

Zjednodušené základy ARP, TCP/IP

Jiri Kubina

jiri.kubina@osu.cz

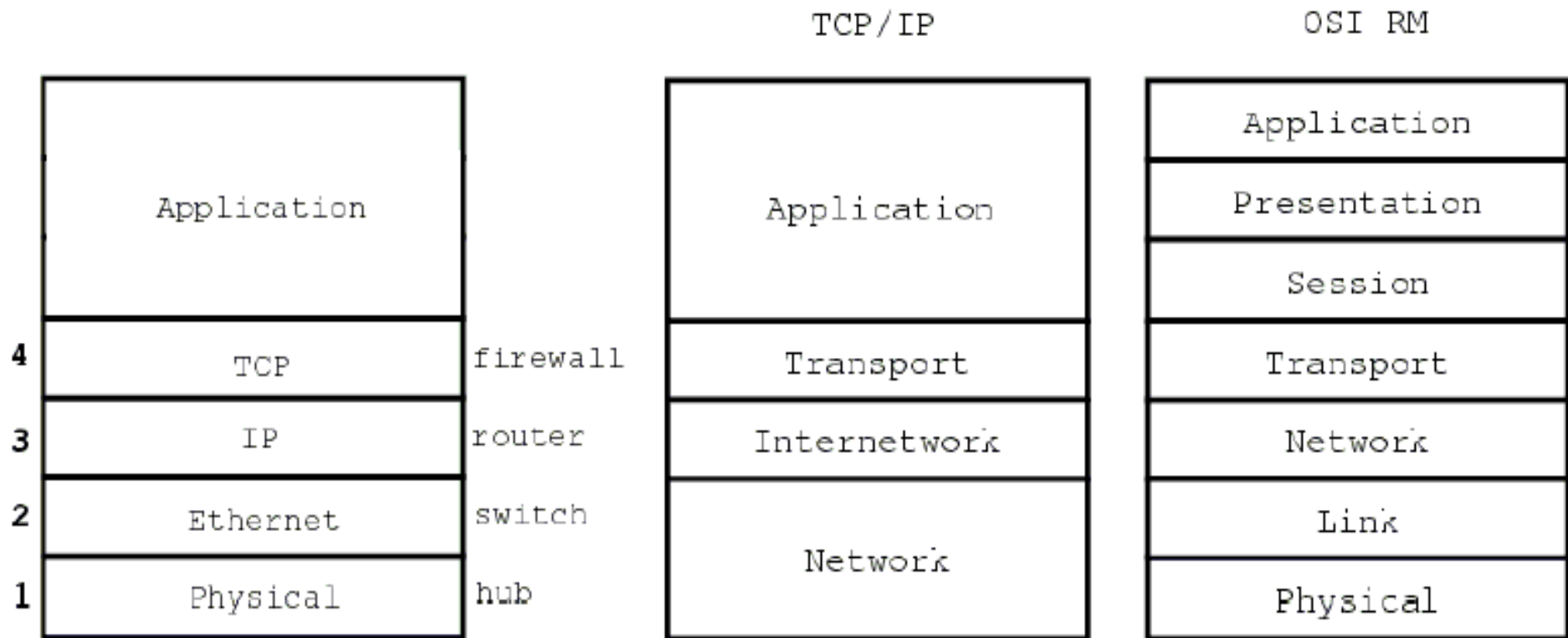
Ver. 1.0

leden 2006

Obsah

1. ARP - zjednodusený popis metody prekladu IP na MAC
2. Stručný prehľad IP protokolu
3. Hlavicka TCP
4. Navazani spojeni - TCP
5. Datová výmena - TCP
6. Ukončení spojeni - TCP
7. Reset spojeni - TCP
8. Použité zdroje a nástroje

Upozorneni: Vetsina popisovanych principu je zjednodusena, nebo zkracena z vyukovych duvodu. Pro ziskani kompletnich informaci ctete RFC 791, 1042, 894, 826, 1027, 793



1. ARP - zjednodusený popis prekladu IP na MAC

ARP - Address Resolution Protocol

- Host A chce komunikovat pomocí IP s Host B na stejné síti. Pro komunikaci znám IP-odesilatele a IP-příjemce a tudíž jsem schopen sestavit IP-datagram /3 vrstva/. Problem je v tom že IP-datagram musí být zabalen v linkovém rámci /2 vrstva/ např. ethernet. Aby jsem mohl vytvořit ethernetový rámec potřebuji znát linkovou /MAC/ adresu odesilatele a příjemce. Odesilatel jsem já a svou MAC adresu znám, zbyvá tedy zjistit MAC příjemce. Toto řeší protokol ARP.

