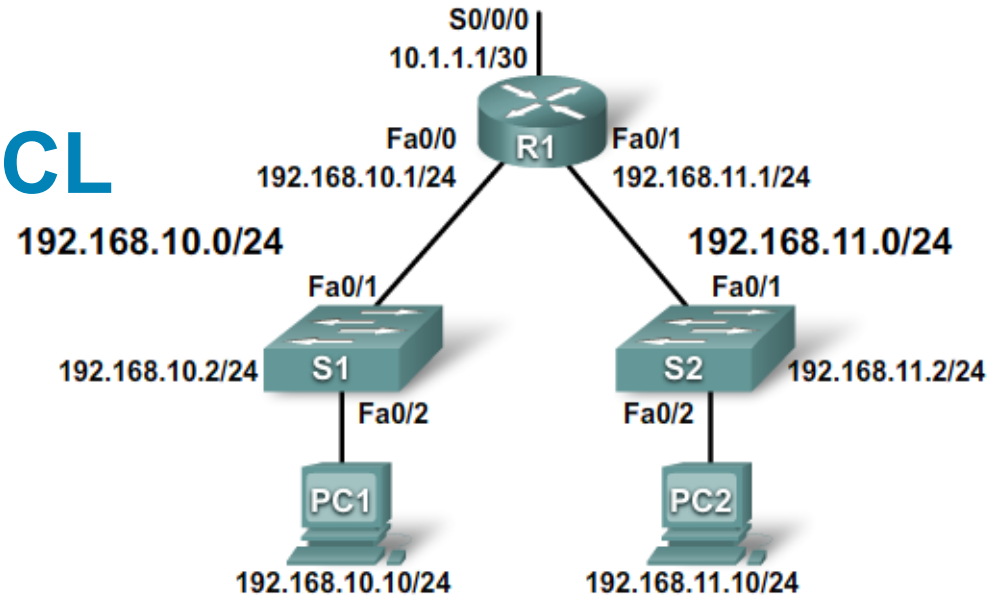


Konfigurácia extended ACL

```
access-list access-list-number {deny | permit | remark} protocol source [source-wildcard]
[operator operand] [port port-number or name] destination [destination-wildcard] [operator
operand] [port port-number or name] [established]
```

Parameter	Description
access-list-number	Identifies the access list using a number in the range 100 to 199 (for an extended IP ACL) and 2000 to 2699 (expanded IP ACLs).
deny	Denies access if the conditions are matched.
permit	Permits access if the conditions are matched.
remark	Indicates whether this entry allows or blocks the specified address. Could also be used to enter a remark.
protocol	Name or number of an Internet protocol. Common keywords include **icmp**, **ip**, **tcp**, or **udp**. To match any Internet protocol (including ICMP, TCP, and UDP) use the **ip** keyword.
source	Number of the network or host from which the packet is being sent.
source-wildcard	Wildcard bits to be applied to source.
destination	Number of the network or host to which the packet is being sent.
destination-wildcard	Wildcard bits to be applied to the destination.
operator	(Optional) Compares source or destination ports. Possible operands include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).
port	(Optional) The decimal number or name of a TCP or UDP port.
established	(Optional) For the TCP protocol only: Indicates an established connection.

Príklad extended ACL



- ACL úloha:

Vytvor ACL, ktorý povolí všetkým hostom zo siete 192.168.10.0/24 HTTP a HTTPS kamkoľvek

```
Router(config)#access-list 101 remark Povol HTTP
Router(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 80
Router(config)#access-list 101 remark Povol HTTPS
Router(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 443
```

Aplikovanie na rozhranie, ktoré?

```
Router(config)#int s 0/0/0
Router(config-if)#ip access-group 101 out
```

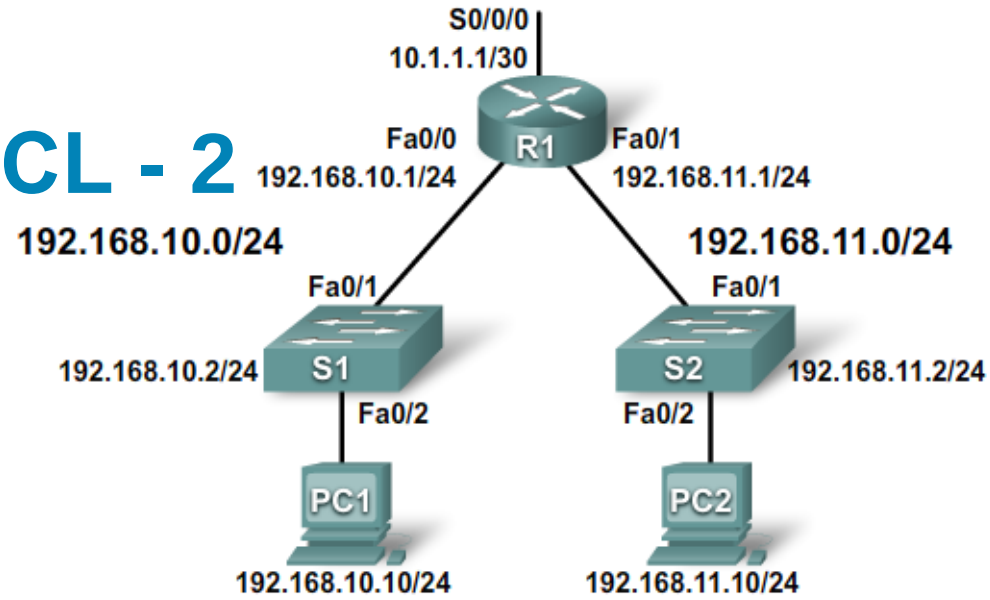
Áno, ale

```
Router(config)#int fa0/0
Router(config-if)#ip access-group 101 in
```

