MINISTERSTVO
ŠKOLSTVA, VEDY,
VÝSKUMU A ŠPORTU
SLOVENSKEJ REPUBLIKY

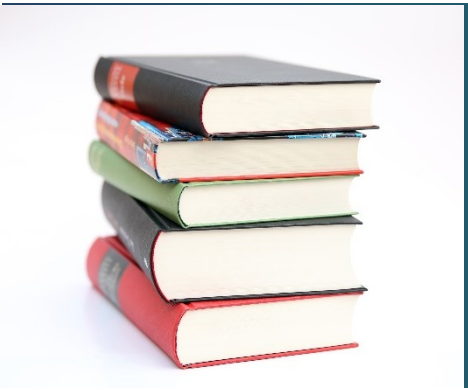
UNIVERSITY OF ŽILINA
Faculty of Management Science
and Informatics

ACL
Access Control Lists

Počítačové siete 1

Mgr. Jana Uramová, PhD.
Katedra informačných sietí
Fakulta riadenia a informatiky, ŽU

Vytvorené v rámci projektu KEGA 026TUKE-4/2021.



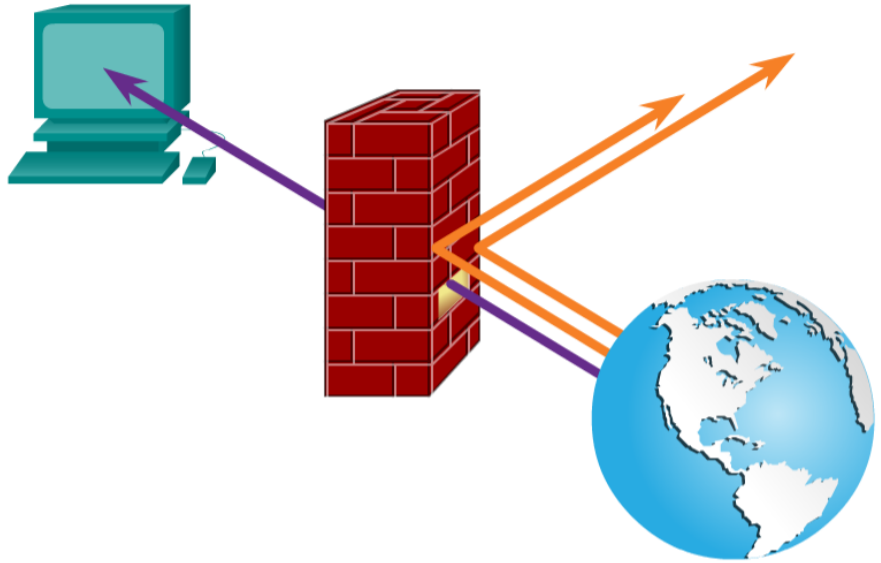
Čo sa dozvieme

- **Účel, nasadenie a činnosť ACL**
- **Typy ACL**
- **Umiestnenie ACL a odporúčania**
- **Konfigurácia ACL (štandardný, rozšírený)**
- **Diagnostika chýb pri nasadzovaní ACL**

- **ENSA_04 ACL Concepts**
- **ENSA_05 ACLs for IPv4 Configuration**

A toto až na ďalšej prednáške:

- Komplexné ACL (dynamické, reflexívne, časové)
- IPv6 ACL



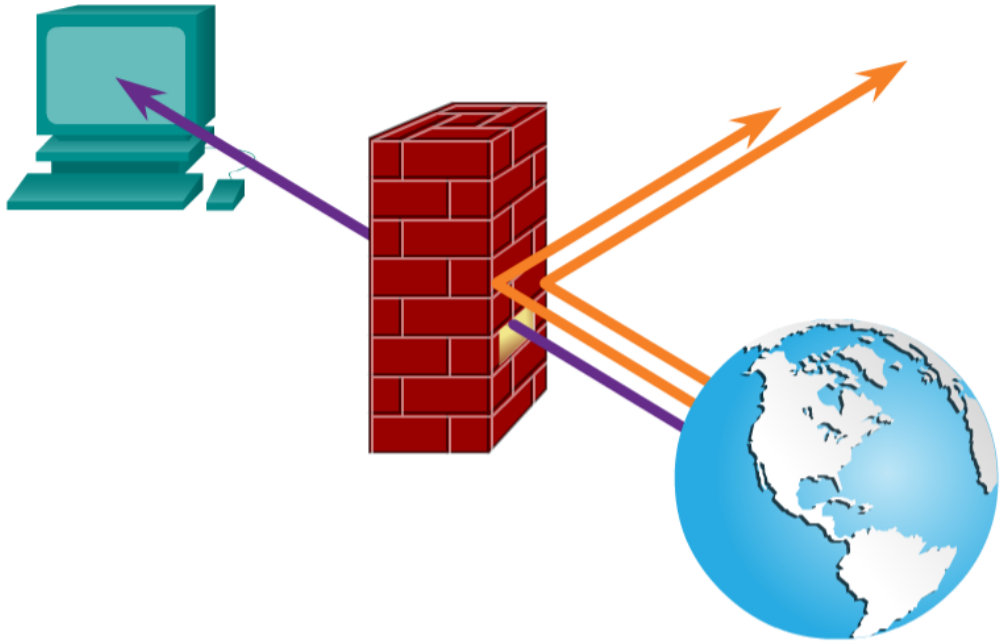
Účel, nasadenie a činnosť ACL

„ACLs are among the most commonly used features of Cisco IOS software.“

Čo je to Access Control List

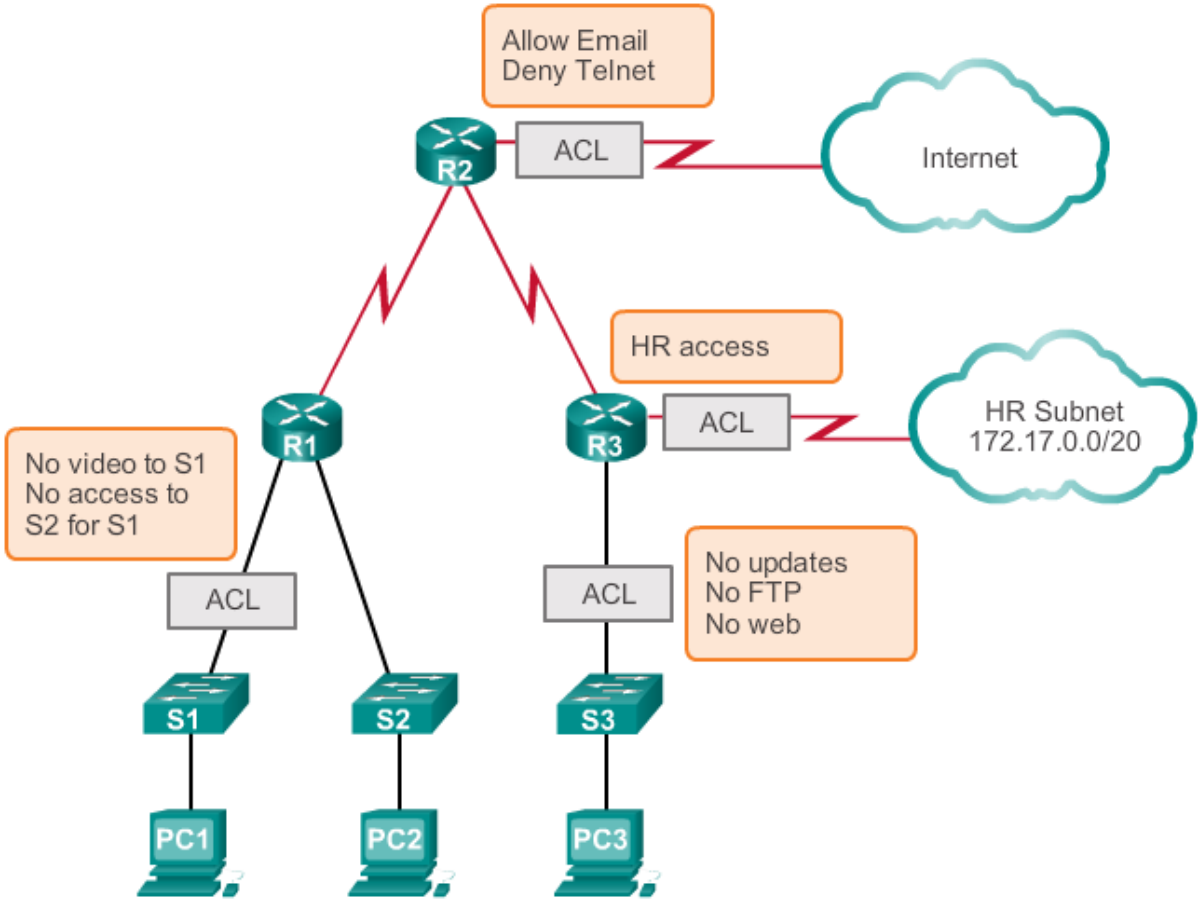
Cisco ACL
(Triediace a kontrolné zoznamy)

- 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 sieťovej prevádzky
  - Povolenie určitého typu prevádzky, služby a zakázanie iného
    - Povoľ SMTP a zakáž telnet
- Riadenie procesov
  - Napr. príjem a zasielanie updates, riadenie smerovania
- Poskytnutie základnej bezpečnosti
  - Riadenie kto môže kam pristupovať



ACL paketový filter

- Zoznam testovacích podmienok = záznamov
  (ACEs Access Control Entries = ACL statements), ktoré sa aplikujú na IP prevádzku prechádzajúcu rozhraniami smerovača
- Každý záznam obsahuje **testovaciu podmienku** a **akciu**, ktorá sa má vykonať, ak podmienka platí:
  - Povoľ (**Permit**) danú prevádzku
  - Zakáž (**Deny**) danú prevádzku
- Defaultne smerovače nemajú implementované ACL filtre

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

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

Legend

Registered TCP Ports:

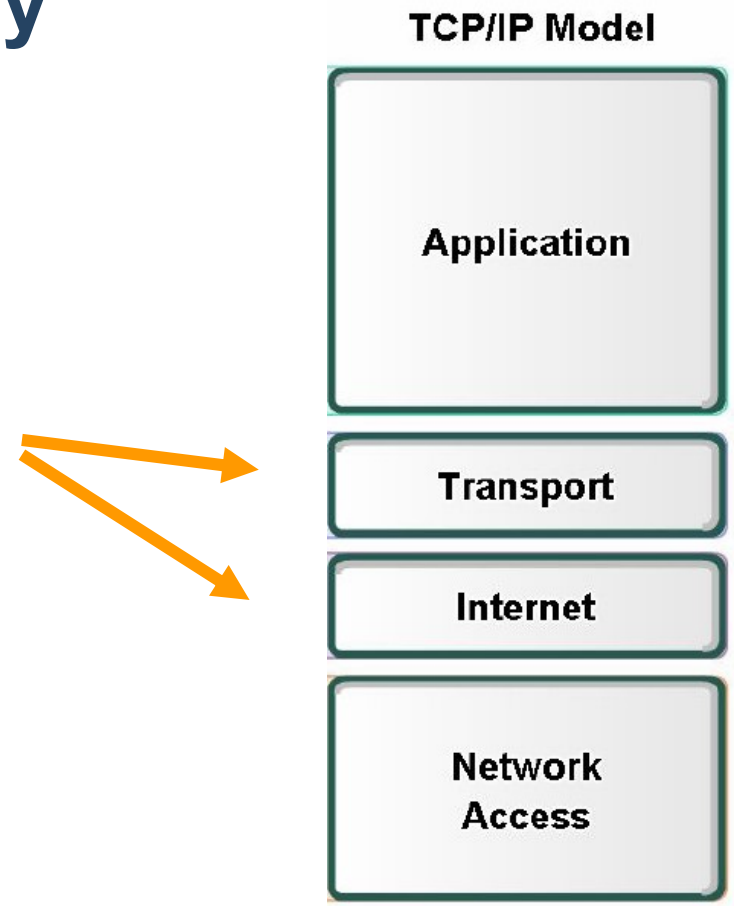
1863	MSN Messenger
2000	Cisco SCCP (VoIP)
8008	Alternate HTTP
8080	Alternate HTTP

Well Known TCP Ports:

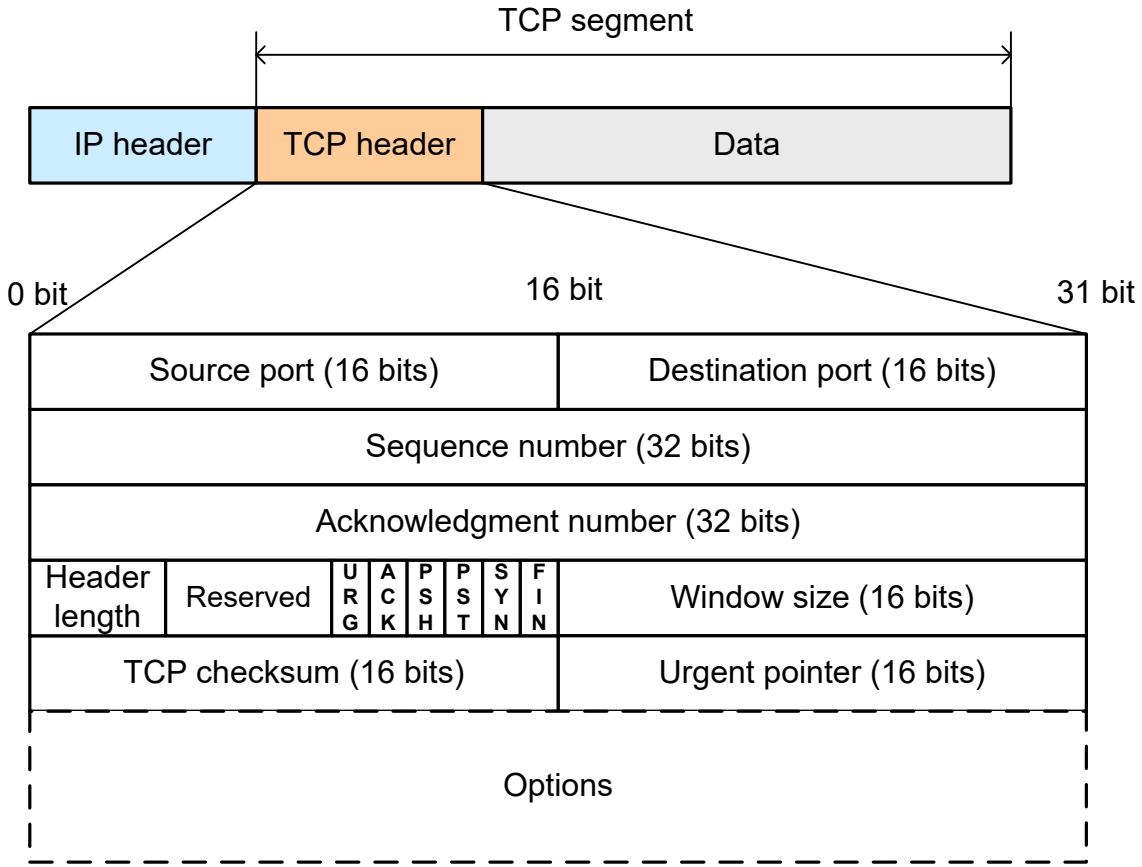
21	FTP
23	Telnet
25	SMTP
80	HTTP
143	IMAP
194	Internet Relay Chat (IRC)
443	Secure HTTP (HTTPS)