- ARP resi problem zjistení MAC adresy jiného PC na stejné síti ze znalosti jeho IP. Do sítě se vysle linkový broadcast typ 1 /2 vrstva/ s cílovou adresou FF:FF:FF:FF:FF:FF. V tomto broadcastu uvede svoji MAC svoji IP a IP příjemce. Protože je cílová MAC FF:FF:FF:FF:FF:FF dostane se ke všem PC na tomto segmentu sítě. To PC které má IP uvedenou v broadcastu se ozve linkovým datagramem typ 2 /opět 2 vrstva/ ale už ne broadcastem ale na konkrétní MAC adresu, kterou obdržel v předchozím broadcastu.

- Vypis programu tcpdump

```
09:54:17.063770 arp who-has 195.113.106.168 tell 195.113.106.167
09:54:17.063875 arp reply 195.113.106.168 is-at 00:11:11:35:1d:bf
```

ARP broadcast - request

arp - Ethereal

File Edit View Go Capture Analyze Statistics Help

Filter: + Expression... Clear Apply

No.	Time .	Source	Destination	Protocol	Info
15	10.521036	neo.firmal.intr	Broadcast	ARP	Who has 195.113.106.129? Tell 195.113.106.168
16	10.521206	Cisco_23:8c:00	neo.firmal.intr	ARP	195.113.106.129 is at 00:0d:65:23:8c:00
17	10.521214	195.113.106.168	195.113.106.14	ICMP	Echo (ping) request

Frame 15 (42 bytes on wire, 42 bytes captured)

Arrival Time: Jan 27, 2006 10:01:05.205960000
[Time delta from previous packet: 0.517623000 seconds]
[Time since reference or first frame: 10.521036000 seconds]
Frame Number: 15
Packet Length: 42 bytes
Capture Length: 42 bytes
[Protocols in frame: eth:arp]

Ethernet II, Src: neo.firmal.intr (00:11:11:35:1d:bf), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Source: neo.firmal.intr (00:11:11:35:1d:bf)
Type: ARP (0x0806)

Address Resolution Protocol (request)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (0x0001)
Sender MAC address: neo.firmal.intr (00:11:11:35:1d:bf)
Sender IP address: 195.113.106.168 (195.113.106.168)
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 195.113.106.129 (195.113.106.129)

Frame (frame), 42 bytes P: 31 D: 31 M: 0

1 2 Mozilla Download - Mozilla Fi arp - Ethereal root@neo:~ - Shell - Konsole 10:16
3 4 mc - /var/named/chroot/var/na Inbox for jura@firmal.intr - Mc 2006-01-27

ARP broadcast - reply

The image shows a Wireshark capture of network traffic. The main pane displays a list of packets:

No.	Time	Source	Destination	Protocol	Info
15	10.521036	neo.firmal.intr	Broadcast	ARP	Who has 195.113.106.129? Tell 195.113.106.168
16	10.521206	Cisco_23:8c:00	neo.firmal.intr	ARP	195.113.106.129 is at 00:0d:65:23:8c:00
17	10.521214	195.113.106.168	195.113.106.14	ICMP	Echo (ping) request

The details pane for Frame 16 (64 bytes on wire, 64 bytes captured) shows the following structure:

- Arrival Time: Jan 27, 2006 10:01:05.206130000
[Time delta from previous packet: 0.000170000 seconds]
[Time since reference or first frame: 10.521206000 seconds]
Frame Number: 16
Packet Length: 64 bytes
Capture Length: 64 bytes
[Protocols in frame: eth:arp]
- Ethernet II, Src: Cisco_23:8c:00 (00:0d:65:23:8c:00), Dst: neo.firmal.intr (00:11:11:35:1d:bf)
Destination: neo.firmal.intr (00:11:11:35:1d:bf)
Source: Cisco_23:8c:00 (00:0d:65:23:8c:00)
Type: ARP (0x0806)
Trailer: 00000000000000000000000000000000
Frame check sequence: 0x00000000 [incorrect, should be 0x2ddbba34]
- Address Resolution Protocol (reply)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (0x0002)
Sender MAC address: Cisco_23:8c:00 (00:0d:65:23:8c:00)
Sender IP address: 195.113.106.129 (195.113.106.129)
Target MAC address: neo.firmal.intr (00:11:11:35:1d:bf)
Target IP address: 195.113.106.168 (195.113.106.168)