Áno, ale

Zadanie nie je celkom došpecifikované

Príklad extended ACL - 2



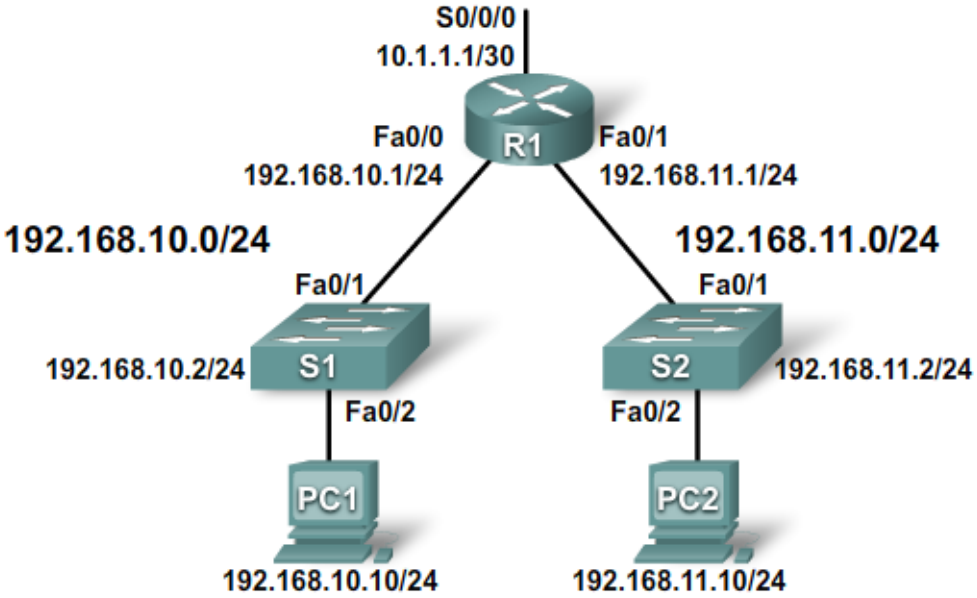
- ACL úloha:
  - Vytvor ACL, ktorý povolí všetkým hostom zo siete 192.168.10.0/24 HTTP a HTTPS kamkoľvek
  - A do vnútra tejto siete nepovolí žiaden prístup z von

```
Router(config)#access-list 101 remark Povol HTTP
Router(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 80
Router(config)#access-list 101 remark Povol HTTPS
Router(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 443
Router(config)#int fa 0/0
Router(config-if)#ip access-group 101 in
```

Riešenie bodu 2?

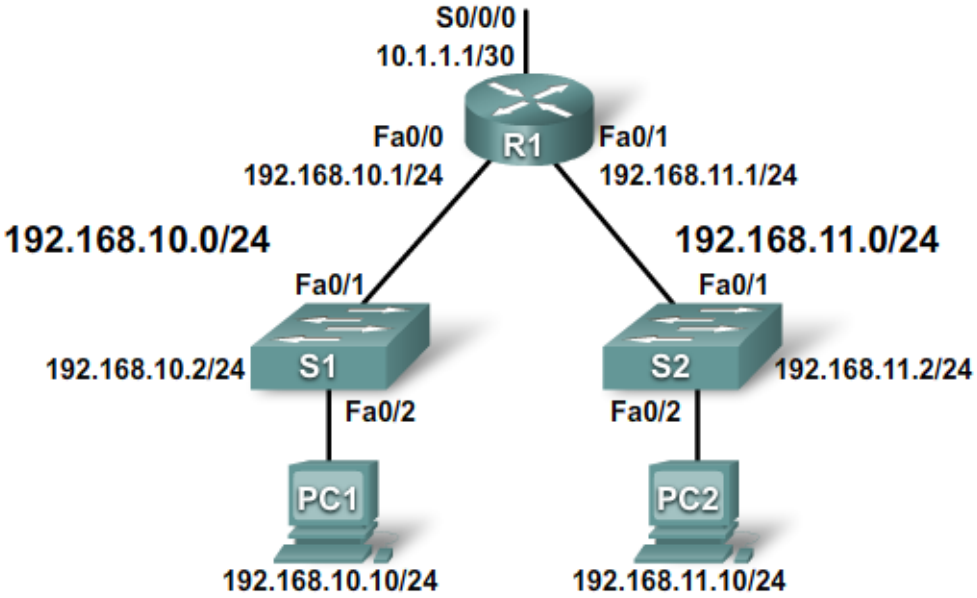
```
! Iny ACL, ktory bude riesit vstup do siete
Router(config)#access-list 102 remark Povol len zalozene TCP spojenia
Router(config)#access-list 102 permit tcp any any established
Router(config)#int fa 0/0
Router(config-if)#ip access-group 102 out
```

