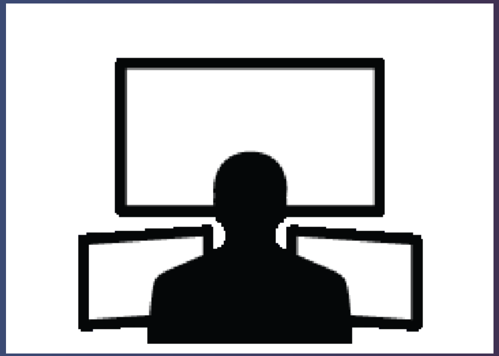
UNIVERSITY OF ŽILINA
Faculty of Management Science
and Informatics

Prednáška 3
Sieťová bezpečnostná infraštruktúra



Riešenie bezpečnostných incidentov
(CyberOps Associate v1.02)

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

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

Ktorý výsledok pokrýva táto prednáška

Výsledky vzdelávania

Študent po absolvovaní predmetu získa vedomosti a zručnosti potrebné na úspešné zvládnutie úloh, povinností a zodpovedností bezpečnostného analytika v operačnom centre bezpečnosti.
Študent po absolvovaní predmetu bude vedieť:

- Vysvetliť rolu analytika v rámci kybernetickej bezpečnosti
- Vysvetliť prostriedky operačného systému Windows
a Linux a charakteristiky pre podporu analýzy v rámci kybernetickej bezpečnosti
- Analyzovať operácie v rámci sieťových protokolov a služieb
- Vysvetliť operácie sieťovej infraštruktúry
- Klasifikovať rôzne typy sieťových útokov
- Použiť sieťové monitorovacie nástroje na identifikáciu útokov proti sieťovým protokolom a službám
- Použiť rôzne metódy na prevenciu škodlivého prístupu do počítačových sietí, k používateľom a k dátam
- Vysvetliť vplyvy kryptografie v rámci monitorovania bezpečnostných sietí
- Vysvetliť, ako skúmať zraniteľnosti a útoky koncových zariadení
- Identifikovať hlásenia v rámci sieťovej bezpečnosti
- Analyzovať sieťovú prevádzku na overenie potencionálneho zneužitia siete
- Aplikovať reakčné modely na incident, a získať prostriedky na manažovanie sieťových bezpečnostných incidentov

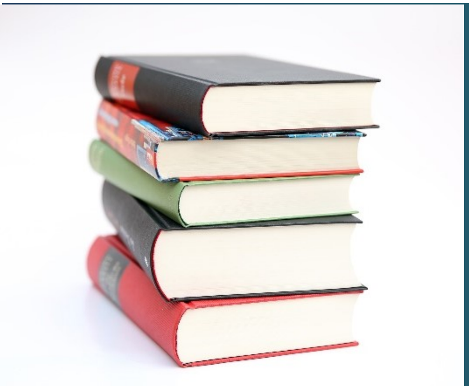
- Prerekvizity:
  - Princípy IKS, Počítačové siete 1, Úvod do OS

Preliminary version of topics for lectures

Planning

Week	CyberOps Modules in lectures	Exam from:
1	Chapter 1 The Danger Chapter 2 Fighters in the War Against Cybercrime Chapter 3: The Windows Operating System	none
2	Chapter 4: Linux Overview Chapter 5 Network Protocols Chapter 6 Ethernet and Internet Protocol (IP) Chapter 7 Connectivity Verification Chapter 8 Address Resolution Protocol Chapter 10 Network Services Chapter 11 Network Communication Devices	1-2
3	Chapter 9 The Transport Layer (+nmap) Chapter 12 Network Security Infrastructure	3-4
4	Chapter 13 Attackers and Their Tools Chapter 14 Common Threats and Attacks Chapter 15 Network Monitoring and Tools *(SIEM, SOAR)*	5-10

Week	CyberOps Modules in Lectures	Exam from:
5	Chapter 16 Attacking the Foundation *(L2, L3 protocols vulnerabilities and attacks)* Chapter 17 Attacking What We Do *(L7 vulnerabilities and attacks)*	11-12
6	Chapter 18 Understanding Defense *(security management)* Chapter 19 Access Control *(AAA)* Chapter 20 Threat Intelligence *(commercials, CVE database)*	13-17
7	Chapter 21 Cryptography Chapter 22 Endpoint Protection	18-20
8	Chapter 23 Endpoint Vulnerability Assessment Chapter 24 Technologies and Protocols	none
9	Chapter 25 Network Security Data Chapter 26 Evaualting Alerts (in Security Onion)	21-23
10	Chapter 27 Working with Network Security Data (Security Onion and ELK)	24-25
11	Chapter 28 Digital Forensics and Incident Analysis and Response	none
12	Expert talk (invited lecture)	26-28



Obsah dnešnej prednášky

- **Chapter 9 The Transport Layer**
  - Refresh znalostí o TCP protokole (slajdy 5 – 47 len ako opakovanie)
  - Ako vznikajú duplikáty TCP segmentov, a ako vie pomôcť SACK
  - Skenovanie siete s nmap, hping3 a massscan
- **Chapter 12 Network Security Infrastructure**
  - Network topologies
  - Network security devices
  - Network security services
  - Disaster recovery, bussiness continuity, RPO, RTO

Modul 9
TCP

Module Objective: Explain how transport layer protocols support network functionality.

Topic Title	Topic Objective
Transport Layer Characteristics	Explain how transport layer protocols support network communication.
Transport Layer Session Establishment	Explain how the transport layer establishes communication sessions.
Transport Layer Reliability	Explain how the transport layer establishes reliable communications.

Úlohy transportnej vrstvy v TCP/IP

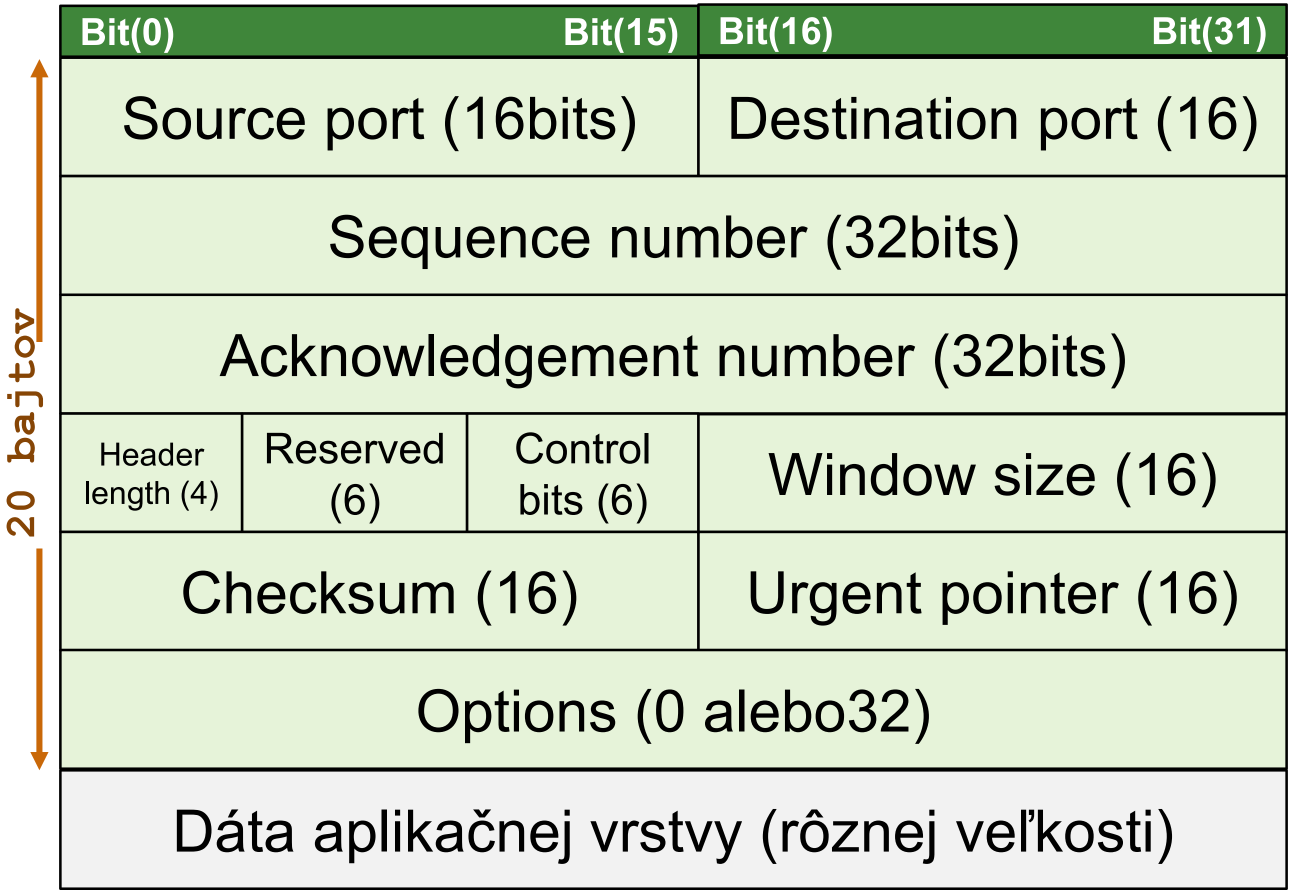
..základné funkcie

- **Oddelenie konverzácií**
  (Track individual conversations)
- **Segmentácia dát**
  (Segment data)
- **Spätná rekonštrukcia** pôvodných dát zo segmentov
  (Reassemble segments)
- **Identifikácia komunikujúcich aplikácií**
  (Identify the applications)
- **Spojovanosť**
  (Connection-oriented data stream support)
- **Spoľahlivosť**
  (Reliability)
  - **Usporiadanosť**
    (Delivery ordering)
- **Riadenie toku dát**
  (Flow control)

..rozširujúce funkcie
závisí od aplikácie, či ich potrebuje

Polia TCP hlavičky

- **Source Port**
  - Zdrojový TCP port
- **Destination Port**
  - Cieľový TCP port
- **Sequence Number**
  - Poradové číslo odoslaného segmentu
- **Acknowledgement Number**
  - Potvrdenie prijatých segmentov (vyjadrené v B)
- **Header Length**
  - Veľkosť hlavičky v 4B slovách
- **Reserved**
  - Rezervované pole, nepoužité bity, vždy s hodnotou 0

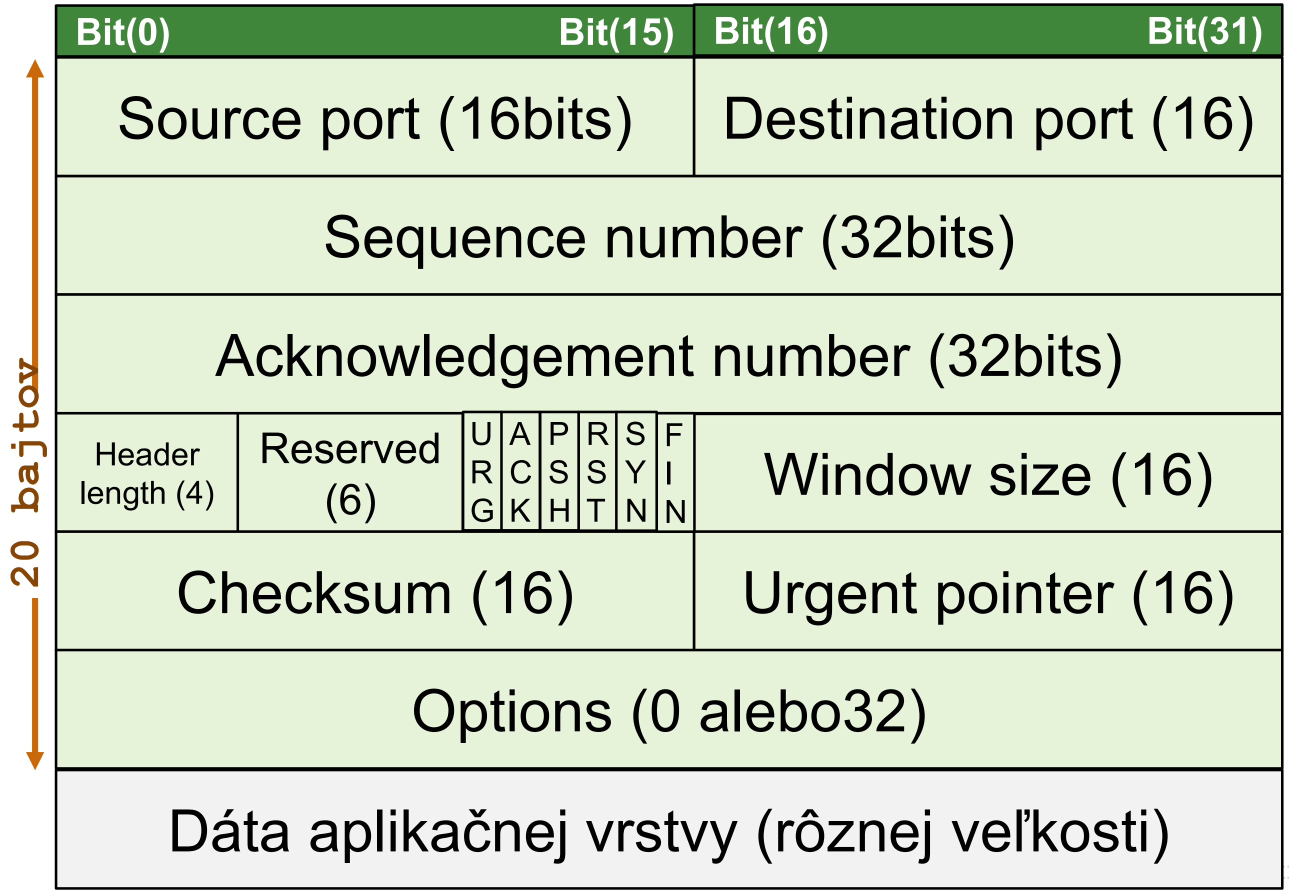


Polia TCP hlavičky

- **Window size**
  - Oznamovaná veľkosť okna pre príjemcu tohto segmentu (neposielaj mi segmenty rýchlejšie)
- **TCP Checksum**
  - Kontrolná suma celého TCP segmentu
- **Urgent Pointer**
  - Ukazateľ na posledný bajt urgentných dát
- **Options**
  - Doplňuj. voľby, nepovinné
- **Control bits**
  - 6 jednobitových príznakov (Flags)
    - URG, ACK,PSH, RST, SYN, FIN

- **ACK = acknowledgement**
  Segment potvrdzuje prijaté dáta
- **SYN = synchronization**
  Synchronizačná značka pri vytváraní spojenia
- **RST = reset**
  Reset spojenia (odmietnutie alebo neočakavané ukončenie)

- PSH = push
  Pošli dáta hneď, nečakaj na naplnenie MSS
- URG = urgent
  Segment obsahuje urgentné dáta
- FIN = finish
  Nemám viac dát, končím.



Hlavičky

- **Každý segment TCP aj UDP obsahuje samostatnú hlavičku, ktorá nesie informácie potrebné pre správne spracovanie segmentu u príjemcu**
- **Oba majú 2 adresné polia a kontrolný súčet (checksum)**

TCP (20 bajtov)

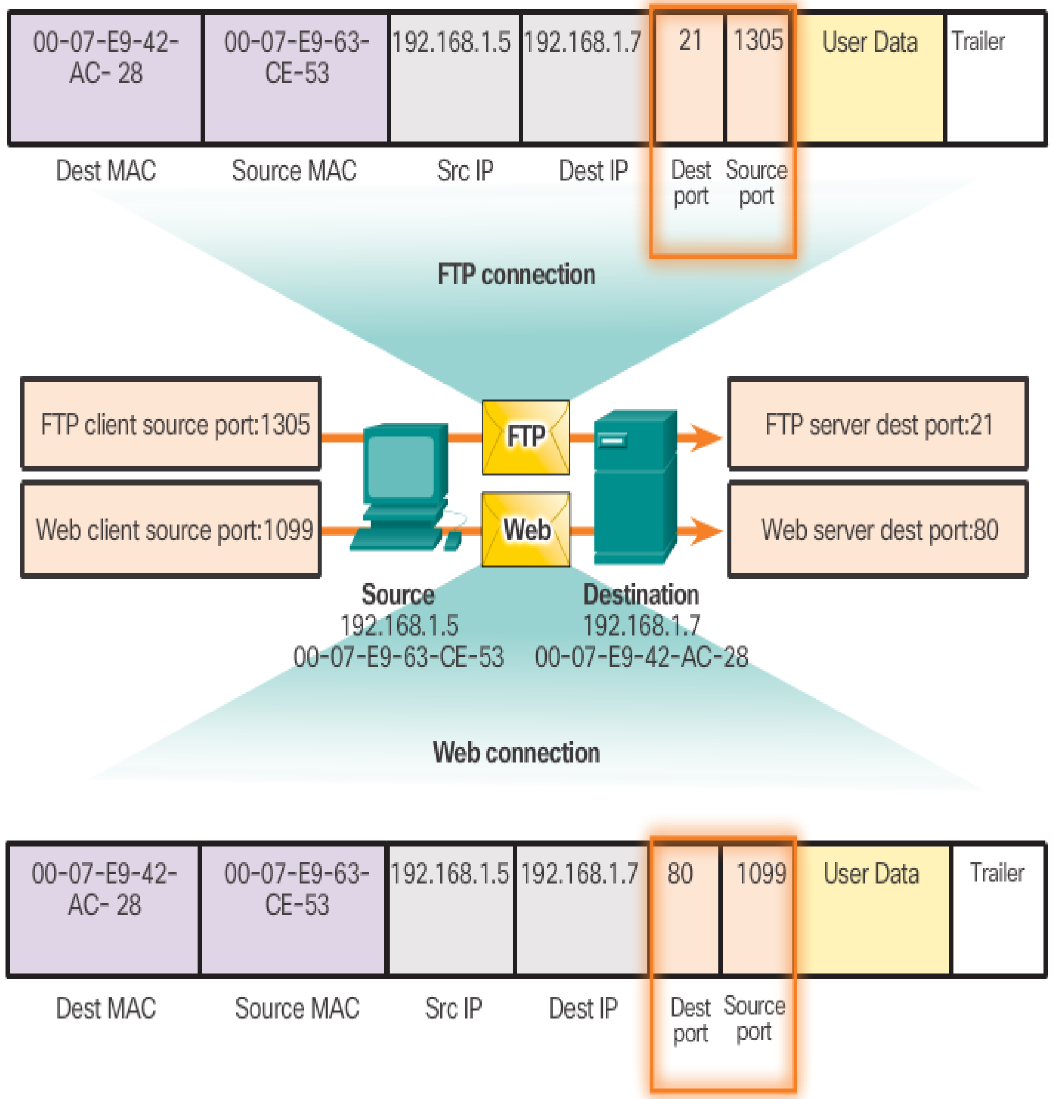
Bit(0) … Bit(15)			Bit(16) … Bit(31)
Source port (16bits)			Destination port (16)
Sequence number (32bits)			
Acknowledgement number (32bits)			
Header length (4)	Reserved (6)	Control bits (6)	Window (16)
Checksum (16)			Urgent pointer (16)
Options (0 alebo32)			
Dáta aplikačnej vrstvy (rôznej veľkosti)			

UDP (8 bajtov)

Bit(0) … Bit(15)	Bit(16) … Bit(31)
Source port (16bits)	Destination port (16)
Length (16)	Checksum (16)
Dáta aplikačnej vrstvy (rôznej veľkosti)	

Socket

- Kombinácia IP adresy uzla, transportného protokolu a portu sa nazýva **socket** a uvádza sa v tvare **IPadresa:port**
  - t.j. zdrojová IP adresa:zdrojový port je jeden socket
    - zdrojový port slúži ako spätná adresa pre odpoveď klientovi
  - cieľová adresa:cieľový port je druhý socket
    - používa sa na identifikáciu servera a služby, ktorú požaduje daný klient.
- Dvojica socketov **jednoznačne** identifikuje pár komunikujúcich procesov
  - 192.168.1.5:1099 192.168.1.7:80
  - Sockety umožňujú odlíšiť od seba viaceré procesy bežiace na klientovi a viaceré spojenia na danom serveri.
- Je úlohou transportnej vrstvy udržiavať zoznam aktívnych socketov.



Skupiny čísiel transportných portov

- Čísla portov v TCP a UDP sú 2-bajtové a ich rozsah (2^{16}=65536) je rozdelený do **3 skupín**
- Porty z 1. a 2. skupiny prideľuje **IANA** (Internet Assigned Numbers Authority)
- Porty z 3. skupiny prideľuje konkrétny operačný systém pre aplikáciu/službu, ktorá o to žiada:
  - Buď dynamicky, ak proces nežiada žiadne špecifické číslo portu (DYNAMIC)
  - Alebo ak proces požiada o nejaké špecifické číslo, pridelí to (PRIVATE)

Skupiny

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

Well-Known Port Numbers

Port	Protocol	Application	Acronym
20	TCP	File Transfer Protocol (data)	FTP
21	TCP	File Transfer Protocol (control)	FTP
22	TCP	Secure Shell	SSH
23	TCP	Telnet	–
25	TCP	Simple Mail Transfer Protocol	SMTP
53	UDP, TCP	Domain Name Service	DNS
67	UDP	Dynamic Host Configuration Protocol (server)	DHCP
68	UDP	Dynamic Host Configuration Protocol (client)	DHCP
69	UDP	Trivial File Transfer Protocol	TFTP
80	TCP	Hypertext Transfer Protocol	HTTP
110	TCP	Post Office Protocol version 3	POP3
143	TCP	Internet Message Access Protocol	IMAP
161	UDP	Simple Network Management Protocol	SNMP
443	TCP	Hypertext Transfer Protocol Secure	HTTPS

Verifikácia otvorených socketov

- Pre zistenie otvorených socketov a spojení na nich je možné použiť v OS Windows i Linux príkaz `netstat`
- Nevysvetliteľné TCP spojenia môžu indikovať bezpečnostnú hrozbu.
- Defaultne sa utilita **`netstat`** snaží preložiť IP adresu na doménové meno a číslo portu na „well-known“ aplikáciu.
  - Prepínačom **-n** si možno zobraziť zoznam IP adries a čísiel portov v numeric. form.

```
C:\> netstat

Active Connections

Proto   Local Address   Foreign Address          State
TCP     kenpc:3126      192.168.0.2:netbios-ssn  ESTABLISHED
TCP     kenpc:3158      207.138.126.152:http     ESTABLISHED
TCP     kenpc:3159      207.138.126.169:http     ESTABLISHED
TCP     kenpc:3160      207.138.126.169:http     ESTABLISHED
TCP     kenpc:3161      sc.msn.com:http          ESTABLISHED
TCP     kenpc:3166      www.cisco.com:http       ESTABLISHED
```

Parametre príkazu netstat

Príkazový riadok

```
NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-x] [-t] [interval]

  -a            Displays all connections and listening ports.
  -b            Displays the executable involved in creating each connection or
                listening port. In some cases well-known executables host
                multiple independent components, and in these cases the
                sequence of components involved in creating the connection
                or listening port is displayed. In this case the executable
                name is in [] at the bottom, on top is the component it called,
                and so forth until TCP/IP was reached. Note that this option
                can be time-consuming and will fail unless you have sufficient
                permissions.
  -e            Displays Ethernet statistics. This may be combined with the -s
                option.
  -f            Displays Fully Qualified Domain Names (FQDN) for foreign
                addresses.
  -n            Displays addresses and port numbers in numerical form.
  -o            Displays the owning process ID associated with each connection.
  -p proto      Shows connections for the protocol specified by proto; proto
                may be any of: TCP, UDP, TCPv6, or UDPv6.  If used with the -s
                option to display per-protocol statistics, proto may be any of:
                IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
```

Parametre príkazu netstat

Príkazový riadok

```
C:\Users\janau>netstat -h

Displays protocol statistics and current TCP/IP network connections.

NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-x] [-t] [interval]
```

Príkazový riadok

```
  -q            Displays all connections, listening ports, and bound
                nonlistening TCP ports. Bound nonlistening ports may or may not
                be associated with an active connection.
  -r            Displays the routing table.
  -s            Displays per-protocol statistics.  By default, statistics are
                shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6;
                the -p option may be used to specify a subset of the default.
  -t            Displays the current connection offload state.
  -x            Displays NetworkDirect connections, listeners, and shared
                endpoints.
  -y            Displays the TCP connection template for all connections.
                Cannot be combined with the other options.
  interval      Redisplays selected statistics, pausing interval seconds
                between each display.  Press CTRL+C to stop redisplaying
                statistics.  If omitted, netstat will print the current
                configuration information once.
```

Parametre príkazu netstat

Príkazový riadok

```
C:\Users\janau>netstat -s

IPv4 Statistics

  Packets Received                   = 3842688
  Received Header Errors             = 0
  Received Address Errors            = 211
  Datagrams Forwarded                = 0
  Unknown Protocols Received         = 0
```

. . . . . (skrátený výpis) . . . . .

```
UDP Statistics for IPv4

  Datagrams Received    = 503223
  No Ports              = 12270
  Receive Errors        = 188
  Datagrams Sent        = 343244
```

```
TCP Statistics for IPv4

  Active Opens                        = 16364
  Passive Opens                       = 1041
  Failed Connection Attempts          = 850
  Reset Connections                   = 1147
  Current Connections                 = 20
  Segments Received                   = 3505199
  Segments Sent                       = 940146
  Segments Retransmitted              = 51410

TCP Statistics for IPv6

  Active Opens                        = 582
```

. . . . . (skrátený výpis) . . . . .

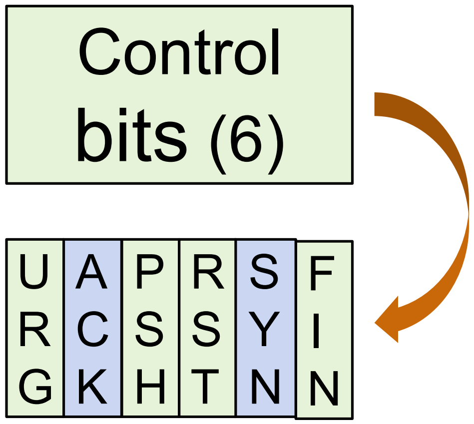
Procesy TCP servera

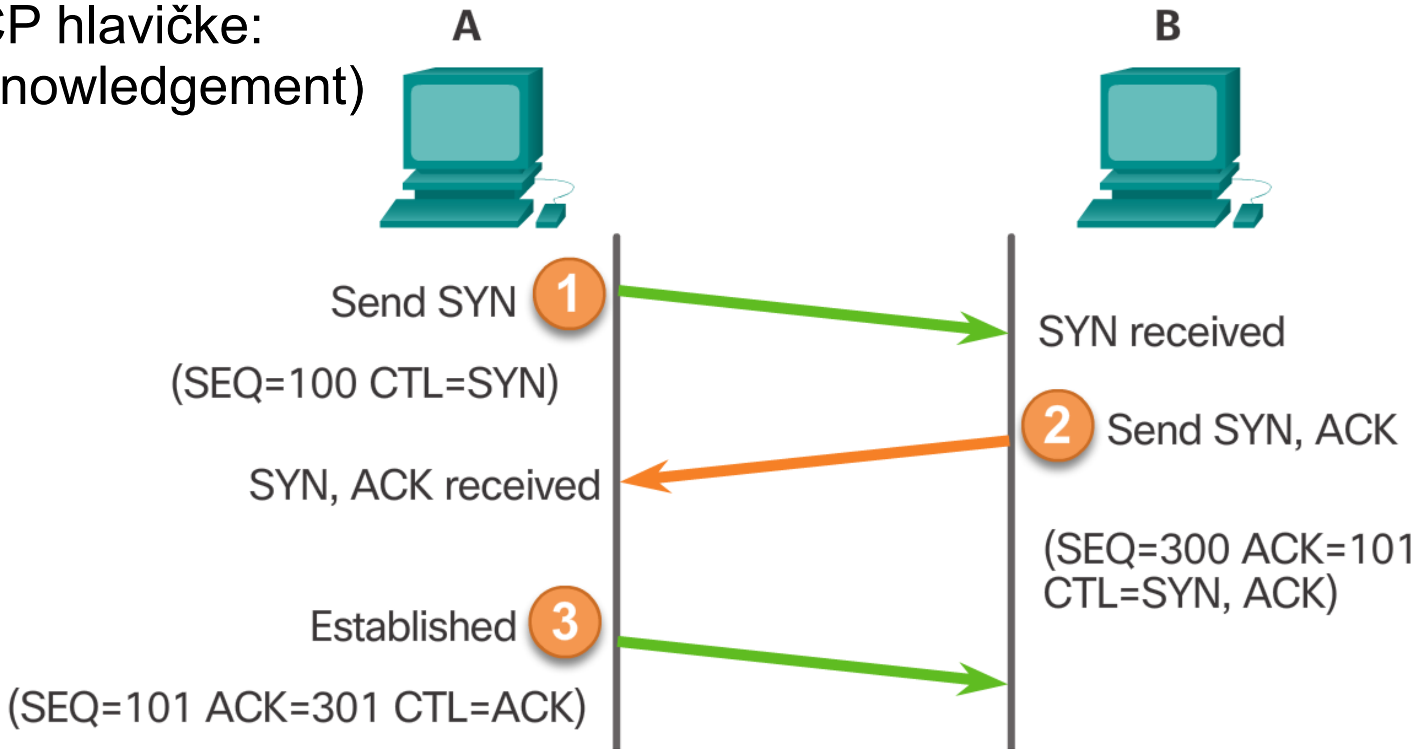
- Každý aplikačný proces bežiaci na serveri používa číslo portu.
- Jeden server nemôže mať dve služby priradené tomu istému číslu portu vrámci jednej služby transportnej vrstvy (TCP, UDP).
- Aktívne aplikácie daného servera priradené špecifickému portu sa považujú za **otvorené** (open).
- Akákoľvek klientská požiadavka adresovaná na otvorený port je akceptovaná a spracovaná aplikáciou servera, ktorá je zviazaná s daným portom.
- Na jednom serveri môže byť viacero otvorených portov, jeden pre každú aktívnu aplikáciu na serveri.