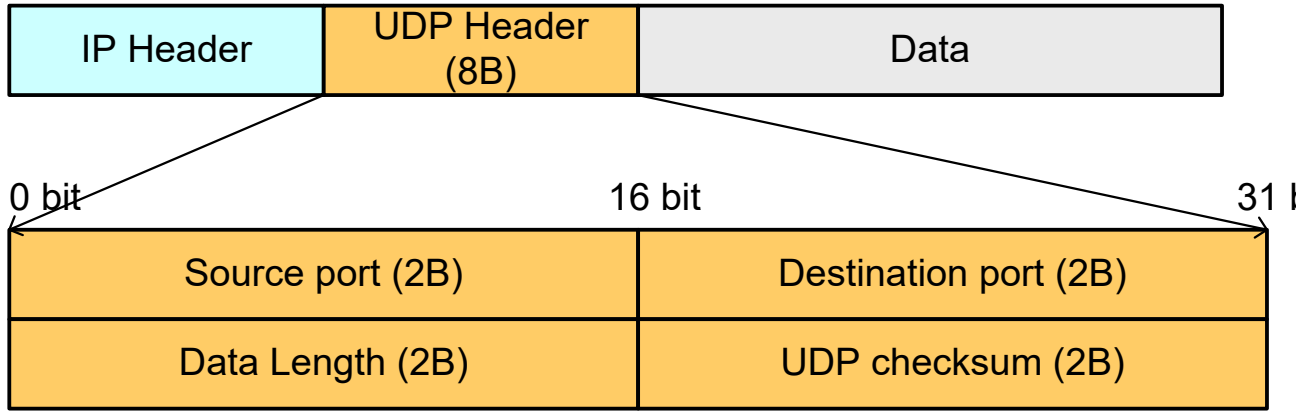
Vlastnosti TCP/IP architektúry

- TCP/IP model je vrstvový!!
- Filtrovanie paketov pracuje na L3 a L4



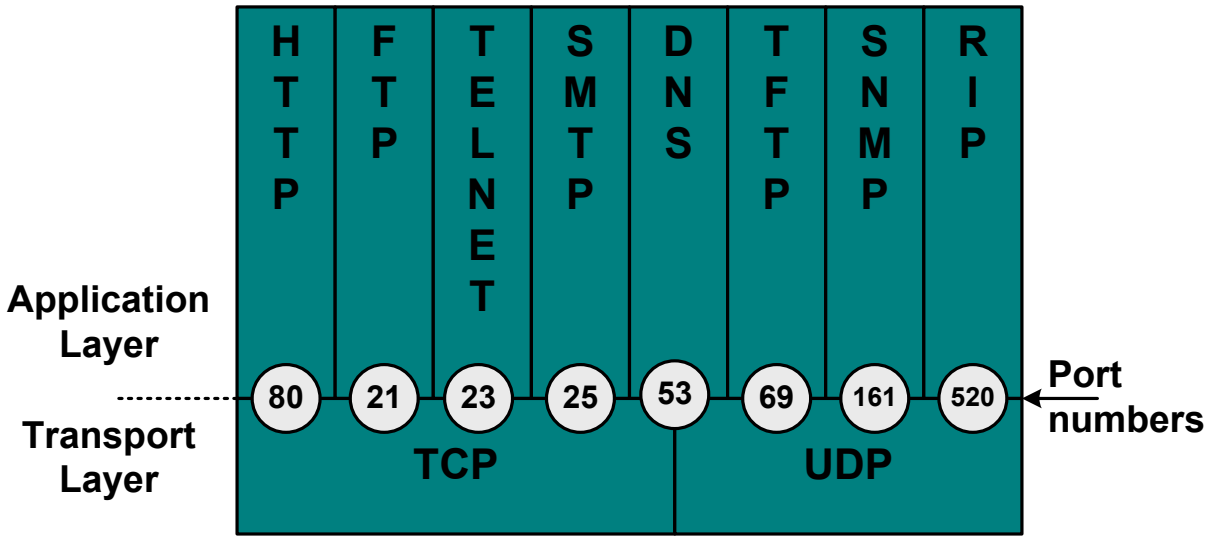
Formát TCP segmentu a UDP datagramu





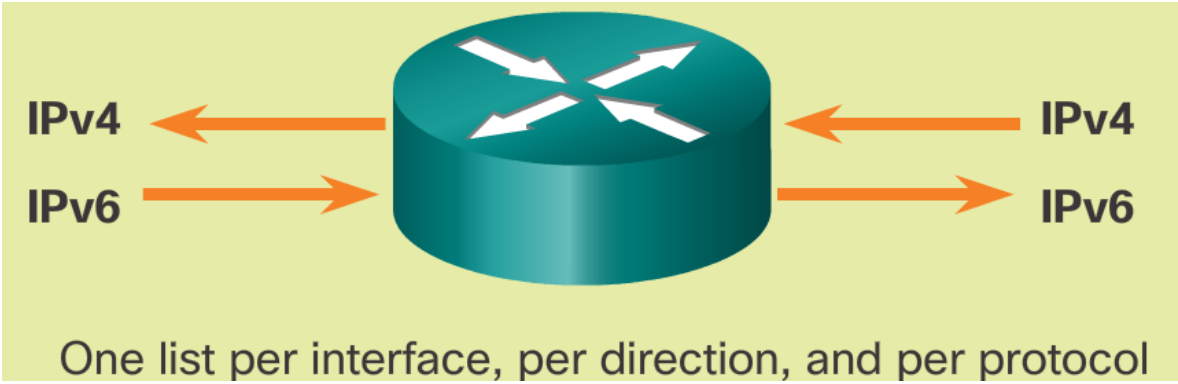
Vlastnosti TCP - 3way handshake, ACK a ukončenie

Čísla portov



- FTP: 20 (data), 21
- Telnet: 23
- SMTP: 25
- WINS replication: 42
- DNS: 53 (UDP aj TCP)
- BOOTP, DHCP: 67 (server), 68 (klient)
- TFTP: 69
- HTTP: 80
- Kerberos: 88 (UDP aj TCP)
- POP3: 110
- NNTP: 119
- NTP: 123
- RPC Locator: 135 (TCP aj UDP)
- IMAPv2: 143
- SNMP: 161
- IMAPv3: 220
- HTTPS: 443
- MS SQL: 1433 (UDP aj TCP)
- AOL, IRC: 531 (UDP aj TCP)

Nasadenie ACL

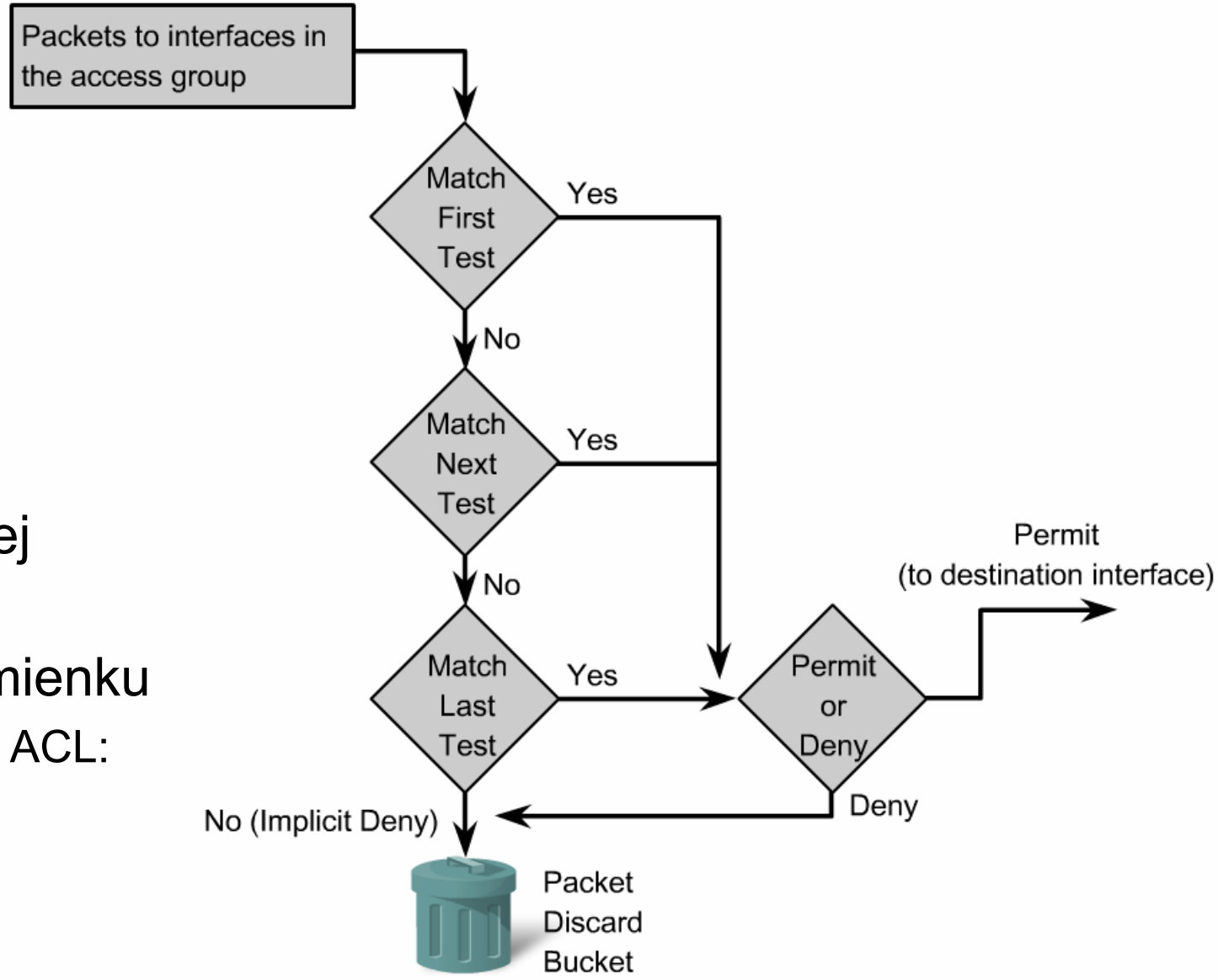


- Jeden ACL per **interface**
  - Jeden ACL per **protokol**
    ACL je definovaný pre každý podporovaný protokol zvlášť
    - Jeden ACL per **smer**
      - ACL riadi tok iba v jednom smere, nie v oboch
- Komplikovanejšie riešenia ACL vyžadujú implementáciu ACL na viac rozhraniach (filter IN a OUT smer)

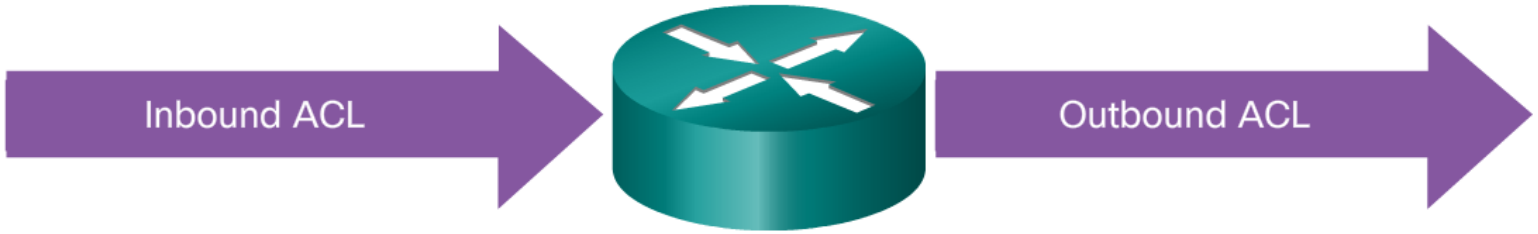
Ako ACL pracuje

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

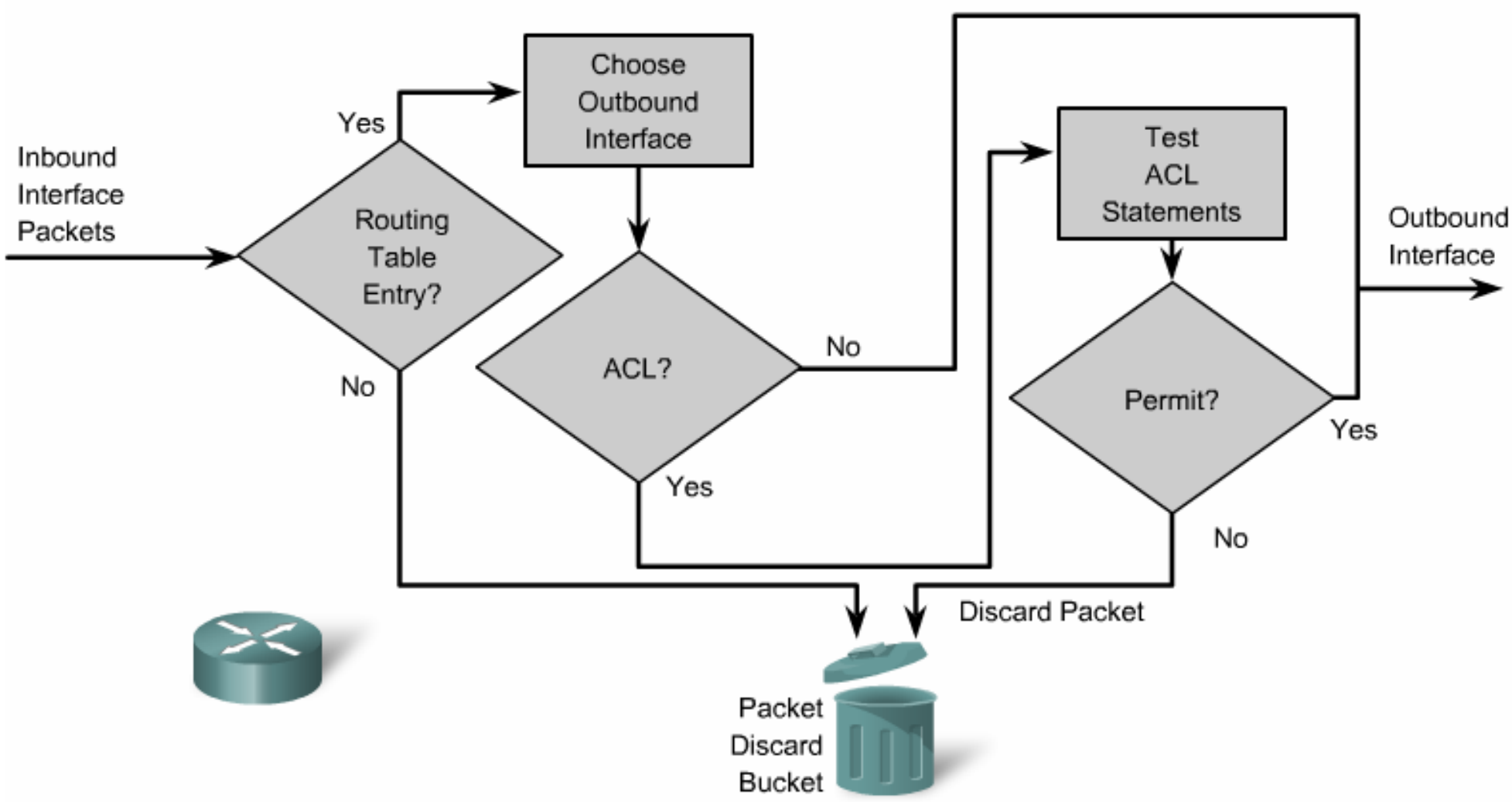


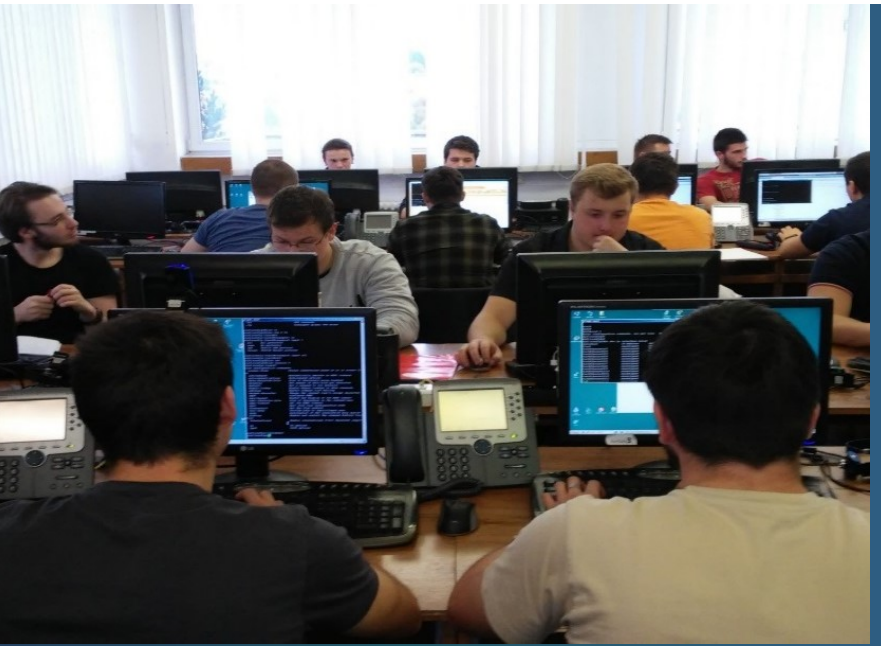
Nasadenie ACL



- **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
  - Vhodný ako filter pre pakety, ktoré môžu prísť **iba cez** dané rozhranie
- **Outbound ACLs**
  - Smer paketov je **von (out)** zo smerovača
    - Vystupujú cez rozhranie
  - Skôr ako je paket postúpený ACL je vykonané smerovanie
  - Vhodný ako filter prevádzky z **rôznych smerov**, idúcu von 1 rozhraním