Iné príklady




```
! Zakaz zo siete 192.168.11.0 telnet a povol ostatne
Router(config)#access-list 104 deny tcp 192.168.11.0 0.0.0.255 any eq 23
Router(config)#access-list 104 permit ip any any
Router(config)#int fa 0/1
Router(config-if)#ip access-group 104 in
```

Iné príklady




```
! Zakaz zo siete 192.168.11.0 ftp do siete 10.0 a povol ostatne
Router(config)#access-list 105 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0 0.0.0.255 eq 21
Router(config)#access-list 105 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0 0.0.0.255 eq 20
Router(config)#access-list 105 permit ip any any
Router(config)#int fa 0/1
Router(config-if)#ip access-group 105 in
```

Konfigurácia pomenovaného extended ACL

- Výhoda pomenovaných ACL
  - Jednoduchšia identifikácia
  - V možnosti ich neskoršej editácie
    - Pridávanie podmienok aj na iné miesto ako na koniec ACL
    - Zmena podmienok

```
! Vytvorenie menneho ACL
! Meno je alfa numericky retazec, ktory nesmie zacinat cislom

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

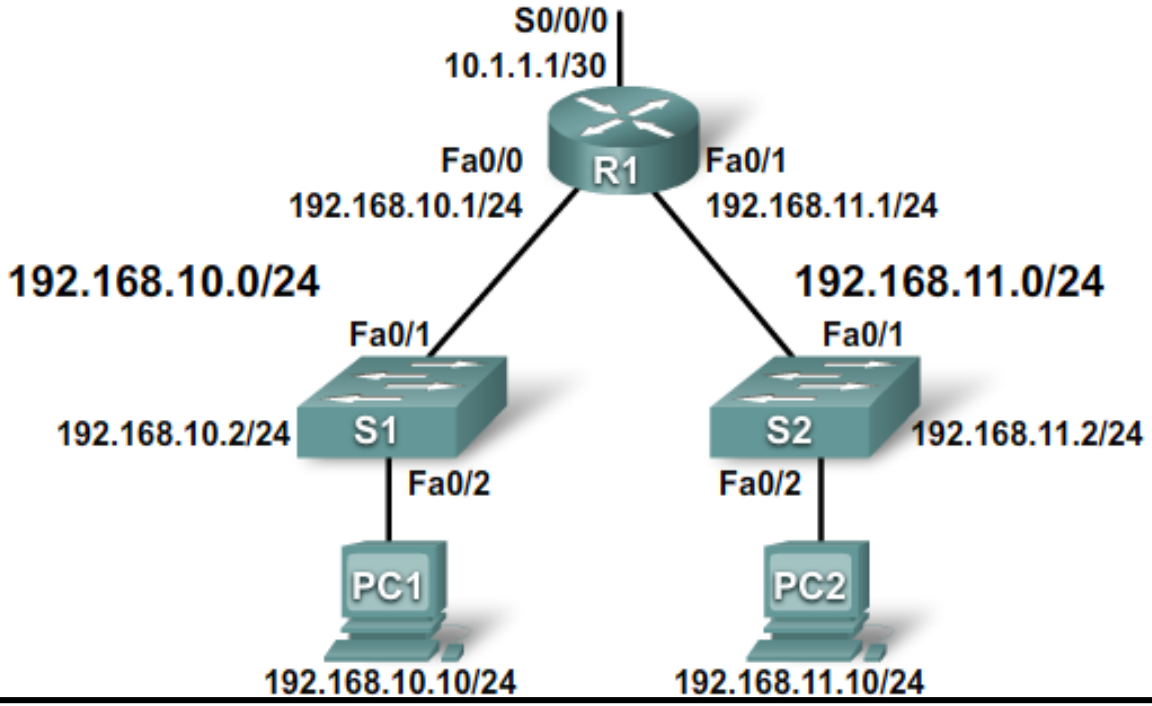
```
! Zadanie testovacich podmienok menneho ACL
! Poradie testovacich podmienkok je dane defaultne od 10 s krokom 10
! Zadanie "no" a cislo riadku acl vyhodi podmienku