Otvorenie TCP spojenia

- Pred výmenou dát v TCP je nutné zostaviť spojenie
  - Zostavením spojenia sa komunikujúce strany navzájom dohodnú na poradových číslach, počnúc ktorými budú číslovať svoje segmenty
  - Až po tejto sekvencii môže začať prenos užitočných dát
  - Využívajú sa na to 2 príznaky v TCP hlavičke: SYN (synchronization) a ACK (acknowledgement)
    - SYN: „*Svoje segmenty budem číslovať počnúc touto hodnotou*“

      Pozor, toto je len príznak (0 alebo 1), konkrétnu hodnotu uvedie v poli SEQ (sequence number)
    - ACK: „*Potvrdzujem prijatie Tvojho segmentu*“
    - CTL je skratka pre „Control bits“, jedno bitové príznaky (Flags)

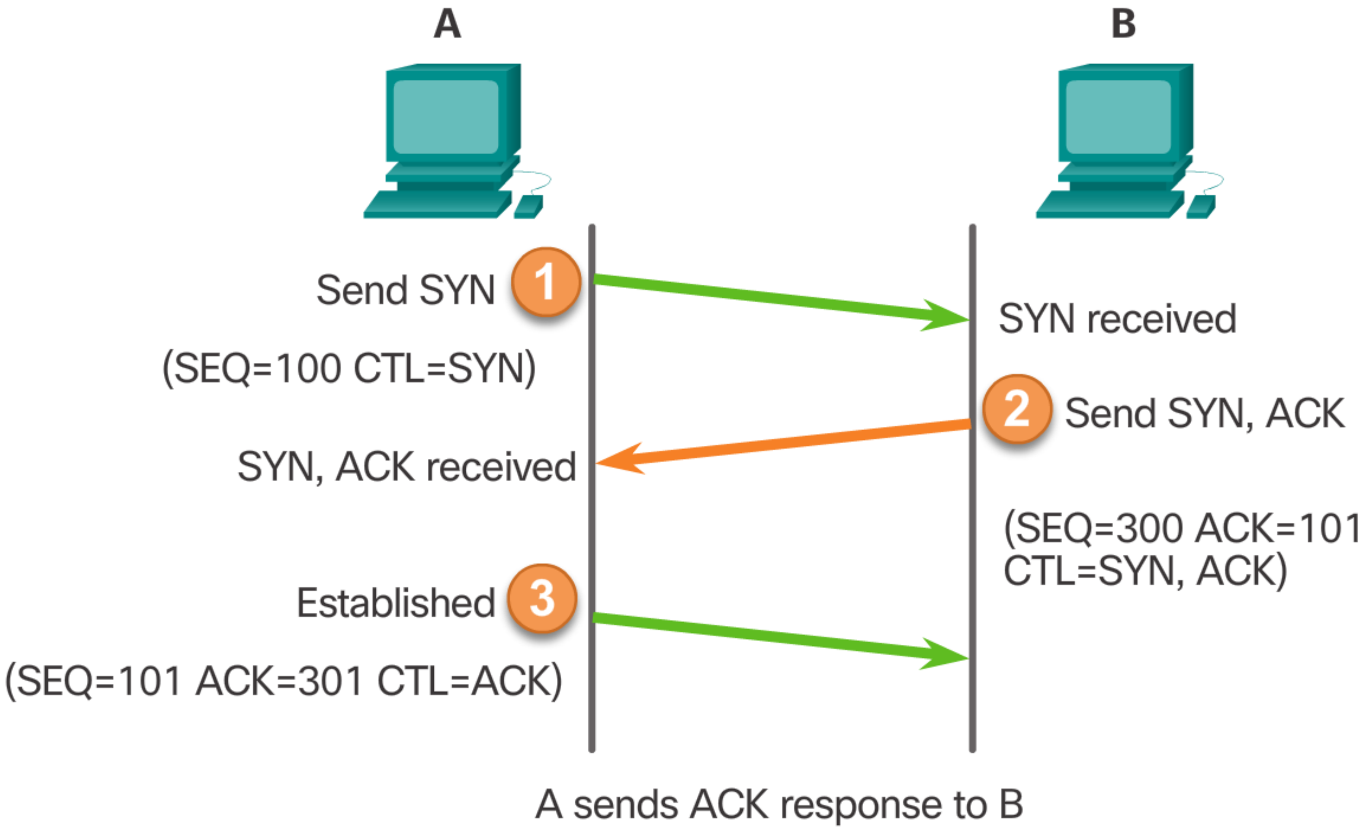




Otvorenie TCP spojenia

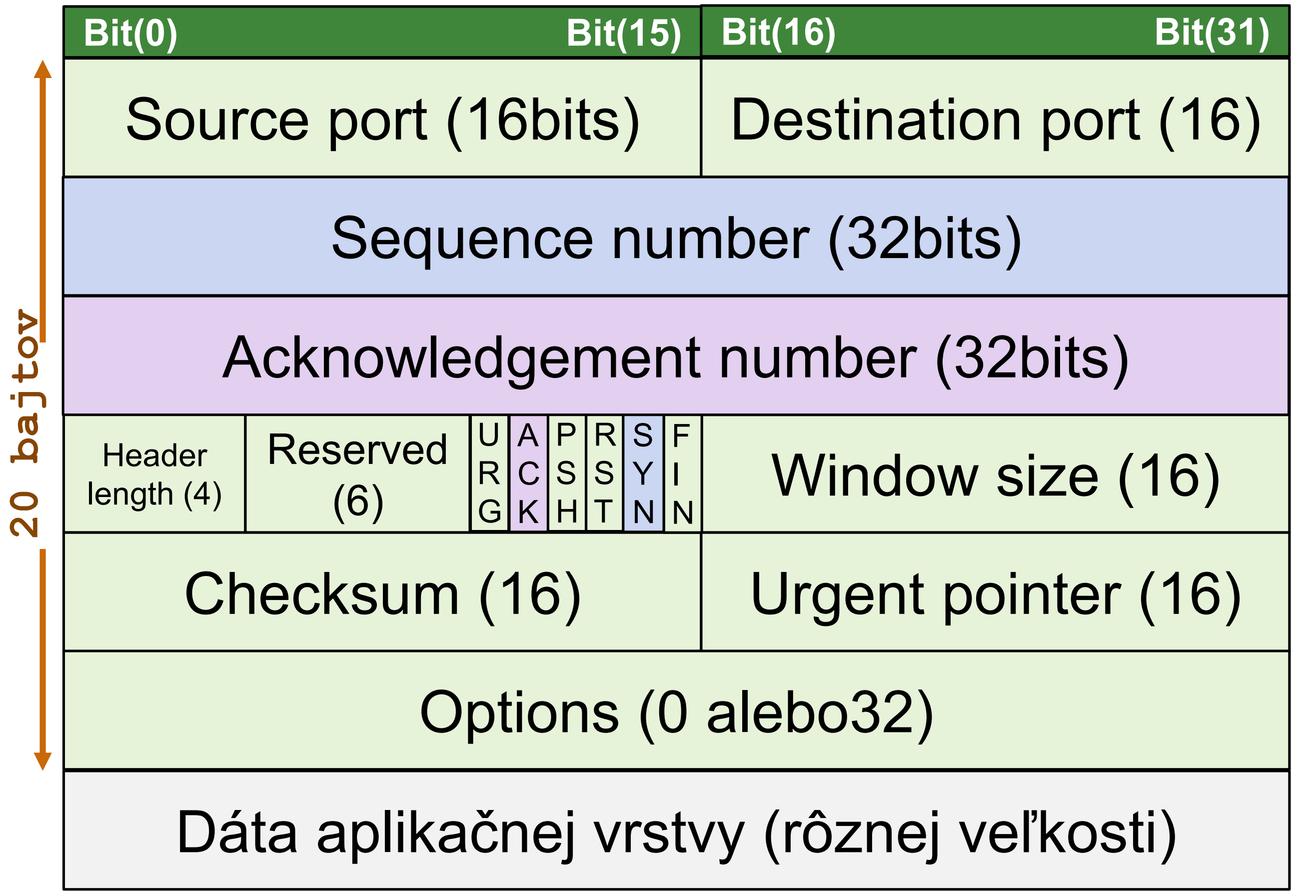
TCP spojenie je vytvorené v 3 krokoch (tzv. **3-way-handshake**):

1. Klient iniciuje vytvorenie spojenia – požiada o client-to-server komunikačnú reláciu (session) so serverom.
2. Server potvrdí túto reláciu a požiada klienta o server-to-client kom. reláciu.
3. Klient potvrdí server-to-client komunikačnú reláciu.



Analýza TCP 3-Way Handshake

- **Potvrdí**, že cieľové zariadenie je v sieti prítomné.
- **Overí**, že cieľové zariadenie má aktívnu službu a prijíma požiadavky na danom cieľovom porte, ktorý chce daný klient použiť.
- **Informuje** cieľové zariadenie, že zdroj (klient) chce vytvoriť komunikačnú reláciu na danom porte.
- **Príznaky:**
  - **SYN – v tomto segmente ti oznamujem, od akej hodnoty budem číslovať svoje segmenty**
  - **ACK – v tomto segmente ti posielam aj potvrdenie (čo mi už od teba úspešne prišlo)**



Wireshark: 3-way handshake [SYN]

No.	Source	Destination	Protocol	Length	Info
480	192.168.100.3	54.173.68.175	TCP	66	52305→80 [SYN] Seq=0 Win=65535 Len=0
527	54.173.68.175	192.168.100.3	TCP	66	80→52305 [SYN, ACK] Seq=0 Ack=1 Win=1
528	192.168.100.3	54.173.68.175	TCP	54	52305→80 [ACK] Seq=1 Ack=1 Win=26214

```
⊞ Ethernet II, Src: IntelCor_e7:0e:37 (d0:7e:35:e7:0e:37), Dst: HuaweiTe_be:0b:
⊞ Internet Protocol Version 4, Src: 192.168.100.3 (192.168.100.3), Dst: 54.173.
⊟ Transmission Control Protocol, Src Port: 52305 (52305), Dst Port: 80 (80), Se
    Source Port: 52305 (52305)
    Destination Port: 80 (80)
    [Stream index: 19]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 0
    Header Length: 32 bytes
  ⊟ .... 0000 0000 0010 = Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ⊞ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
```

Wireshark: 3-way handshake [SYN, ACK]

No.	Source	Destination	Protocol	Length	Info
480	192.168.100.3	54.173.68.175	TCP	66	52305→80 [SYN] Seq=0 Win=65535 Len=0
527	54.173.68.175	192.168.100.3	TCP	66	80→52305 [SYN, ACK] Seq=0 Ack=1 Win=1
528	192.168.100.3	54.173.68.175	TCP	54	52305→80 [ACK] Seq=1 Ack=1 Win=26214

```
Frame 527: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on inter
Ethernet II, Src: HuaweiTe_be:0b:27 (fc:e3:3c:be:0b:27), Dst: IntelCor_e7:0e:
Internet Protocol Version 4, Src: 54.173.68.175 (54.173.68.175), Dst: 192.168
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 52305 (52305), Se
  Source Port: 80 (80)
  Destination Port: 52305 (52305)
  [Stream index: 19]
  [TCP Segment Len: 0]
  Sequence number: 0    (relative sequence number)
  Acknowledgment number: 1    (relative ack number)
  Header Length: 32 bytes
  .... 0000 0001 0010 = Flags: 0x012 (SYN, ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
  Window size value: 14600
```

Wireshark: 3-way handshake [ACK]

No.	Source	Destination	Protocol	Length	Info
480	192.168.100.3	54.173.68.175	TCP	66	52305→80 [SYN] Seq=0 Win=65535 Len=0
527	54.173.68.175	192.168.100.3	TCP	66	80→52305 [SYN, ACK] Seq=0 Ack=1 Win=1
528	192.168.100.3	54.173.68.175	TCP	54	52305→80 [ACK] Seq=1 Ack=1 Win=26214
529	192.168.100.3	54.173.68.175	HTTP	537	GET / HTTP/1.1

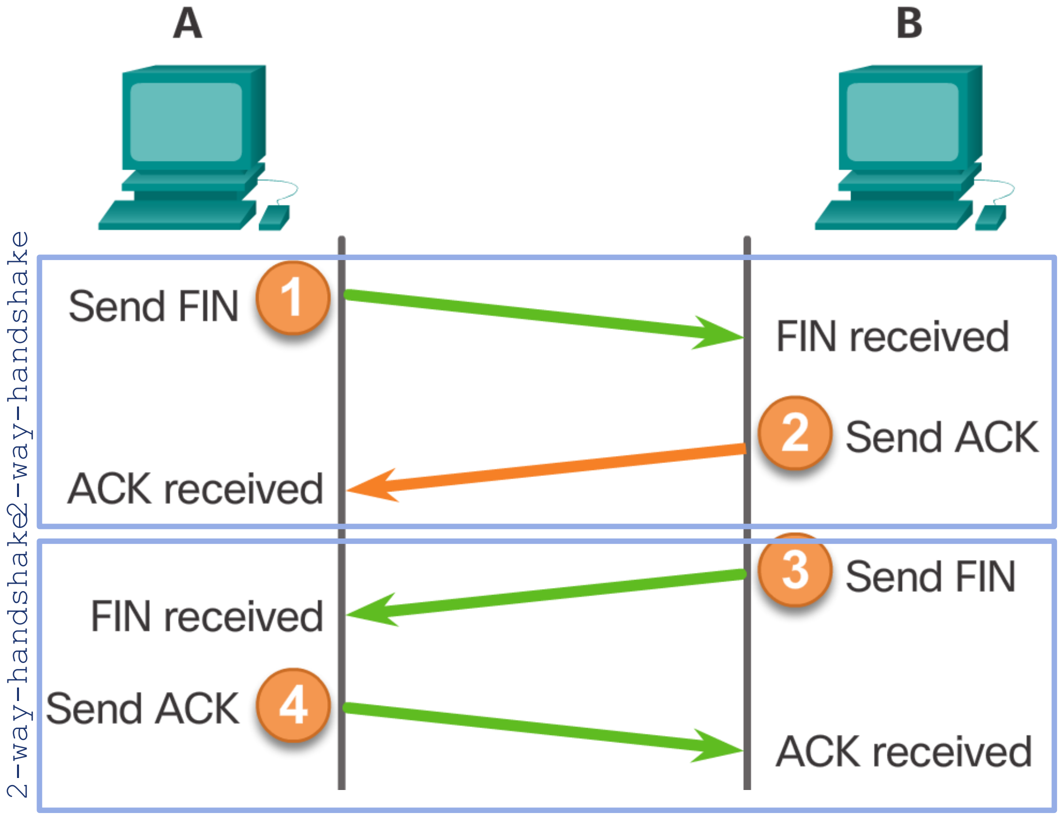
```
⊞ Frame 528: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on inter
⊞ Ethernet II, Src: IntelCor_e7:0e:37 (d0:7e:35:e7:0e:37), Dst: HuaweiTe_be:0b:
⊞ Internet Protocol Version 4, Src: 192.168.100.3 (192.168.100.3), Dst: 54.173.
⊟ Transmission Control Protocol, Src Port: 52305 (52305), Dst Port: 80 (80), Se
   Source Port: 52305 (52305)
   Destination Port: 80 (80)
   [Stream index: 19]
   [TCP Segment Len: 0]
   Sequence number: 1    (relative sequence number)
   Acknowledgment number: 1    (relative ack number)
   Header Length: 20 bytes
 ⊟ .... 0000 0001 0000 = Flags: 0x010 (ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
   Window size value: 1024
```

Uzatvorenie TCP spojenia

- Po konci komunikácie je potrebné TCP spojenie uzatvoriť
- Tzv. 2x   2-way-handshake
- Využíva sa príznak FIN (finish)
  - FIN: „*Nemám viac dát na odoslanie, za seba môžem skončiť*"

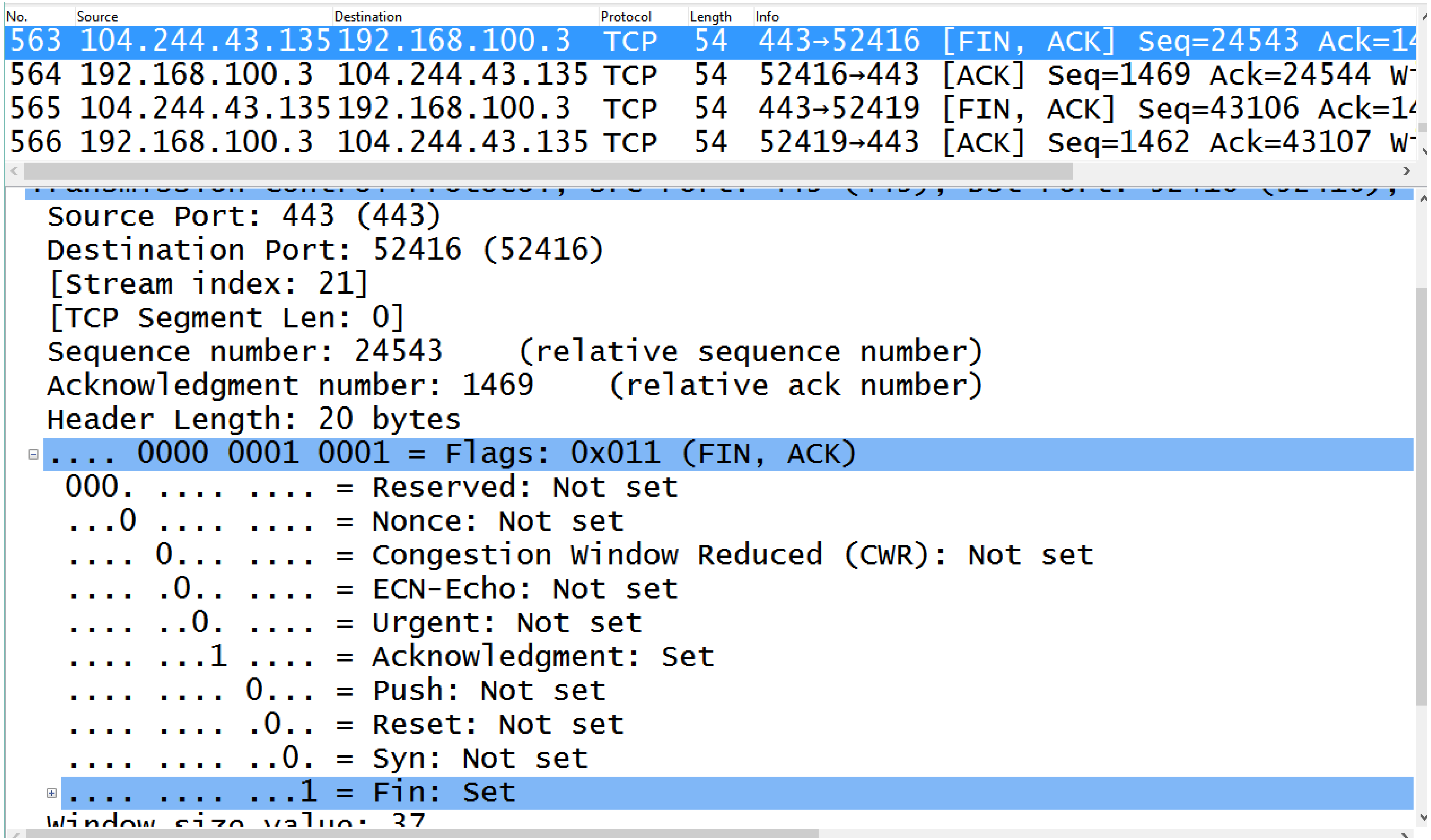
1. Keď klient poslal z daného TCP toku už všetky segmenty, pošle segment s nastaveným príznakom FIN.
2. Server pošle ACK, čím potvrdí prijatie FIN, že klient žiada o ukončenie danej relácie client-to-server.
3. Server pošle klientovi FIN, že súhlasí a ukončuje reláciu server-to-client.
4. Klient odpovie segmentom s ACK, čím potvrdí prijatie FIN od servera.

Po skončení týchto krokov je relácia zatvorená.



Wireshark: uzatvorenie spojenia [FIN, ACK]

Prvý 2-way-handshake



Wireshark: uzatvorenie spojenia [ACK]

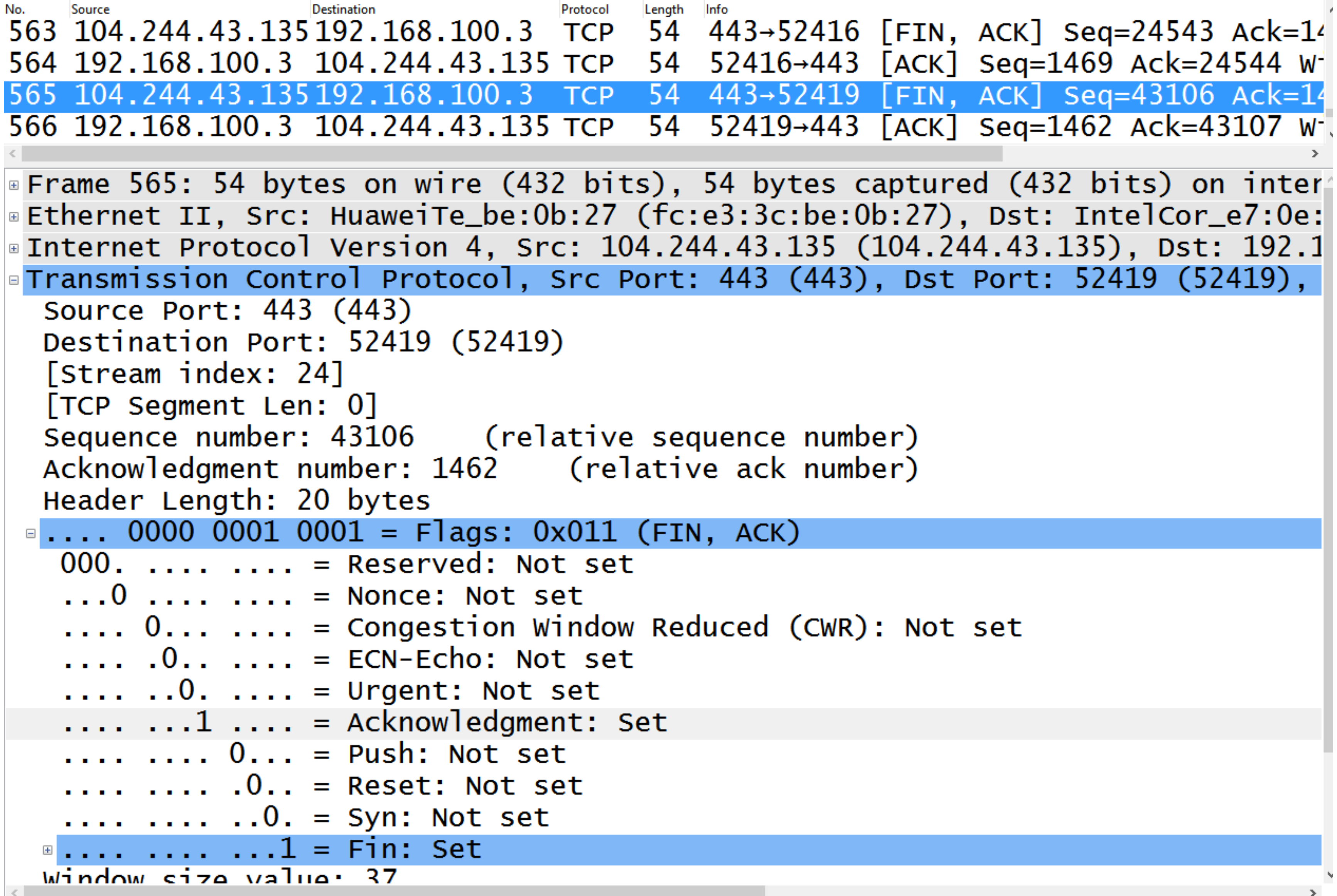
Prvý 2-way-handshake

```
No.  Source          Destination     Protocol Length Info
563 104.244.43.135 192.168.100.3  TCP  54  443→52416 [FIN, ACK] Seq=24543 Ack=14
564 192.168.100.3  104.244.43.135 TCP  54  52416→443 [ACK] Seq=1469 Ack=24544 W
565 104.244.43.135 192.168.100.3  TCP  54  443→52419 [FIN, ACK] Seq=43106 Ack=14
566 192.168.100.3  104.244.43.135 TCP  54  52419→443 [ACK] Seq=1462 Ack=43107 Wi

⊞ Frame 566: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on inter
⊞ Ethernet II, Src: IntelCor_e7:0e:37 (d0:7e:35:e7:0e:37), Dst: HuaweiTe_be:0b:
⊞ Internet Protocol Version 4, Src: 192.168.100.3 (192.168.100.3), Dst: 104.244
⊟ Transmission Control Protocol, Src Port: 52419 (52419), Dst Port: 443 (443),
   Source Port: 52419 (52419)
   Destination Port: 443 (443)
   [Stream index: 24]
   [TCP Segment Len: 0]
   Sequence number: 1462    (relative sequence number)
   Acknowledgment number: 43107    (relative ack number)
   Header Length: 20 bytes
 ⊟ .... 0000 0001 0000 = Flags: 0x010 (ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
   Window size value: 64
```

Wireshark: uzatvorenie spojenia [FIN, ACK]

Druhý 2-way-handshake



Wireshark: uzatvorenie spojenia [ACK]

Druhý 2-way-handshake

No.	Source	Destination	Protocol	Length	Info
563	104.244.43.135	192.168.100.3	TCP	54	443→52416 [FIN, ACK] Seq=24543 Ack=14
564	192.168.100.3	104.244.43.135	TCP	54	52416→443 [ACK] Seq=1469 Ack=24544 W
565	104.244.43.135	192.168.100.3	TCP	54	443→52419 [FIN, ACK] Seq=43106 Ack=14
566	192.168.100.3	104.244.43.135	TCP	54	52419→443 [ACK] Seq=1462 Ack=43107 W

```
⊞ Frame 565: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on inter
⊞ Ethernet II, Src: HuaweiTe_be:0b:27 (fc:e3:3c:be:0b:27), Dst: IntelCor_e7:0e:
⊞ Internet Protocol Version 4, Src: 104.244.43.135 (104.244.43.135), Dst: 192.1
⊟ Transmission Control Protocol, Src Port: 443 (443), Dst Port: 52419 (52419),
   Source Port: 443 (443)
   Destination Port: 52419 (52419)
   [Stream index: 24]
   [TCP Segment Len: 0]
   Sequence number: 43106    (relative sequence number)
   Acknowledgment number: 1462    (relative ack number)
   Header Length: 20 bytes
 ⊟ .... 0000 0001 0001 = Flags: 0x011 (FIN, ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
  ⊞ .... .... ...1 = Fin: Set
   Window size value: 37
```

Wireshark: náhle ukončenie spojenia [RST]

```
No.  Source           Destination    Protocol  Length  Info
853  192.168.100.3    54.173.68.175  TCP       54      52305→80 [RST, ACK] Seq=484 Ack=770 W
860  192.168.100.3    54.173.68.175  TCP       54      52309→443 [RST, ACK] Seq=1209 Ack=694

⊞ Frame 853: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on inter
⊞ Ethernet II, Src: IntelCor_e7:0e:37 (d0:7e:35:e7:0e:37), Dst: HuaweiTe_be:0b:
⊞ Internet Protocol Version 4, Src: 192.168.100.3 (192.168.100.3), Dst: 54.173.
⊟ Transmission Control Protocol, Src Port: 52305 (52305), Dst Port: 80 (80), Se
    Source Port: 52305 (52305)
    Destination Port: 80 (80)
    [Stream index: 19]
    [TCP Segment Len: 0]
    Sequence number: 484    (relative sequence number)
    Acknowledgment number: 770    (relative ack number)
    Header Length: 20 bytes
  ⊟ .... 0000 0001 0100 = Flags: 0x014 (RST, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
    ⊞ .... .... .1.. = Reset: Set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
    Window size value: 0
    [Calculated window size: 0]
    [Window size scaling factor: 256]
```

TCP stavový diagram prechodov

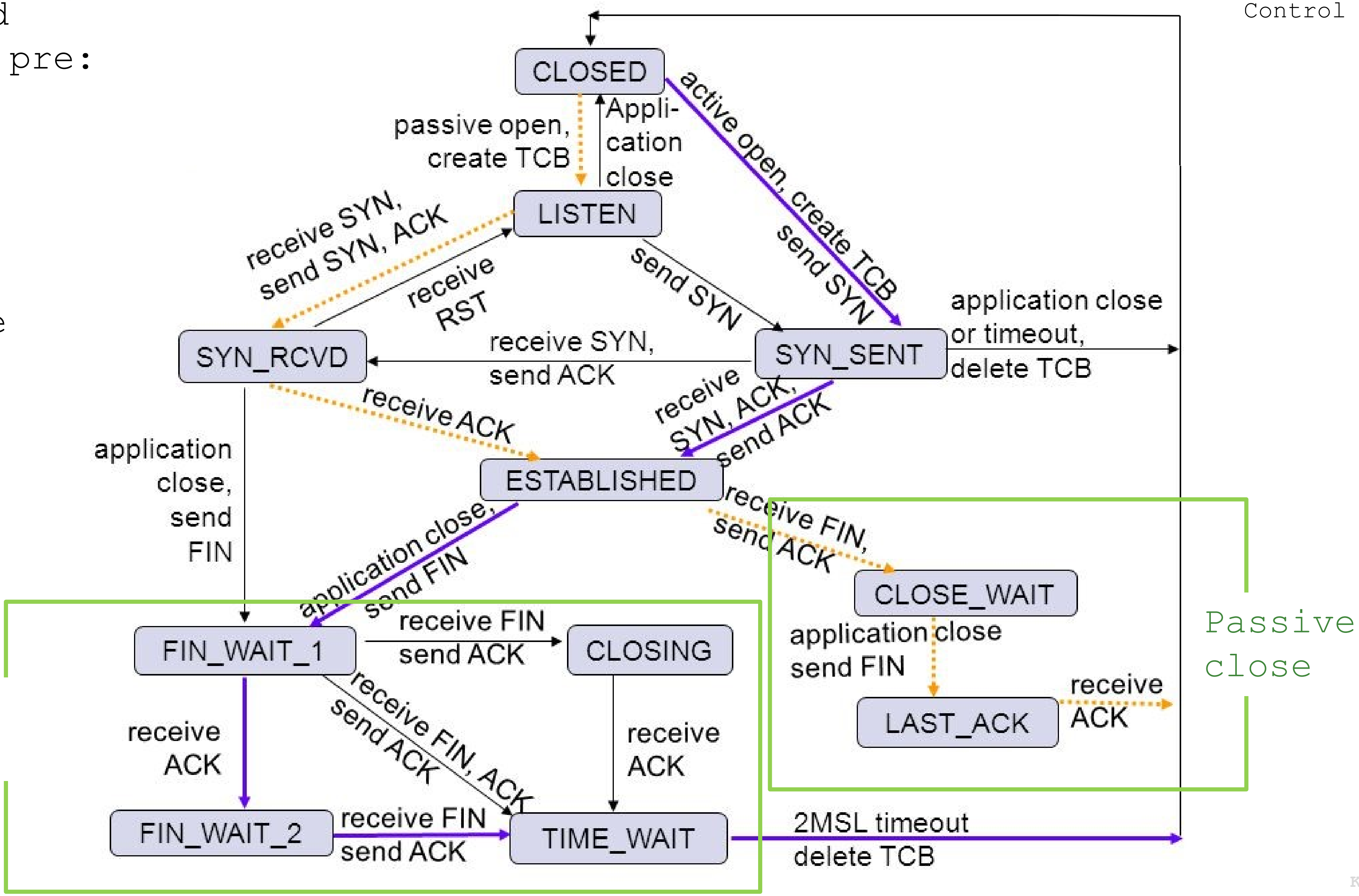
(skratka TCB = Transmission Control Block)