File: "/root/arp" 3294 Bytes 0 P: 31 D: 31 M: 0

2. Stručný přehled IP protokolu

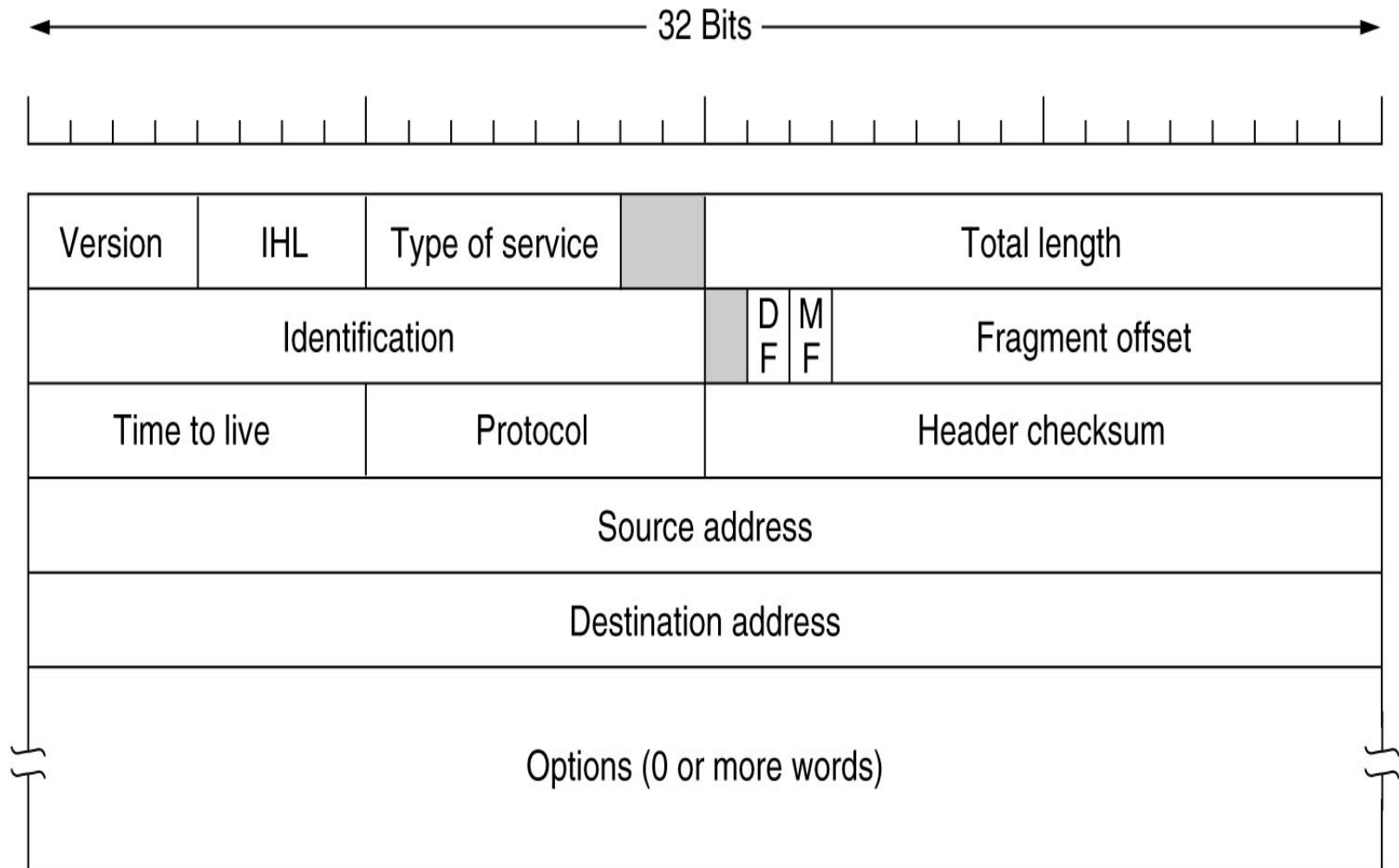
3. vrstva, síťová nespojovaná služba posílání nezávisle směrovaných paketů

Některé **linkové** protokoly jsou určeny pro dopravu dat v rámci lokální sítě, jiné linkové protokoly dopravují data mezi sousedními směrovací rozsáhlých sítí.

IP-protokol na rozdíl od linkových protokolů dopravuje data mezi dvěma libovolnými počítači v Internetu, tzn. i přes několik LAN.

Pakety jsou od odesílatele k příjemci dopravovány-směrovány přes směrovací /router/. Po cestě může být celá řada směrovaců. Každý směrovac řeší samostatně směrování k následujícímu směrovací.

IP - datagram



- **version** - verze protokolu bud Ipv4 nebo IPv6
- **header length** - informuje o delce hlavicky datagramu, protoze header obsahuje options s promennou delkou
- **type of service** - urcuje prioritu paketu pri posilani siti /dnes **Differentiated Services Field** /
- **total length** - delka IP paketu (max. 64kB)
- **identification** - ID paketu (docasne) jednoznacne v ramci zdroje, vsechny fragmenty paketu maji ID stejne
- **flags** - povoleni/zakaz fragmentace, indikace posledniho fragmentu fragmentovaneho paketu
- **Time to Live** - pocitadlo snizovane pri kazdem pruchodu smerovacem, proti zbloudilym paketum cyklujicim ve smerovaci smyccce. Pri docitani do nuly se paket zahodi.
- **protocol** - cislo protokolu vyssi vrstvy neseneho v paketu
- **header checksum** - kontrolni soucet vsech bytu hlavicky datagramu, nezahrnuje datovou cast paketu (!)
- **source & destination IP address** - IP adresa zdrojoveho a ciloveho pocitace (zarizeni)
- **options** - volitelne, promenna delka