Router(config-std-nacl)#[permit | deny | remark ] TEST_CONDITION WM [log]
```

```
! Priadenie menneho ACL na rozhranie

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

Konfigurácia pomenovaného extended ACL




```
! Vytvorenie pomenovaneho ACL
Router(config)#ip access-list extended WEB-SERVICES-ONLY
Router(config-std-nacl)# remark Povol HTTP
Router(config-std-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
Router(config-std-nacl)# remark Povol HTTPS
Router(config-std-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
! Priradenie ACL
Router(config)#int fa 0/0
Router(config-if)#ip access-group WEB-SERVICES-ONLY in
```

Komplexné ACL

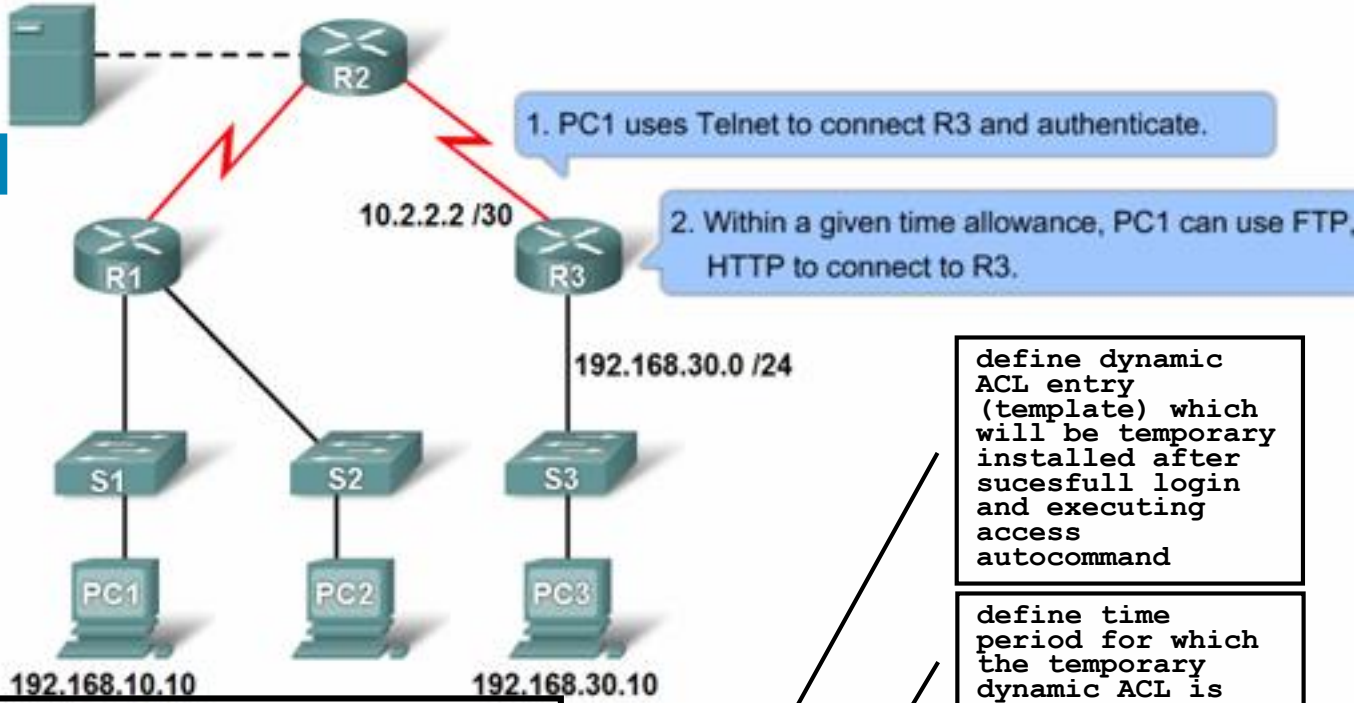
Komplexné ACL

- Máme tri typy komplexných ACL
  - **Dynamické (Dynamic) ACL**
    - Používatelia, ktorí chcú komunikovať cez router sa musia najskôr naň prihlásiť cez telnet
  - **Reflexívne (Reflexive) ACL**
    - Umožňuje prevádzke prechádzať smerom z dnu von, v opačnom smere obmedzuje komunikáciu
  - **Časové (Time-Based) ACL**
    - Riadenie prevádzky podľa času