Bežný prechod medzi stavmi pre:

Klienta

Server

Neštandardné správanie



Spoľahlivosť a kontrola toku dát v TCP

Spoľahlivost TCP – Usporiadanie prijatých dát

- TCP segmenty používajú **sekvenčné čísla** (sequence numbers - SN):
  - na jednoznačné identifikovanie a potvrdzovanie každého segmentu

    „Strata každého segmentu je odhalená.“

  - aby si udržiavali správne poradie segmentov.
  - aby druhej strane dali vedieť v akom poradí má zrekonš--truovať jednotlivé segmenty do celej správy

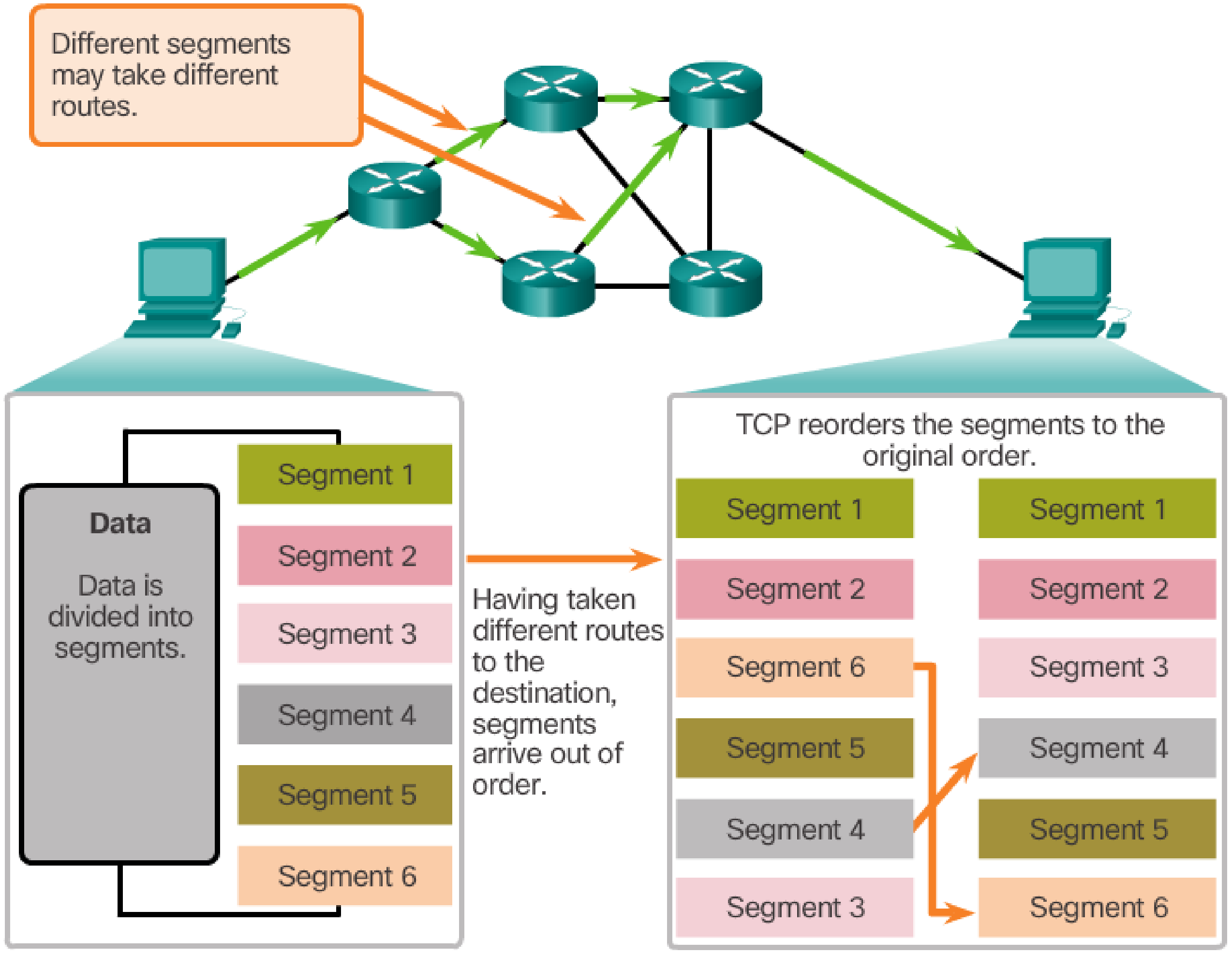
20 bajtov

Bit(0) … Bit(15)	Bit(16) … Bit(31)
Source port (16bits)	Destination port (16)
Sequence number (32bits)	
Acknowledgement number (32bits)	
Header length (4) | Reserved (6) | URG | ACK | PSH | RST | SYN | FIN	Window size (16)
Checksum (16)	Urgent pointer (16)
Options (0 alebo32)	
Dáta aplikačnej vrstvy (rôznej veľkosti)	

- Úvodné/prvé SN sa volí **náhodne** (v minulosti sa brala 0) počas vytvorenia spojenia (**3-way-handshake**), a **inkrementuje** sa vždy **o počet odoslaných bajtov**

Spoľahlivost TCP – Usporiadanie prijatých dát

- TCP proces u príjemcu si ukladá prijaté segmenty do frontu - buffer, segmenty ktoré prídu mimo poradia sú odložené na neskoršie spracovanie.
- Až keď príjemca dostane všetky segmenty (slušne sa ukončí spojenie), začne robiť rekonštrukciu prijatých segmentov do pôvodnej správy.
- Ak sa rekonštrukcia podarí, dáta predá na spracovanie aplikačnej vrstve.



Spoľahlivosť TCP – Potvrdzovanie prijatých dát

- TCP proces u príjemcu **potvrdzuje** každý segment, ktorý prijme od TCP procesu odosielateľa. Táto metóda sa nazýva **Stop and wait**:

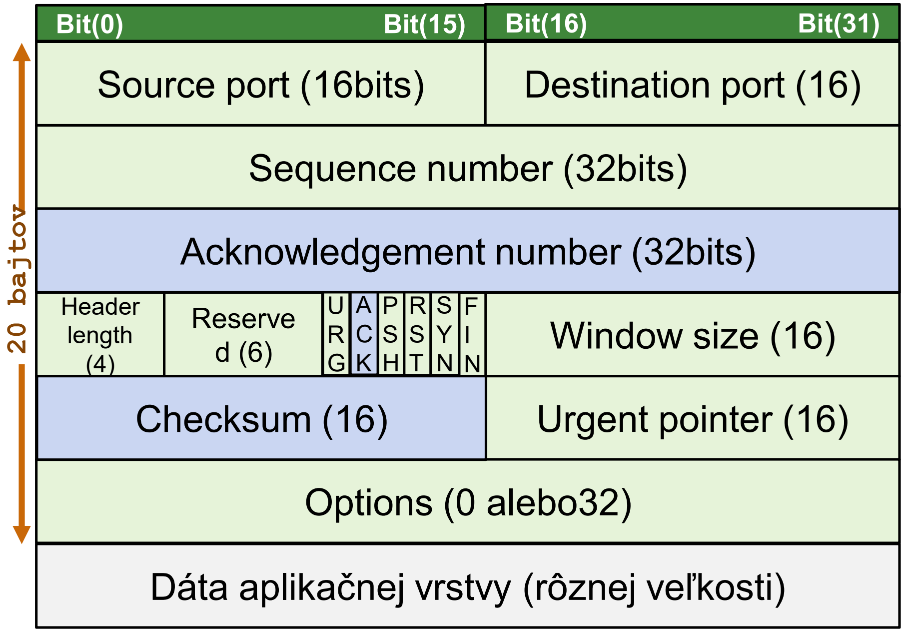
„*Pošli segment a zastav posielanie ďalších segmentov (Stop), čakaj na potvrdenie - ACK (Wait).*“

  - Používa sa na to pole **Acknowledgement Number** v hlavičke TCP

- Aby sa odosielateľ nedostal do slepej uličky (deadlock), kedy by čakal na ACK do nekonečna:

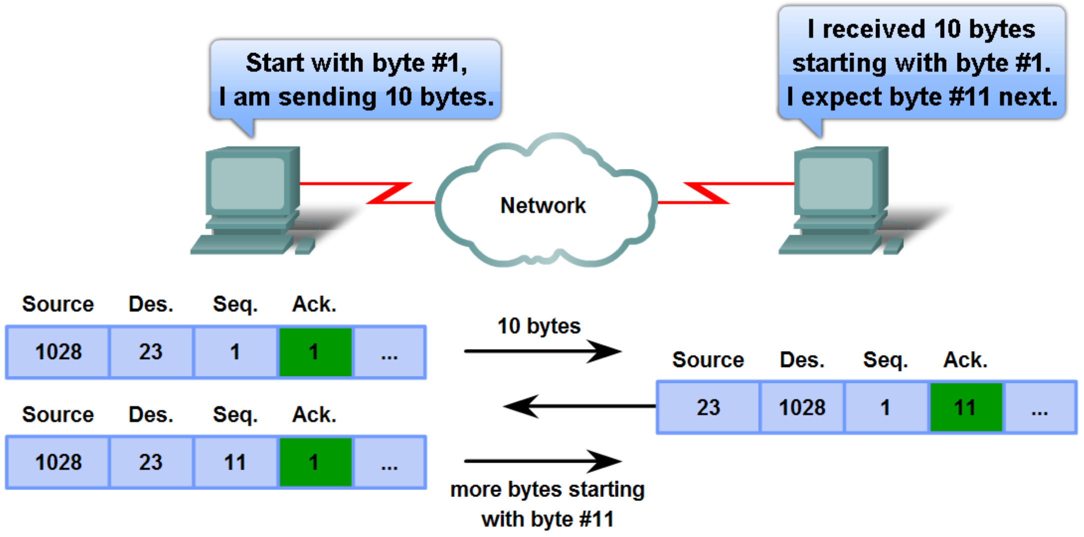
1. **pri strate segmentu**
2. **pri strate potvrdenia**
3. **keď príjemcovi segment príde, ale poškodený (zistí.. checksum), a neposiela späť ACK**

- tak sa používa časovač, tzv. **Retransmission Timeout (RTO)**:
  - spúšťa sa pri odoslaní každého segmentu
  - keď vyprší a nepríde ACK, odosielateľ **zopakuje odoslanie** toho istého segmentu
  - ak do tohto času príde ACK, odosielateľ vyšle ďalší segment, pre ktorý znovu spúšťa RTO



Spoľahlivosť TCP – Potvrdzovanie prijatých dát

- Potvrdzovanie je tzv. pozitívne alebo dopredné: ak 1 strana pošle potvrdzovacie číslo ***n***, znamená to, že správne prijala všetky **bajty** až po ***n-1***
  - Potvrdzovacie číslo ***n*** teda znamená:
    „*Pokračuj bajtom n, pretože všetky bajty od počiatku až po n-1 už mám*“
  - Potvrdenie hovorí o prvom bajte, ktorý očakávame (resp. ktorý chýba)
- Potvrdzovanie a prenos dát sa môže diať v jednom TCP segmente súčasne – tzv. **piggybacking** – ACK správy nepošlem samostatne, ale vložím do segmentu, ktorý mám pripravený na odoslanie druhej strane, tzv. nesamostatné potvrdzovanie



Riadenie toku dát – technika posuvného okna

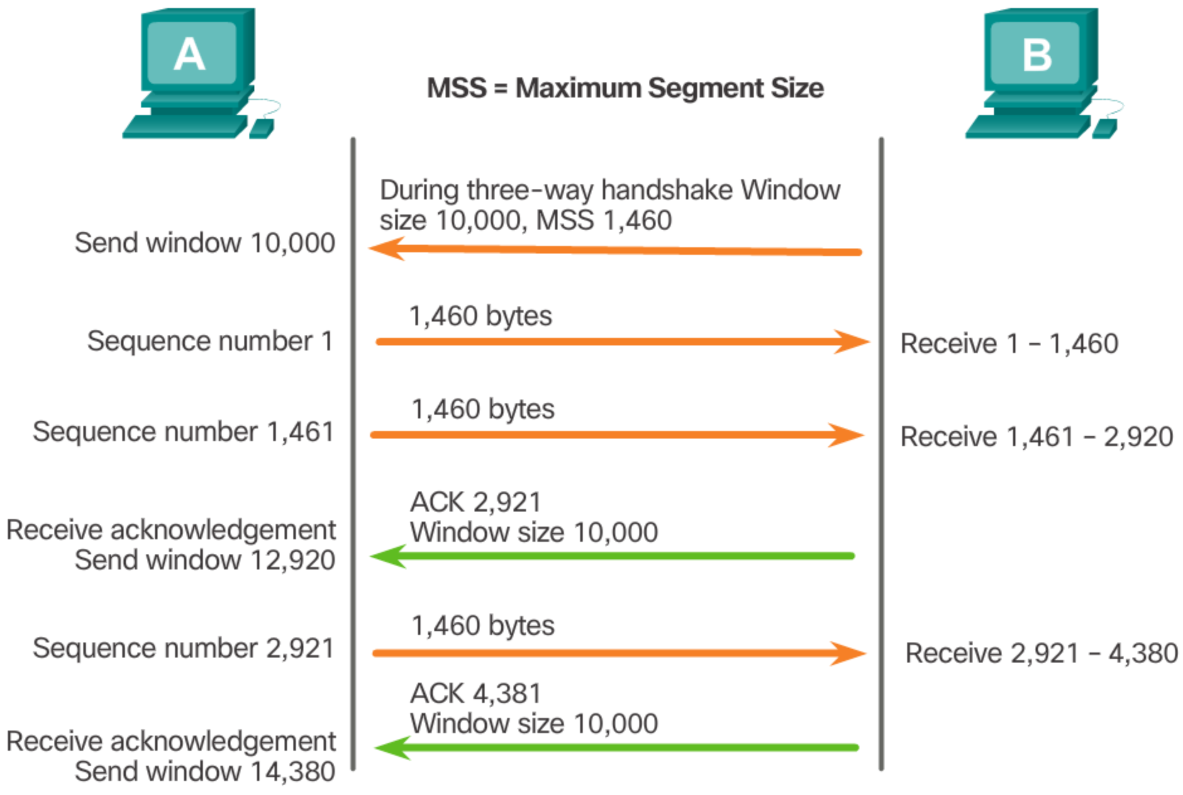
Window size (16)

- Zhodnotenie Stop&Wait:
  - výhoda: je jednoduché implementovať
  - nevýhoda: nevyužíva dostatočne prenosové pásmo, vzniká oneskorenie – za čas, kedy sa čaká na potvrdenie, by sa mohol odoslať ďalší segment
- Obmedzenia Stop&Wait rieši mechanizmus pre riadenie toku tzv. **sliding window** (plávajúce/posuvné okno), ktorého cieľom je, aby sa dáta vysielali čo najrýchlejšie, ale tak, aby oba konce TCP komunikácie stíhali prijímať a spracovávať segmenty spoľahlivo
- Používa na to pole **Window size (WS)** v hlavičke TCP
  - 16 bitové pole (2^{16} možných veľkostí okna)
  - min. WS = 0 B, max. WS = 2^{16} -1 = 65 535 B
  - Je to maximálny objem dát (v bajtoch), ktoré mi môže druhá strana poslať ešte pred tým, ako jej doručím potvrdenie o ich prijatí (t.j. nemusí čakať na potvrdenie)
    - potvrdenia prísť nakoniec musia, ale odosieľateľ na ne nemusí čakať, kým súčet bajtov zo všetkých odvysielaných segmentov nepresiahne veľkosť okna
  - Je to počet bajtov, ktoré som ako príjemca schopný prijať, preto veľkosť môjho okna musím oznámiť druhej strane, aby sa mi vedela prispôsobiť

Riadenie toku dát – technika posuvného okna

Window size (16)

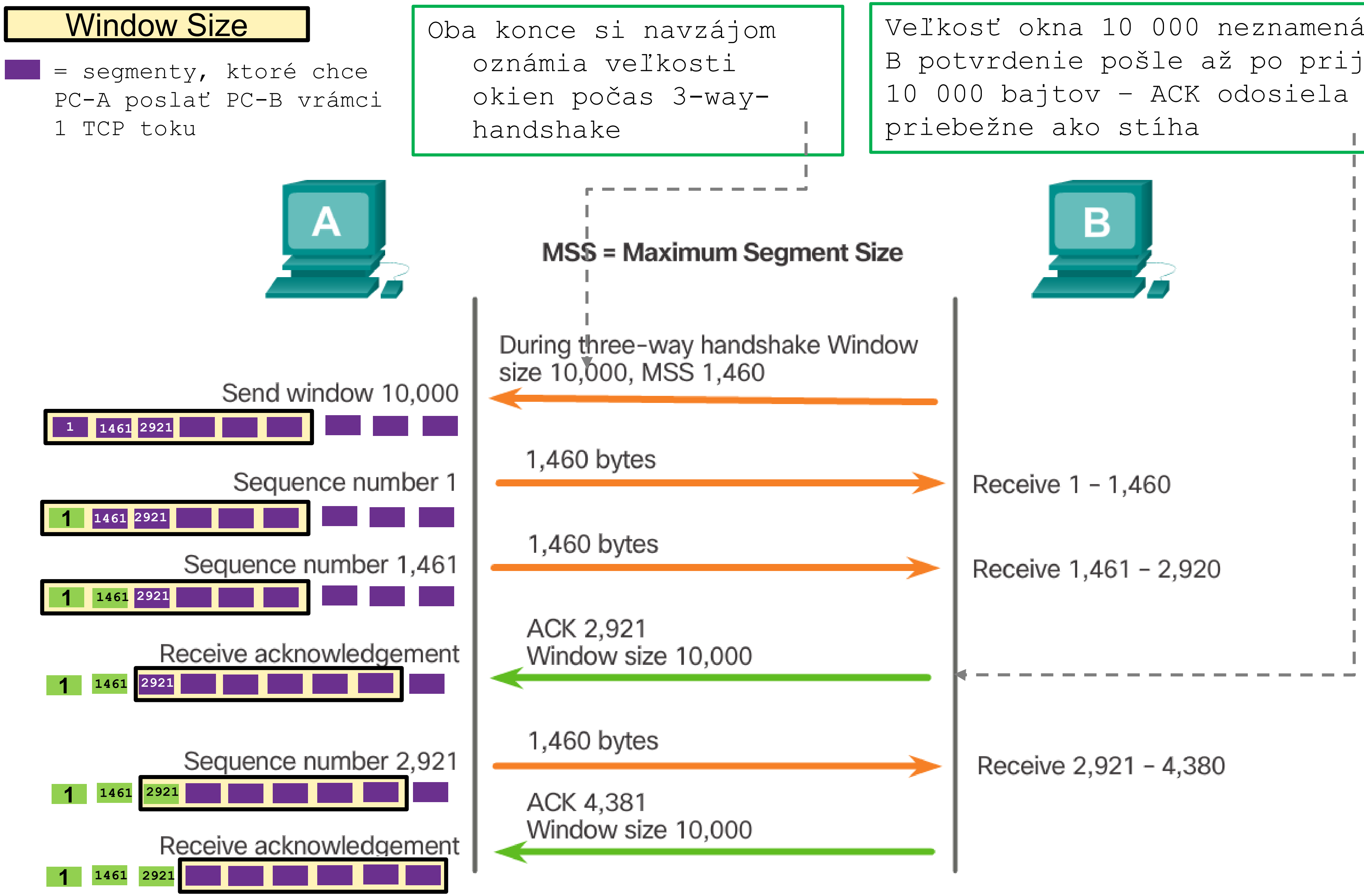
- Oba konce si navzájom oznámia veľkosti okien
  počas 3-way-handshake
- Po vytvorení spojenia môžu veľkosť tohto okna meniť - veľkosť okna stanica uvádza v **každom** odoslanom TCP segmente v poli **Window Size**
- Veľkosti okna **N** neznamená, že potvrdenie pošleme až po prijatí N bajtov – odosielanie potvrdení nemusí byť (a zväčša nie je) veľkosťou okna ovplyvnené



Riadenie toku dát – technika posuvného okna

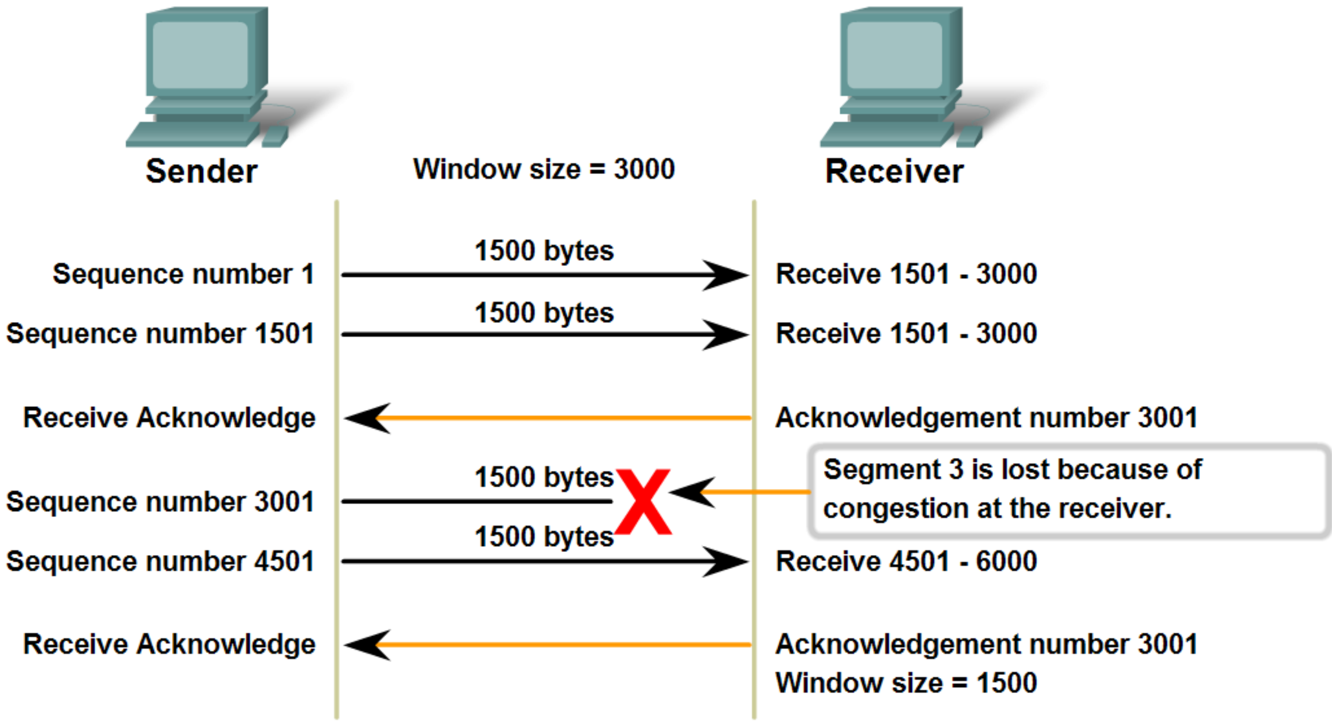
Oba konce si navzájom oznámia veľkosti okien počas 3-way-handshake

Veľkosť okna 10 000 neznamená, že PC-B potvrdenie pošle až po prijatí 10 000 bajtov – ACK odosiela priebežne ako stíha



Riadenie toku dát – technika posuvného okna

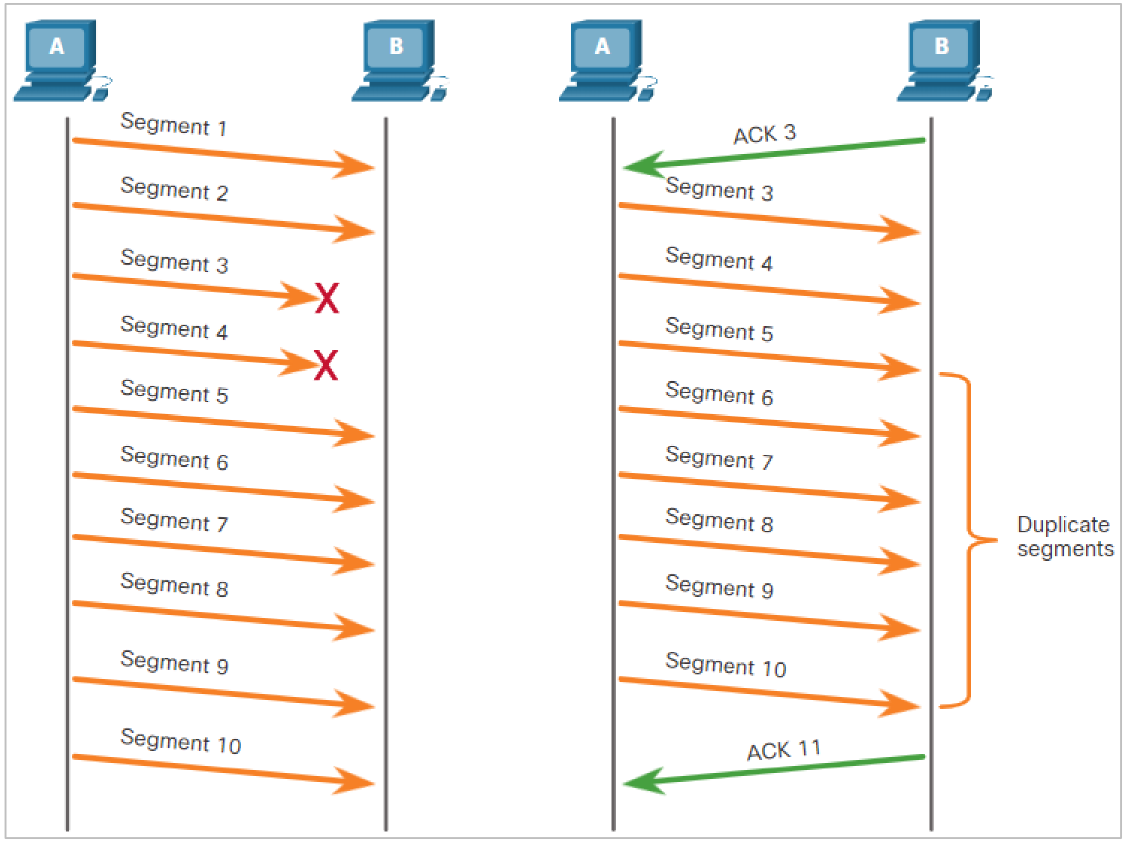
- Pri strate segmentu so SN=3001, príjemca potvrdí znova posledný prijatý bajt – aby odosielateľ vedel, čo posledné mu prišlo a čo ďalšie očakáva
  - T.j. nepotvrdzuje segment so SN=4501, aj keď mu tento prišiel
  - Následne odosielateľ bude musieť zopakovať prenos segmentov SN=3001, SN=4501
- Navyše odosielateľ si **zmenší** veľkosť okna WS=3000, reaguje na stratené segmenty (alebo stratené ACK, alebo poškodené segmenty), v tomto príklade na ½, t.j. WS=1500
- Keď sa situácia zlepší a odosielateľ začne dostávať ACK, zase si okno **zväčší**



Riadenie toku dát – predchádzanie zahlteniu

Congestion Avoidance

- **Zahltenie** v sieti (congestion) zvyčajne vedie k **stratám** paketov
- Nedoručené TCP segmenty sa musia **preposielať znova**, čo môže situáciu ešte viac zhoršiť.
- Odosielateľ môže **detegovať** zahltenie v sieti tým, že **rýchlosť** akou vysiela dáta, neodpovedá rýchlosti akou **mu chodia ACK**, alebo mu vôbec nechodia.
- Vtedy odosielateľ môže **zmenšiť rýchlosť odosielania dát** – zníži si veľkosť okna ešte pred tým, ako mu zníženie oznámi príjemca, ktorý takéto zahltenie zväčša zistí až neskôr, alebo ho vôbec nezaregistruje.
- Keď sa situácia zase zlepší, a odosielateľovi začnú chodiť ACK, znova si zväčší okno.. zväčšuje ho postupne
- Takto sa okno dynamicky „otvára“ a „zatvára“
- O ktorom okne je reč? ....