Činnosť outbound ACL

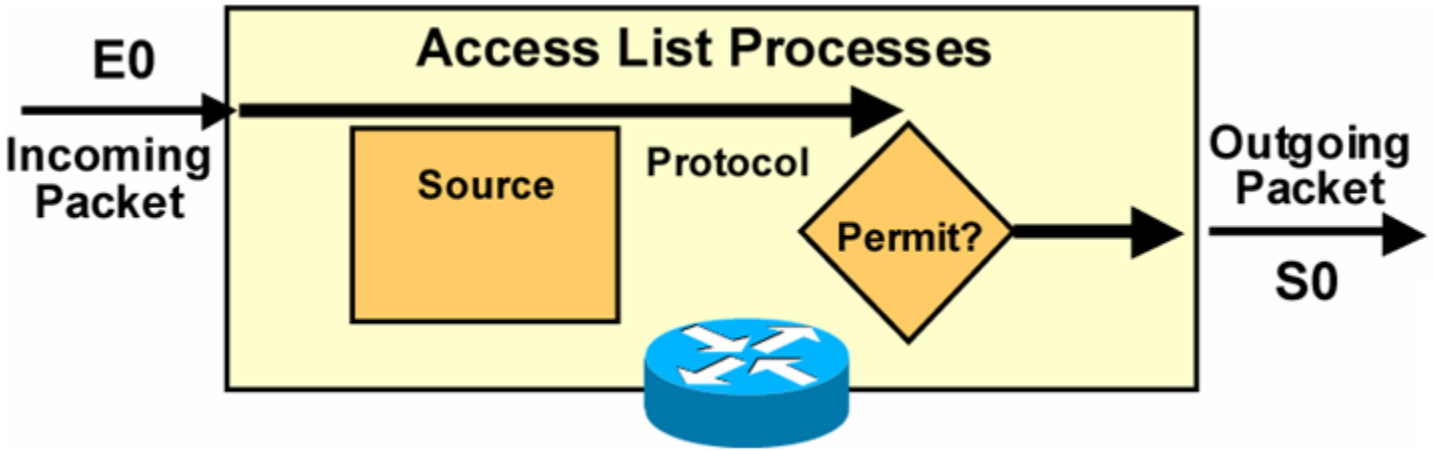




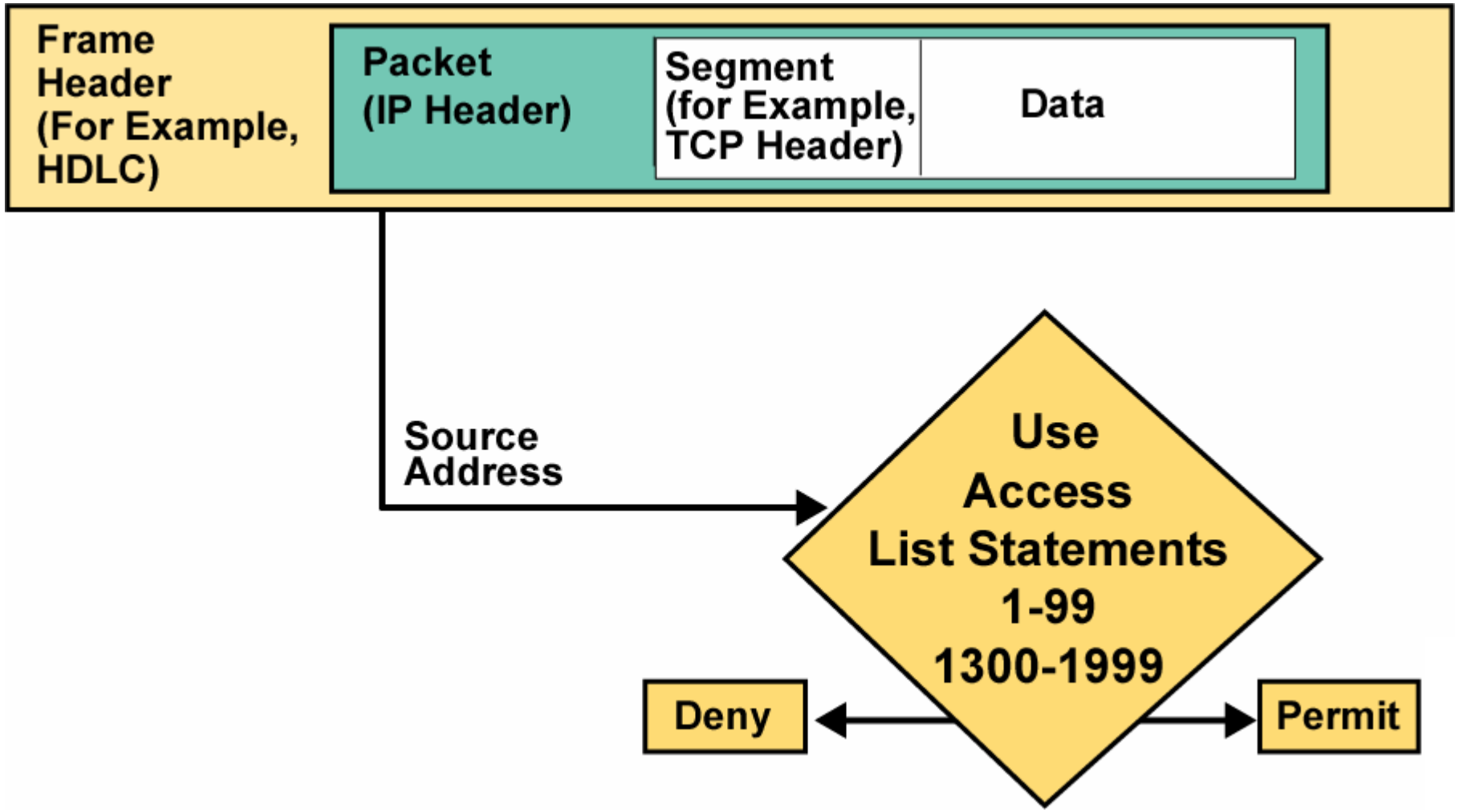
Typy ACL

Štandardné (standard) ACL

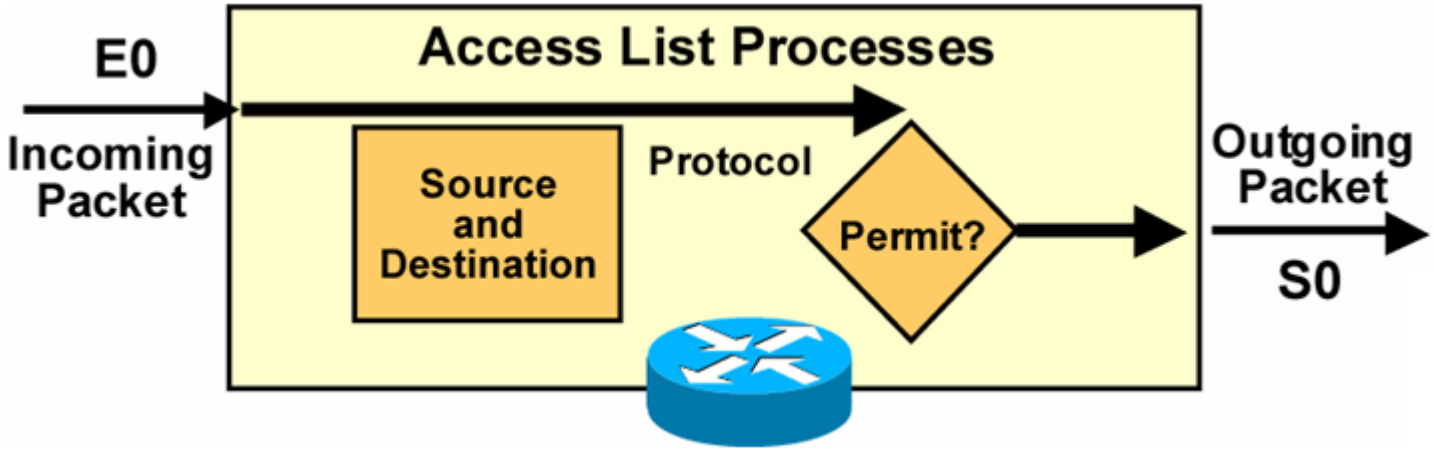
- Všeobecne povoľujem a zakazujem celý protokolový stack
  - Napr. celé IPv4 a pod.
- Na podmienku sa kontroluje len zdrojová adresa



Štandardné (standard) ACL

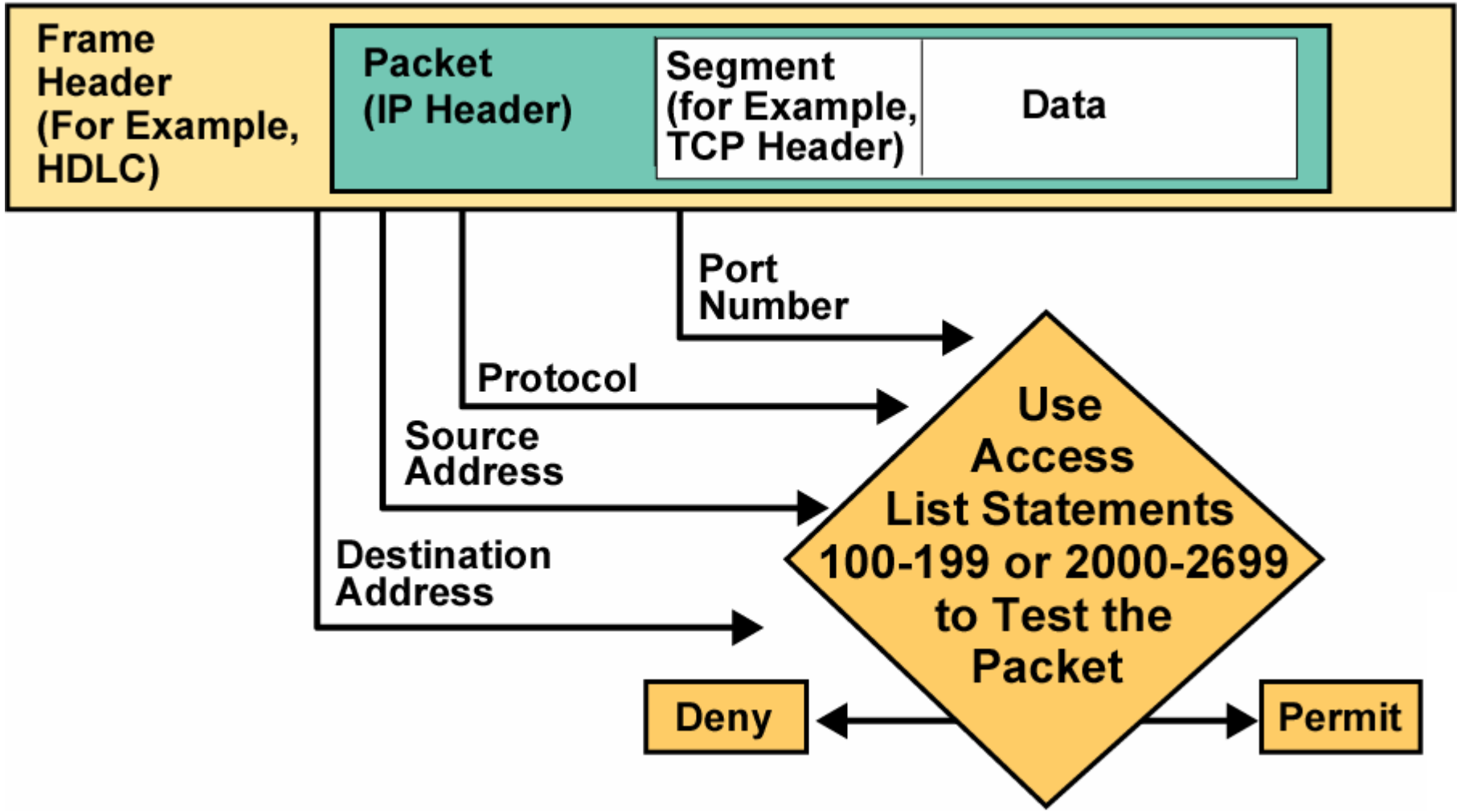


Rozšírené (extended) ACL



- Voči podmienke sa kontroluje:
  - zdrojová aj cieľová **adresa**
  - zdrojový a cieľový **port**
- Povoľujem a zakazujem konkrétny protokol alebo komunikáciu definovanú portom

Rozšírené (extended) ACL



Číslované a pomenované ACL

- **Číslované ACL**
  - **Standard IP ACL**
    - 1 – 99
    - 1300 - 1999
  - **Extended IP ACL**
    - 100 – 199
    - 2000 – 2699
  - **Nevýhoda (iba na starších IOSoch)**: Neviem mazať podmienky, pridávať viem len na koniec zoznamu podmienok (v novších IOSoch to už neplatí)

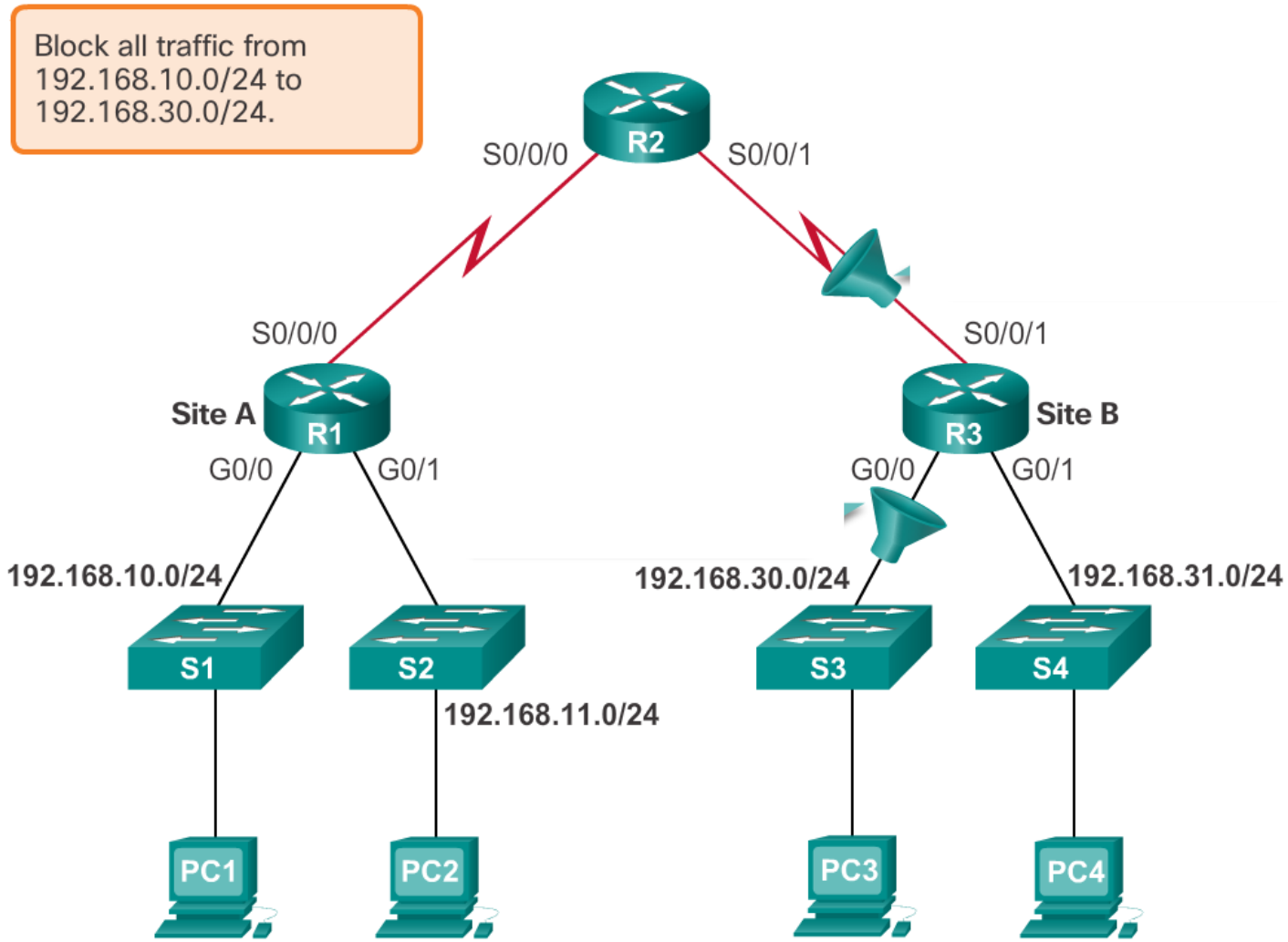
- **Pomenované ACL**
  - ACL (standard alebo extended) je identifikované menom, alfanumerickým (zvykom je používať KAPITÁLKY)
  - Od verzie Cisco IOS 11.2
  - Meno nesmie obsahovať medzery ani interpunkciu
  - **Výhoda**:
  - Môžem mazať a pridávať podmienky (.. už to majú všetky)
  - Lepšie identifikujem dané ACL, keď má meno