Dynamické ACL

- Princíp zámka-kľúč (lock and key)
- Dostupné len pre IP prevádzku
  - Využíva extended ACL
- Používateľ, ktorý chce „prechádzať“ cez smerovač, sa musí naň najprv prihlásiť a autentifikovať (telnet)
  - Do extended ACL je pridaná dočasná položka, ktorá mu umožní dočasne komunikovať cez smerovač
- Použitie
  - Poskytnutie dočasnej konektivity do siete pre vzdialených používateľov
- Výhody
  - Mechanizmus autentifikácie používateľov
  - Zjednodušený manažment prístupu vo veľkých sieťach
  - Obmedzenie prielomov do siete hackermi
  - Vytvorenie dynamických prechodov cez FW, bez obmedzenia iných bezpečnostných reštrikcií

Dynamické ACL - príklad




```
! Vytvorenie uctu pre telnet
username palo password my_password
```

```
! Vytvorenie ACL
access-list 111 permit tcp any host 192.168.10.10 eq telnet
access-list 111 dynamic my_dynamic permit ip any any
access-list 111 dynamic my_dynamic timeout 120 permit ip any any
```

```
int s x/y/z
Ip access-group 111 in
```

```
line vty 0 4
autocommand access-enable timeout 5
login local
```

define dynamic ACL entry (template) which will be temporary installed after sucesfull login and executing access autocommand

define time period for which the temporary dynamic ACL is created

The autocommand creates a temporary inbound access list entry at the serial x/y/z interface, based on the second access-list entry (my_dynamic). This temporary entry will expire after 5 minutes, as specified by the timeout.