Riadenie toku dát – predchádzanie zahlteniu
Congestion Avoidance

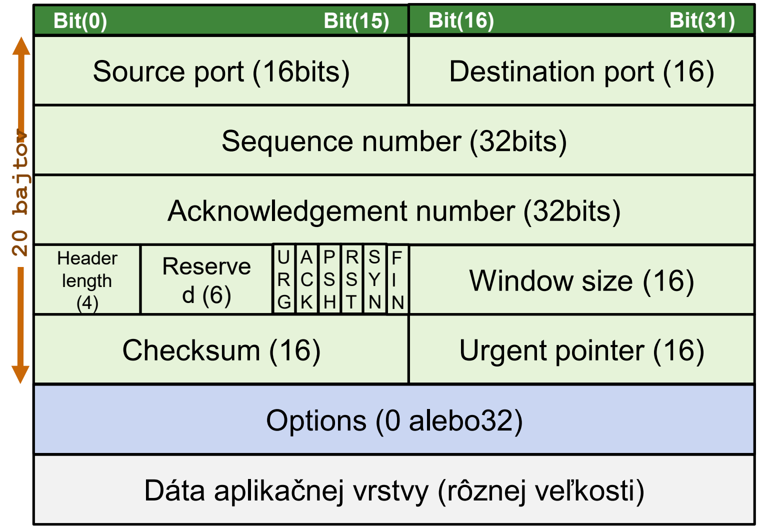
- V skutočnosti má odosielateľ svoje vlastné okno, **Sender Window,** tiež nazývané ako **Congestion Window**, ktoré si prispôsobuje podľa situácie v sieti (či mu nechýbajú ACK alebo dostáva chybné segmenty)
  - Okno, ktoré mu ohlasuje príjemca v TCP hlavičke v poli **Window Size,** potom môžeme nazvať aj ako **Receiver Window** - to čo príjemca ohlasuje odosielateľovi:

    „*Neposielaj mi dáta rýchlejšie ako je Window size.*“
  - Rýchlosť akou odosielateľ nakoniec skutočne vysiela dáta sa potom berie ako minimum z:

min { Sender window, Receiver window }

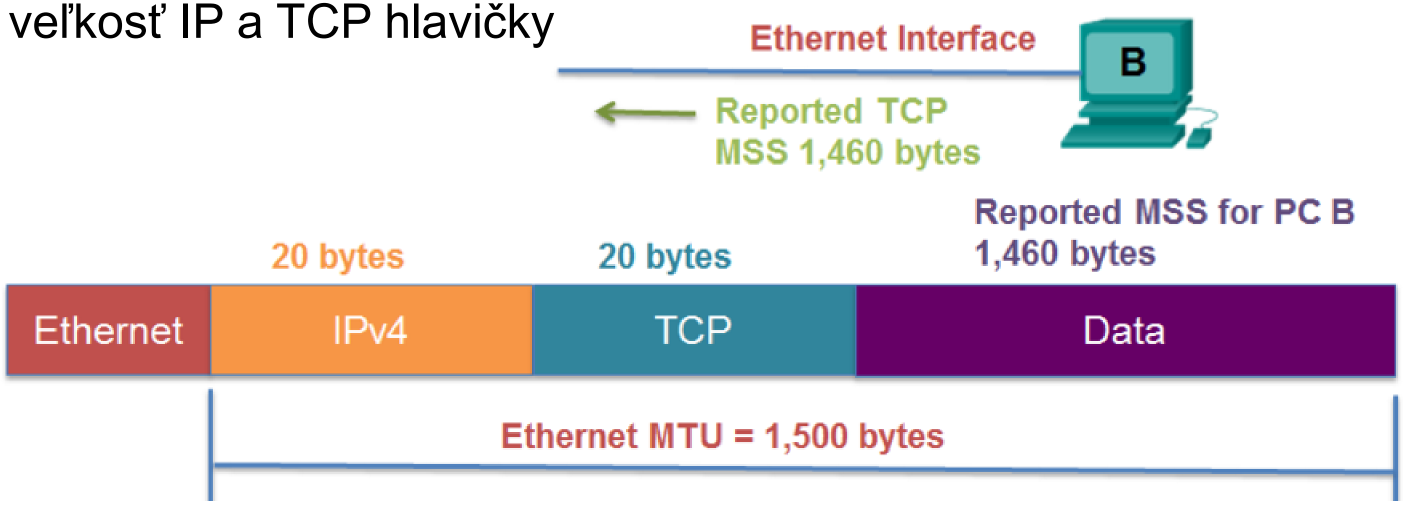
- T.j. nemôžem vysielať dáta rýchlejšie ako si to praje príjemca

Doplňujúce voľby v TCP - Options



Dohadovanie MSS (maximum segment size) pri nadväzovaní spojenia

- 16 bitová hodnota
- Najväčší počet dát (v bajtoch), ktoré je schopné dané zariadenie prijať v jednom TCP segmente (bez hlavičky)
  - Obe komunikujúce zariadenia si to oznámia počas 3-way-handshake
  - Zväčša je to MTU (Maximum Transmission Unit) výstupného rozhrania, ktorým odíde daný segment zo zariadenia, mínus veľkosť IP a TCP hlavičky
  - Ethernetové rozhranie má MTU = 1500 B (najväčšie množstvo dát ktoré možno zabaliť do Ethernetového rámca).
    - Potom MSS = 1500–20–20 = 1460 B



Doplňujúce voľby v TCP - Options

Násobky veľkosti okna (window scale)

- Pre niektoré aplikácie nestačí ani max. veľkosť okna (64kB = 65535 B)
- Napr. na trasách s veľkým oneskorením (satelitné siete), alebo siete s veľmi nízkym oneskorením, ale veľkou šírkou pásma (bandwidth), t.j. siete, pre ktoré súčin delay*bandwith vychádza vysoký (viac ako 10^5)
- Skutočná veľkosť okna = Window Size * $2^{WindowScale}$
  - Násobky (W.size) mocniny (W.Scale) so základom 2
    - $2^{WindowScale}$ je tzv. Scale factor
  - t.j. W.Size sa posunie – spraví sa shift doľava na vyššie mocniny

Príklad: W.size = 65 535 = 11111111 11111111, W.Scale = 14, potom:

2^0

11111111 11111111 00000000000000 = 1 073 725 440 = 1GB = 65 535 * 2^{14}

2^{14}

(viac info v RFC 1323)

Doplňujúce voľby v TCP - Options

Násobky veľkosti okna (window scale)

- hodnota **Window scale** sa uvedie v hlavičke TCP v poli **Options**
  - nastaví sa počas 3-way handshake v SYN segmente, potom je fixná počas celého spojenia
  - max. hodnota je 14
- hodnota **Window size** sa uvedie v poli **Window size** v hlavičke
- toto uvidíme aj pri snifovaní cez Wireshark na cvičení

(viac info v RFC 1323)

Doplňujúce voľby v TCP - Options

Násobky veľkosti okna - Wireshark sniff: pohľad na Window Scale

```
[Next sequence number: 6806    (relative sequence number)]
Acknowledgment number: 218    (relative ack number)
Header Length: 20 bytes
> Flags: 0x018 (PSH, ACK)
Window size value: 512
[Calculated window size: 131072]
[Window size scaling factor: 256]
Checksum: 0x7179 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
```

Časové značkovanie paketov (timestamps)

- Kvôli výpočtu RTT (round trip time)
- a následnej kalkulácii RTO (retransmission timeout) pre znovuopakovanie vysielania, pri strate segmentu a pod.

Doplňujúce voľby v TCP - Options

SACK – iná technika potvrdzovania segmentov

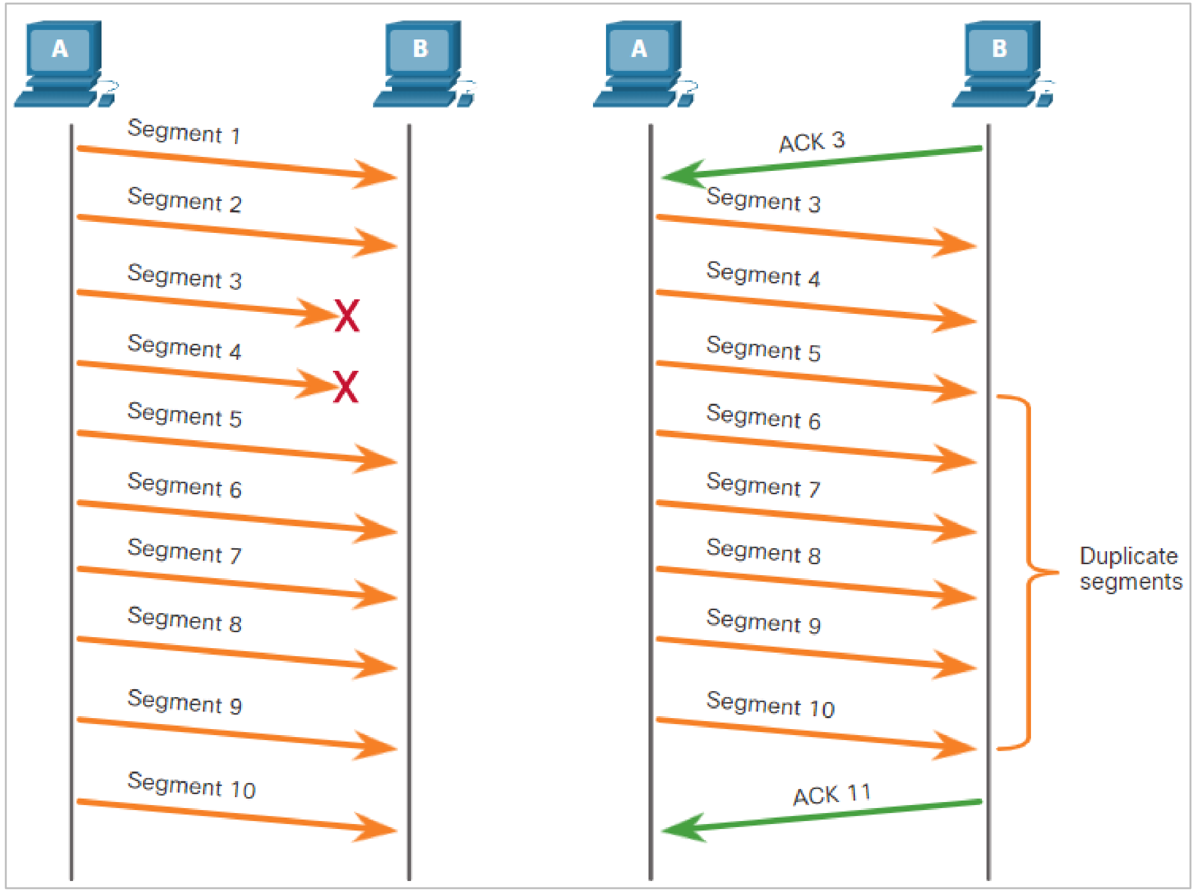
V základnom TCP sa pri strate 1 alebo viac segmentov, ktoré odosielateľ posiela bez potvrdenia v rámci dohodnutého okna, musia preposlať všetky segmenty od posledného potvrdeného bajtu (nielen 1-2 stratené) – toto rieši SACK – voliteľná implementácia TCP :

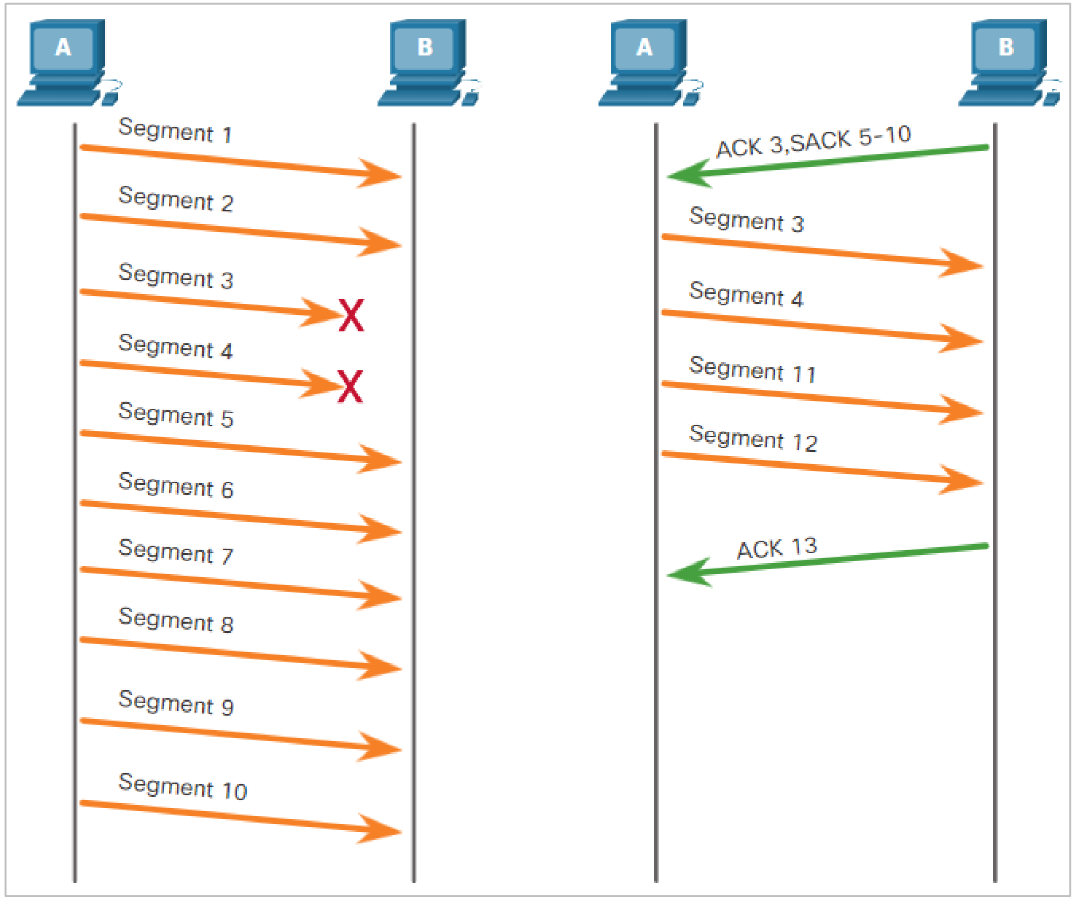
Selective Acknowledgement (SACK)

- pri strate segmentu, môže odosielateľ zopakovať prenos iba strateného segmentu, nemusí preposielať celé okno
  - Príjemca mu totiž vie oznámiť, ktorý segment mu chýba aj ktoré segmenty mu medzi časom prišli (po danom stratenom)
    - ACK = „*Toto očakávam že pošleš.*“ (hodnota sa uvedie v poli Acknowledgement number)
    - SACK= „*Toto mi už ale medzi časom prišlo, nepreposielaj.*“ (hodnoty sa uvedú v poli Options ako SACK)
- Použitie SACK si zariadenia dohodnú počas 3-way-handshake
  - Ak oba konce podporujú SACK, tak sa použije počas celého prenosu

(viac info v RFC 2018)

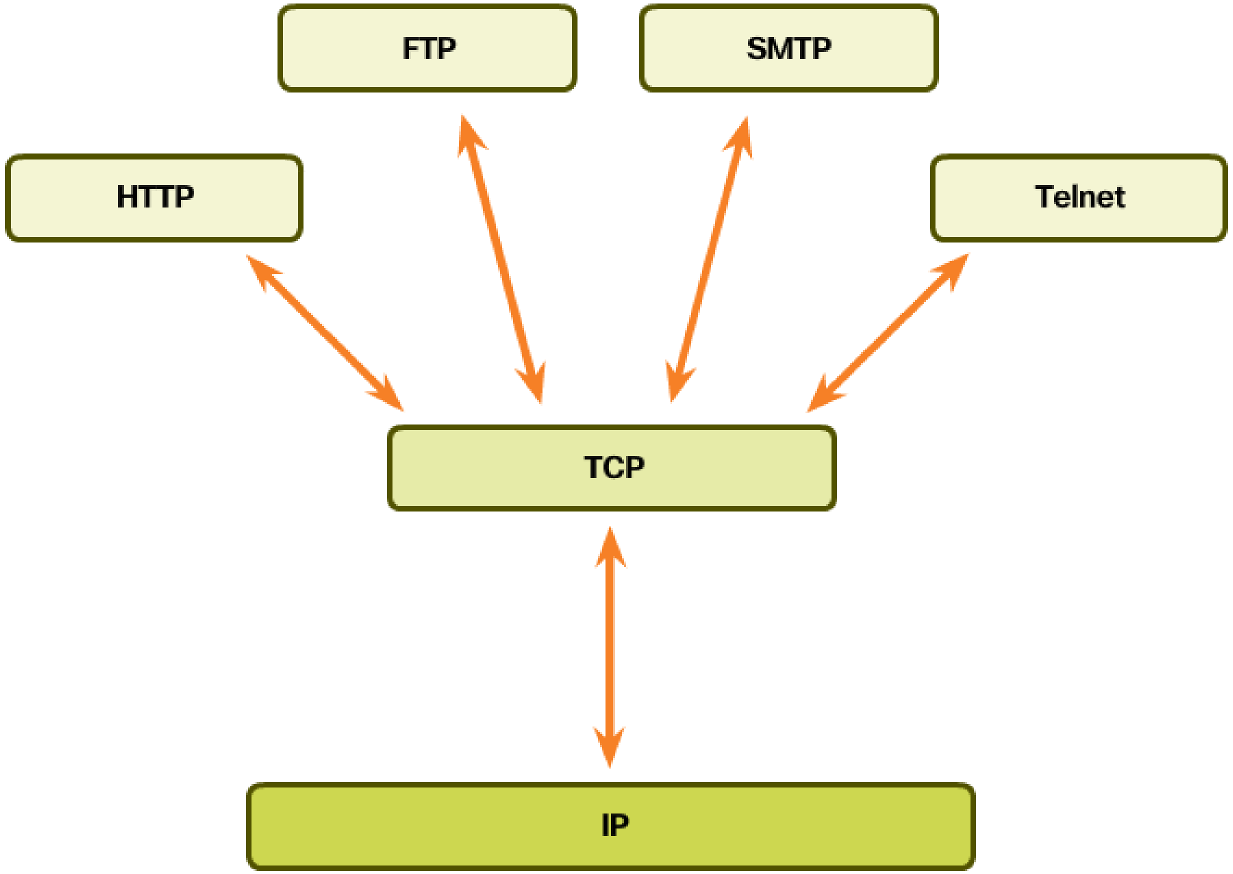
Doplňujúce voľby v TCP - Options
SACK – iná technika potvrdzovania segmentov

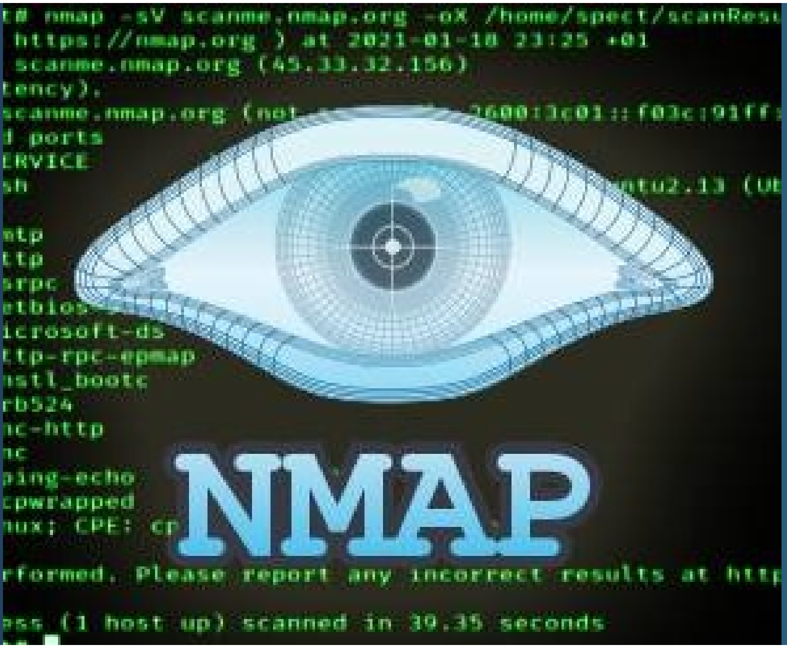




Využitie TCP

- Všade kde potrebujem **spoľahlivú**, **spojovo orientovanú** službu doručovania tokov dát (streams) **s riadením toku** (flow control)



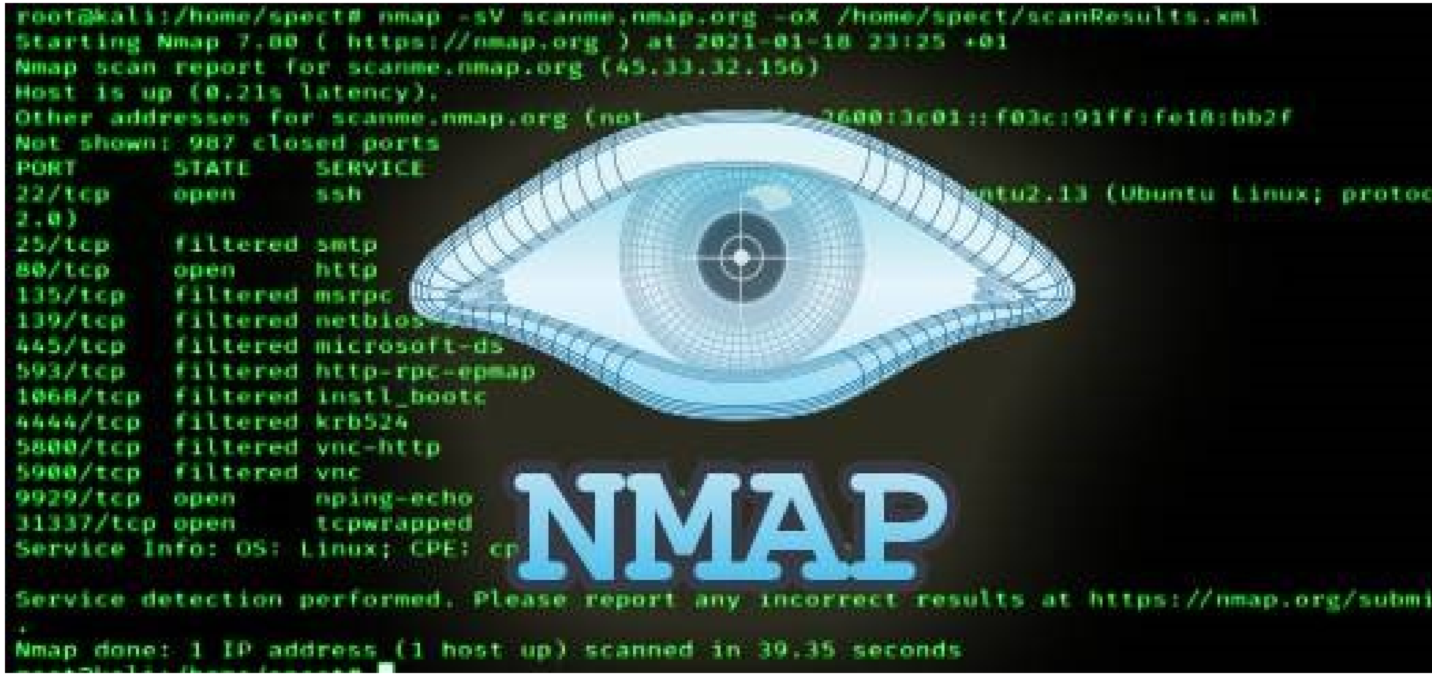


Skenovanie siete

Nmap, hping, massscan

Nástroj NMAP

- Výkonný open-source sieťový CLI nástroj, ktorý sa používa na
  - mapovanie siete
  - audit bezpečnosti
- Súčasťou Kali D
- Doku, ukážky, riešenie problémov: https://nmap.org/
- Official Nmap reference guide https://nmap.org/book/man.html



sudo apt-get install nmap

Funkcionality nmap-u

- Dokáže zistiť:
  - Živých hostov v sieti
  - Otvorené porty
  - Bežiace služby
  - Verzie bežiacich služieb na zariadení
  - IP adresu aktívneho zariadenia
  - MAC adresu zariadenia
  - Masku siete zariadenia
  - Výrobcu
  - Model
  - Typ zariadenia
  - Typ operačného systému
  - Názov zariadenia

- Nie je potrebný privilegovaný (admin) režim pre spustenie, ale bez neho nedostaneme toľko informácií zo skenu:

  nmap [*options*] ***target***

- Rozsah skenu špecifikujeme IP adresným rozsahom (target)

- máme možnosť počas priebehu skenu, si overiť koľko % skenu je už vykonaného stlačením tlačidla “Enter”

Prepínače pre nmap

- Rýchlosť skenu:
  - Porty 1-1024 nad 1 IP add => 2,5 s
- Prepínače
  - Veľa rôznych a je možná ich kombinácia

```
spect@kali:~$ nmap scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-18 13:55 +01
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.25s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 982 filtered ports
PORT      STATE  SERVICE
22/tcp    open   ssh
25/tcp    open   smtp
80/tcp    open   http
110/tcp   open   pop3
111/tcp   closed rpcbind
119/tcp   open   nntp
143/tcp   open   imap
199/tcp   closed smux
443/tcp   closed https
465/tcp   open   smtps
563/tcp   open   snews
587/tcp   open   submission
993/tcp   open   imaps
995/tcp   open   pop3s
1720/tcp  closed h323q931
3389/tcp  closed ms-wbt-server
9929/tcp  open   nping-echo
31337/tcp open   Elite

Nmap done: 1 IP address (1 host up) scanned in 39.96 seconds
spect@kali:~$
```

Host discovery with nmap

- **-sn** prepínač

```
spect@kali:~$ nmap -sn 192.168.1.0-255
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-18 14:22 +01
Nmap scan report for 192.168.1.1
Host is up (0.0032s latency).
Nmap scan report for 192.168.1.100
Host is up (0.0090s latency).
Nmap scan report for 192.168.1.101
Host is up (0.0073s latency).
Nmap scan report for 192.168.1.119
Host is up (0.0077s latency).
Nmap scan report for 192.168.1.125
Host is up (0.000075s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.46 seconds
spect@kali:~$
```

Port scaning with nmap

- **-p** prepínač

```
spect@kali:~$ nmap -p 21-23,53,80,443 scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-18 22:13 +01
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.46s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT    STATE  SERVICE
21/tcp  closed ftp
22/tcp  open   ssh
23/tcp  closed telnet
53/tcp  closed domain
80/tcp  open   http
443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 1.98 seconds
spect@kali:~$ 
```

Stealth Scan – Tajný sken – Pasívny TCP sken

TCP SYN scan with nmap

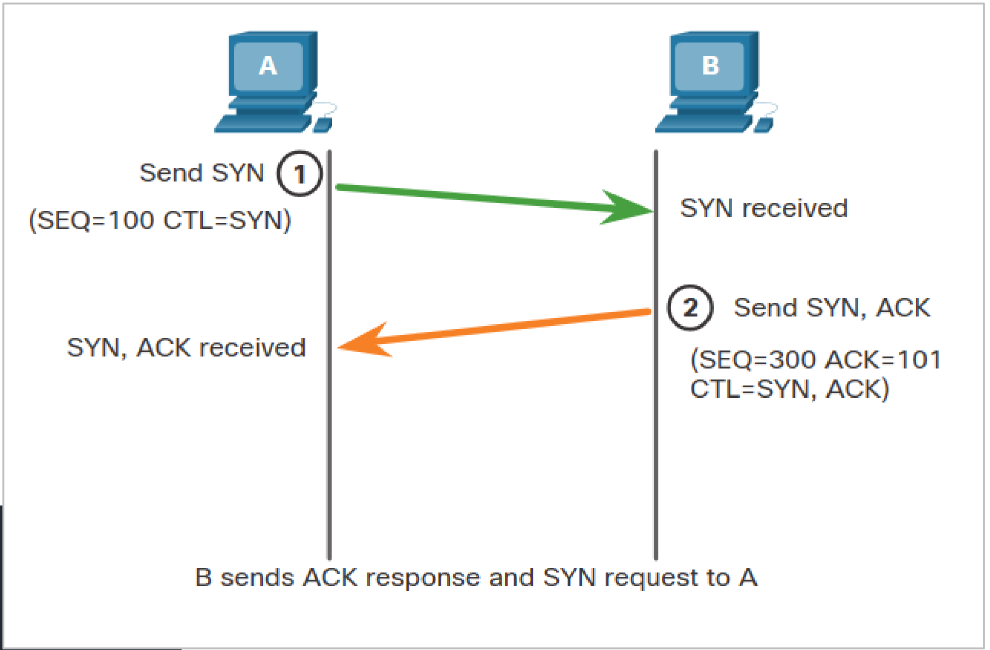
-sS prepínač

potrebné sú root/admin pr8ca

-> SYN, <- SYN, ACK, … close connection

```
root@kali:/home/spect# nmap -sS scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-18 22:40 +01
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.30s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 987 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    filtered smtp
80/tcp    open     http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
593/tcp   filtered http-rpc-epmap
1068/tcp  filtered instl_bootc
4444/tcp  filtered krb524
5800/tcp  filtered vnc-http
5900/tcp  filtered vnc
9929/tcp  open     nping-echo
31337/tcp open     Elite

Nmap done: 1 IP address (1 host up) scanned in 33.85 seconds
root@kali:/home/spect# 
```



Aktívny TCP sken

TCP connect scan with nmap

- -sT prepínač
- Nie sú potrebné admin práva

```
spect@kali:~$ nmap -sT scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-18 22:41 +01
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 987 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    filtered smtp
80/tcp    open     http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
593/tcp   filtered http-rpc-epmap
1068/tcp  filtered instl_bootc
4444/tcp  filtered krb524
5800/tcp  filtered vnc-http
5900/tcp  filtered vnc
9929/tcp  open     nping-echo
31337/tcp open     Elite

Nmap done: 1 IP address (1 host up) scanned in 232.82 seconds
spect@kali:~$
```

TCP Flag Scan with nmap

Control bits (6)

URG	ACK	PSH	RST	SYN	FIN

- ak pošleme segment bez príznaku SYN, ACK alebo RST, tak cieľový hostiteľ
  - nebude reagovať, ak je port otvorený
  - navrátii RST segment, ak je port zavretý – toto vieme využiť