Umiestnenie ACL a odporúčania

Umiestnenie ACL – standard ACL

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

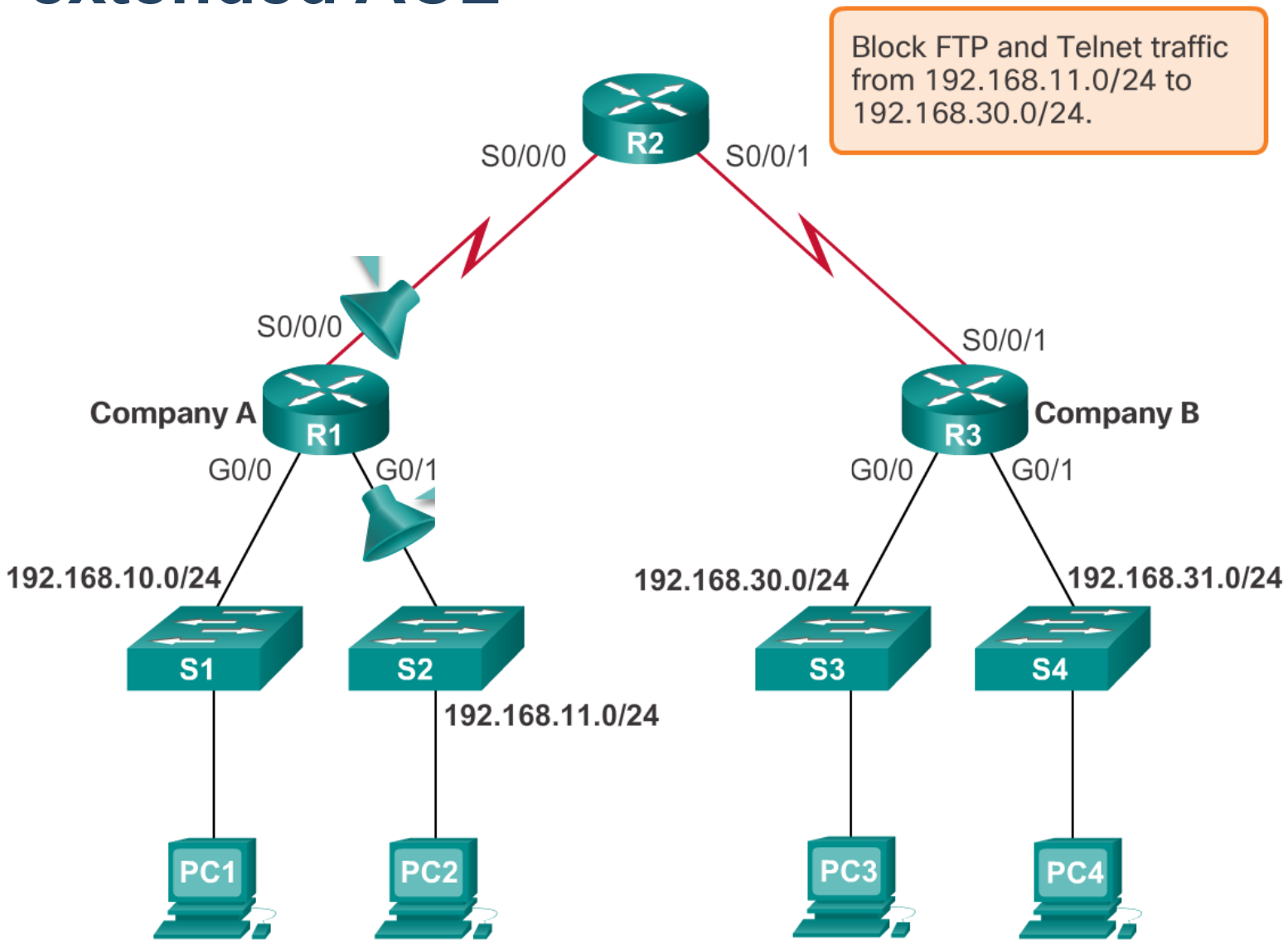
Čo najbližšie k **cieľu !**



Umiestnenie ACL – extended ACL

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

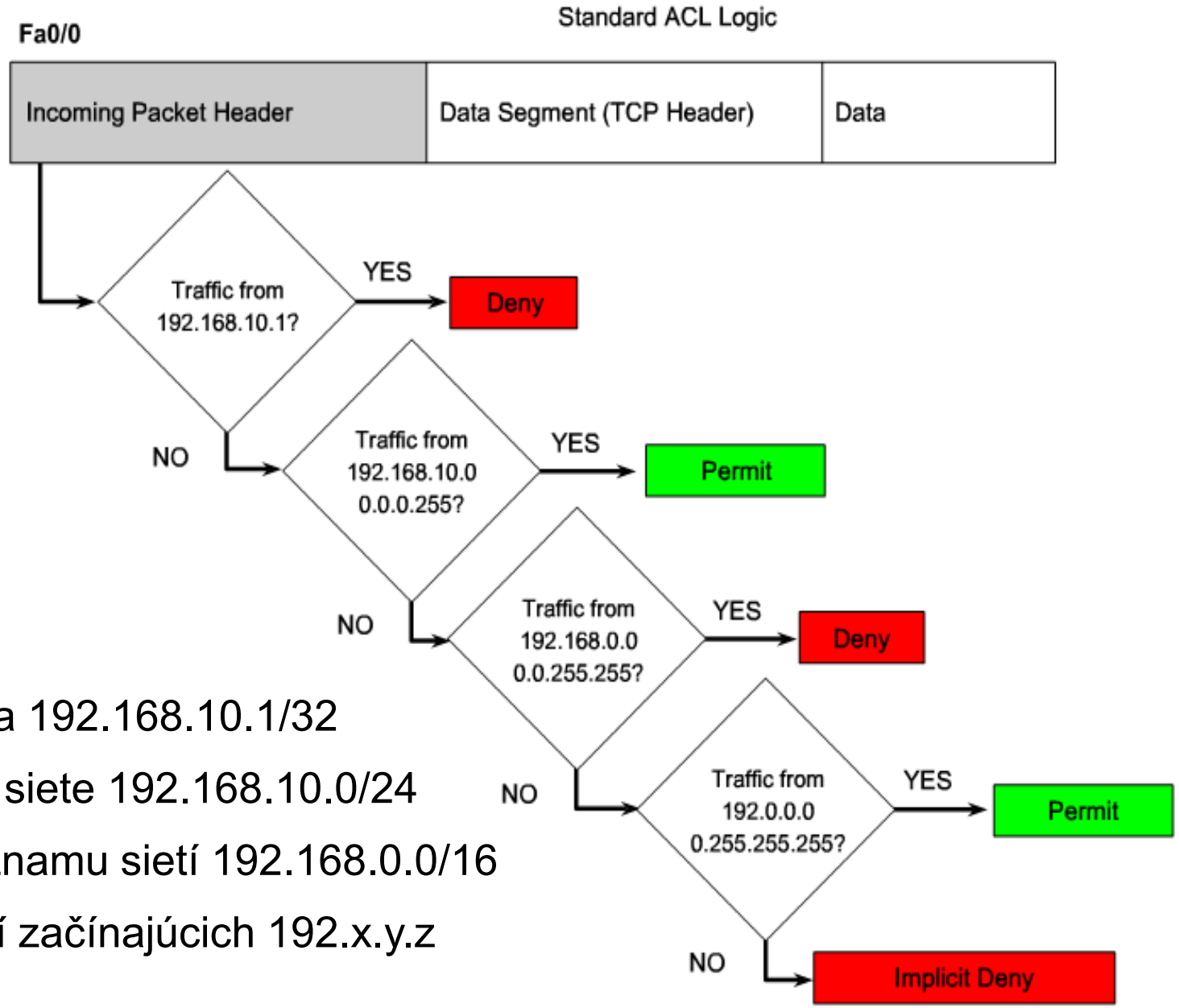
Čo najbližšie k **zdroju !**



Idea ACL

Podmienky ACL

- sekvenčné
- first match
- implicitné deny any



Príklad:

Podm.1: zakáž prevádzku z hosta 192.168.10.1/32

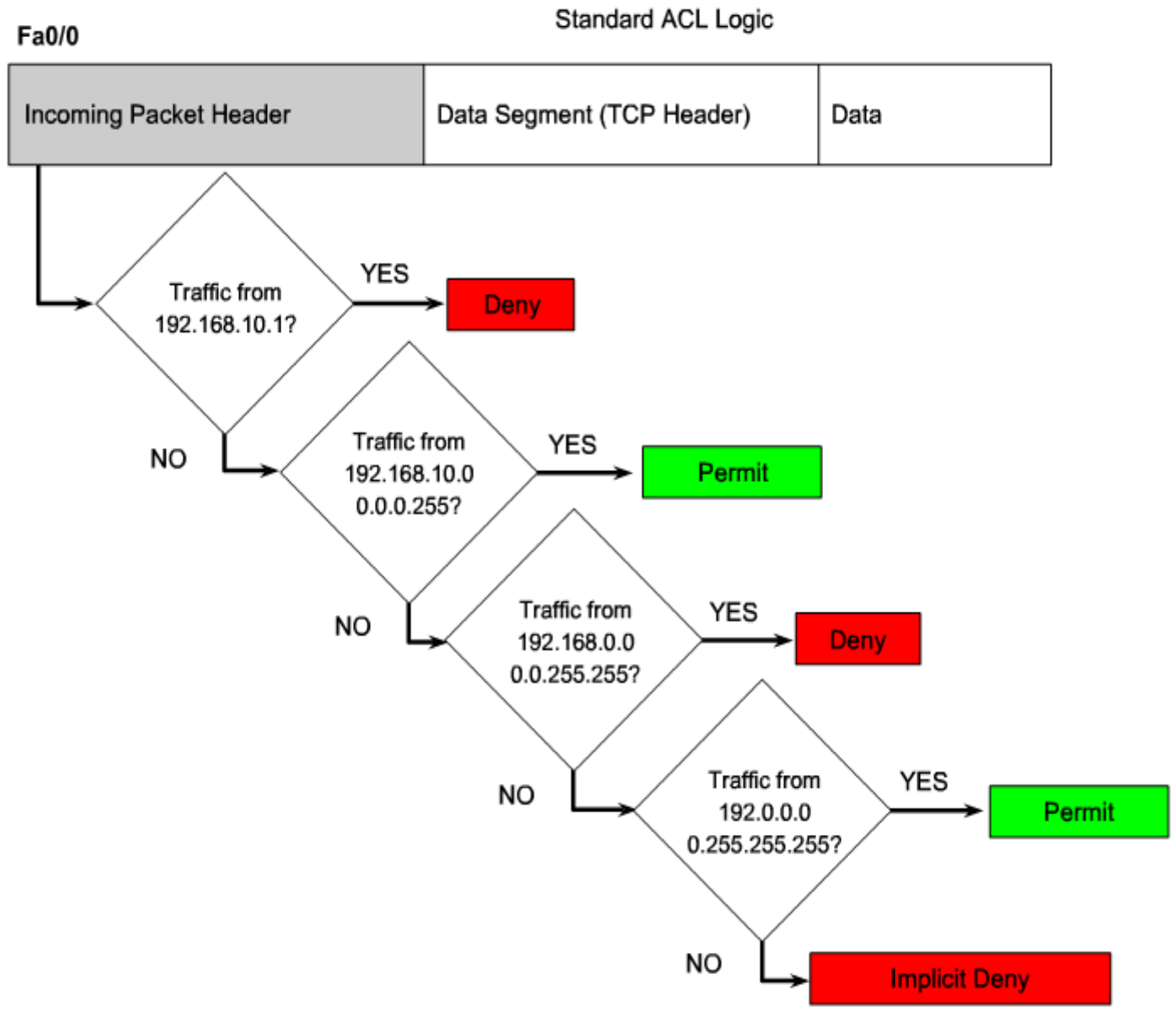
Podm.2: povoľ prevádzku z celej siete 192.168.10.0/24

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

- 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

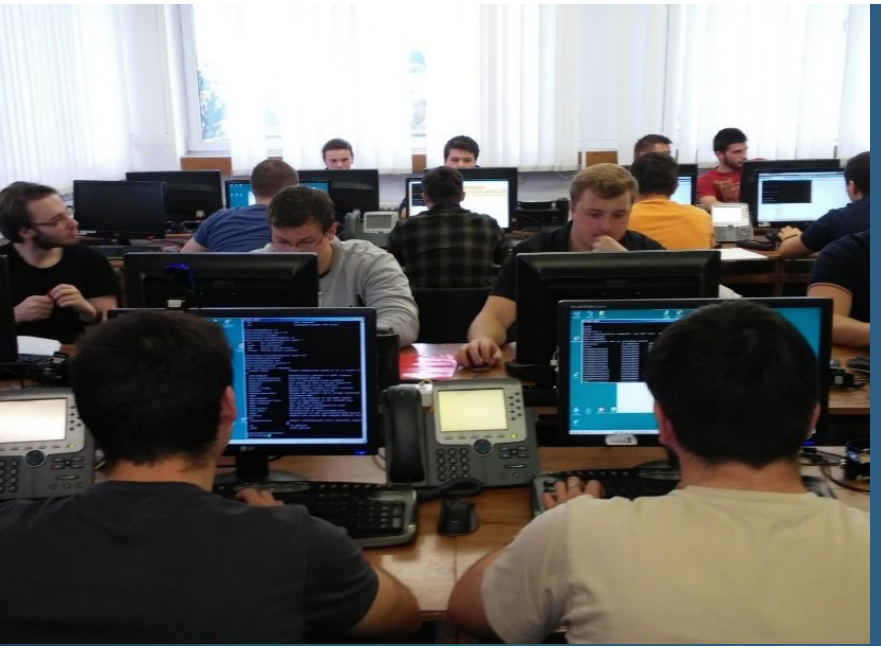


Pri ACL ber do úvahy

- 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

Odporúčanie	Výhoda
ACLs postaviť na základe bezpečnostnej **politiky** organizácie	Budeme si istý, že implementujeme bezpečnostné pravidlá danej **organizácie**
Pripraviť **popis**, čo chceme dosiahnuť danými ACL	Predídeme nechceným **problémom** s konektivitou a prístupom
Použiť **textový editor** na tvorbu, editovanie a ukladanie ACLs	Budujeme si tak **knižnicu** znovu použiteľných ACLs
Testovať ACLs v testovacom labe/sieti pred samotným nasadením v produkčnej sieti	Predídeme tak **chybám**, ktoré môžu viesť k finančnej strate



Konfigurácia číslovaného štandardného ACL

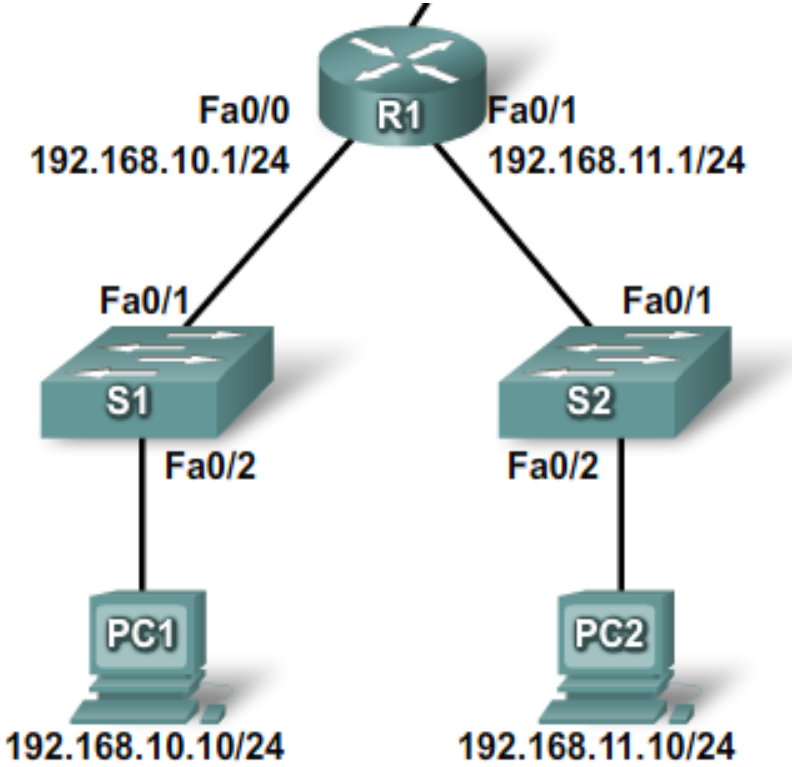
Konfigurácia standard ACL

```
R(config)# access-list CISLO [deny|permit|remark] TEST_PODMIENKA [WILDCARD] [log]
```