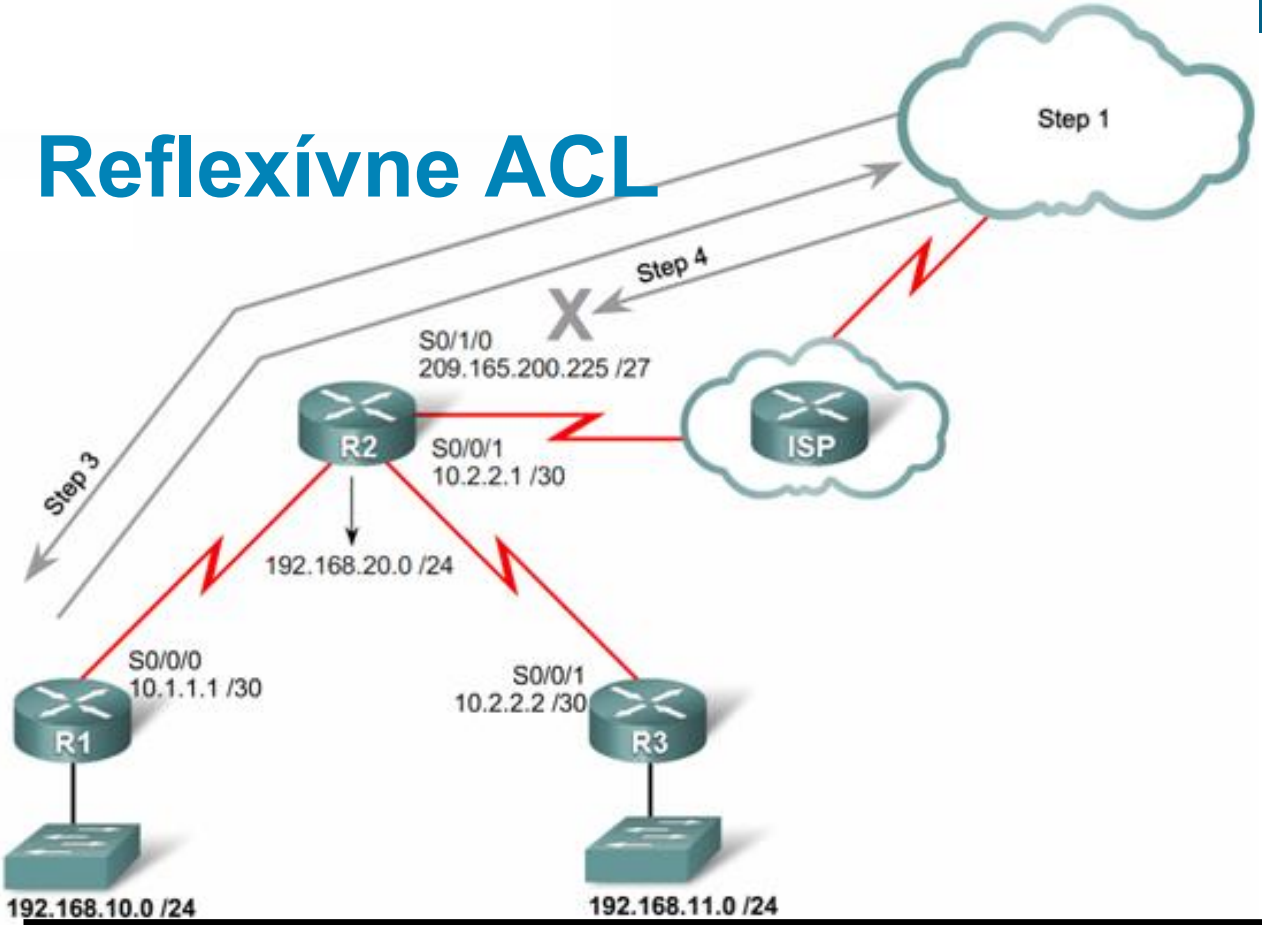
Odporúčanie na použitie dynamických ACL

- Do *not* create more than one dynamic access list for any one access list. The software only refers to the first dynamic access list defined.
- Do *not* assign the same *dynamic-name* to another access list. Doing so instructs the software to reuse the existing list. All named entries must be globally unique within the configuration.
- Assign attributes to the dynamic access list in the same way you assign attributes for a static access list. The temporary access list entries inherit the attributes assigned to this list.
- Configure Telnet as the protocol so that users must open a Telnet session into the router to be authenticated before they can gain access through the router.
- Either define an idle timeout now with the **timeout** keyword in the **access-enable** command in the **autocommand** command, or define an absolute timeout value later with the **access-list** command. You must define either an idle timeout or an absolute timeout—otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated their session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure an idle timeout, the idle timeout value should be equal to the WAN idle timeout value.
- If you configure both idle and absolute timeouts, the idle timeout value must be less than the absolute timeout value.
- If you realize that a job will run past the ACL's absolute timer, use the **access-list dynamic-extend** command to extend the absolute timer of the dynamic ACL by six minutes. This command allows you to open a new Telnet session into the router to re-authentication yourself using lock-and-key.
- The only values replaced in the temporary entry are the source or destination address, depending whether the access list was in the input access list or output access list. All other attributes, such as port, are inherited from the main dynamic access list.
- Each addition to the dynamic list is always put at the beginning of the dynamic list. You cannot specify the order of temporary access list entries.
- Temporary access list entries are never written to NVRAM.

Reflexívne ACL (IP Session Filtering)

- Umožňuje otvárať (povoľovať) IP toky (relácie) z vnútra siete dynamicky
  - Zakazuje, resp. nepovoľuje toky z vonku dnu
- Reflexívne ACL obsahuje len dynamické položky
  - Po reštarte nie sú dostupné
  - Dokonalejšie ako `established` parameter v extended ACL
    - Kontrolujú sa aj iné parametre ako TCP Flag bity
  - Použitie len s pomenovanými extended ACL
  - Inštalované pri štarte relácie z vnútra siete
- Výhody nasadenia
  - Nasadenia na routre na rozhraní Internal/External net
  - Lepšia ochrana siete voči útokom.
  - Lepšia ochrana voči DoS a spoofing útokom.
  - Jednoduchšia obsluha, väčšia kontrola nad prevádzkou.

Reflexívne ACL



we will define reflexive ACL, which will adds dynamic session entries into extended acl for traffic originating in internal network and directed to the external nets

This ACL will compare incoming traffic against entries builded when traffic leaves internal network. The EXTERNAL_IN acl will nest the reflexive acl

```
! Applied on external interface for outbound direction.
R2(config)#ip access-list extended EXTERNAL_OUT
R2(config-ext-nacl)#permit tcp any any reflect TRAFFIC
R2(config-ext-nacl)#permit udp any any eq domain reflect TRAFFIC
```

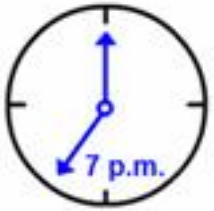
```
! Applied on external interface for inbound direction
R2(config)#ip access-list extended EXTERNAL_IN
R2(config-ext-nacl)#evaluate TRAFFIC
```