-sN : NULL (All flag bits are equal to 0)
-sF : FIN (Only the FIN bit is set to 1)
-sX : Xmas (URG, PSH and FIN bits are all set to 1)

```
root@kali:/home/spect# nmap -sN scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-18 22:47 +01
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.48s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 486.78 seconds
root@kali:/home/spect# 
```

Identifikácia OS s nmap

- dokáže identifikovať OS cieľového počítača porovnaním jeho odpovedí s databázou odtlačkov – OS fingerprints

- **-O** prepínač

```
root@kali:/home/spect# nmap -O scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-18 23:13 +01
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.18s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 987 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    filtered smtp
80/tcp    open     http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
593/tcp   filtered http-rpc-epmap
1068/tcp  filtered instl_bootc
4444/tcp  filtered krb524
5800/tcp  filtered vnc-http
5900/tcp  filtered vnc
9929/tcp  open     nping-echo
31337/tcp open     Elite
Device type: general purpose
Running (JUST GUESSING): Linux 3.X|4.X (85%)
OS CPE: cpe:/o:linux:linux_kernel:3.8 cpe:/o:linux:linux_kernel:4.4
Aggressive OS guesses: Linux 3.8 (85%), Linux 4.4 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 22 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/
 .
Nmap done: 1 IP address (1 host up) scanned in 44.53 seconds
root@kali:/home/spect# 
```

Identifikácia verzií s nmap

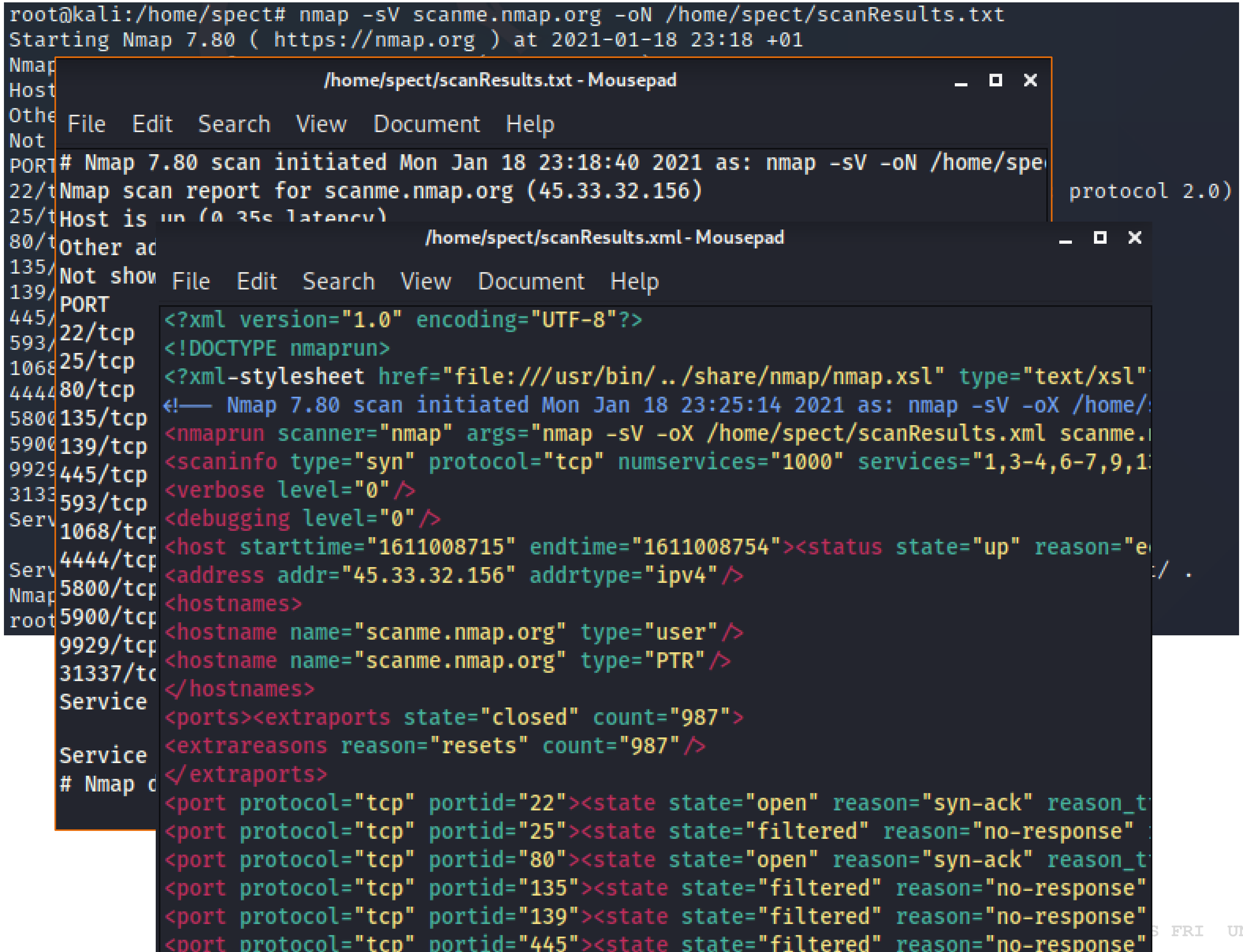
- **-sV** prepínač

```
root@kali:/home/spect# nmap -sV scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-18 23:16 +01
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.21s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 987 closed ports
PORT      STATE    SERVICE        VERSION
22/tcp    open     ssh            OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2
.0)
25/tcp    filtered smtp
80/tcp    open     http           Apache httpd 2.4.7 ((Ubuntu))
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
593/tcp   filtered http-rpc-epmap
1068/tcp  filtered instl_bootc
4444/tcp  filtered krb524
5800/tcp  filtered vnc-http
5900/tcp  filtered vnc
9929/tcp  open     nping-echo     Nping echo
31337/tcp open     tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.32 seconds
root@kali:/home/spect#
```

Modifikácia výstupu v nmap

- Pre modifikáciu výstupu:
  - **-oN** pre normálny formát
  - **–oX** pre XML formát



sudo nmap -A 158.193.139.100

```
Nmap scan report for b303-teacher.netlab.kis.fri.uniza.sk (158.193.139.100)
Host is up (0.00081s latency).
Not shown: 994 filtered ports
PORT    STATE SERVICE       VERSION
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
443/tcp open  ssl/https     VMware Workstation SOAP API 15.1.0
| fingerprint-strings:
|   SIPOptions:
|     HTTP/1.1 400 Bad Request
|     Date: Sat, 4 Dec 2021 10:07:35 GMT
|     Connection: close
|     Content-Type: text/html
|     Content-Length: 50
|_    <HTML><BODY><H1>400 Bad Request</H1></BODY></HTML>
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
| ssl-cert: Subject: commonName=VMware/countryName=US
| Not valid before: 2016-10-11T08:30:54
|_Not valid after:  2017-10-11T08:30:54
|_ssl-date: TLS randomness does not represent time
```

-A:
enable OS detection,
version detection,
script scanning,
and traceroute

sudo nmap -A 158.193.139.100

```
| vmware-version:
|   Server version: VMware Workstation 15.1.0
|   Build: 13591040
|   Locale version: INTL
|   OS type: win32-x86
|_  Product Line ID: ws
445/tcp open  microsoft-ds?
902/tcp open  ssl/vmware-auth VMware Authentication Daemon 1.10 (Uses VNC, SOAP)
912/tcp open  vmware-auth     VMware Authentication Daemon 1.0 (Uses VNC, SOAP)
MAC Address: 40:8D:5C:C1:36:86 (Giga-byte Technology)
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP|7|2008 (87%)
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2
Aggressive OS guesses: Microsoft Windows XP SP2 (87%), Microsoft Windows 7 (85%), Microsoft Windows Server 2008 SP1 or Windows Server 2008 R2 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows, cpe:/o:vmware:Workstation:15.1.0
```

UDP scan with nmap

- UDP nemá 3way handshake
- Poslaný UDP paket na cieľový port nebude potvrdený ACK
- Ale využije sa iná vlastnosť ICMP: „port unreachable“
- -sU

```
root@kali:/home/spect# nmap -sU scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-18 22:43 +01
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.28s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 985 closed ports
PORT       STATE         SERVICE
68/udp     open|filtered dhcpc
69/udp     open|filtered tftp
123/udp    open          ntp
135/udp    open|filtered msrpc
137/udp    open|filtered netbios-ns
138/udp    open|filtered netbios-dgm
139/udp    open|filtered netbios-ssn
443/udp    open|filtered https
445/udp    open|filtered microsoft-ds
593/udp    open|filtered http-rpc-epmap
1433/udp   open|filtered ms-sql-s
1434/udp   open|filtered ms-sql-m
8181/udp   open|filtered unknown
27444/udp  open|filtered Trinoo_Bcast
31337/udp  open|filtered BackOrifice

Nmap done: 1 IP address (1 host up) scanned in 1128.73 seconds
root@kali:/home/spect#
```

Hping3

- sieťový nástroj schopný odosielať vlastné pakety a zobrazovať cieľové odpovede
- Účel - testovať pravidlá firewall, vykonávať spoofing portov, simulovať útoky...
- Nedokáže skenovať celú sieť, len po jednotlivých IP adresách
  - buď musíme poznať IP adresu cieľa
  - alebo pomocou iných nástrojov ako netdiscover, nmap ju najskôr nájsť
- Po oskenovaní konkrétnej IP adresy, môžeme dostať nasledovné údaje:
  - Otvorené porty
  - Počet odoslaných paketov
  - Systémový uptime
- Ak sa pomýlime v syntaxi, nástroj nevypíše žiadny hint a musí byť striktne dodržaná
  - Náročný na používanie, ale rozsiahly nástroj
- Je potrebný privilegovaný režim pre spustenie
- Na to ako je náročný tak je pomerne rozsiahly, hlavne vďaka jeho možnosti

Hping3

- **hping3 -scan 1-512 -S 158.193.139.100**
- **–scan** znamená **ktoré porty** ideme skenovať a **-S** reprezentuje **SYN flag**

```
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
  139 netbios-ssn: .S..A... 128 58100  8192    46
  443 https      : .S..A... 128 58356 65392    46
  445 microsoft-d: .S..A... 128 58612 65392    46
  135 epmap      : .S..A... 128 58868 65392    46
All replies received. Done.
```

Masscan

- Jeho použitie (parametre, výstup) sa môže podobať na Nmap
  - ale má jedno z najlepších intuitívnych rozhraní
  - no nemá toľko možností, a ani nie je tak rozšírený a populárny
- Jedným z jeho najväčších zameraní je kontrola bannerov.
- Zvládne skenovať ľubovoľný rozsah siete, aj jednotlivé IP adresy
- Dokážeme zistiť nasledovné informácie zo skenu:
  - Zistenie živých hostov v sieti
  - Zistenie otvorených portov
  - Banner informácie
- Upozorní na syntaktické chyby

Masscan

- modifikovať výstup:
  -oG ktorý reprezentuje grapable formát
  -oX ktorý je XML formát
  -oJ reprezentuje JSon formát
- Jeho záťaž na procesor je takmer minimálna

masscan -p1-512 158.193.139.100 --rate 1000000

-p reprezentuje **ktoré porty chceme skenovať**
-rate reprezentuje **počet paketov za sekundu,**
predvolene je nastavené 100paketov/sekundu

```
Scanning 1 hosts [512 ports/host]
Discovered open port 443/tcp on 158.193.139.100
Discovered open port 445/tcp on 158.193.139.100
Discovered open port 135/tcp on 158.193.139.100
Discovered open port 139/tcp on 158.193.139.100
```

Module 12
Network Security Infrastructure

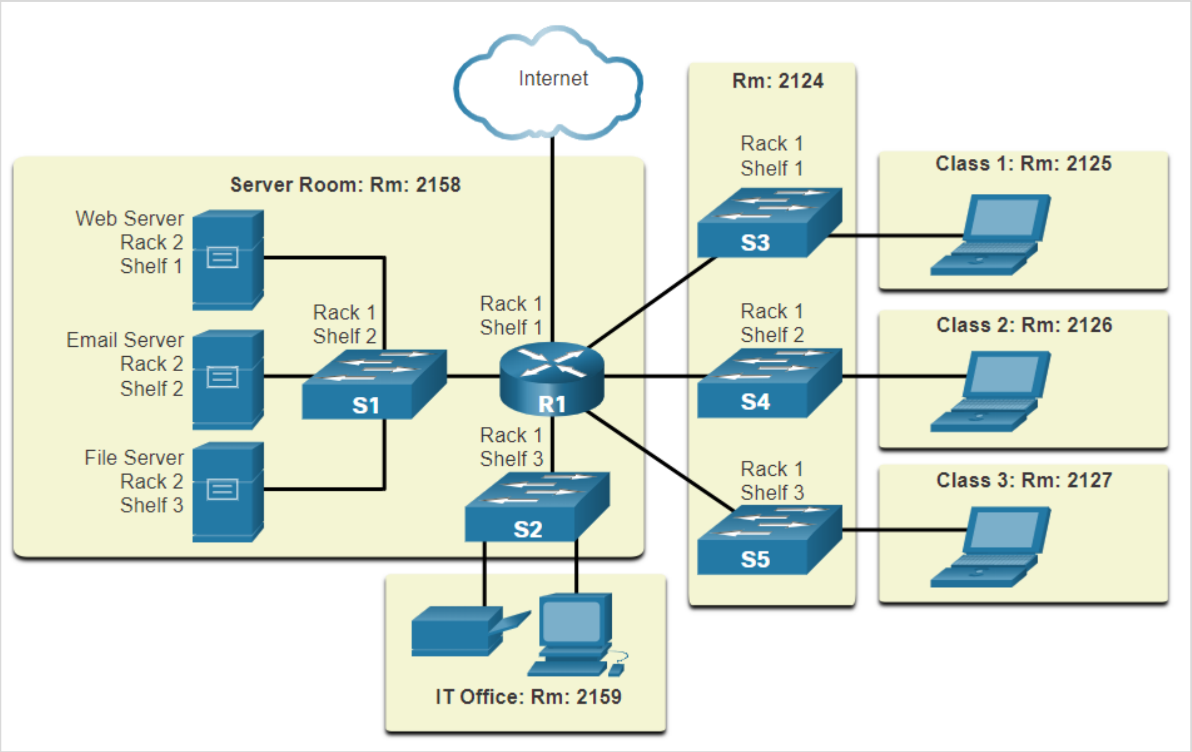
Module Objective: Explain how devices and services are used to enhance network security.

Topic Title	Topic Objective
Network Topologies	Explain how network designs influence the flow of traffic through the network.
Security Devices	Explain how specialized devices are used to enhance network security.
Security Services	Explain how network services enhance network security.

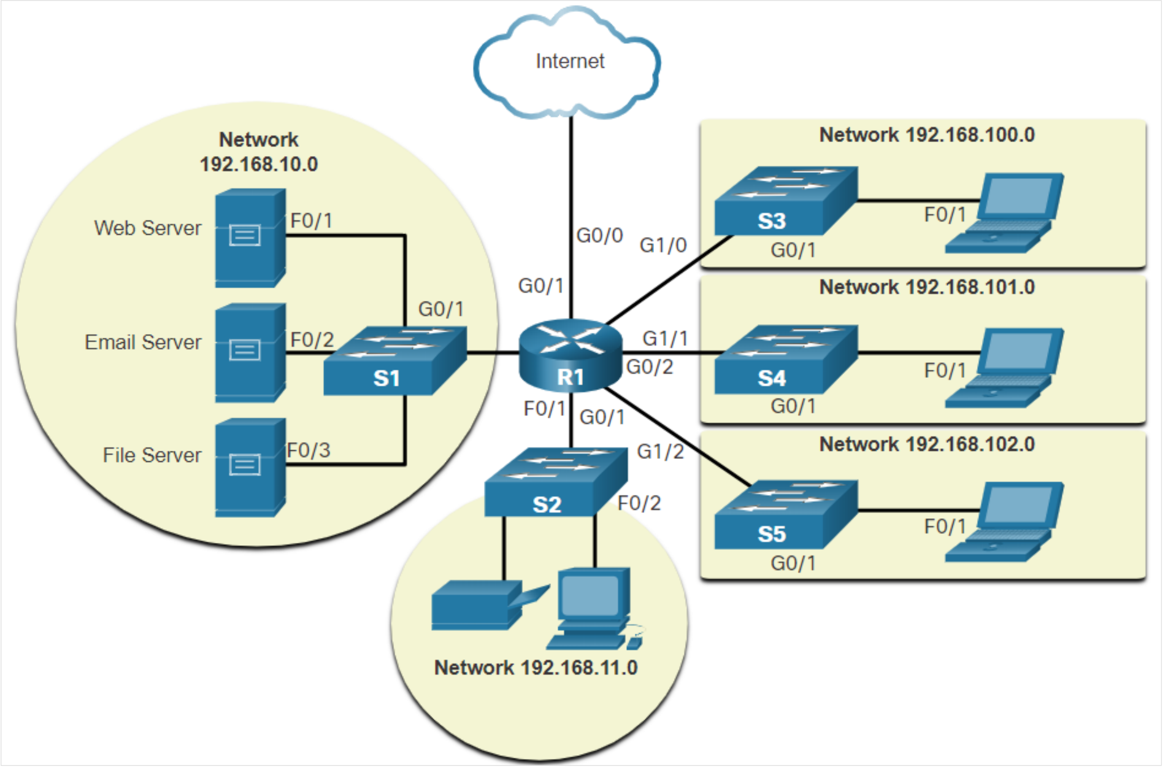
12.1 Network Topologies

Topology Diagrams

Physical topology diagrams illustrate the physical location of intermediary devices and cable installation.



Logical topology diagrams illustrate devices, ports, and the addressing scheme of the network.

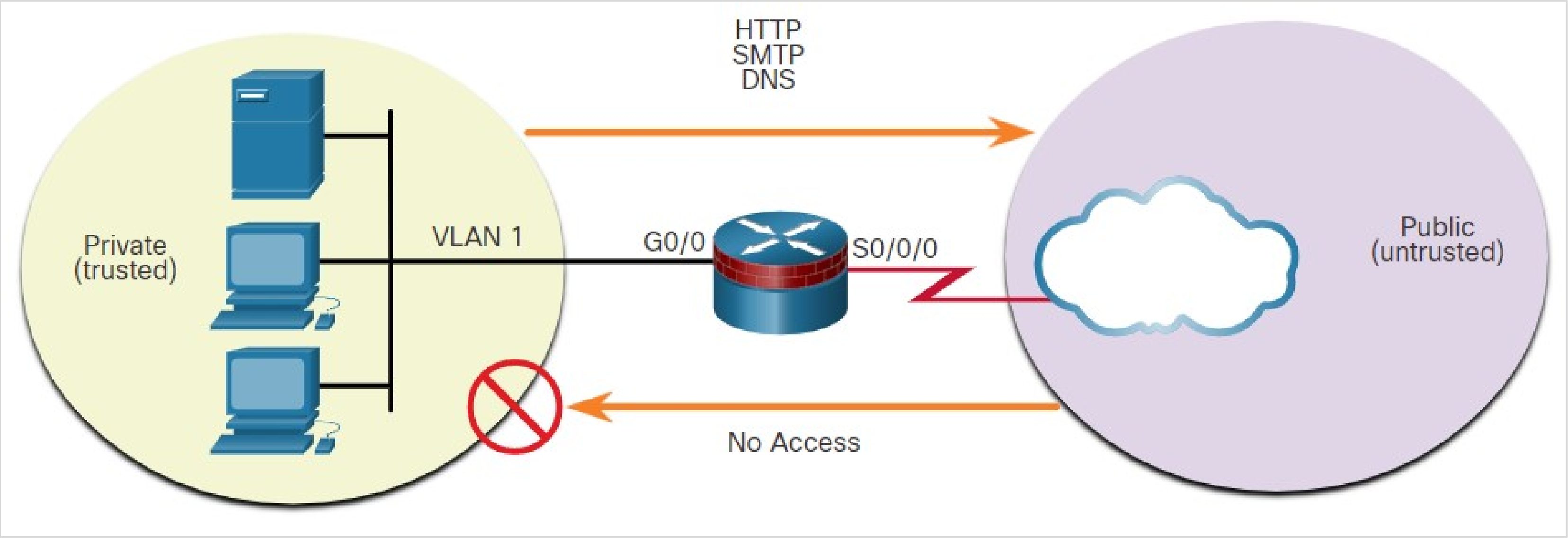


Common Security Architectures

Firewall design is primarily about device interfaces permitting or denying traffic based on the source, the destination, and the type of traffic.

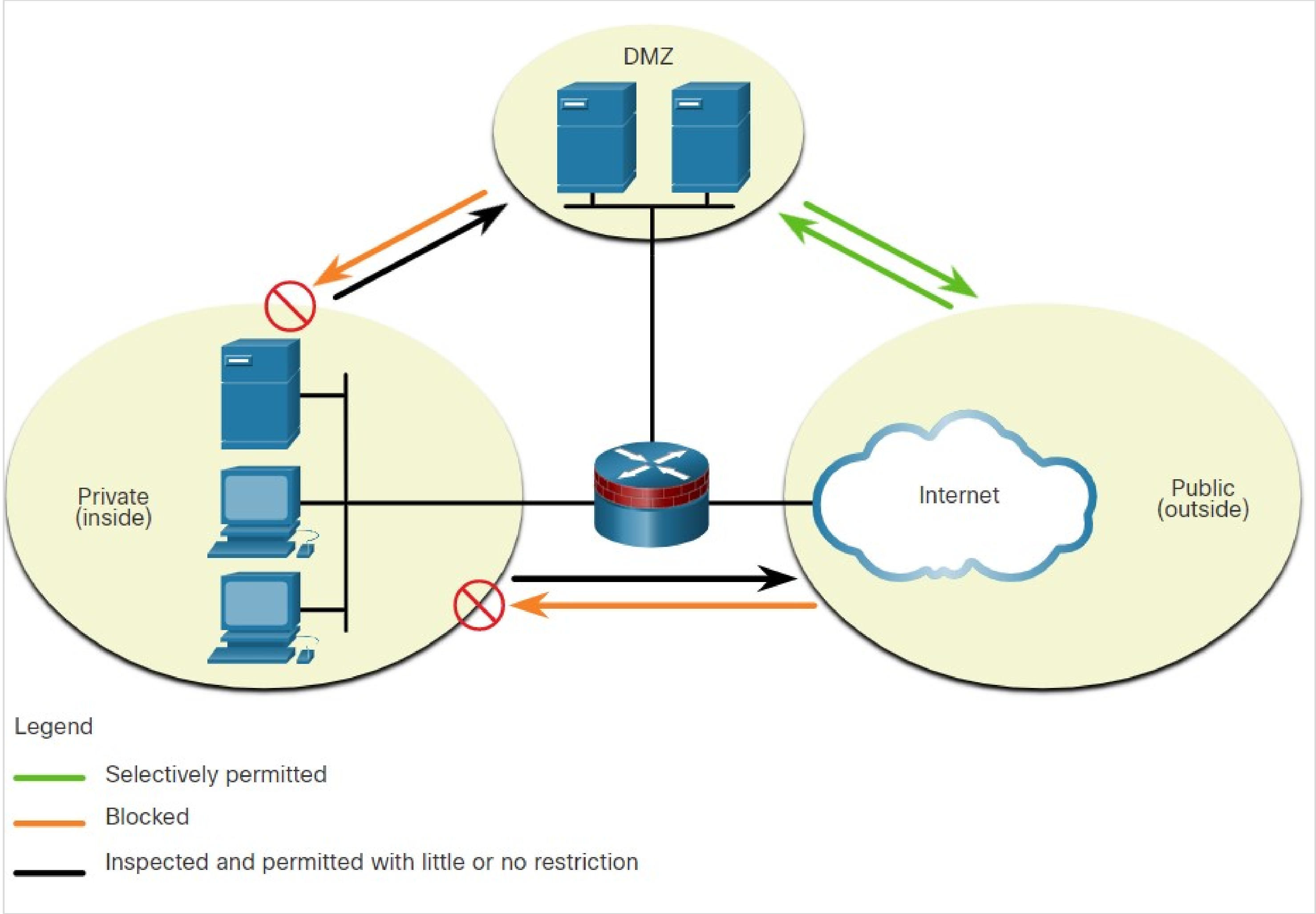
The three firewall designs are:

- **Public and Private**
  - The public network (or outside network) is untrusted, and the private network (or inside network) is trusted.



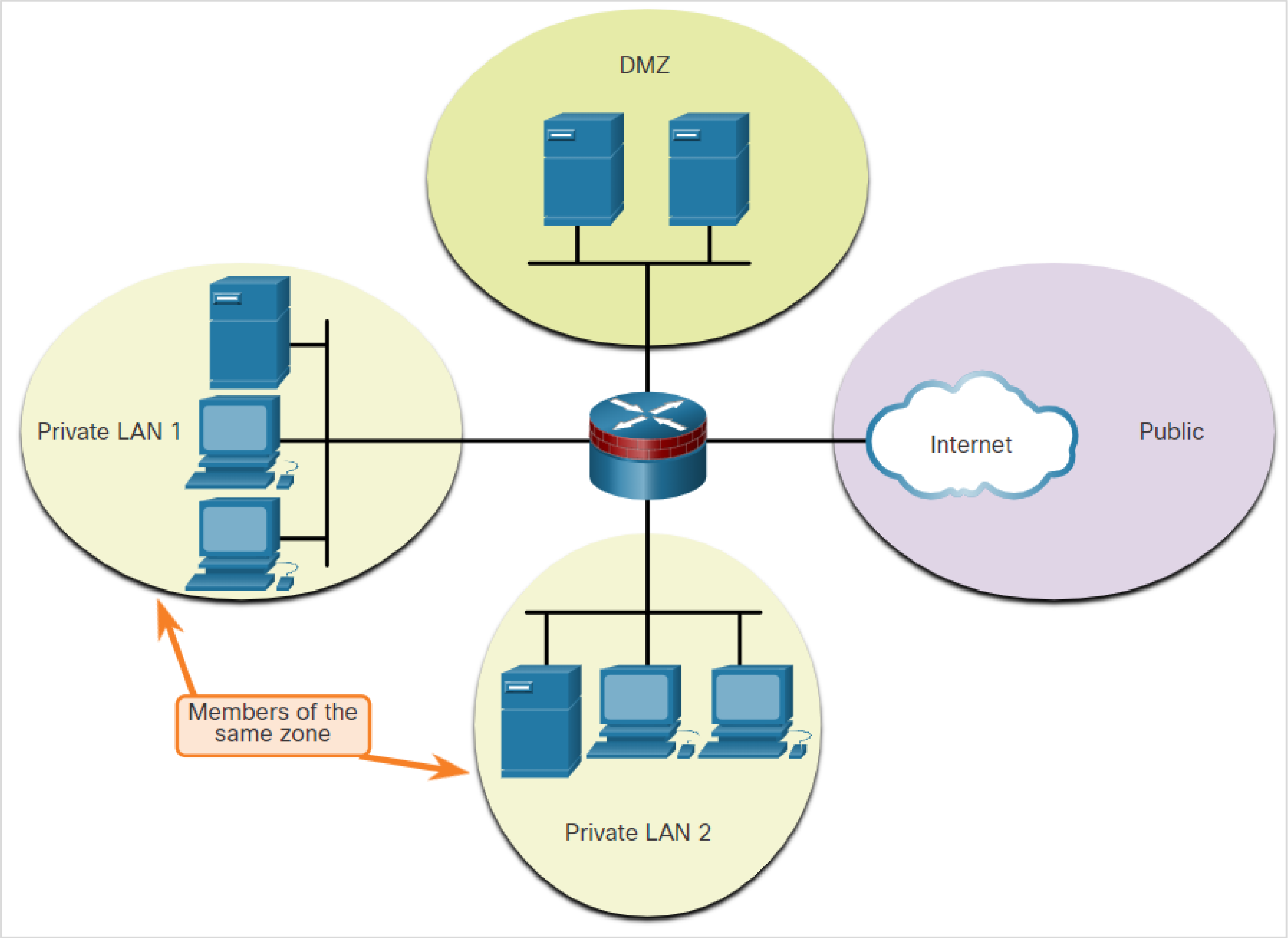
Common Security Architectures (Contd.)