- ***CISLO:*** číslo ACL do ktorého vkladám novú podmienku alebo poznámku
  - ACL môže mať veľa podmienok, ich príslušnosť k danému ACL je uvedená týmto číslom
- **Deny**: zakáž paket spĺňajúci podmienku
- **Permit**: povoľ paket spĺňajúci 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, a vypíše na konzolu správu pre prvý paket, a neskôr v 5 min intervale štatistiku, koľko bolo denied alebo permitted

Príklad jednoduchého ACL

- 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 isté inak:

```
Router(config)#access-list 2 remark Povol hostov 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
```

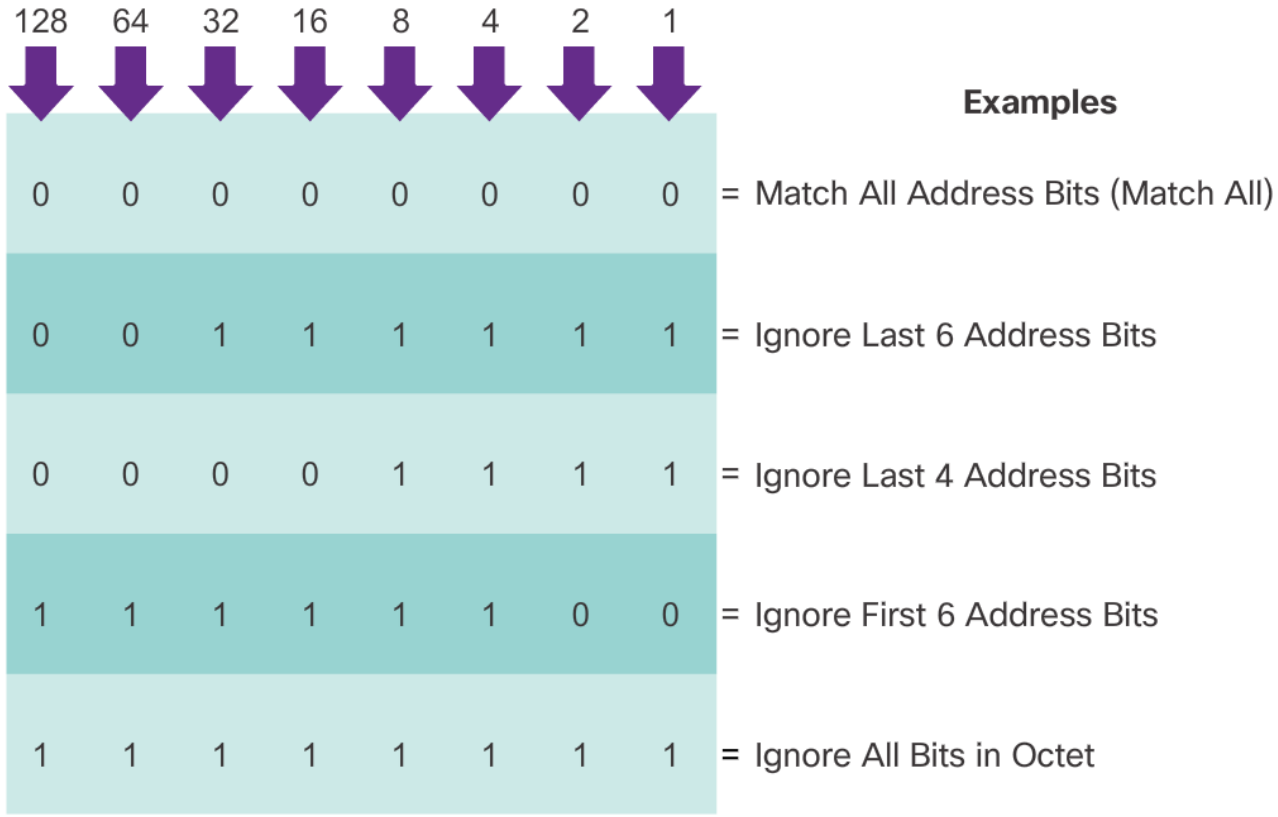
Defaultná podmienka
Netreba písať

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

0 = kontroluj **zhodu** odpovedajúcich bitov IP adresy a podmienky

1 = ignoruj **hodnotu** odpovedajúcich bitov IP adresy



Wildcard Mask - príklady

- Uvažujme sieť 172.16.10.0/24. Aká bude testovacia podmienka (TP) a aká wildcard maska (WM), ak chcem definovať:
  1. Prvú polovicu hostov z danej siete
  2. Druhú polovicu
  3. Iba párne IP adresy z danej siete
  4. Iba nepárne
  5. Presne 1 hosta
  6. Čokoľvek (akákoľvek IP)
- Ako zvoliť TP a WM, ak chcem definovať:
  1. Všetkých hostov zo subsietí 172.16.0.0/24 až 172.16.15.0
  2. Všetkých hostov zo subsietí 172.16.32.0/24 až 172.16.47.0

Uvažujme sieť 172.16.10.0/24. Aká bude testovacia podmienka (TP) a aká wildcard maska (WM), ak chcem definovať: **Prvú polovicu** hostov z danej siete

Uvažujme sieť 172.16.10.0/24. Aká bude testovacia podmienka (TP) a aká wildcard maska (WM), ak chcem definovať: Iba **párne/nepárne IP adresy** z danej siete

Uvažujme sieť 172.16.10.0/24. Aká bude testovacia podmienka (TP) a aká wildcard maska (WM), ak chcem definovať:

- presne **1 hosta**
- **čokoľvek** (akákoľvek IP)

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	

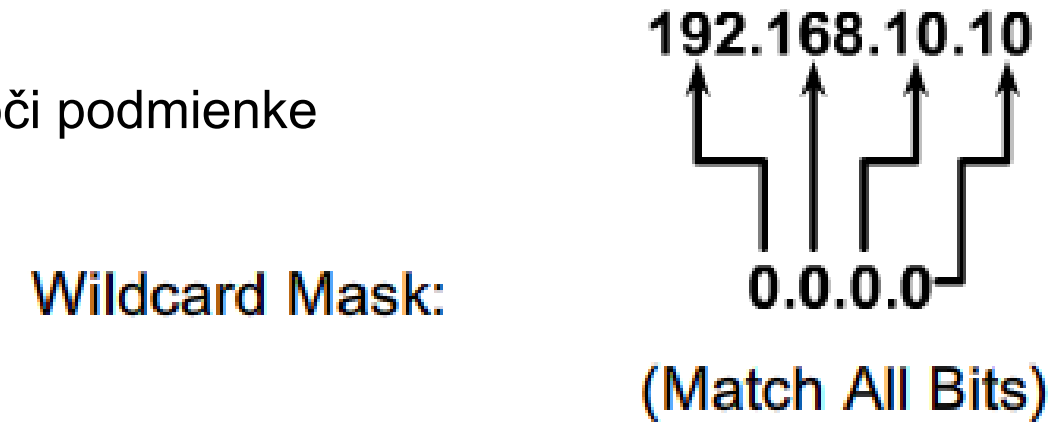
Počítanie WM masky môže byť zjednodušené odčítaním masky siete od 255.255.255.255.

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

Kľúčové slová „host“ a „any“

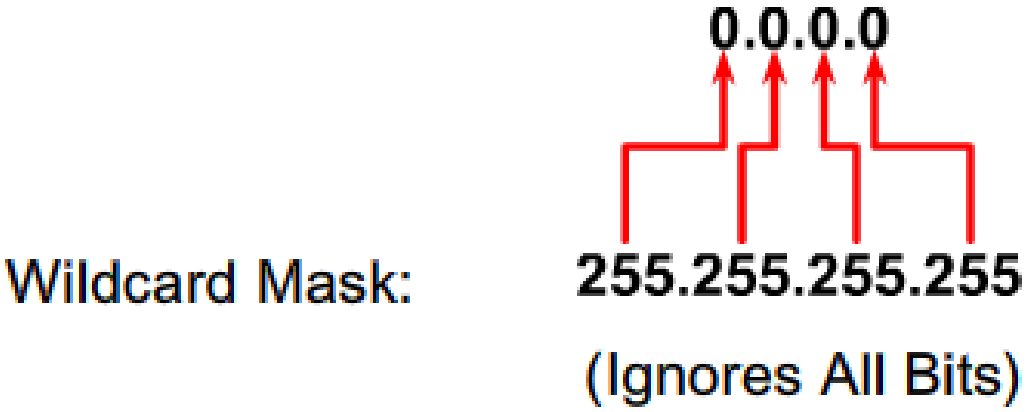
- Podmienka **192.168.10.10 0.0.0.0**
  - vyžaduje kontrolu všetkých 32 bitov adresy voči podmienke
  - zjednodušenie:

    host 192.168.10.10



- Podmienka **0.0.0.0 255.255.255.255**
  - ignoruje porovnávanie na všetkých bitoch
  - zjednodušenie:

    any



Kľúčové slová „host“ a „any“

- Príklad na host:

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

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

- Príklad na any:

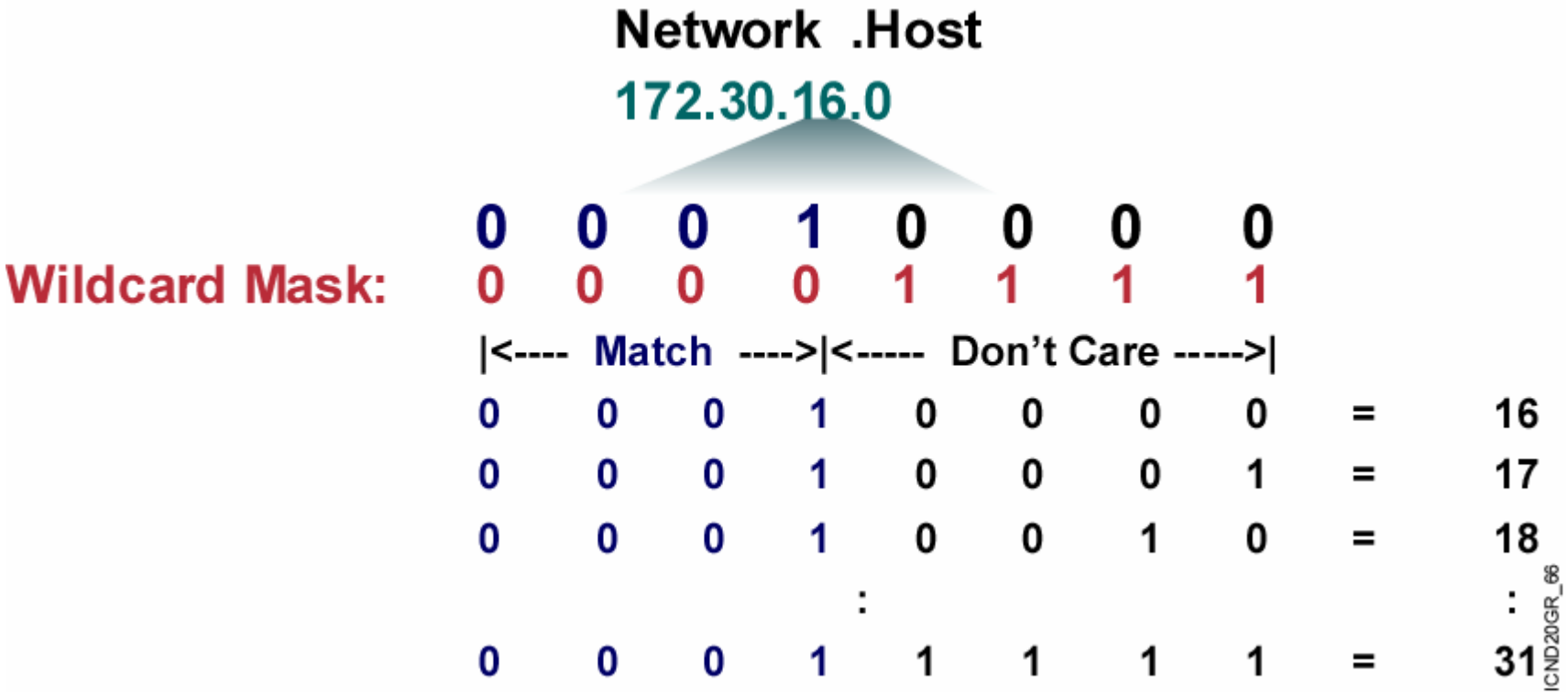
```
Router(config)#access-list 3 permit|deny 0.0.0.0 255.255.255.255

! Alebo – ako zdrojová IP môže byť uvedené úplne čokoľvek:
Router(config)#access-list 3 permit|deny X.X.X.X 255.255.255.255

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

Ako vyčleniť subnet

Príklad:

Kontroluj zoznam sietí od 172.30.16.0/24 do 172.30.31.0/24.

- ACL testovacia podmienka (adresa): 172.30.16.0
- Wildcard mask: 0.0.15.255

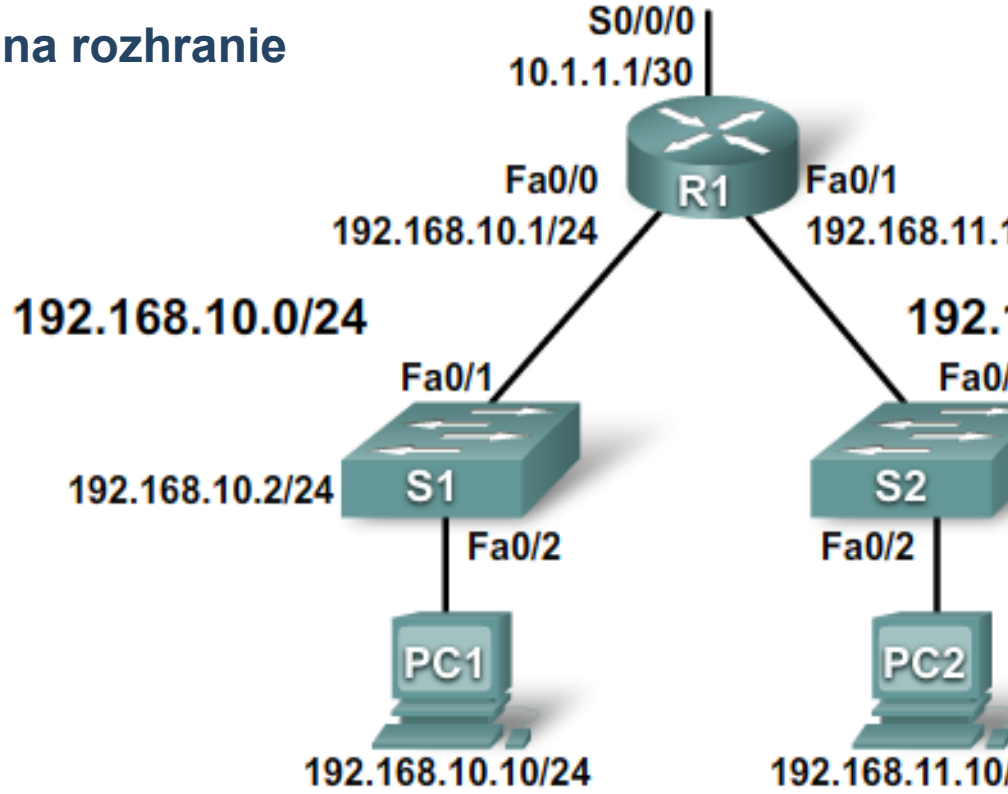


Priradenie ACL na rozhranie

```
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

Príklad 1

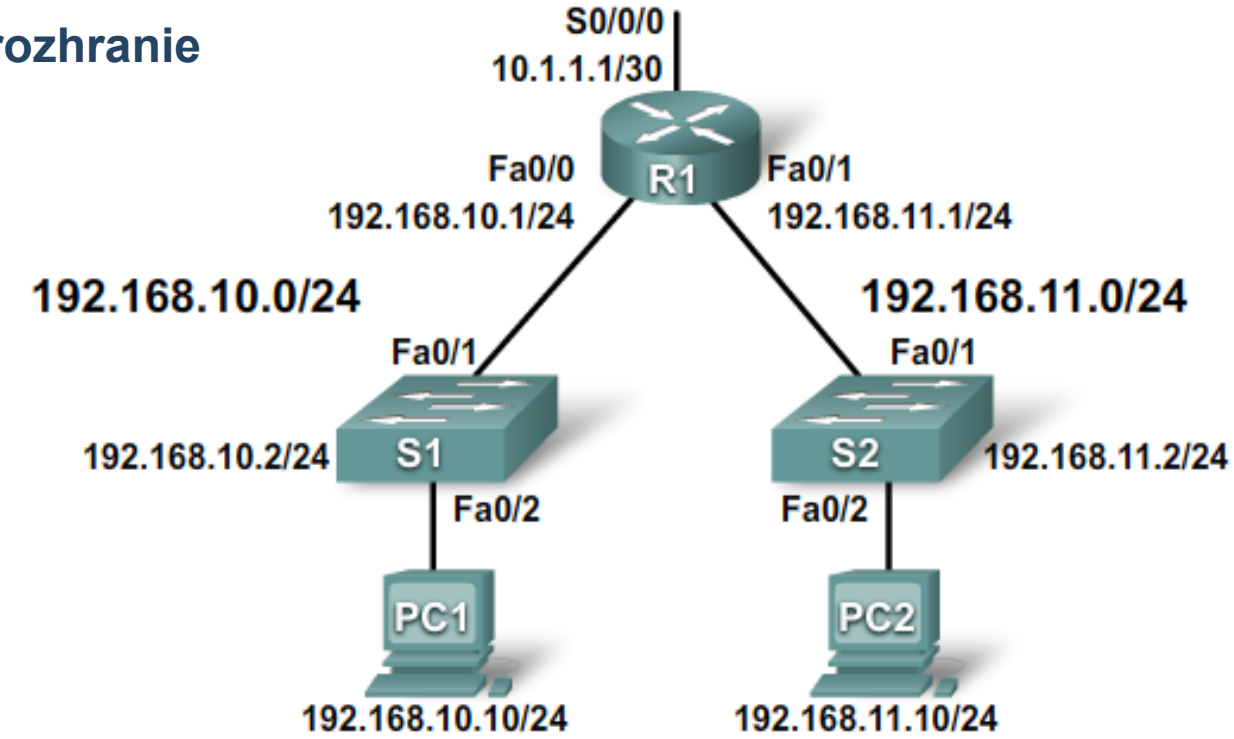



```
! Vytvorenie ACL
R1(config)#access-list 2 permit 192.168.10.0 0.0.0.255
! Priradenie ACL
R1(config)#interface fa 0/1
R1(config-if)#ip access-group 2 out
```

Čo robí ACL?