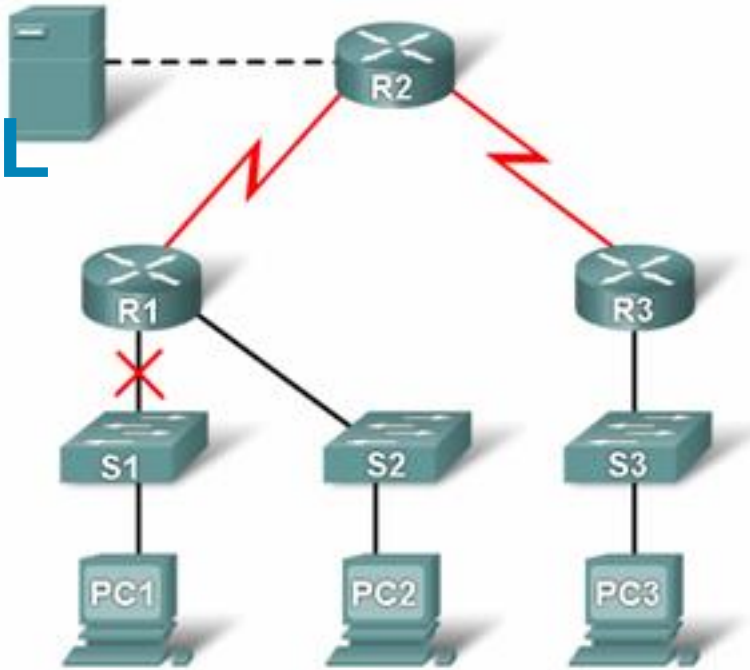
```
R2(config)#int se 0/1/0
R2(config-if)#ip access-group EXTERNAL_IN in
R2(config-if)#ip access-group EXTERNAL_OUT out
```

Časové (Time-Based) ACL

- V činnosti podobné extended ACL
  - Len riadenie prístupu môže byť definované časom
    - Obdobie dňa, deň a podobne + nadefinovaná funkcia

Časové (Time-Based) ACL





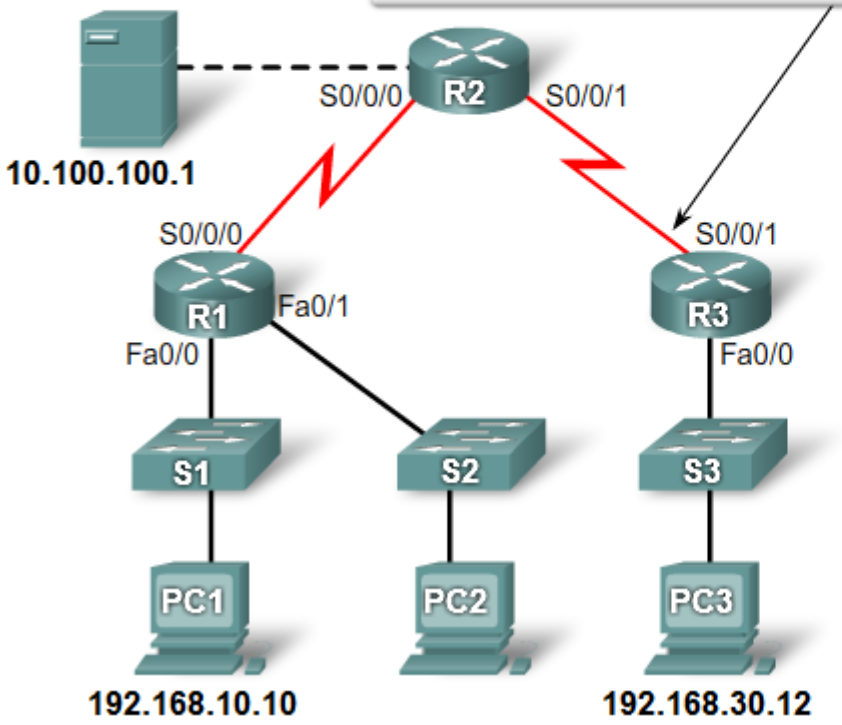
Time-based ACLs:
Allow for access control based on the time of day and week

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

Diagnostika ACL – chyba 1

sh access-list

```
# show access-lists 110
Extended IP access list 110
    10 deny tcp 192.168.10.0 0.0.0.255 any
    20 permit tcp 192.168.10.0 0.0.0.255 any eq telnet
    30 permit ip any any
```



Error 1:

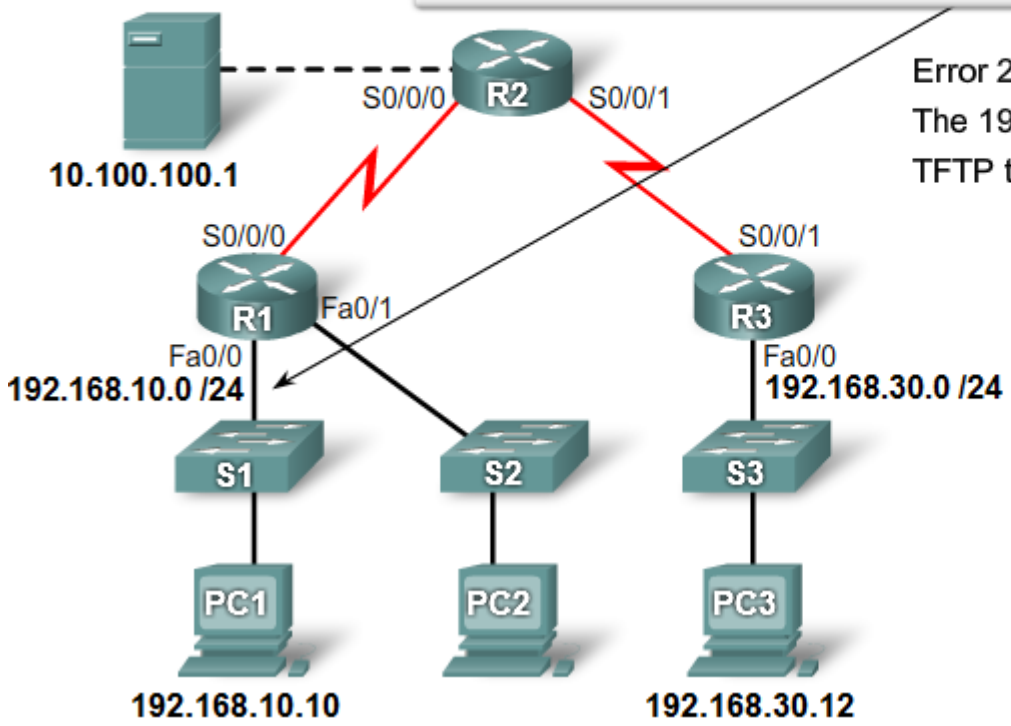
Host 192.168.10.10 has no connectivity with 192.168.30.12

Riešenie

- Skontroluj poradie ACL podmienok

Diagnostika ACL – chyba 2

```
# show access-lists 120
Extended IP access list 120
    10 deny tcp 192.168.10.0 0.0.255.255 any eq telnet
    20 deny tcp 192.168.10.0 0.0.0.255 host 10.100.100.1 eq smtp
    30 permit tcp any any
```



Error 2:

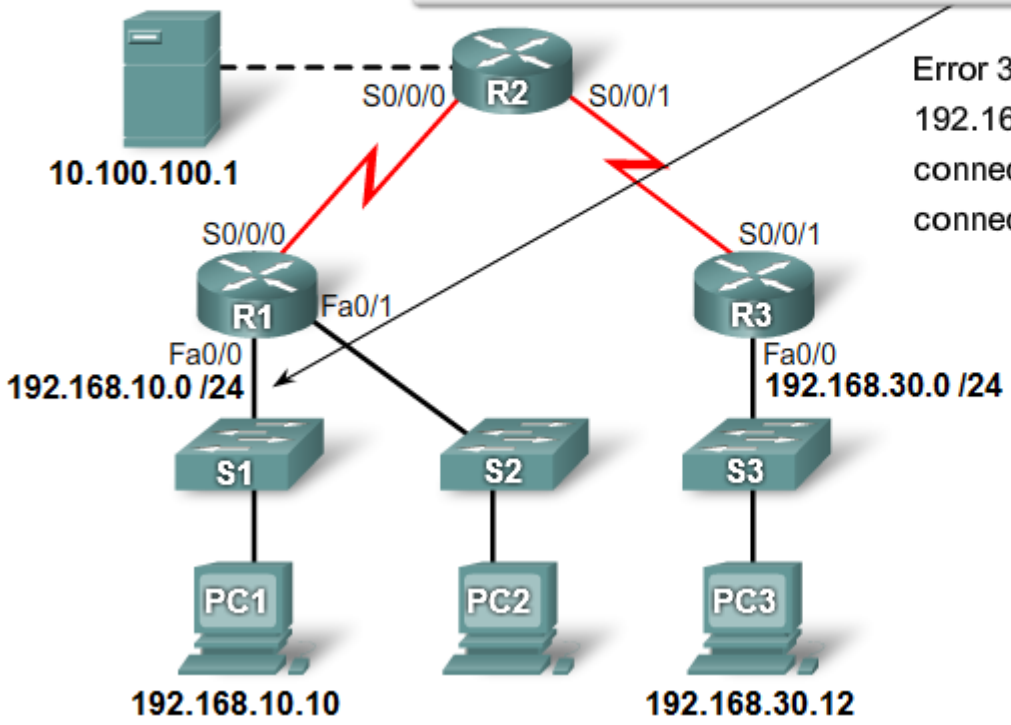
The 192.168.10.0 /24 network cannot use TFTP to connect to the 192.168.30.0 /24.

Riešenie

- TFTP používa UDP

Diagnostika ACL – chyba 3

```
# show access-lists 130
Extended IP access list 130
    10 deny tcp any eq telnet any
    20 deny tcp 192.168.1.0 0.0.0.255 host 192.168.30.0 eq smtp
    30 permit ip any any
```



Error 3:

192.168.10.0 /24 network can use Telnet to connect to 192.168.30.0 /24, but this connection should not be allowed.

Riešenie

- Telnet pravidlo zle zadefinovane, zakazujem source port a nie destination