- **Demilitarized Zone (DMZ)**
  - A firewall design where there is typically one:
    - Inside interface connected to the private network
    - Outside interface connected to the public network
    - DMZ interface

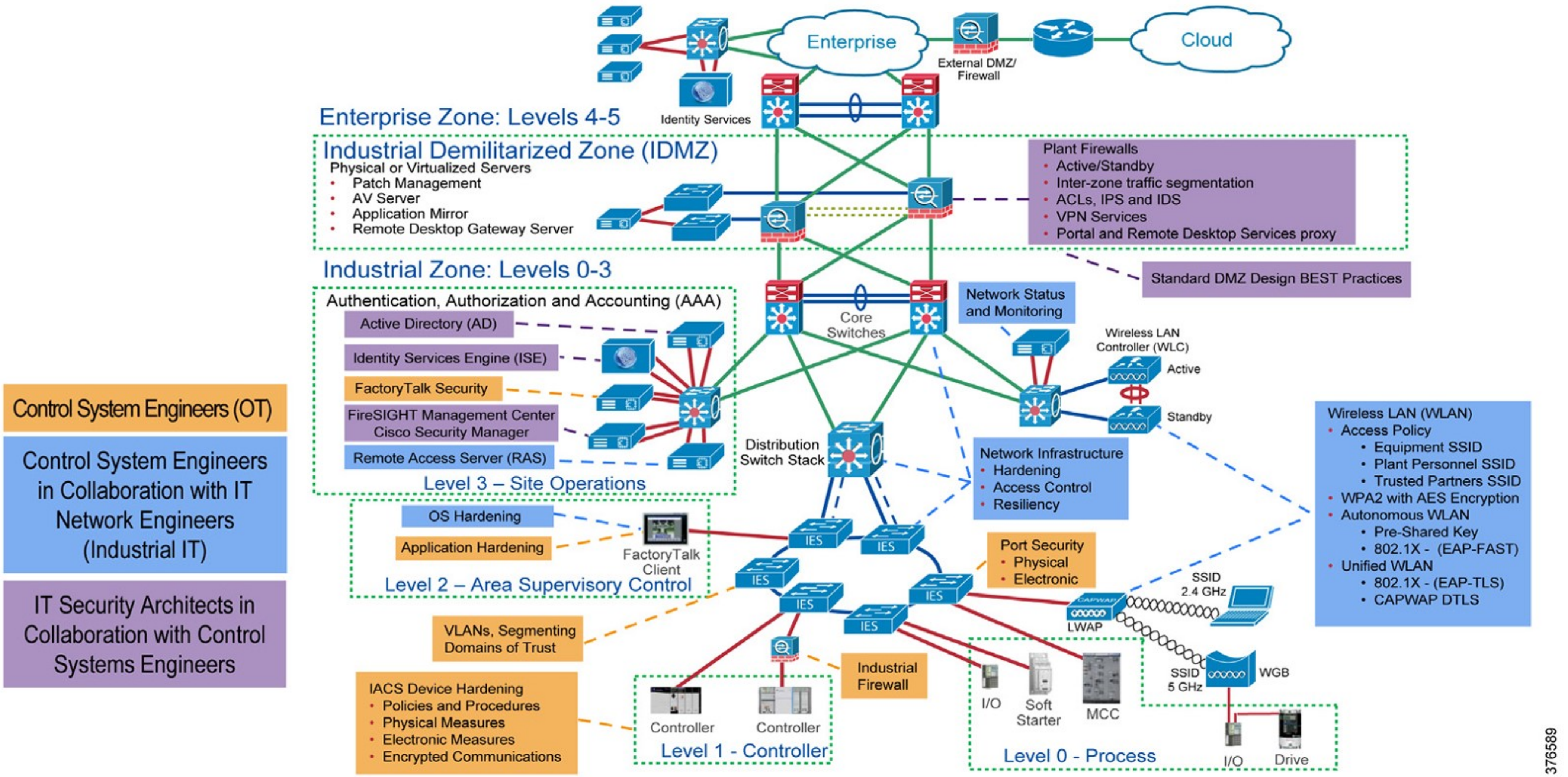


Common Security Architectures (Contd.)

- **Zone-based Policy Firewalls (ZPFs)**
  - ZPFs use the concept of zones to provide additional flexibility.
  - A zone is a group of one or more interfaces that have similar functions or features.
  - Zones help to specify where a Cisco IOS firewall rule or policy should be applied.



Common Security Architectures (Contd.)



Securely Traversing IACS Data across the Industrial Demilitarized Zone Design and Implementation Guide, March 2022

12.2 Security Devices

Firewalls

A firewall is a system, or group of systems, that enforces (*uplatňuje, vymáha*) an access control policy between networks.

Common Firewall Properties:

- Resistant to network attacks
- The only transit point between internal corporate networks and external networks because all traffic flows through the firewall
- Enforce the access control policy

Allow traffic from any external address to the web server.

Allow traffic to FTP server.

Allow traffic to SMTP server.

Allow traffic to internal IMAP server.

Deny all inbound traffic with network addresses matching internal-registered IP addresses.

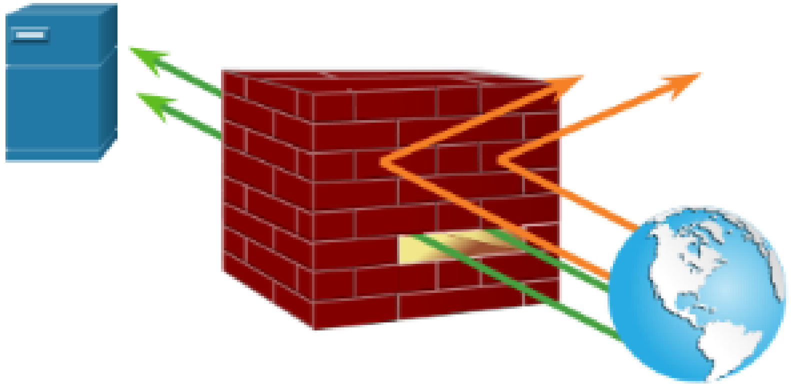
Deny all inbound traffic to server from external addresses.

Deny all inbound ICMP echo request traffic.

Deny all inbound MS Active Directory queries.

Deny all inbound traffic to MS SQL server queries.

Deny all MS Domain Local Broadcasts.



Firewalls (Contd.)

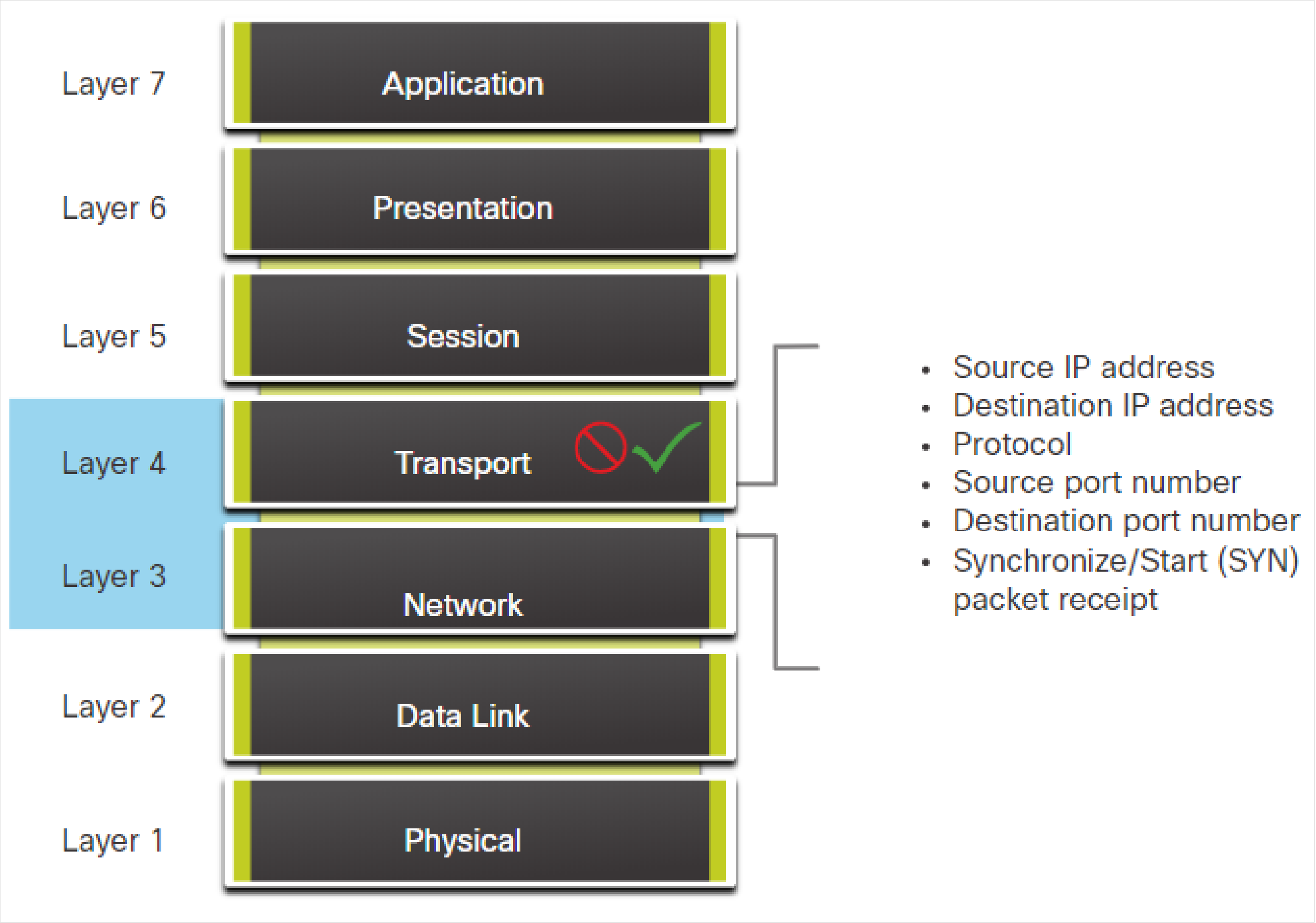
Following are the benefits and limitations of firewalls:

Firewall Benefits	Firewall Limitations
Prevent the exposure (*odhalenie*) of sensitive hosts, resources, and applications to untrusted users.	A misconfigured firewall can have serious consequences for the network, such as becoming a single point of failure.
Sanitize protocol flow, which prevents the exploitation of protocol flaws.	The data from many applications cannot be passed over firewalls securely.
Block malicious data from servers and clients.	Users might proactively search for ways around the firewall to receive blocked material, which exposes the network to potential attack.
Reduce security management complexity.	Network performance can slow down.
	Unauthorized traffic can be tunnelled or hidden as legitimate traffic through the firewall.

Firewall Type Descriptions

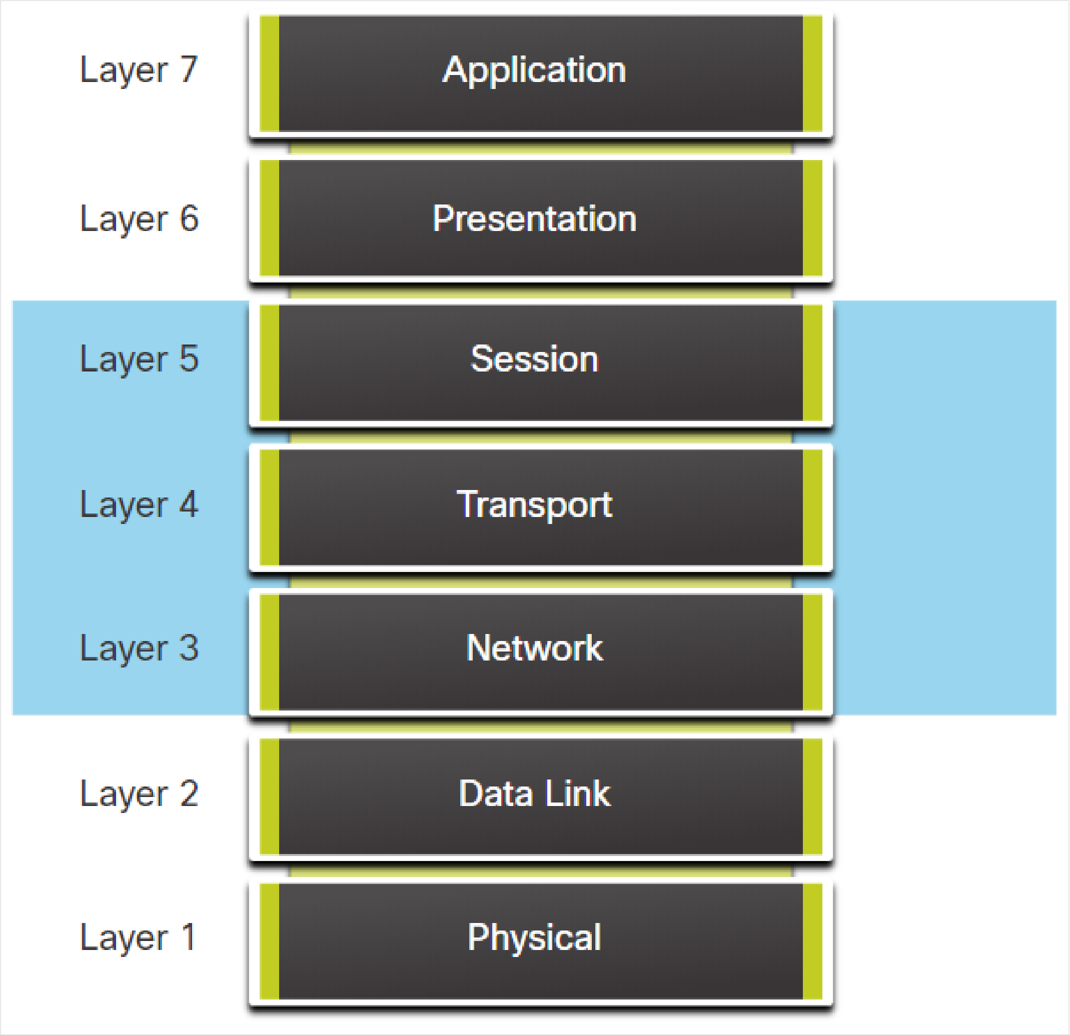
The different types of firewalls are:

- **Packet Filtering (Stateless) Firewall**
  - Packet Filtering firewalls are part of a router firewall, which permits or denies traffic based on Layer 3 and Layer 4 information.
  - They are stateless firewalls that use a simple policy table look-up that filters traffic based on specific criteria.



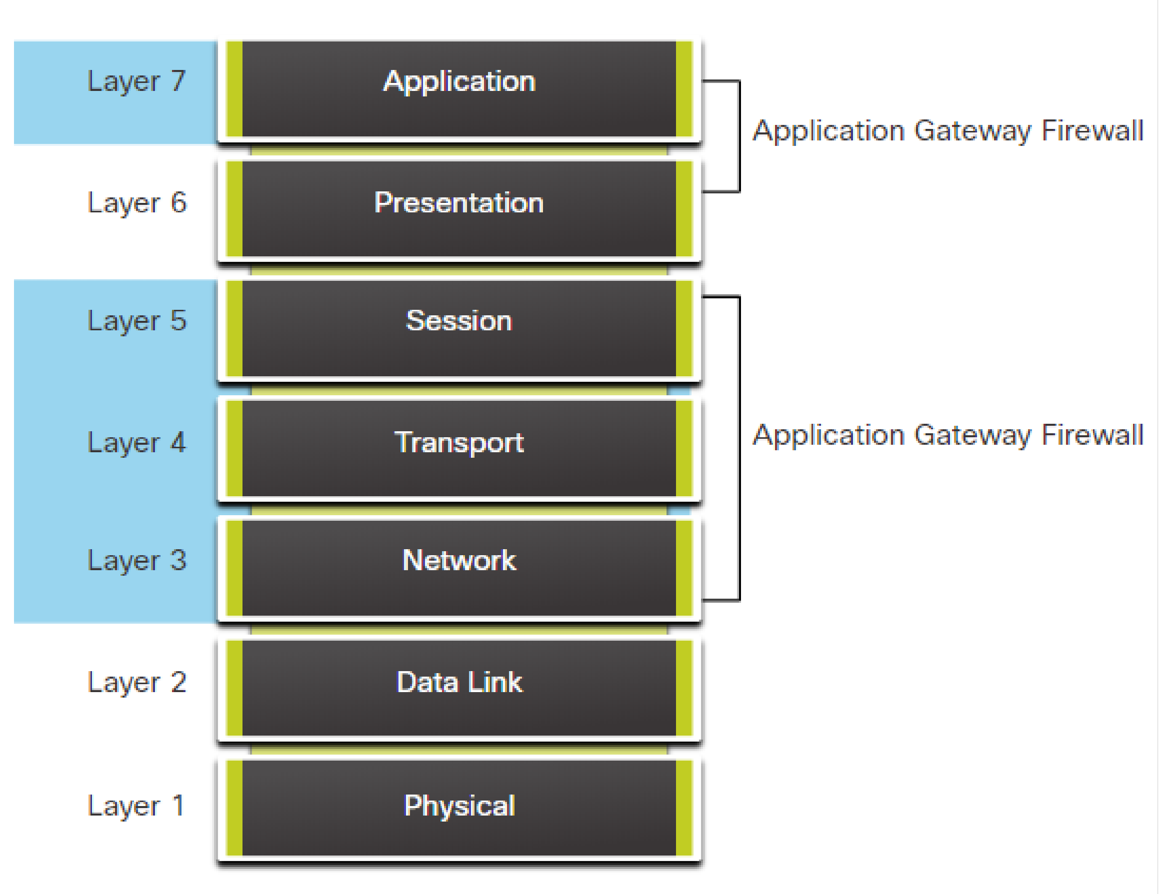
Firewall Type Descriptions (Contd.)

- **Stateful Firewalls**
  - najuniverzálnejšie a najbežnejšie používané firewallové technológie.
  - poskytujú stavové filtrovanie paketov pomocou informácií o spojeniach udržiavaných v tabuľke stavov (state table)



Firewall Type Descriptions (Contd.)

- **Application gateway firewall (proxy firewall)**
  - Application gateway firewall filters information at Layers 3, 4, 5, and 7 of the OSI reference model.
  - Most of the firewall control and filtering is done in the software.



Firewall Type Descriptions (Contd.)

- **Next-generation firewalls (NGFW)**
  - NGFW ide nad rámec stavových brán firewall tým, že poskytuje:
    - Integrovaná prevencia vniknutia
    - Povedomie o aplikáciách (application awareness) a ich kontrola na zobrazenie a blokovanie rizikových aplikácií
    - Inovujte cesty tak, aby zahŕňali budúce informačné kanály
    - Techniky na riešenie vyvíjajúcich sa bezpečnostných hrozieb

Firewall Type Descriptions (Contd.)

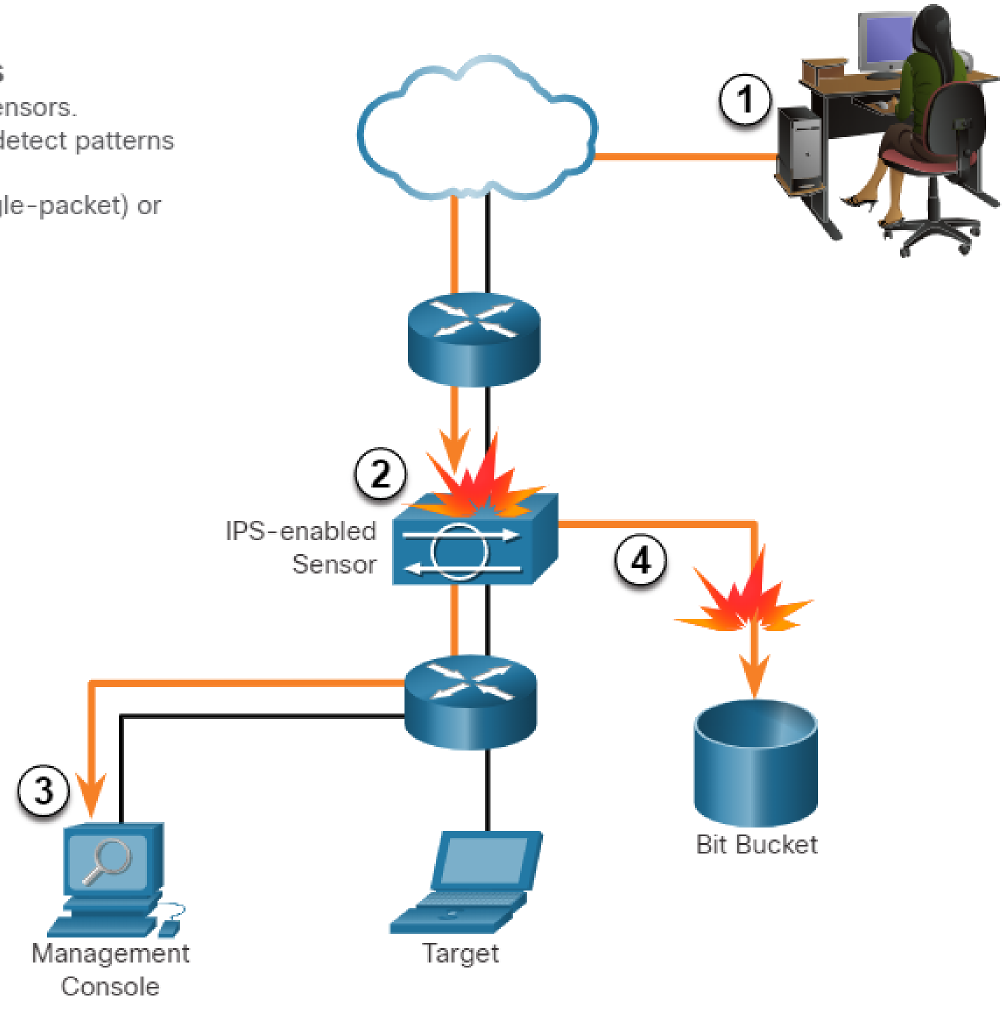
- Other methods of implementing firewalls include:
  - **Host-based (server and personal) firewall** - A PC or server with firewall software running on it.
  - **Transparent firewall** - Filters IP traffic between a pair of bridged interfaces.
  - **Hybrid firewall** - A combination of various firewall types.

Intrusion Prevention and Detection Devices

- A networking architecture paradigm shift is required to defend against fast-moving and evolving attacks. This must include cost effective and prevention systems such as:
  - Intrusion Detection Systems (IDS)
  - Intrusion Prevention Systems (IPS)
- The network architecture integrates these solutions into the entry and exit points of the network.
- The figure shows how an IPS device handles malicious traffic.

Common Characteristics of IDS and IPS

- Both technologies are deployed as sensors.
- Both technologies use signatures to detect patterns of misuse in network traffic.
- Both can detect atomic patterns (single-packet) or composite patterns (multi-packet).



1. Malicious traffic is sent to the target host that is inside the network.
2. The traffic is routed into the network and received by an IPS-enabled sensor where it is blocked.
3. The IPS-enabled sensor sends logging information regarding the traffic to the network security management console.
4. The IPS-enabled sensor kills the traffic. (It is sent to the “Bit Bucket.”)

Advantages and Disadvantages of IDS and IPS

The table lists the advantages and disadvantages of IDS and IPS:

Solution	Advantages	Disadvantages
IDS	• No Impact on network (latency, jitter) • No Network impact if there is a sensor failure • No network impact if there is sensor overload	• Response action cannot stop trigger packets • Correct tuning required for response actions • More vulnerable to network security evasion techniques
IPS	• Stops trigger packets • Can use stream normalization techniques	• Sensor issues might affect network traffic • Sensor overloading impacts the network • Some impact on network (latency, jitter)

Deployment Consideration:

- IPS and IDS technologies can complement each other.
- Deciding which implementation to use is based on the security goals of the organization as stated in their network security policy.

Types of IPS

There are two primary kinds of IPS :

- Host-based IPS
- Network-based IPS

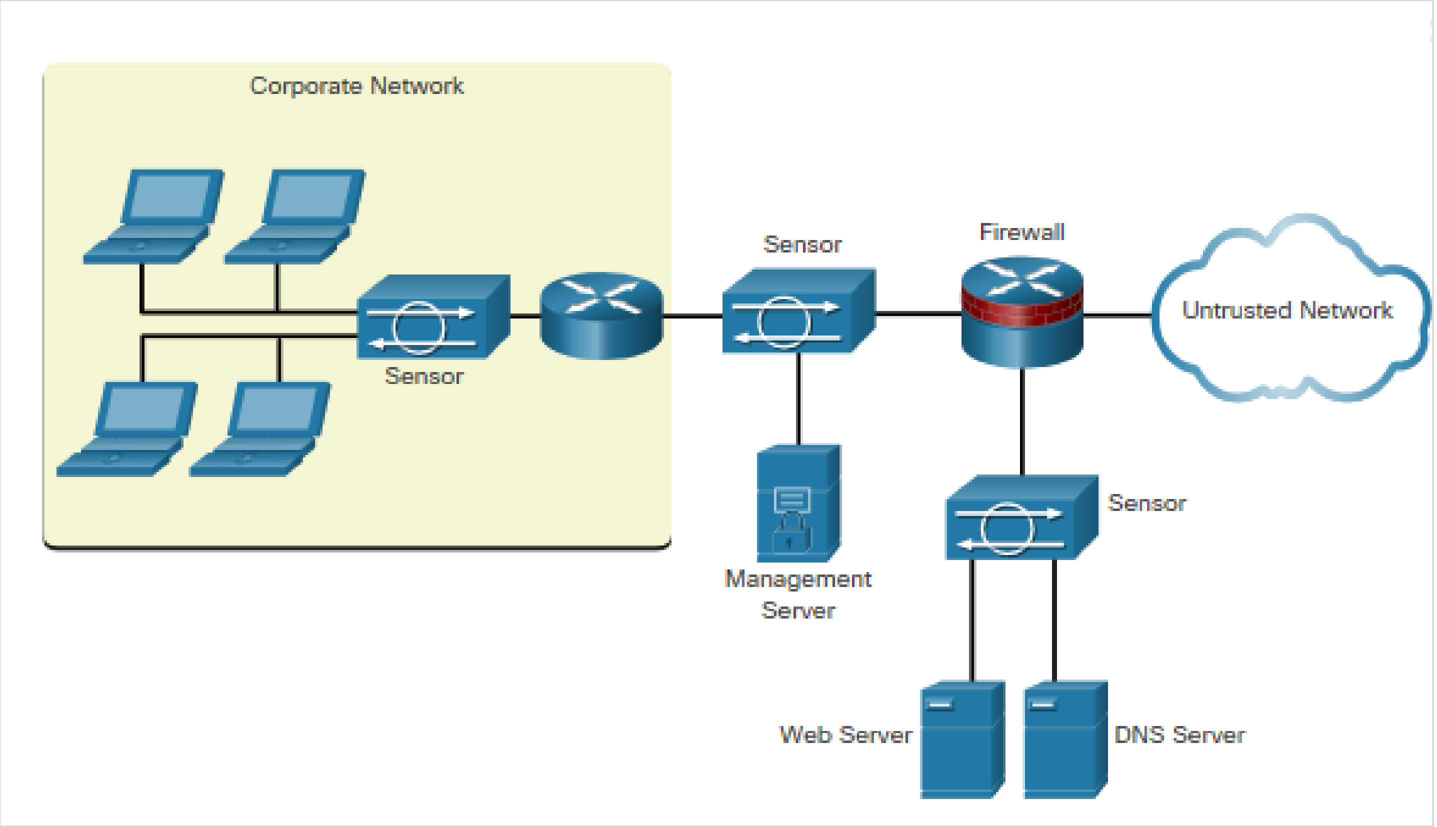
- **Host-based IPS (HIPS)**

HIPS is a software installed on a host to monitor and analyze suspicious activity.

Advantages	Disadvantages
• Provides protection specific to a host operating system • Provides operating system and application level protection • Protects the host after the message is decrypted	• Operating system dependent • Must be installed on all hosts

Types of IPS (Contd.)

- **Network-based IPS**
  - Network-based IPS are Implemented using a dedicated or non-dedicated IPS device.
  - Host-based IDS/IPS solutions are integrated with a network-based IPS implementation to ensure a robust security architecture.
  - Sensors detect malicious and unauthorized activity in real time and can take action when required.



Specialized Security Appliances

Few examples of specialized security appliances.

Cisco Advanced Malware Protection (AMP)	Cisco Web Security Appliance (WSA)	Cisco Email Security Appliance (ESA)
An enterprise-class advanced malware analysis and protection solution	A secure web gateway that combines leading protections to help organizations address the growing challenges of securing and controlling web traffic	ESA/Cisco Cloud Email Security helps to mitigate email-based threats and the ESA defends mission-critical email systems
It provides comprehensive malware protection for organizations **before**, **during**, and **after** an attack	Protects the network by automatically **blocking risky sites** and **testing unknown sites before allowing users to access** them	Constantly **updated by real-time feeds** from Cisco Talos, which detects and correlates threats using a **worldwide database monitoring system**
		Features: Global threat intelligence, Spam blocking, Advanced Malware Protection, Outbound Message Control

12.3 Security Services

Security Services

IDS

SPAN

Syslog

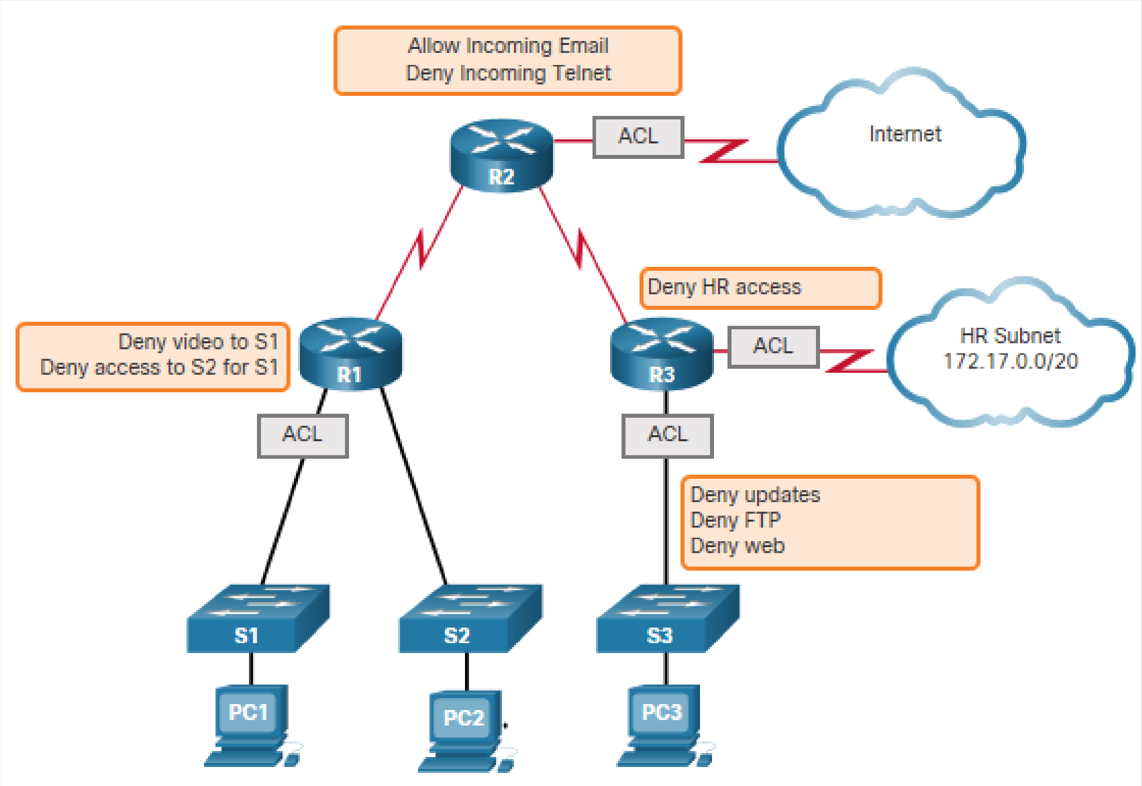
AAA

NetFlow

SNMP

Traffic Control with ACLs

- An Access Control List (ACL) is a series of commands that control whether a device forwards or drops packets based on information found in the packet header.
- When configured, ACLs perform the following tasks:
  - Limit network traffic to increase network performance.
  - Provide traffic flow control.
  - Provide basic level of security for network access.
  - Filter traffic based on traffic type.
  - Screen hosts to permit or deny access to network services.