Pozor na default **deny any** na konci

Príklad 2a

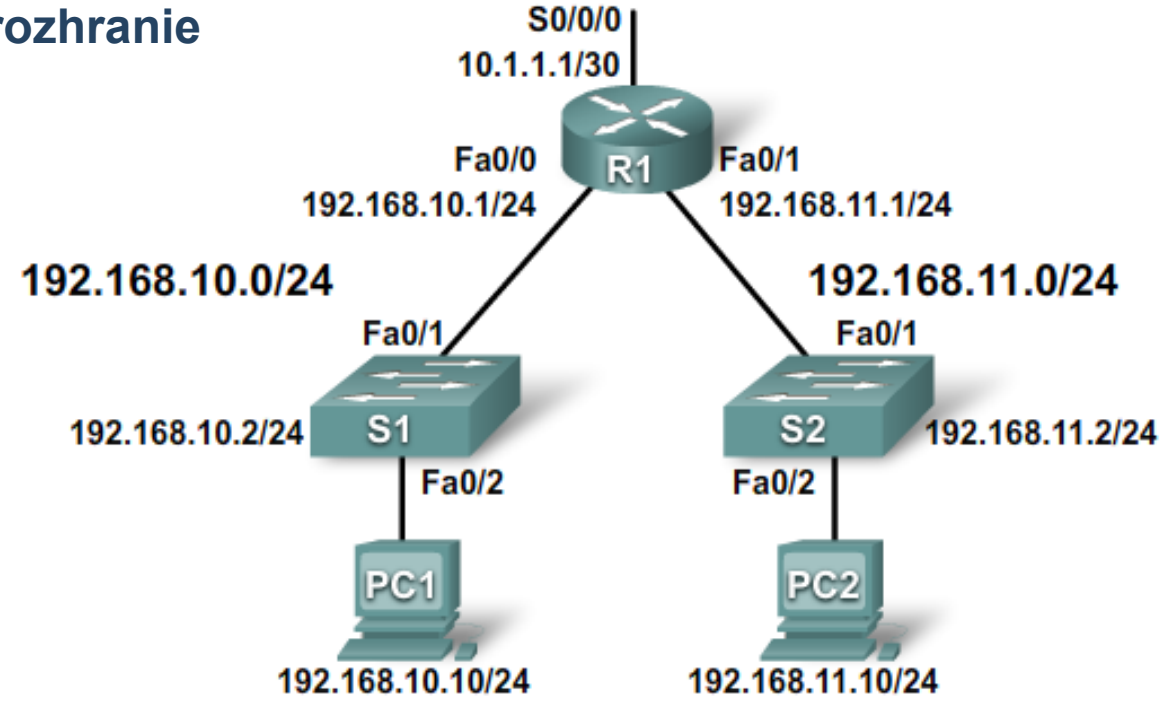



```
! Vytvorenie ACL
R1(config)#access-list 2 deny host 192.168.10.10
R1(config)#access-list 2 permit 192.168.10.0 0.0.0.255
! Priradenie ACL
R1(config)#interface s 0/0/0
R1(config-if)#ip access-group 2 out
```

Čo robí ACL?

Pozor na default **deny any** na konci

Príklad 2b

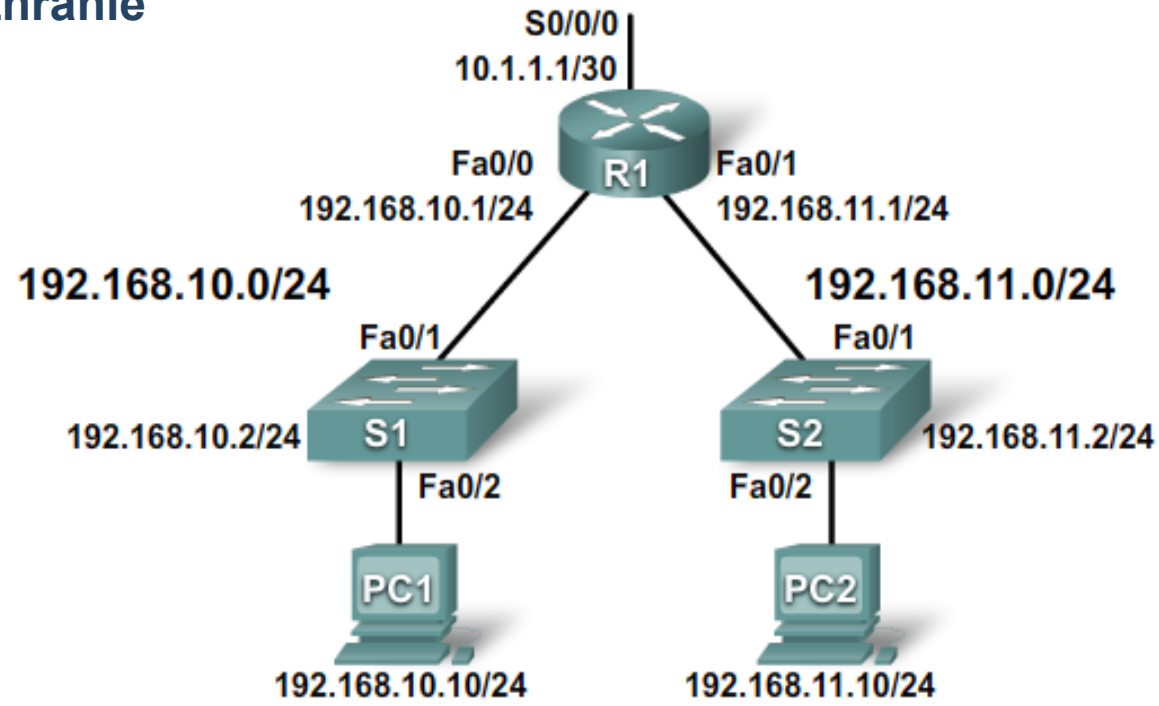



```
! Vytvorenie ACL
R1(config)#access-list 2 permit 192.168.10.0 0.0.0.255
R1(config)#access-list 2 deny host 192.168.10.10
! Priradenie ACL
R1(config)#interface s 0/0/0
R1(config-if)#ip access-group 2 out
```

Čo robí ACL?

Čo sa stalo, keď sme vymenili poradie podmienok?

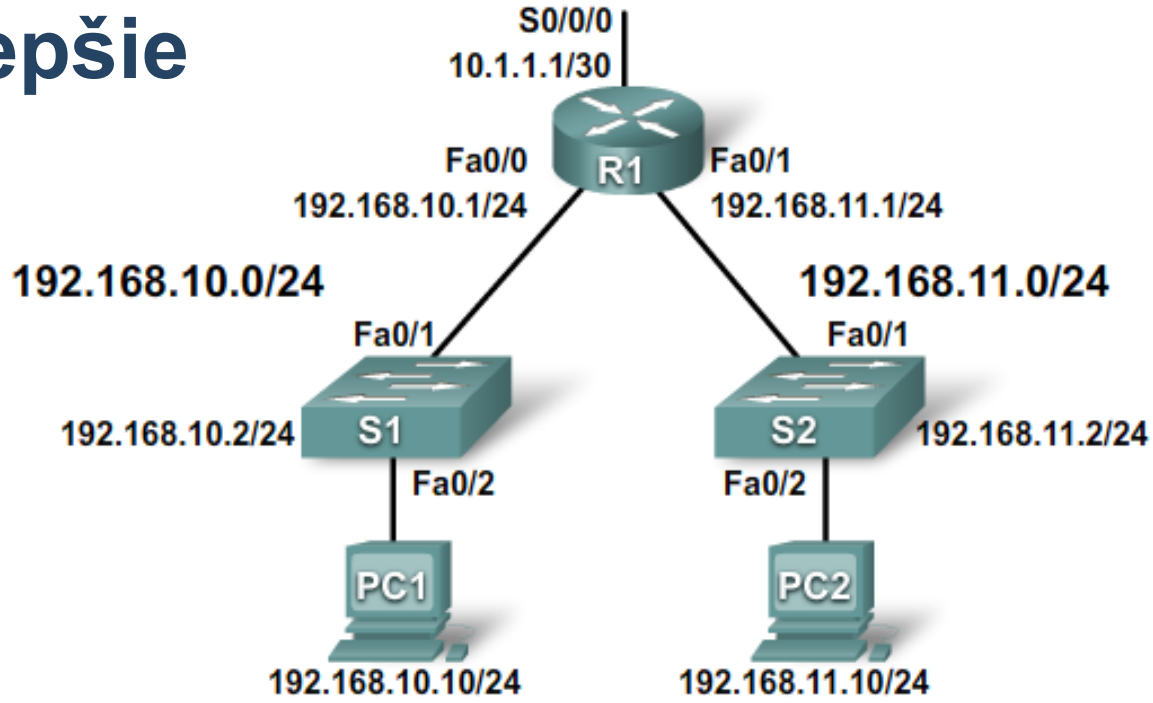
Príklad 3




```
! Vytvorenie ACL
R1(config)#access-list 2 deny host 192.168.10.10
R1(config)#access-list 2 permit 192.168.0.0 0.0.255.255
R1(config)#access-list 2 permit any
! Priradenie ACL
R1(config)#interface s 0/0/0
R1(config-if)#ip access-group 2 out
```

Čo robí ACL?

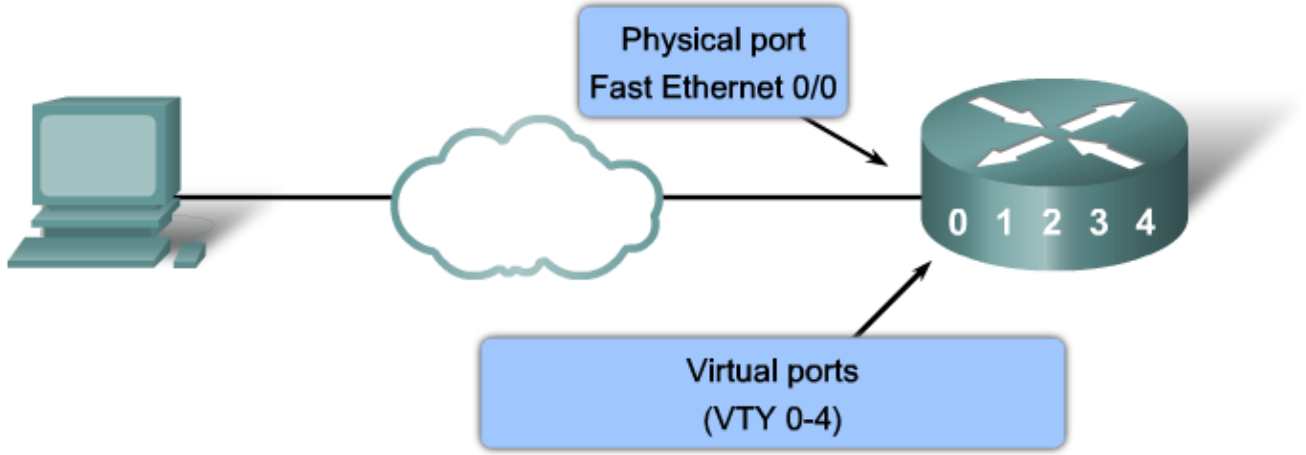
Príklad 3 – lepšie




```
! Vytvorenie ACL
R1(config)#access-list 2 deny host 192.168.10.10
R1(config)#access-list 2 permit any
! Priradenie ACL
R1(config)#interface s 0/0/0
R1(config-if)#ip access-group 2 out
```

Kontrola prístupu na VTY cez ACL

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




```
! Vytvorenie ACL
R1(config)#access-list 21 permit host 192.168.10.10
R1(config)#access-list 21 permit host 158.193.152.108

! Priradenie ACL na vty line
R1(config)#line vty 0 4
R1(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 deny host 192.168.10.10
access-list 23 permit 192.168.10.0 0.0.0.255
```

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

Použitie poznámok v ACL

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

Editovanie standard ACL – novšie IOSy

- V novších IOS je možné doeditovať zmeny

```
R1(config)# access-list 1 deny host 192.168.10.99
R1(config)# access-list 1 permit 192.168.0.0 0.0.255.255
```

```
R1# show access-lists 1
Standard IP access list 1
    10 deny   192.168.10.99
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
R1#
```

```
R1# conf t
R1(config)# ip access-list standard 1
R1(config-std-nacl)# no 10
R1(config-std-nacl)# 10 deny host 192.168.10.10
R1(config-std-nacl)# end
R1#
```

```
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
R1#
```

Pozor na mazanie ACL !

```
! 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
! Priradenie ACL
Router(config)#interface s 0/0/0
Router(config-if)#ip access-group 2 out
Router(config-if)#end
Router#.... {nieco konfigurujem}
Router#conf t

! Zmazanie ACL
Router(config)#no access-list 2
```

- Čo sa stane?
- Ako to spraviť lepšie?
  - Najprv zrušiť ACL z daného rozhrania
  - Až potom ho zmazať

Konfigurácia pomenovaného štandardného ACL

Konfigurácia pomenovaného standard ACL

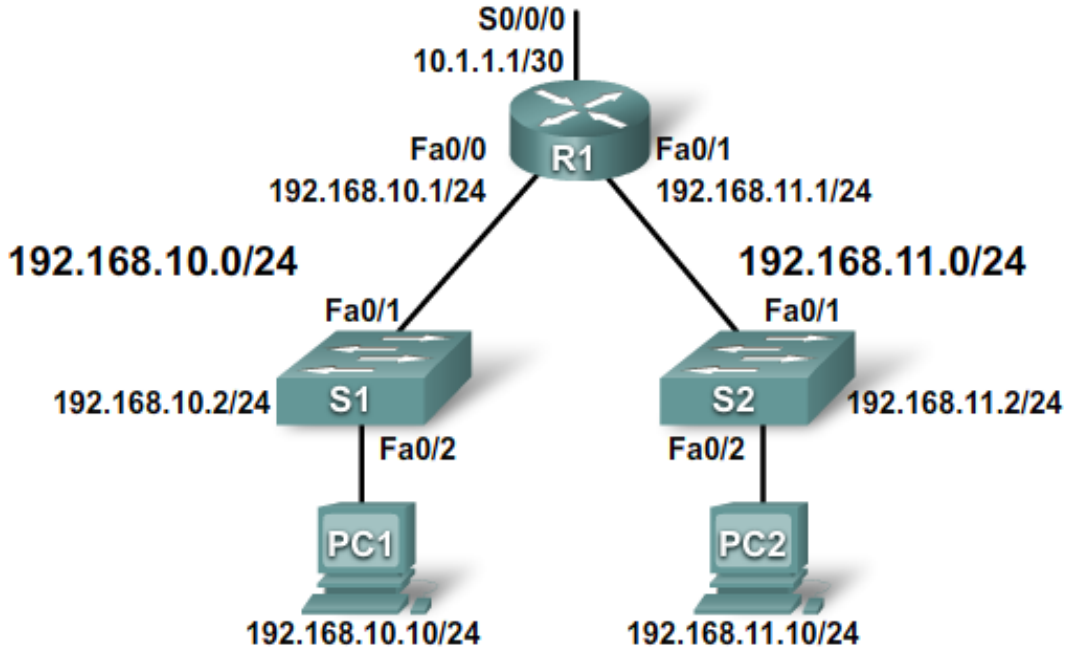
- Výhoda pomenovaných ACL
  - Jednoduchšia identifikácia
  - V možnosti ich neskoršej editácie
    - Pridávanie podmienok aj na iné miesto ako na koniec ACL
    - Zmena podmienok

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

R1(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 zmaze podmienku

R1(config-std-nacl)#[permit | deny | remark ] TEST_CONDITION WM [log]
```

```
! Priradenie menneho ACL na rozhranie

R1(config-if)#ip access-group NAME [in | out]
```

Príklad




```
! 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
Router(config-std-nacl)# remark Povol vsetko ostatne
Router(config-std-nacl)# permit any
! Priradenie ACL
Router(config)#interface s 0/0/0
Router(config-if)#ip access-group MOJ-ACL out
```

Overenie ACL

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

```
R1# show access-lists 1
Standard IP access list 1
    10 deny   192.168.10.99
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
```

```
R1# show access-lists
Standard IP access list NO_ACCESS
    10 deny   192.168.11.10
    15 deny   192.168.11.11
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
```

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

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

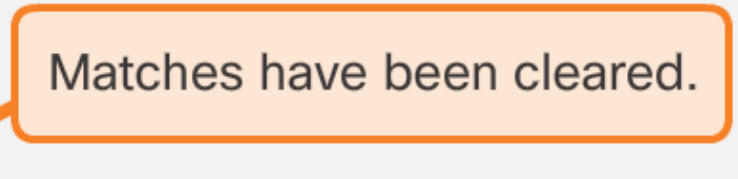
```
R1# show running-config | include access-list 1
access-list 1 deny host 192.168.10.99
access-list 1 permit 192.168.0.0 0.0.255.255
```

Post-editácia pomenovaného standard ACL

```
R1# show access-lists
Standard IP access list NO_ACCESS
    10 deny   192.168.11.10
    20 permit 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 NO_ACCESS
R1(config-std-nacl)# 15 deny host 192.168.11.11
R1(config-std-nacl)# end
R1# show access-lists
Standard IP access list NO_ACCESS
    10 deny   192.168.11.10
    15 deny   192.168.11.11
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
```

ACL štatistiky a resetovanie počítadiel

```
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10 (8 match(es))
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
Standard IP access list NO_ACCESS
    15 deny   192.168.11.11
    10 deny   192.168.11.10 (4 match(es))
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1# clear access-list counters 1
R1#
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
Standard IP access list NO_ACCESS
    15 deny   192.168.11.11
    10 deny   192.168.11.10 (4 match(es))
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
```

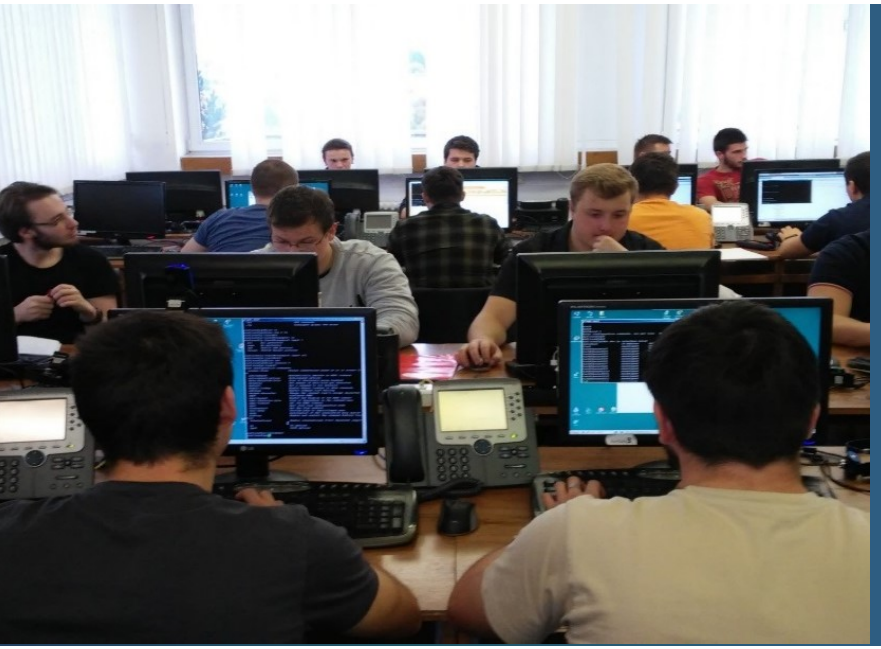


Sekvenčné čísla v standard ACLs

```
R1# show access-lists 1
Standard IP access list 1
 50 permit 10.0.0.2
 60 permit 10.0.0.3
 40 permit 10.0.0.1
 70 permit 10.0.0.4
 80 permit 10.0.0.5
 10 deny   192.168.10.0, wildcard bits 0.0.0.255
 20 deny   192.168.20.0, wildcard bits 0.0.0.255
 30 deny   192.168.30.0, wildcard bits 0.0.0.255
R1# copy running-config startup-config
R1# reload
R1# show access-lists 1
Standard IP access list 1
 10 permit 10.0.0.2
 20 permit 10.0.0.3
 30 permit 10.0.0.1
 40 permit 10.0.0.4
 50 permit 10.0.0.5
 60 deny   192.168.10.0, wildcard bits 0.0.0.255
 70 deny   192.168.20.0, wildcard bits 0.0.0.255
 80 deny   192.168.30.0, wildcard bits 0.0.0.255
```

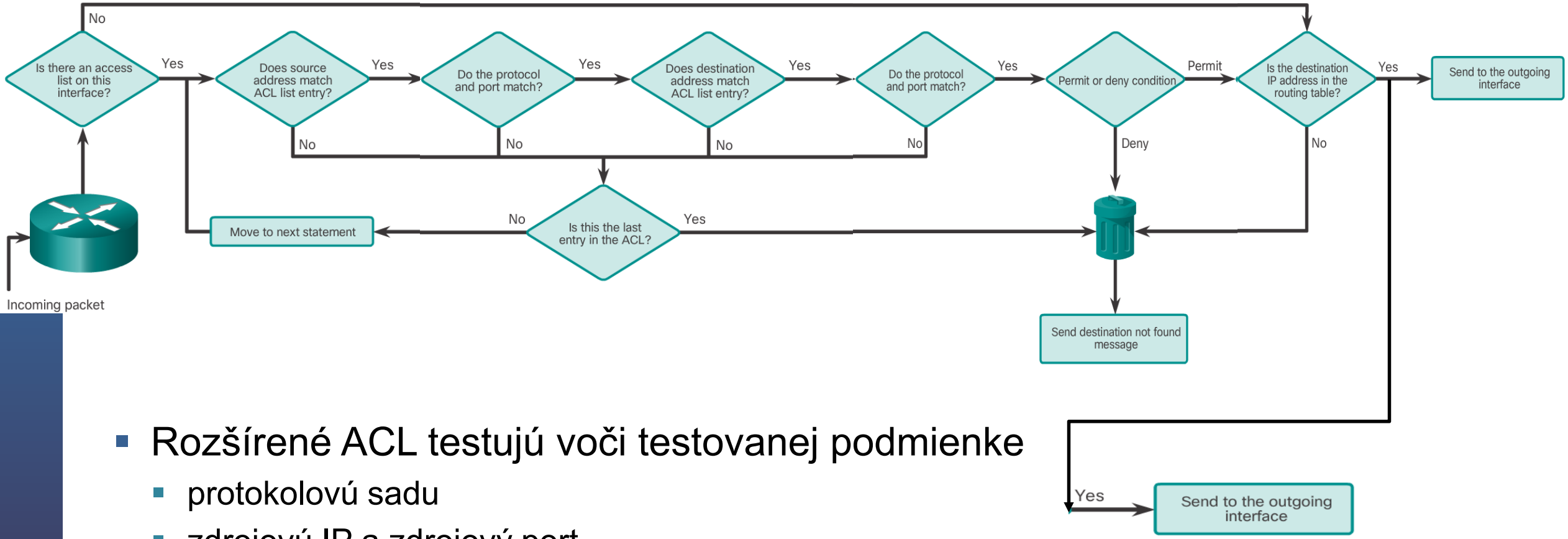
Host statements are listed first, in an order to be efficiently processed by the IOS.

Range statements are listed after host statements, in the order they were entered.



Konfigurácia rozšíreného ACL

Rozšírené (extended) ACL



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

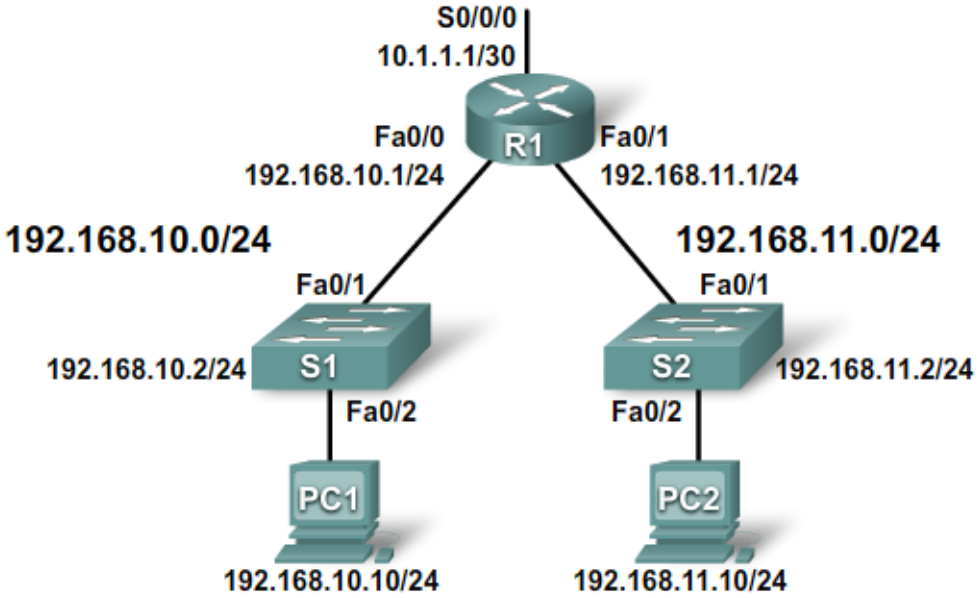
Konfigurácia extended ACL

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

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	Used to enter a remark or comment.
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 1

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




```
R1(config)#access-list 101 remark Povol HTTP
R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)#access-list 101 remark Povol HTTPS
R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 443
```

Aplikovanie na rozhranie, ktoré?

```
R1(config)#int s 0/0/0
R1(config-if)#ip access-group 101 out
```

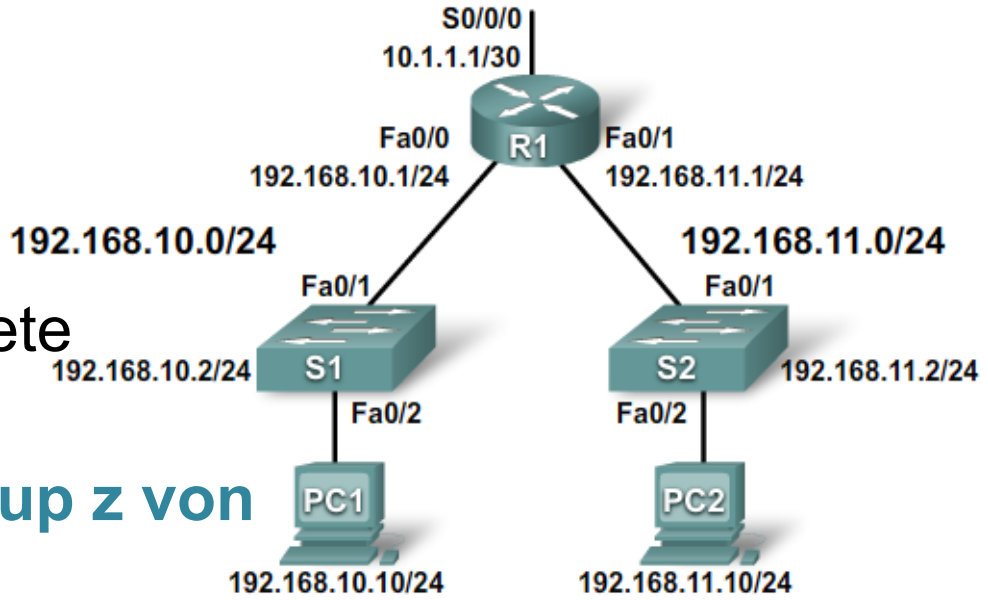
Áno, ale...

```
R1(config)#int fa0/0
R1(config-if)#ip access-group 101 in
```

Áno, ale...

...zadanie nie je celkom došpecifikované

Príklad 1 - došpecifikovanie



- 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**

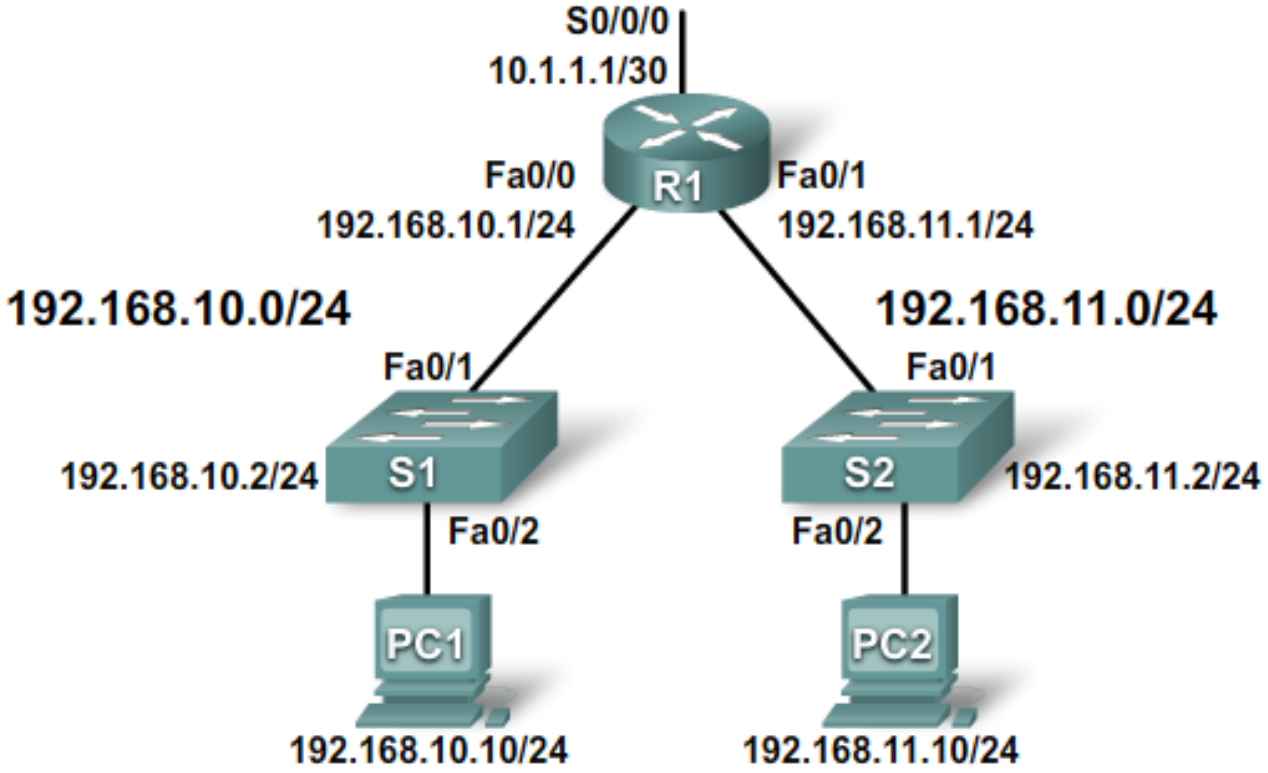
```
R1(config)#access-list 101 remark Povol HTTP
R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)#access-list 101 remark Povol HTTPS
R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)#int fa 0/0
R1(config-if)#ip access-group 101 in
```

Riešenie bodu 2?