Sample Topology with ACLs applied to routers R1, R2, and R3.

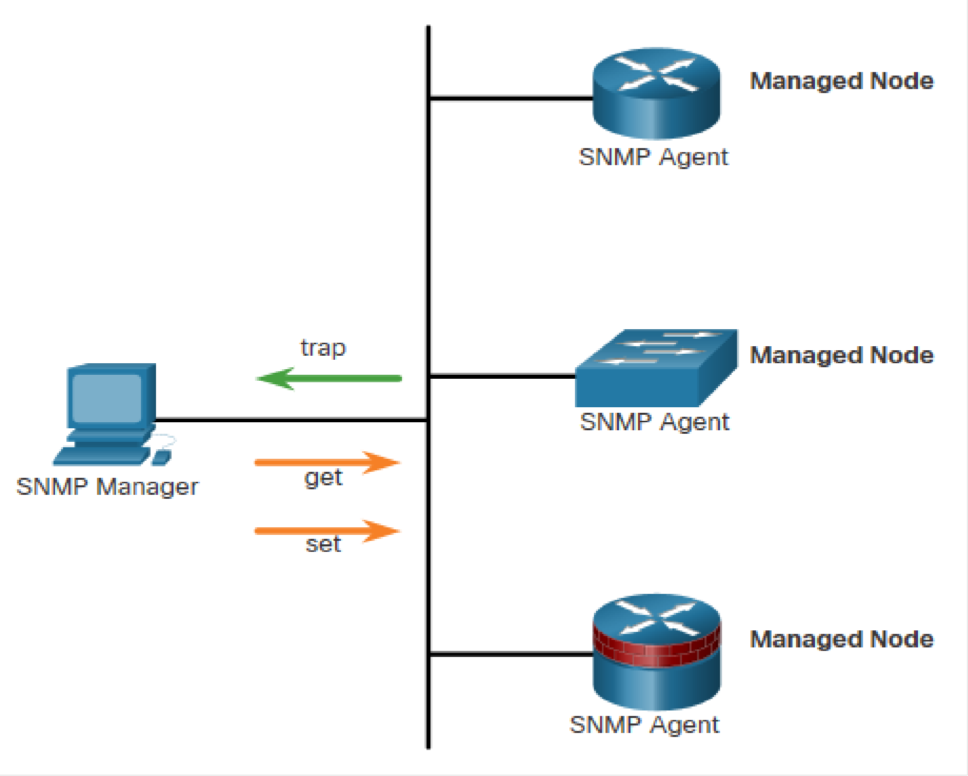
ACLs: Important Features

The two types of Cisco IPv4 ACLs are:

- **Standard ACL** - Used to permit or deny traffic only from source IPv4 addresses.
- **Extended ACL** - Filters IPv4 packets based on several attributes that include:
  - Protocol type
  - Source IPv4 address
  - Destination IPv4 address
  - Source TCP or UDP ports
  - Destination TCP or UDP ports
  - Optional protocol type information for finer control
- Standard and extended ACLs can be created using either a number or a name to identify the ACL and its list of statements.

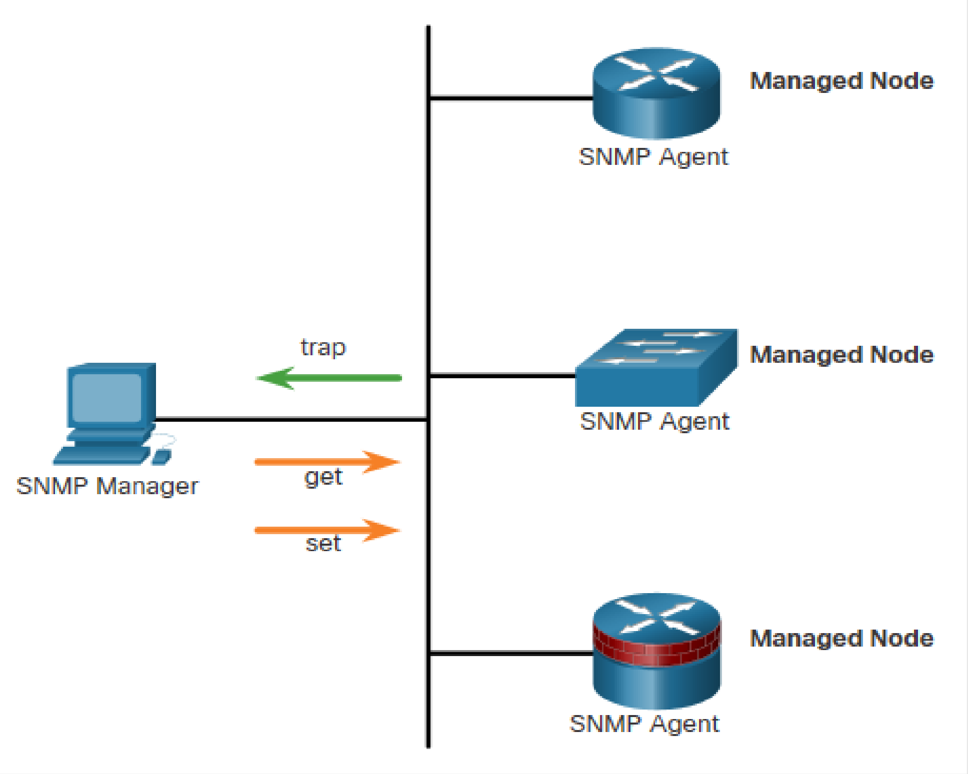
SNMP

- Simple Network Management Protocol (SNMP) is an application layer protocol that provides a message format for communication between managers and agents.
- It allows network administrators to perform the following:
  - Manage end devices such as servers, workstations, routers, switches, and security appliances, on an IP network.
  - Monitor and manage network performance.
  - Find and solve network problems.
  - Plan for network growth.
- The SNMP system consists of two elements:
  - **SNMP manager**: Runs SNMP management software.
  - **SNMP agents**: Nodes being monitored and managed.



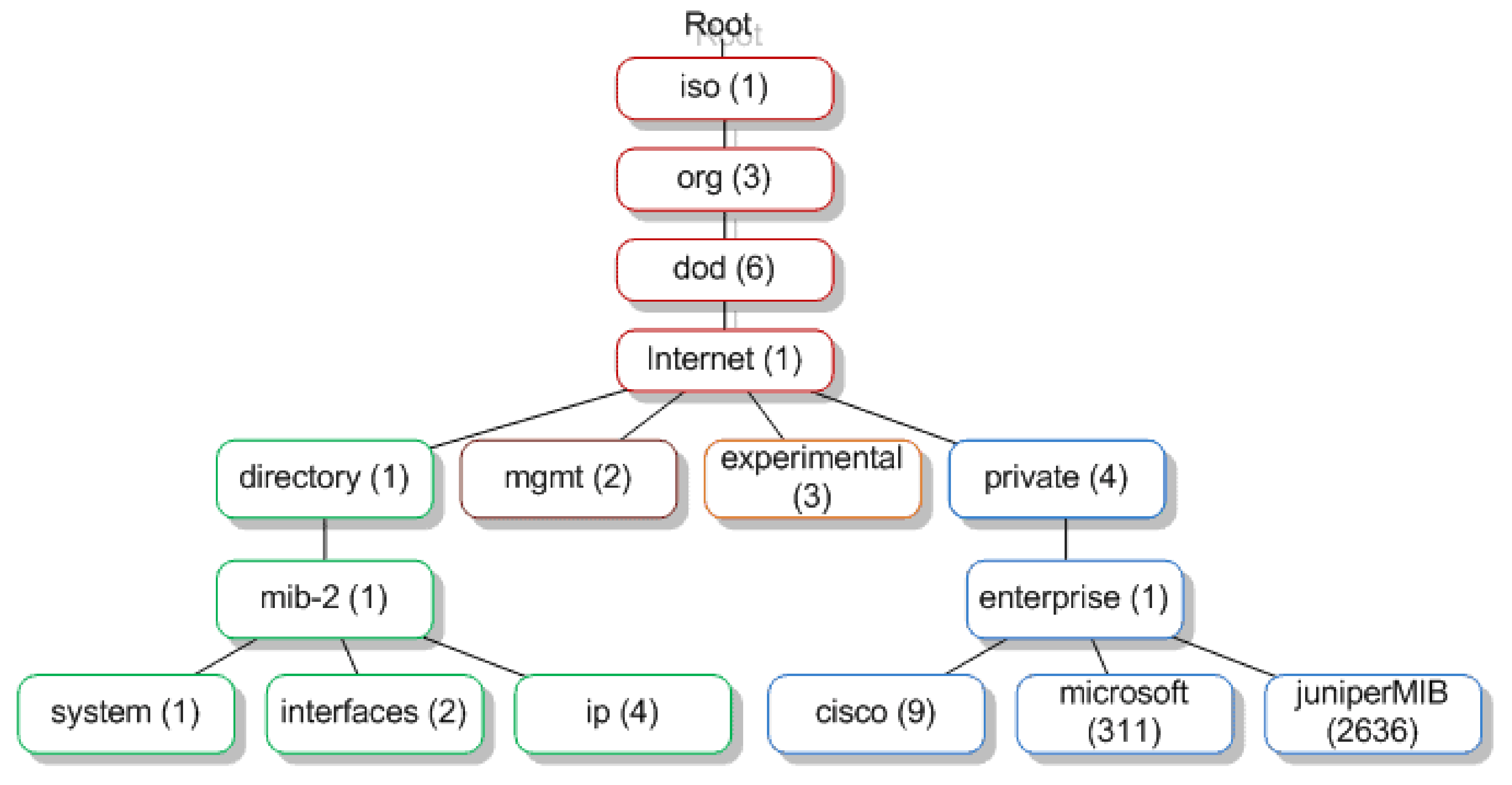
SNMP

- dá sa využiť na zbieranie informácií z danej siete
- definuje operácie nad MIB bázou, ktorá má formu stromu
  - jej jednotlivé prvky sa definujú pomocou OID
  - prvky tejto bázy sa dajú rozdeliť na 2 typy údajov:
    - **Skalárne –** jeden údaj, ktorý obsahuje informáciu na základe daného OID
    - **Tabulárne –** viacero údajov, ktoré sa nachádzajú v predkovi svojho OID
  - Toto OID určuje konkrétny údaj, napríklad IP adresa, Maska siet a podobne, alebo je to vrchol, ako napríklad systém
  - akými volaniami sa pristupuje na jednotlivé OID:
    - **snmget** – ktorý sa používa na prístup skalárnych typov
    - **snmpwalk** – ktorý sa používa primárne na prístup tabulárnych ale môže byť použitý aj na prístup skalárnych



SNMP – MIB – verejná a privátna časť SNMP MIB databázy

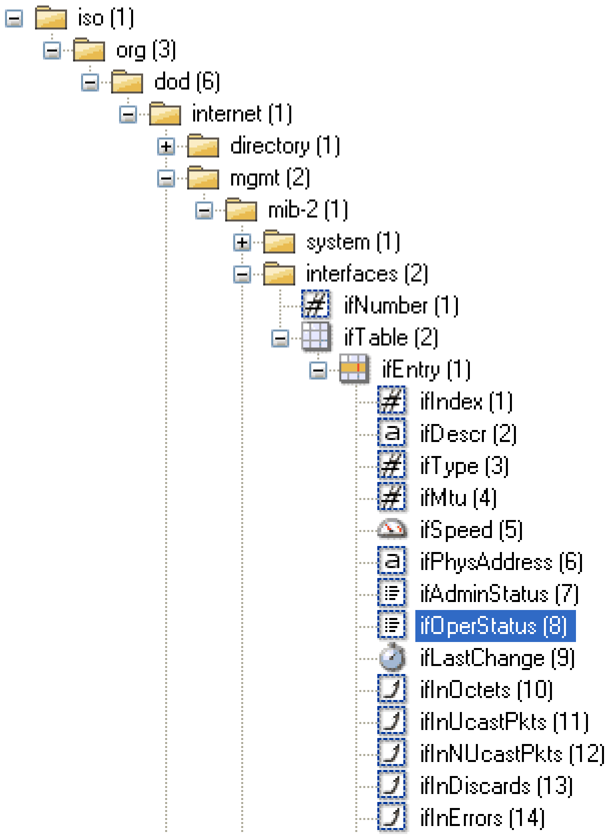
- level ORG, has the number “3”, since it is the 3rd object under ISO
- most SNMP values we’re interested in will always start with the same set of objects – 1.3.6.1
- Interface Status would have an OID of 1.3.6.1.2.1.2.2.1.8
- MIB is like a translator that helps a Management Station to understand SNMP responses obtained from your network devices
- All SNMP devices generally support MIB-2, which is a standard set of objects that can be monitore



OID Tree Example

SNMP – MIB files (not readable)

- Most network management software has the ability to display the OID tree in some way



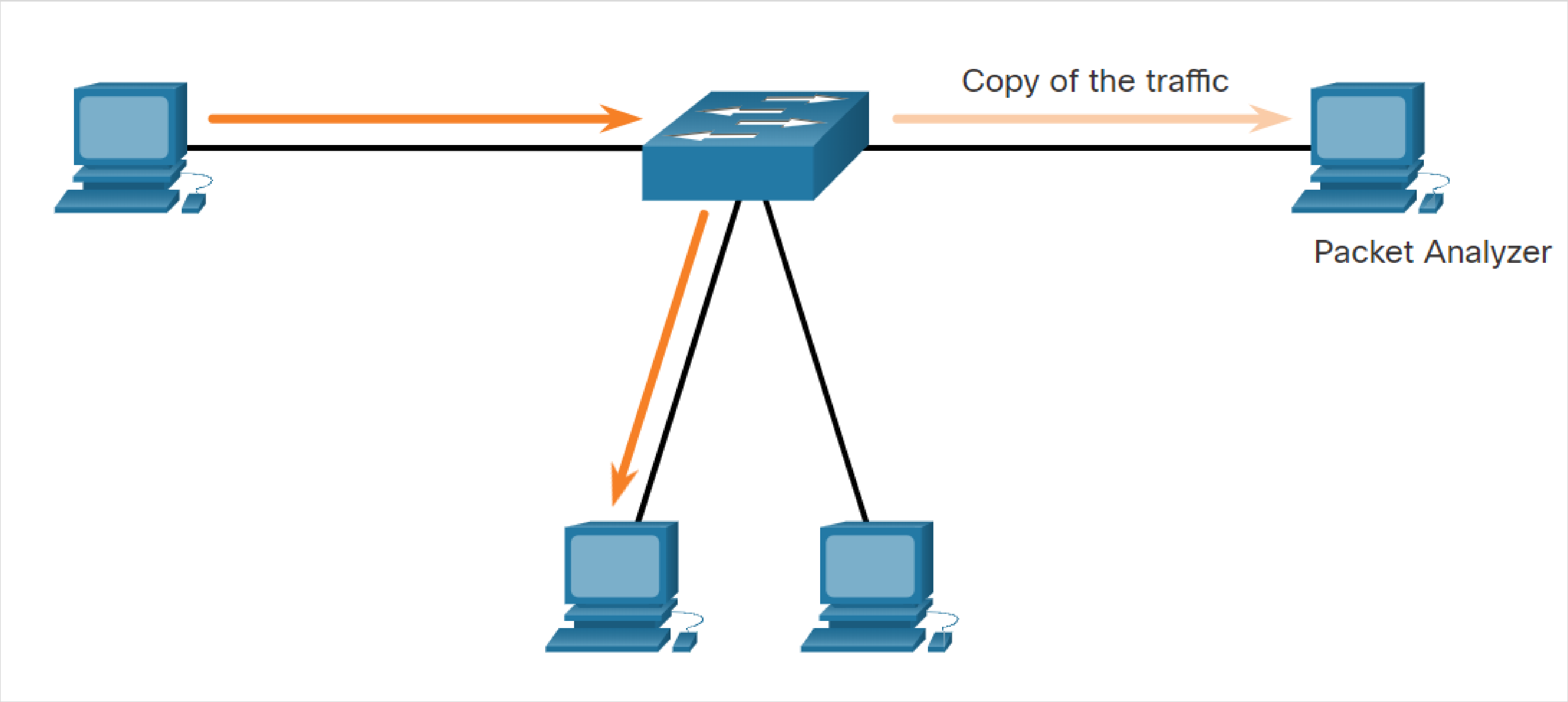
- MIB files themselves are difficult to read, they are only meant to be imported, or “compiled” by a Management Station

```
ciscoEnvMonObjects 2 }▯▯ciscoEnvMonVoltageStatusEntry
OBJECT-TYPE▯        SYNTAX     CiscoEnvMonVoltageStatusEntry▯
    MAX-ACCESS not-accessible▯          STATUS     current▯
DESCRIPTION▯                   "An entry in the voltage status
table, representing the status▯                  of the associated
testpoint maintained by the environmental▯
monitor."▯         INDEX      { ciscoEnvMonVoltageStatusIndex }▯
      ::= { ciscoEnvMonVoltageStatusTable 1 }▯▯
CiscoEnvMonVoltageStatusEntry ::=▯        SEQUENCE {▯
    ciscoEnvMonVoltageStatusIndex   Integer32 (0..2147483647),▯
              ciscoEnvMonVoltageStatusDescr   DisplayString,▯
            ciscoEnvMonVoltageStatusValue   CiscoSignedGauge,▯
             ciscoEnvMonVoltageThresholdLow  Integer32,▯
       ciscoEnvMonVoltageThresholdHigh Integer32,▯
 ciscoEnvMonVoltageLastShutdown  Integer32,▯
ciscoEnvMonVoltageState         CiscoEnvMonState▯        }▯▯
ciscoEnvMonVoltageStatusIndex OBJECT-TYPE▯       SYNTAX
Integer32 (0..2147483647)▯       MAX-ACCESS not-accessible▯
   STATUS     current▯        DESCRIPTION▯
"Unique index for the testpoint being instrumented.▯
   This index is for SNMP purposes only, and has no▯
   intrinsic meaning."▯        ::= {
```

Cisco provides a useful resource for looking up OID values, and downloading MIB files for any of their devices. [You can access it here.]

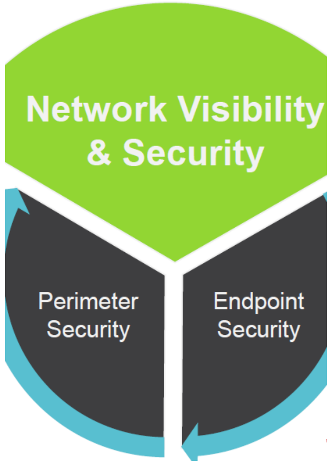
Port Mirroring

Port mirroring is a feature that allows a switch to make duplicate copies of traffic passing through a switch, and then sending it out a port with a network monitor attached.

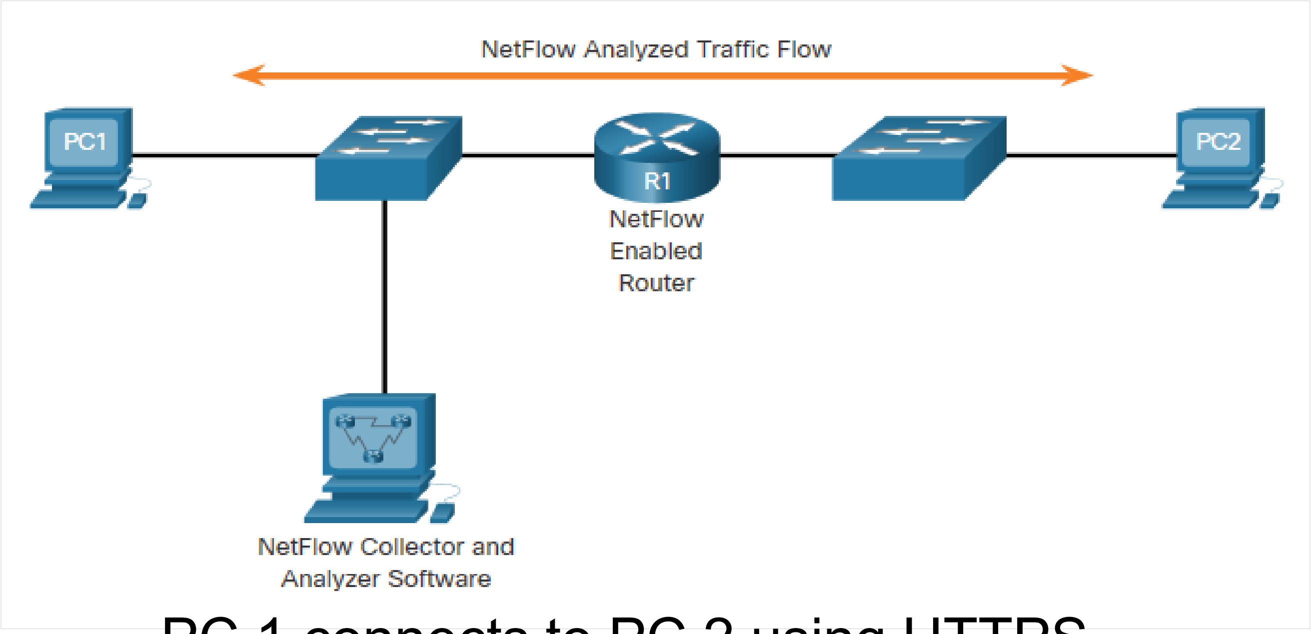


Traffic Sniffing Using a Switch

NetFlow



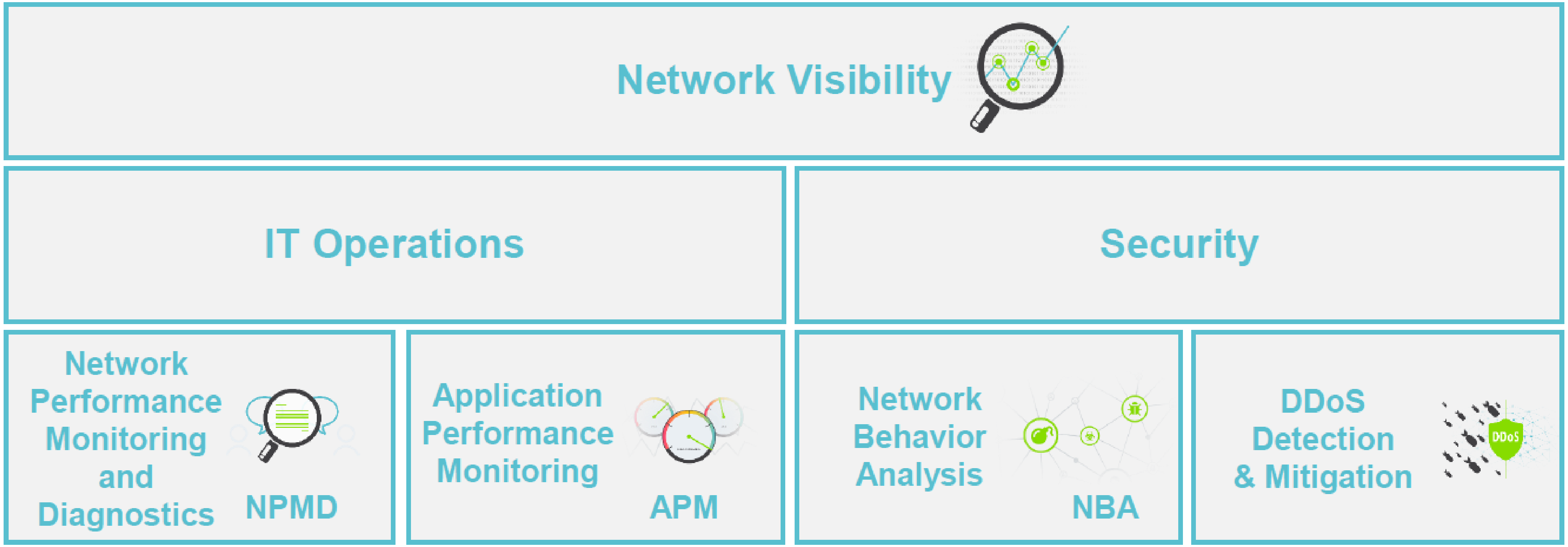
- NetFlow is a feature that was introduced on Cisco routers around 1996 that provides the ability to collect IP network traffic as it enters or exits an interface.
- NetFlow provides data to enable:
  - network and security monitoring,
    - NPMD Network Performance Monitoring & Diagnostics
    - Network visibility & security
      - Perimeter Security
      - Endpoint Security
  - network planning
  - traffic analysis to include identification of network bottlenecks
  - IP accounting for billing purposes.



PC 1 connects to PC 2 using HTTPS

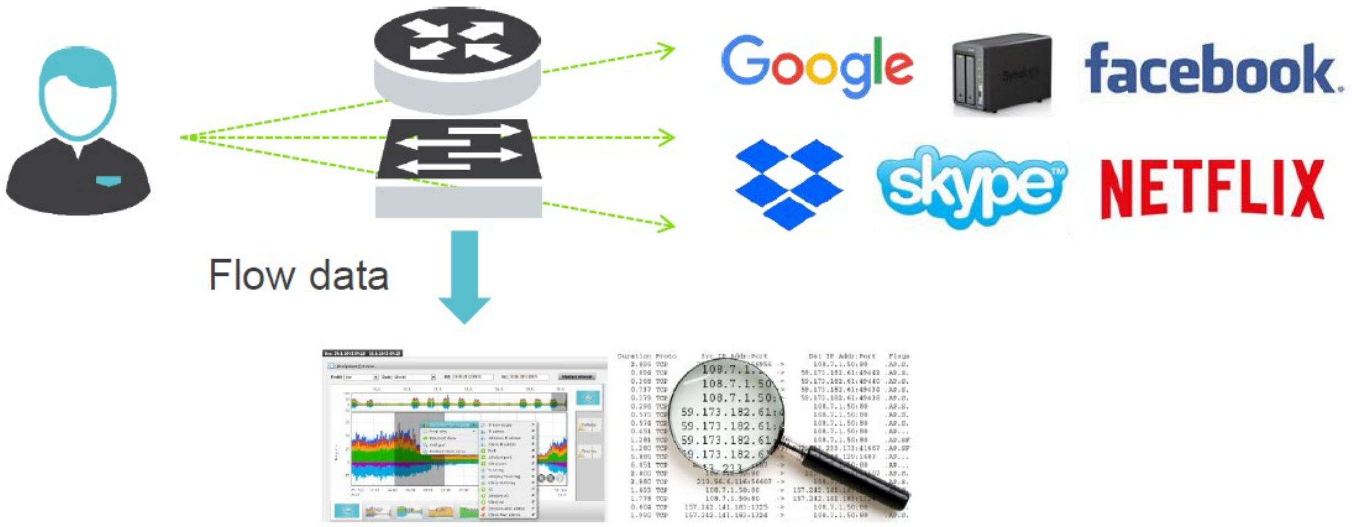
NetFlow

- NetFlow can monitor application connection, tracking byte and packet counts for that individual application flow.
- It then pushes the statistics over to an external server called a NetFlow collector.

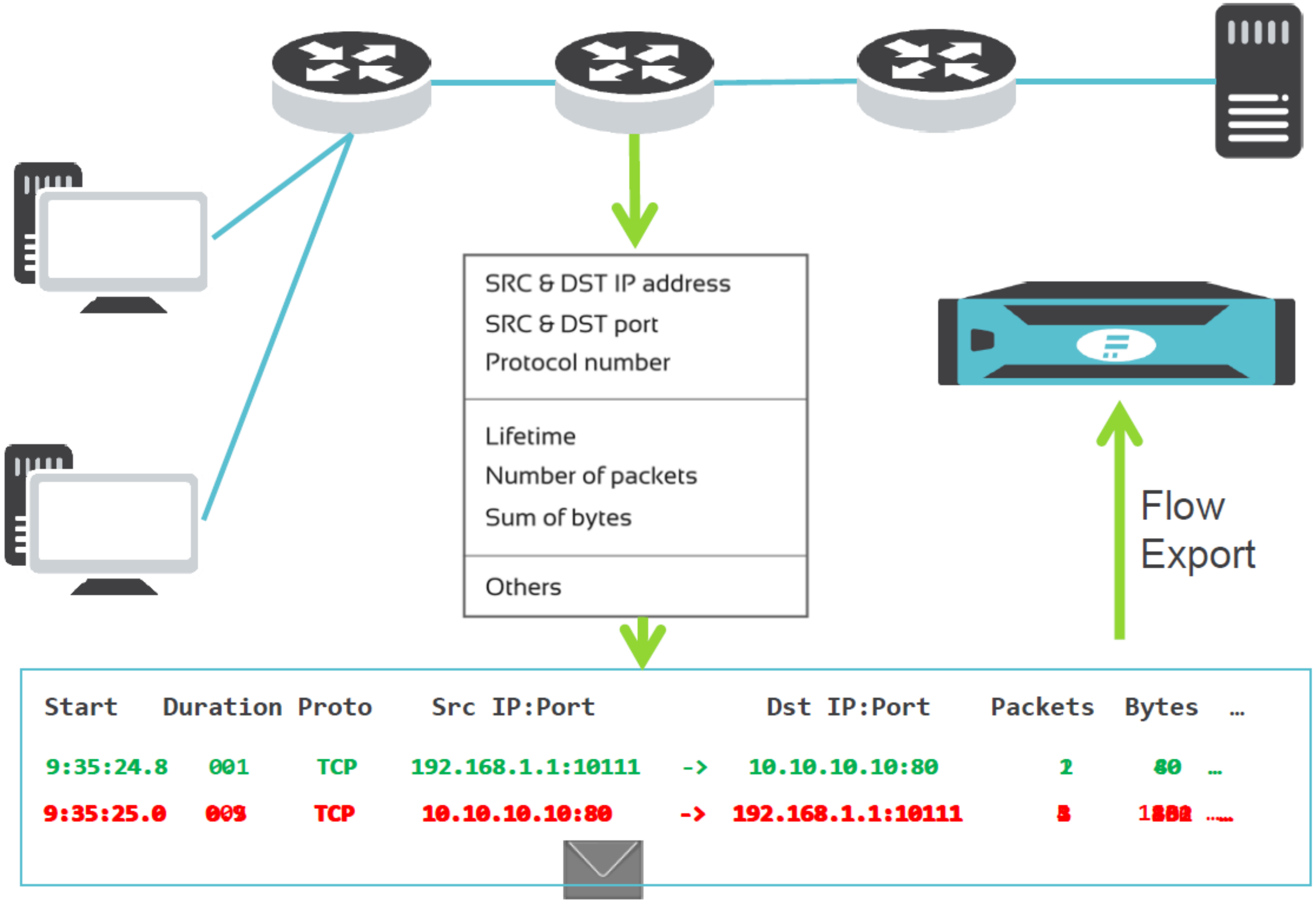


What is Flow Data?