```
! Iny ACL, ktory bude riesit vstup do siete
R1(config)#access-list 102 remark Povol len zalozene TCP spojenia
R1(config)#access-list 102 permit tcp any any established
R1(config)#int fa 0/0
R1(config-if)#ip access-group 102 out
```

Príklad 2

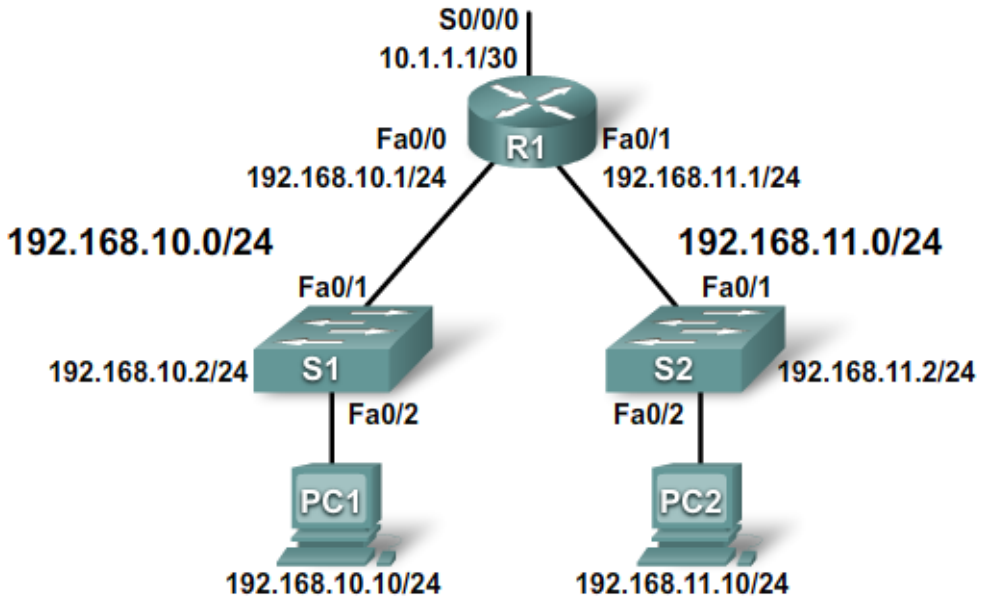
- Zakáž zo siete 192.168.11.0 telnet
- Povol všetko ostatné




```
! Zakaz zo siete 192.168.11.0 telnet a povol ostatne
R1(config)#access-list 104 deny tcp 192.168.11.0 0.0.0.255 any eq 23
R1(config)#access-list 104 permit ip any any
R1(config)#int fa 0/1
R1(config-if)#ip access-group 104 in
```

Príklad 3

- Zakáž zo siete 192.168.11.0 do siete 192.168.10.0 ftp
- Povol všetko ostatné




```
! Zakaz zo siete 192.168.11.0 ftp do siete 10.0 a povol ostatne

R1(config)#access-list 105 deny tcp 192.168.11.0 0.0.0.255
                                    192.168.10.0 0.0.0.255 eq 21

R1(config)#access-list 105 deny tcp 192.168.11.0 0.0.0.255
                                    192.168.10.0 0.0.0.255 eq 20

R1(config)#access-list 105 permit ip any any

R1(config)#int fa 0/1

R1(config-if)#ip access-group 105 in
```

Konfigurácia pomenovaného extended ACL

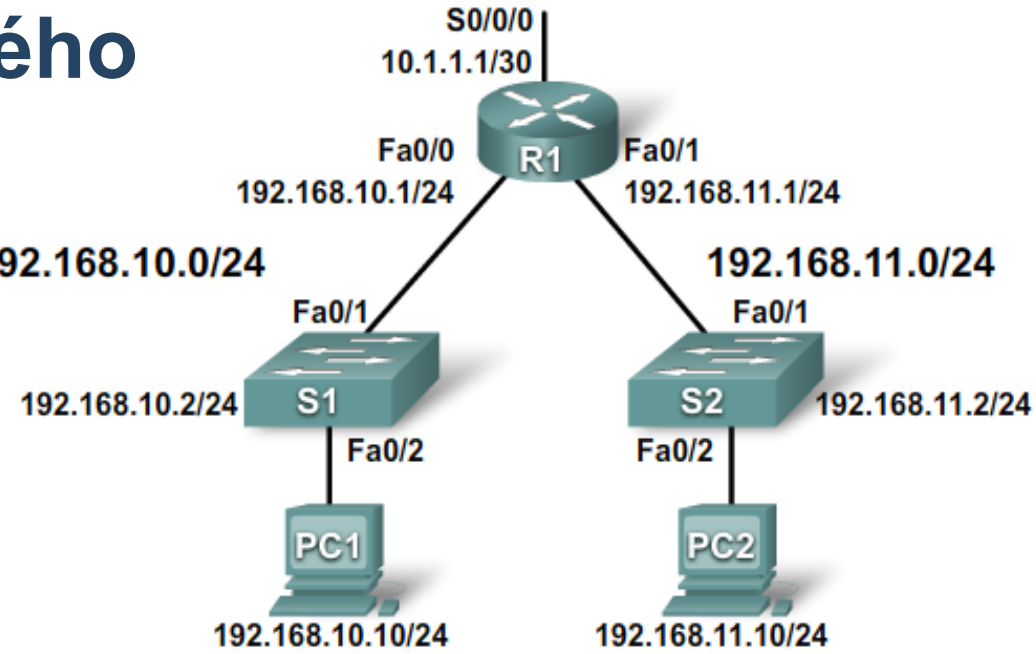
- Výhoda pomenovaných ACL
  - Jednoduchšia identifikácia
  - V možnosti ich neskoršej editácie (toto už dnes umožňuje každý novší IOS)
    - Pridávanie podmienok aj na iné miesto ako na koniec ACL
    - Zmena podmienok

```
! Vytvorenie menneho ACL
! Meno je alfa numericky retazec, ktory nesmie zacinat cislom
R1(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
R1(config-std-nacl)#[permit | deny | remark ] TEST_CONDITION WM [log]
```

```
! Priradenie menneho ACL na rozhranie
R1(config-if)#ip access-group NAME [in | out]
```

Konfigurácia pomenovaného extended ACL

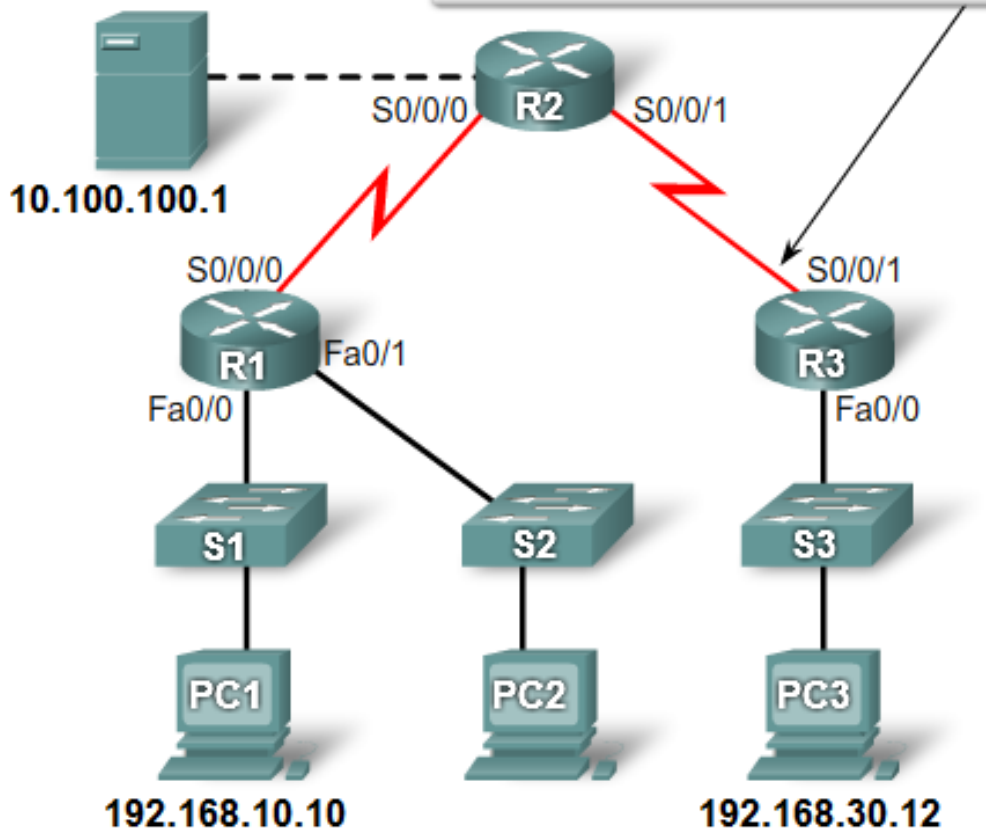



```
! Vytvorenie pomenovaneho ACL
R1(config)#ip access-list extended WEB-SERVICES-ONLY
R1(config-std-nacl)# remark Povol HTTP
R1(config-std-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-std-nacl)# remark Povol HTTPS
R1(config-std-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
! Priradenie ACL
R1(config)#int fa 0/0
R1(config-if)#ip access-group WEB-SERVICES-ONLY in
```

Diagnostika chýb pri nasadzovaní ACL

Diagnostika ACL – chyba 1

```
# 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:

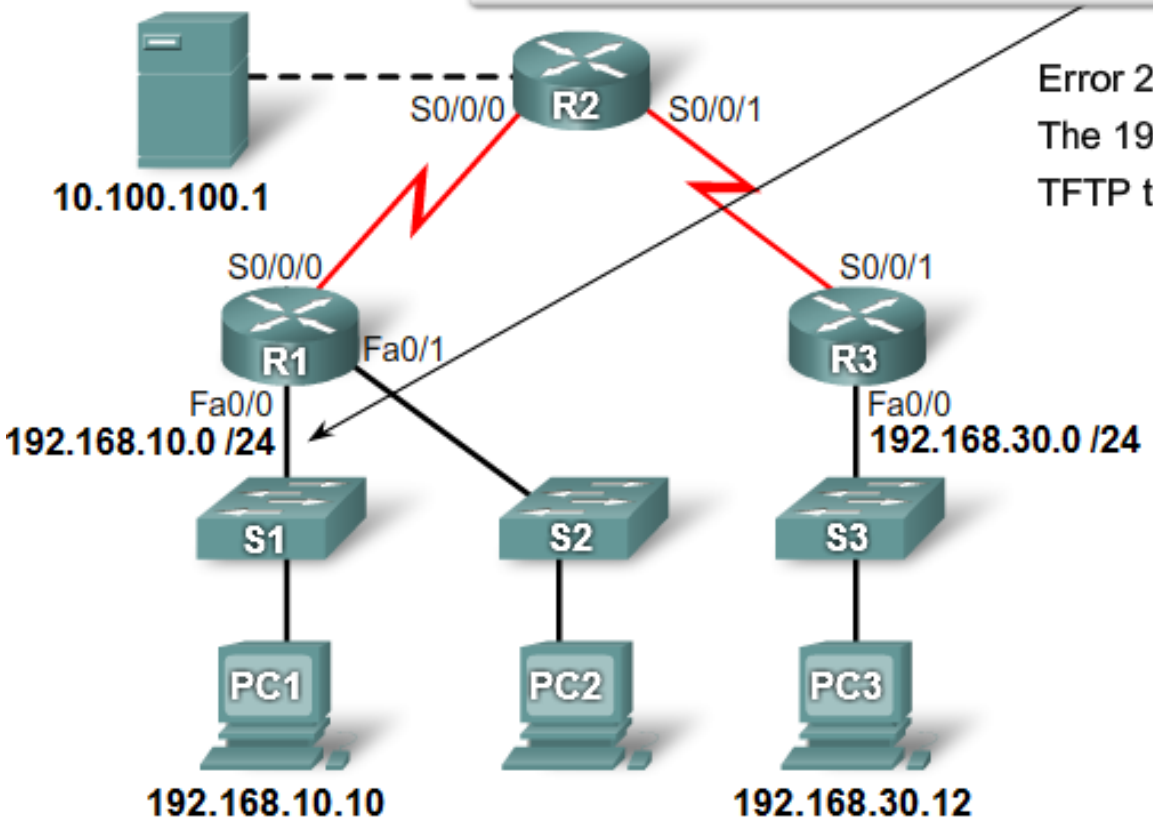
Host 192.168.10.10 has no connectivity with 192.168.30.12

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:

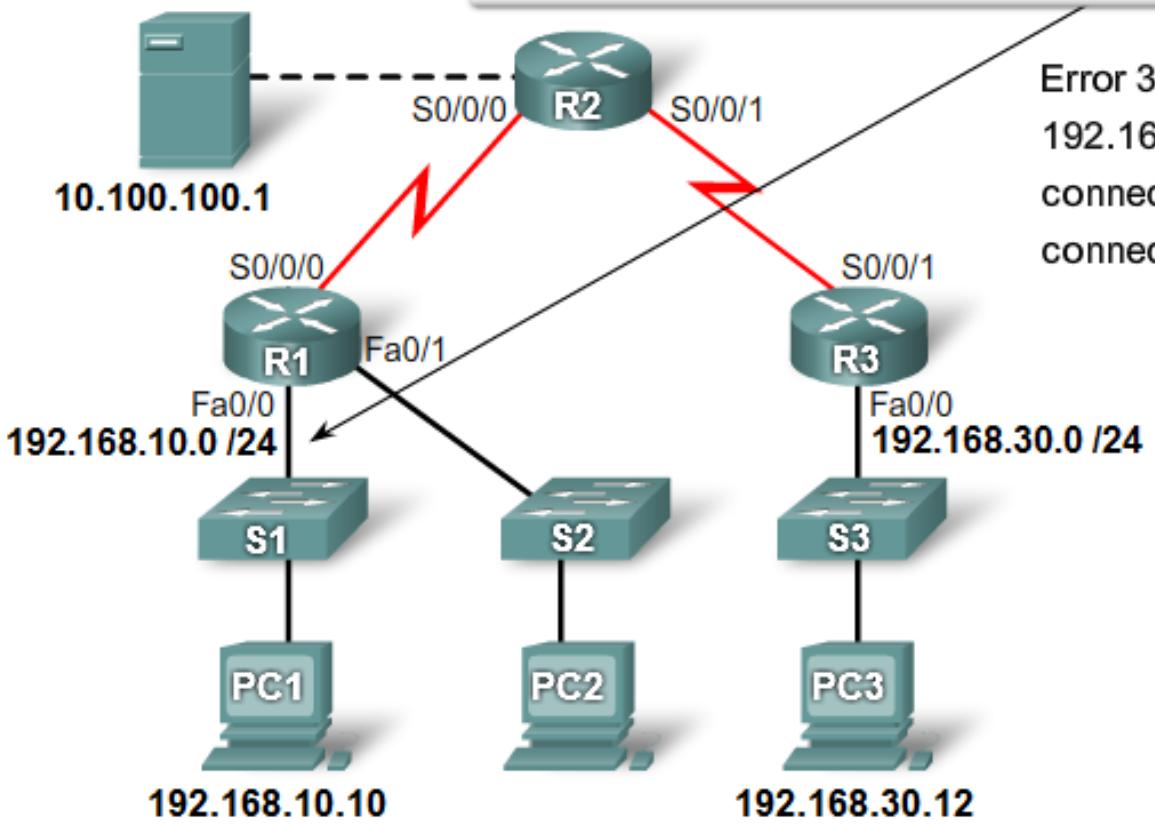
The 192.168.10.0 /24 network cannot use TFTP to connect to the 192.168.30.0 /24.

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 zadefinované, zakazujem source port a nie destination**

UNIVERSITY OF ŽILINA
Faculty of Management Science and Informatics

MINISTERSTVO ŠKOLSTVA, VEDY, VÝSKUMU A ŠPORTU SLOVENSKEJ REPUBLIKY

Ďakujem za pozornosť.

Obsahom boli **kapitoly 4 a 5** kurzu ENSA=Enterprise Networking, Security, and Automation (ccna3).

Na začiatku ďalšej prednášky bude IPv6 ACLs.

Doma pozorne preštudovať a spraviť si **kvízy** z týchto kapitol na Netacad-e.

Ostrý test bude na cvičení v nasledujúcom týždni – 1 otvorená otázka (bez výberu odpovede).

Spätnú väzbu na prednášku alebo cvičenie vyjadrite kedykoľvek anonymne sem.

Vytvorené v rámci projektu KEGA 026TUKE-4/2021.