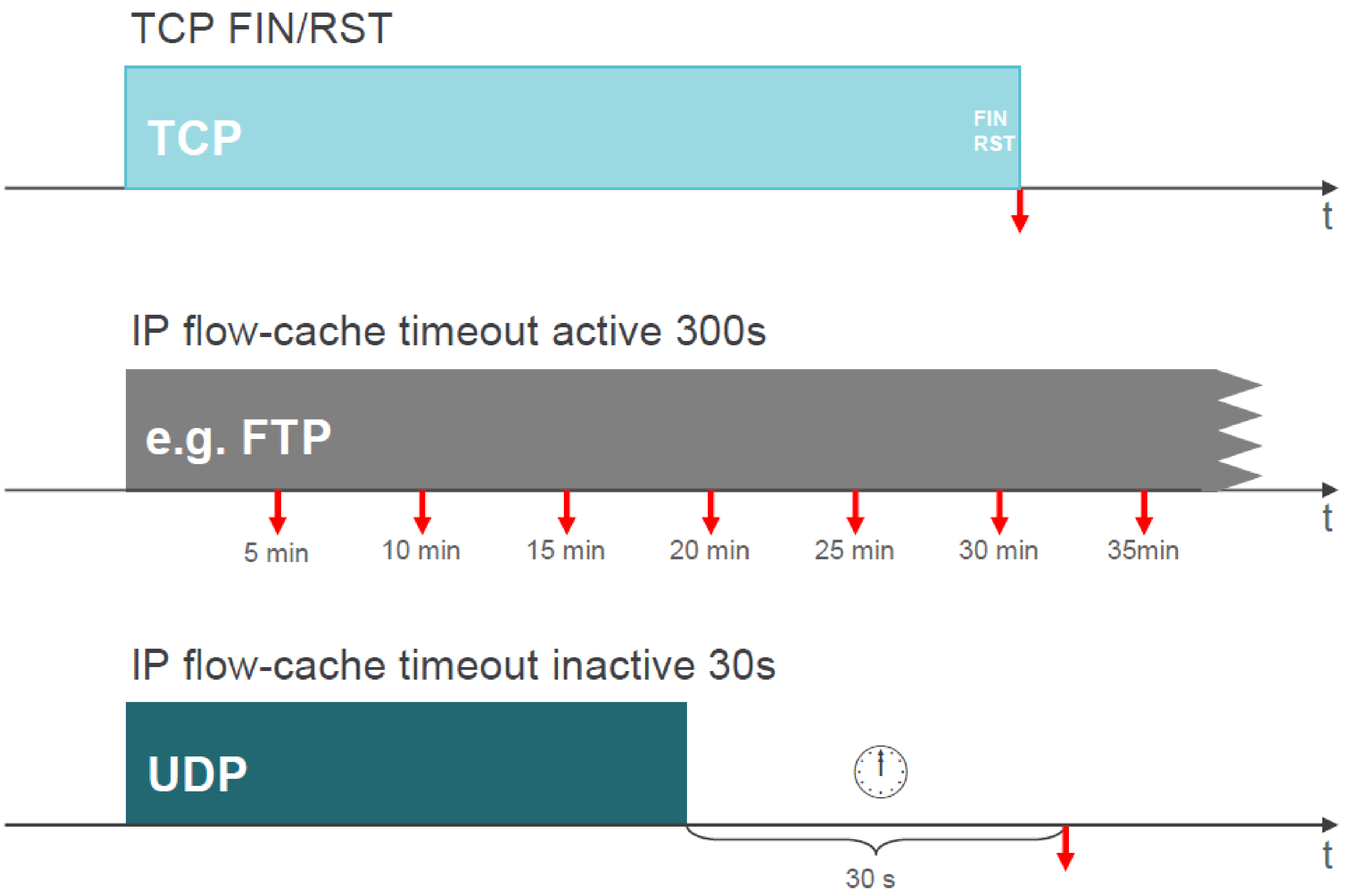
- Modern method for network monitoring –flow measurement
- Cisco standard NetFlow v5/v9, IETF standard IPFIX
- Focused on L3/L4 information and volumetric parameters
- Real network traffic to flow statistics reduction ratio 500:1



Flow Monitoring Principle



Flow Export Principle



Flow Key vs. Non-Key Fields

Flow Key vs. Non-Key Field

- Packet count
- Byte count

- Source IP address
- Destination IP address

- Start sysUpTime
- End sysUpTime

- Source TCP/UDP port
- Destination TCP/UDP port

- Input ifIndex
- Output ifIndex
- Type of service
- TCP flags
- Protocol

- Next hop address
- Source AS number
- Dest. AS number
- Source prefix mask
- Dest. Prefix mask
- …

Flow Standards

Cisco standard	**NetFlow v5**	fixed format only basic items available no IPv6, MAC, VLANs, …
	NetFlow v9 (Flexible NetFlow)	flexible format using templates mandatory for current needs provides IPv6, VLANs, MAC, …
Independent IETF standard	**IPFIX ("NetFlow v10")**	the future of flow monitoring more flexibility than NetFlow v9
Huawei	**NetStream**	same as original Cisco standard NetFlow v9
Juniper	**jFlow**	similar to NetFlow v9 issues in timestamps limited usability

Flow Standards

Related standards	**Cisco – NEL, NSEL**	uses NetFlow protocol to export firewall or NAT events and logs, similar format but different interpretation and use-cases
	sFlow	works on packet sampling basis not a real flow data, limited usability impossible to use for security purposes

- **Trends**
  - New monitored items (L7 application information)
    - NBAR2 (L7 application detection), HTTP, …
  - Number of flow-enabled devices is growing
    - Firewalls, UTMs, virtualization, SMB network equipment, …

Netflow versions

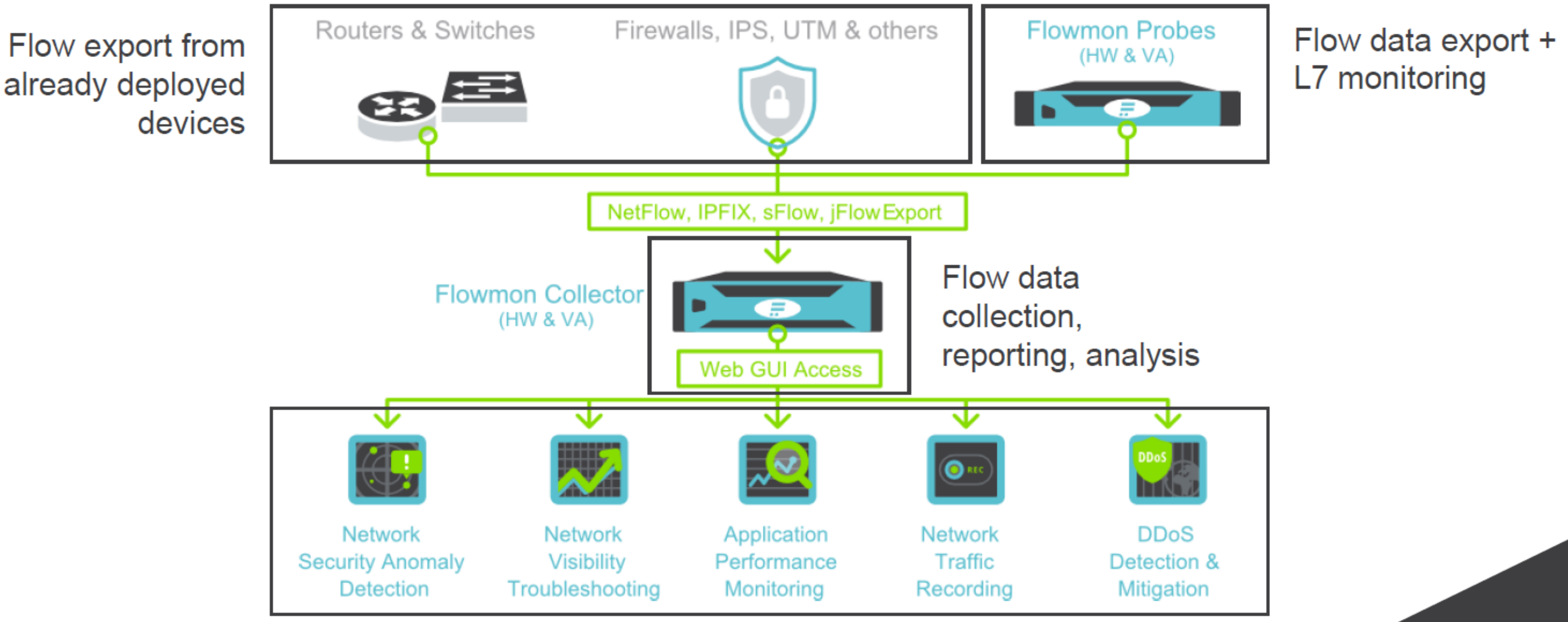
Version	Comment
v1	First implementation, now obsolete, and restricted to IPv4 (without IP mask and AS Numbers).
v2	Cisco internal version, never released.
v3	Cisco internal version, never released.
v4	Cisco internal version, never released.
v5	Most common version, available (as of 2009) on many routers from different brands, but restricted to IPv4 flows.
v6	No longer supported by Cisco. Encapsulation information (?).
v7	Like version 5 with a source router field. Used (only?) on Cisco Catalyst switches.
v8	Several aggregation form, but only for information that is already present in version 5 records
v9	Template Based, available (as of 2009) on some recent routers. Mostly used to report flows like IPv6, MPLS, or even plain IPv4 with BGP nexthop.
v10	Used for identifying IPFIX. Although IPFIX is heavily based on NetFlow, v10 does not have anything to do with NetFlow.

Netflow support by vendors

Vendor and type	Models	NetFlow Version
Cisco IOS-XR routers	CRS, ASR9000 old 12000	v5, v8, v9
Cisco IOS routers	10000, 7200, old 7500	v5, v8, v9
Cisco Catalyst switches	7600, 6500, 4500	v5, v8, v9
Cisco Nexus switches	5600, 7000, 7700	v5, v9
Juniper legacy routers	M-series, T-series, MX-series with DPC	v5, v8
Juniper legacy routers	M-series, T-series, MX-series with DPC	v5, v8, v9
Juniper routers	MX-series with MPC-3D, FPC5 for T4000	v5, IPFIX
Nokia routers	7750SR	v5, v8, v9, v10 IPFIX
Huawei routers	NE5000E NE40E/X NE80E	v5, v9
Enterasys Switches	S-Serie[9] and N-Serie[10]	v5, v9
Flowmon Probes	Flowmon Probe 1000, 2000, 4000, 6000, 10000, 20000, 40000, 80000, 100000	v5, v9, IPFIX
Nortel Switches	Ethernet Routing Switch 5500 Series (ERS5510, 5520 and 5530) and 8600 (Chassis-based)	v5, v9, IPFIX
PC and Servers	Linux FreeBSD NetBSD OpenBSD	v5, v9, IPFIX
VMware servers	vSphere 5.x[16]	v5, IPFIX (>5.1)[17]
Mikrotik RouterOS	RouterOS 3.x, 4.x, 5.x, 6.x [18]	v1, v5, v9, IPFIX (>6.36RC3)

Ukážka jedného nástroja založeného na NetFlow dátach

Flowmon Architecture



Flowmon modules for advanced flow data analysis

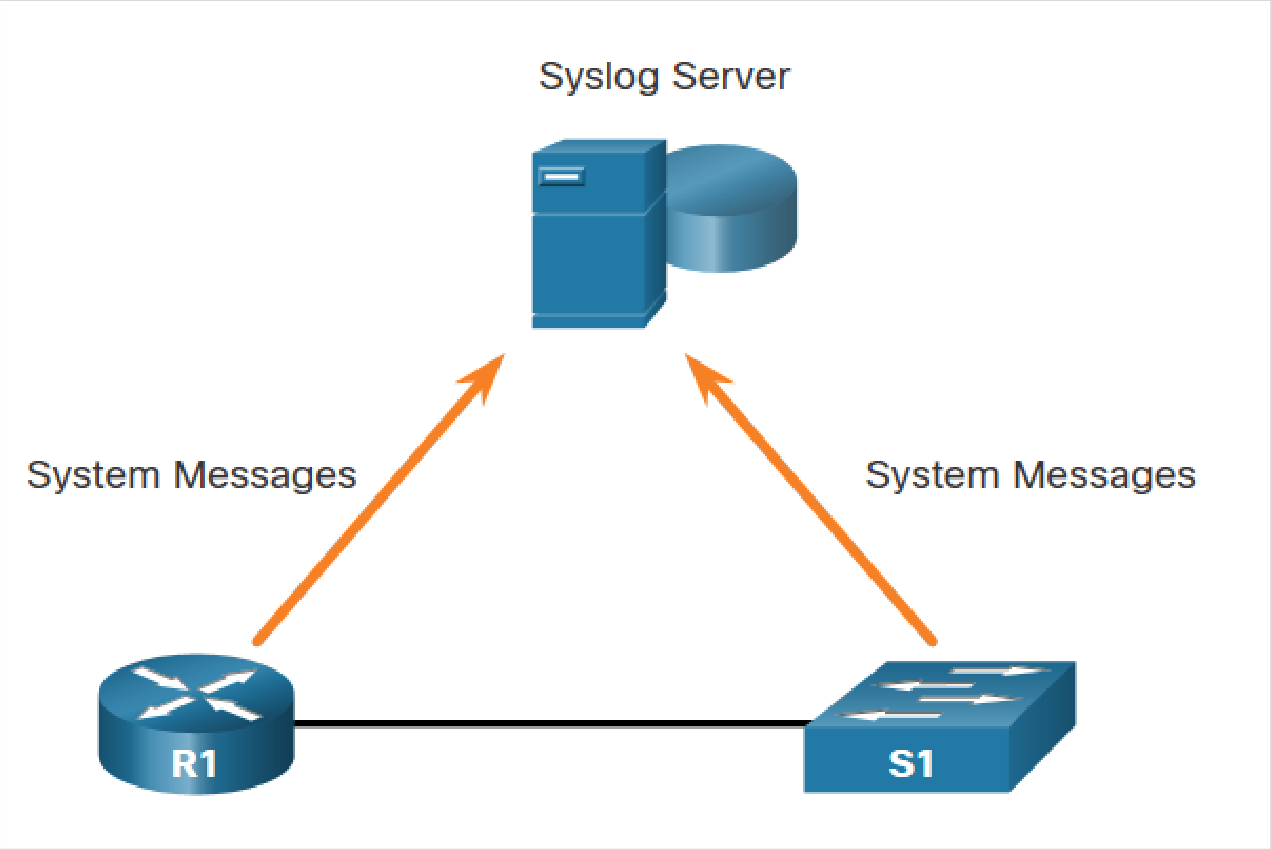
Success story

* Skupina vedcov združenia CESNET v ČR 2002 - začala aktivity v oblasti programovateľného hardvéru s názvom Liberouter project.
* Počas účasti na vývojovom projekte pre GEANT2 (európska akademická sieť), tím Liberouter vyvinul prototyp sieťovej monitorovacej sondy s názvom FlowMon.
* V 2012 – umiestnili sa v Gartner Magic Quadrant v NPMD.
* 2020 - Spoločnosť Flowmon Networks získala spoločnosť Kemp Technologies

Flowmon
Driving Network Visibility

Syslog Servers

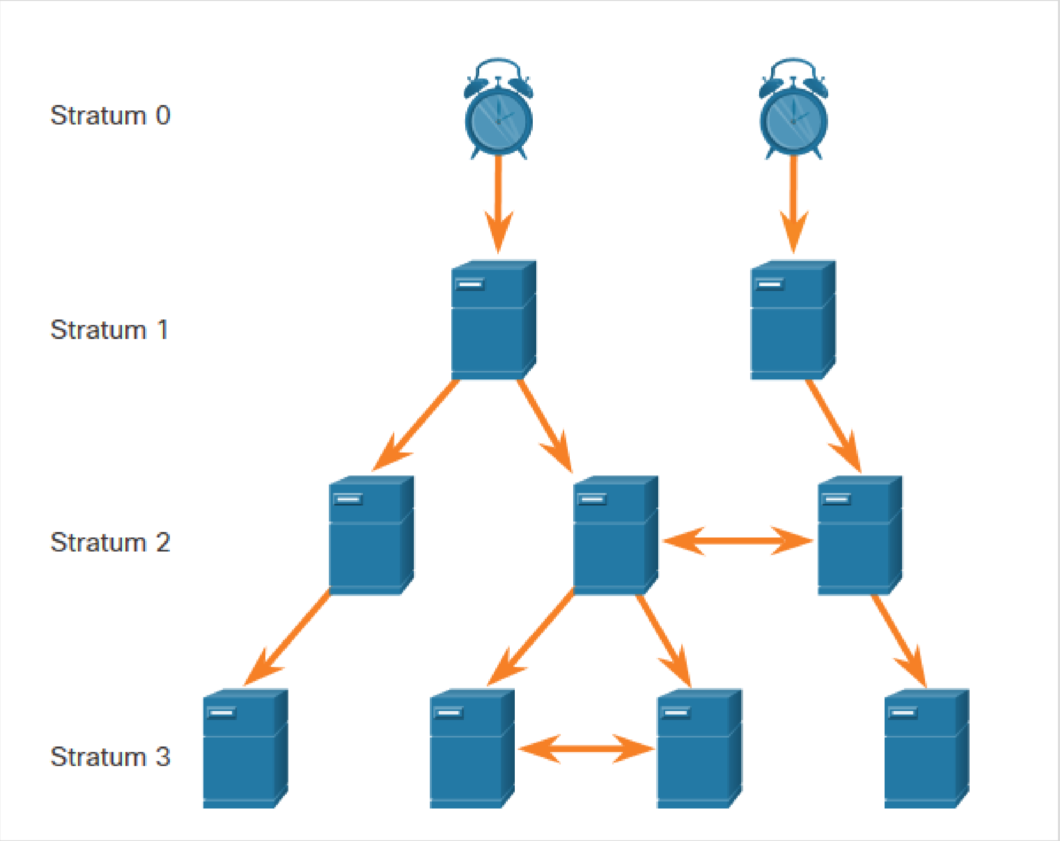
- The most common method of accessing system messages is to use a protocol called syslog.
- The Syslog protocol allows networking devices to send their system messages across the network to syslog servers.
- It provides three primary functions:
  - The ability to gather logging information for monitoring and troubleshooting
  - The ability to select the type of logging information that is captured
  - The ability to specify the destination of captured syslog messages



Syslog

NTP

- It is important to synchronize the time across all devices on the network. The date and time settings on a network device can be set using one of two methods:
  - Manual configuration of the date and time
  - Configuring the Network Time Protocol (NTP)
- NTP networks use a hierarchical system of time sources, where each level in this system is called a stratum. NTP servers are arranged in three levels known as strata:
  - **Stratum 0**: An NTP network gets the time from authoritative time sources.
  - **Stratum 1**: Devices are directly connected to the authoritative time sources.
  - **Stratum 2 and lower strata**: Stratum 2 devices, such as NTP clients, synchronize their time using the NTP packets from stratum 1 servers.



NTP Stratum Levels

AAA Servers

The below table lists the three independent security functions provided by the AAA architectural framework.

Functions	Description
Authentication	• Users and administrators must prove that they are who they say they are. • Authentication can be established using username and password combinations, challenge and response questions, token cards, and other methods. • AAA authentication provides a centralized way to control access to the network.
Authorization	• After the user is authenticated, authorization services determine which resources the user can access and which operations the user is allowed to perform. • An example is “User ‘student’ can access host serverXYZ using SSH only.”
Accounting	• Accounting records what the user does, including what is accessed, the amount of time the resource is accessed, and any changes that were made. • Accounting keeps track of how network resources are used. • An example is "User ‘student’ accessed host serverXYZ using SSH for 15 minutes."

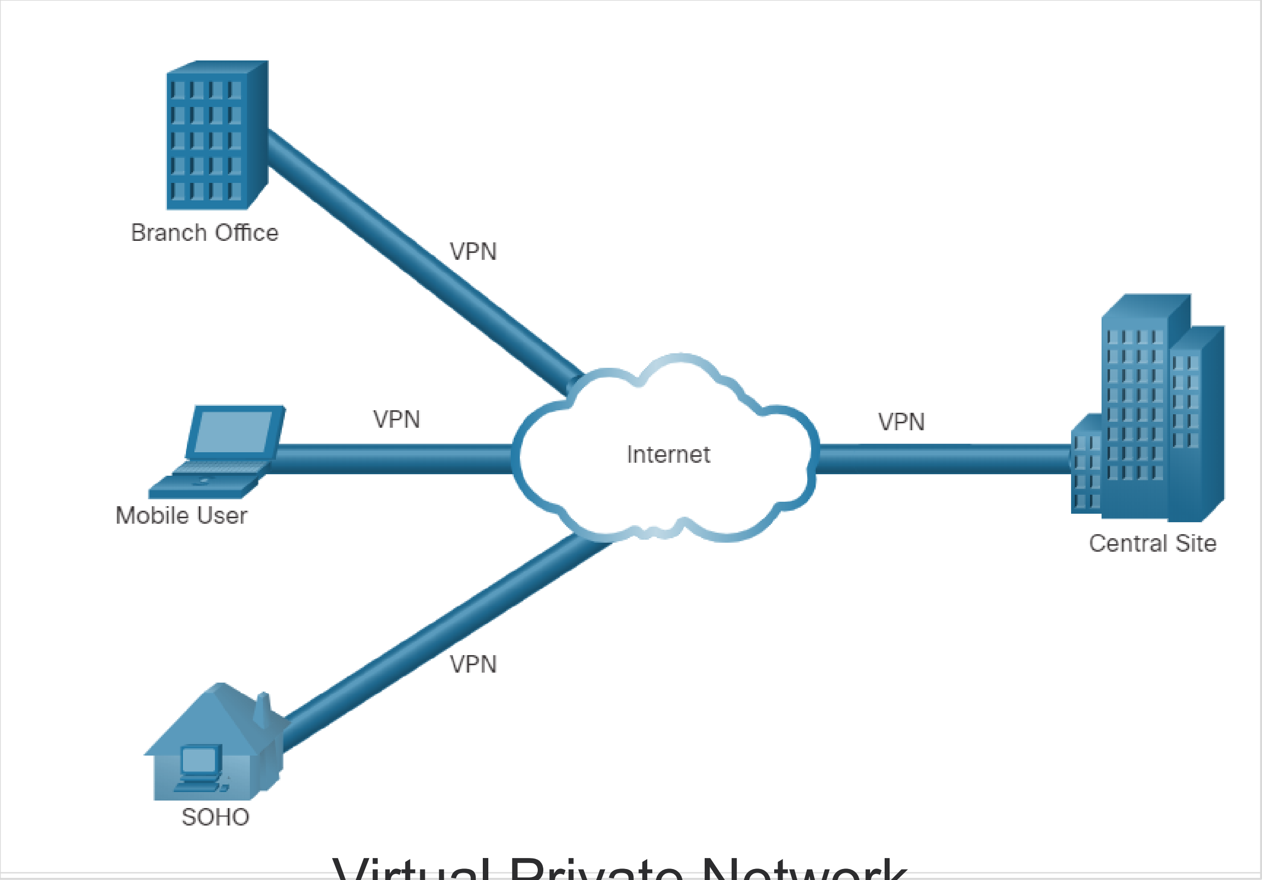
AAA Servers (Contd.)

The below table lists the difference between Terminal Access Controller Access-Control System Plus (TACACS+) and Remote Authentication Dial-In User Service (RADIUS) protocols.

	TACACS+	RADIUS
Functionality	Separates AAA according to the AAA architecture,	Combines authentication and authorization but separates accounting,
Standard	Mostly Cisco supported	Open/RFC standard
Transport	TCP	UDP
Protocol CHAP	Bidirectional challenge and response as used in Challenge Handshake Authentication Protocol (CHAP)	Unidirectional challenge and response from the RADIUS security server to the RADIUS client
Confidentiality	Entire packet encrypted	Password encrypted
Customization	Provides authorization of router commands on per-user or per-group basis	No option to authorize router commands on a per-user or per-group basis
Accounting	Limited	Extensive

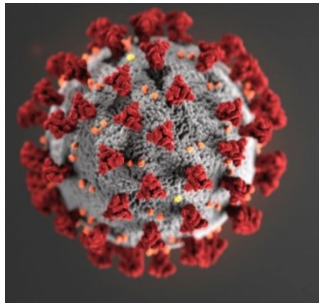
VPN

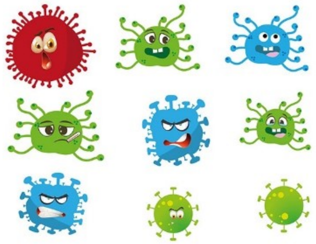
- A VPN is a private network that is created over a public network (usually the internet).
- A VPN uses virtual connections routed through the Internet from the organization to the remote site.
- A VPN is a communications environment in which access is strictly controlled to permit peer connections within a defined community of interest.
- Confidentiality is achieved by encrypting the traffic within the VPN.
- In short, VPN connects two endpoints over a public network, to form a logical connection which can be made at Layer 2 or Layer 3.



Virtual Private Network

Pripomienka hrozieb





Ľudské hrozby

Pôvod hrozby	Motivácia
Heker, kreker	Výzva Ego **Postavenie / status** Peniaze Povstanie
Terorista	Vydieranie Zničenie Finančný zisk **Náboženský fanatizmus** Pomsta **Politický zisk** Mediálne pokrytie
Počítačový kriminálnik	Ničenie informácií Nelegálne zverejňovanie informácií **Finančný zisk** Neoprávnená zmena údajov
Priemyselná špionáž	Finančný zisk Ekonomická špionáž
Členovia (študenti, zamestnanci)	Zvedavosť Neúmyselné chyby (slabé heslá) Ego Spravodajstvo Peňažný zisk Pomsta

TYP	HROZBY	Pôvod
Fyzické poškodenie	Požiar	NUE
	Poškodenie vodou	NUE
	znečistenie	NUE
	závažná havária	NUE
	zničenie zariadenia alebo médií	NUE
	prach, korózia, mrznutie	NUE
Prírodné udalosti	Klimatický jav	E
	Seizmický jav	E
	Vulkanický jav	E
	Meteorologický jav	E
	Povodeň	E
Strata základných služieb	Porucha klimatizácie alebo vodovodu	NU
	Strata energetického napájania	NUE
	Porucha telekomunikačného zariadenia	NU
Narušenie v dôsledku radiácie	Elektromagnetická radiácia	NUE
	Termálna radiácia	NUE
	elektromagnetické impulzy	NUE
Vyzradenie informácií	Veľa rôznych	...
Technické zlyhanie	Zlyhanie zariadenia	N
	Porucha zariadenia	N
	Saturácia informačného systému	NU
	Softvérová porucha	N
	porušenie udržiavateľnosti informačného systému	NU
Neautorizované činnosti	Neautorizované používanie zariadenia	U
	Podvodné kopírovanie systému	U
	Použitie falošného alebo kopírovaného softvéru	NU
	poškodenie dát	U
	Nelegálne spracovanie dát	U
Vyzradenie funkcií	Chyba pri použití	N
	Zneužitie práva	NU
	Falšovanie práv	U
	Odopretie činností	U
	Porušenie dostupnosti personálu	NUE

N – neúmyselný, U – úmyselný, E – enviromentálny pôvod

Disaster recovery – Obnova po katastrofickom scenári

Ideálny svet

- Mám infraštruktúru pre ochranu dát
  - Tá okamžite obnoví všetky aplikácie a dáta
  - Načas a presne od bodu keď nastala kritická udalosť

Reálny svet

- Prepnutie po poruche (failover)
  - Môže byť okamžité
- Replikácia dát
  - Môže byť nepretržitá
- Ale... Veľké ale...
  - Tieto operácie sú veľmi náročné
    - Zdrojovo
    - Finančne

Preto sa zaviedli dva realistické ciele, s ohľadom na rozpočet, zdroje a prioritu aplikácií:

- RTO - Recovery Time Objective
- RPO - Recovery Point Objective

Definícia RTO a RPO

- Oba spolu súvisia, a sú potrebné pre obnovu po kritickej udalosti
- Oba sú rôznymi metrikami s rôznym účelom
- Sú kľúčovými konceptmi v poskytovaní business continuity

RPO

- množstvo údajov, ktorých strata je tolerovaná – pri kritických udalostiach, ktoré spôsobia významné škody
- je vyjadrený ako meranie času od stratovej udalosti po poslednú predchádzajúcu zálohu.
- Zvyčajne 12, 8, 4 hodiny, alebo near-zero RPO (sekundy)
  - Potrebné sú replikácie v odpovedajúcich časoch

RTO

- ako dlho môže byť aplikácia mimo prevádzky bez toho, aby spôsobila značné škody podniku
  - Niektorá dni...
  - Iná niekoľko sekúnd.. „near-zero RTO“
    - Potrebné sú vtedy silné failover služby

UNIVERSITY OF ŽILINA
Faculty of Management Science and Informatics

Ďakujem za pozornosť

Obsahom boli moduly:

Chapter 9 The Transport Layer

Chapter 12 Network Security Infrastructure

Vyjadrite spätnú väzbu na prednášku a/alebo cvičenie v anonymnej ankete cez google form: link

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