IP - datagram

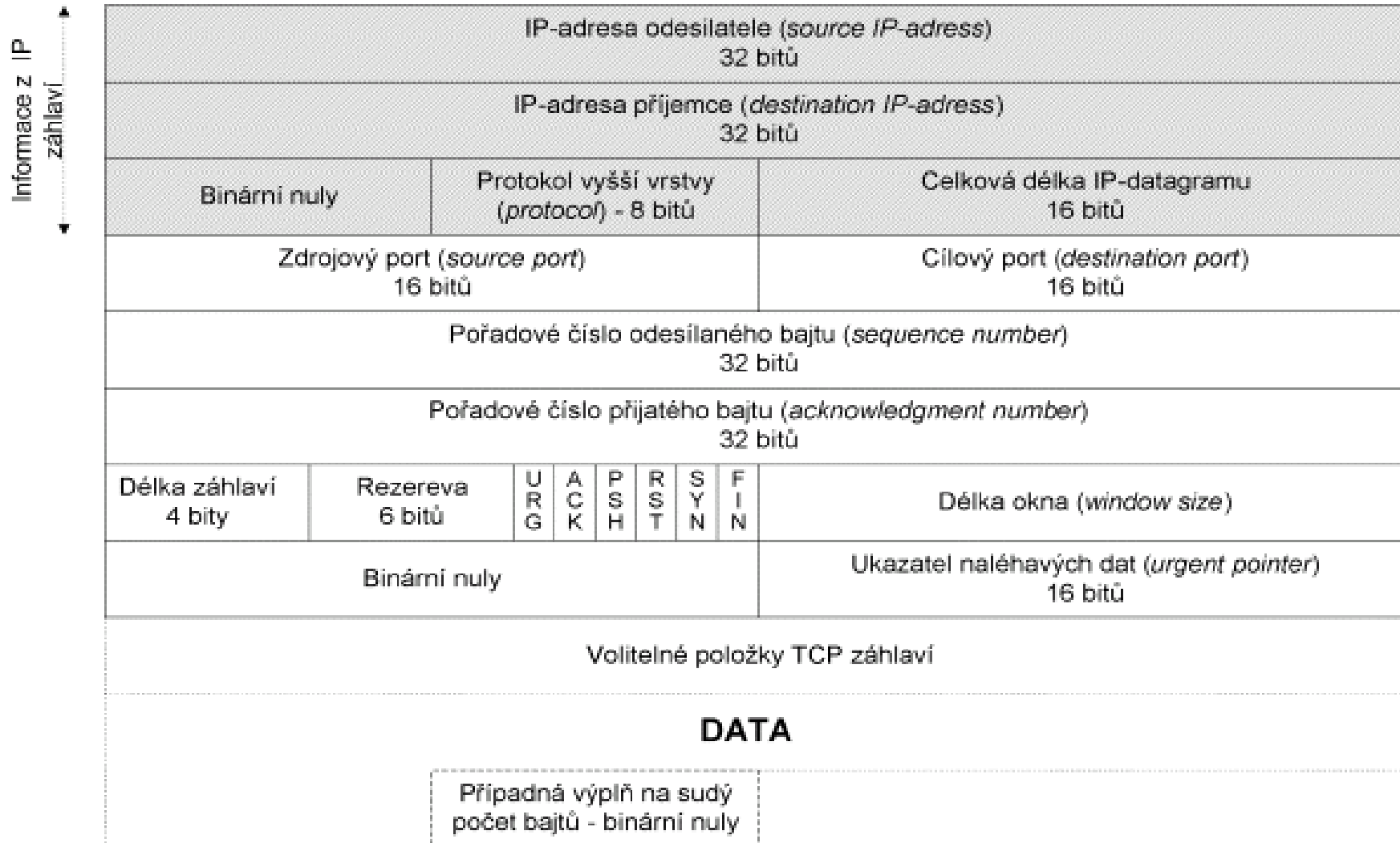
The screenshot displays the Wireshark interface with a network capture of an HTTP SYN-ACK exchange. The packet list shows five packets:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.152.30.50	10.152.30.1	TCP	24773 > http [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=1
2	0.001619	10.152.30.1	10.152.30.50	TCP	http > 24773 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
3	0.001668	10.152.30.50	10.152.30.1	TCP	24773 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=1
4	0.011116	10.152.30.50	10.152.30.1	HTTP	GET / HTTP/1.1
5	0.012614	10.152.30.1	10.152.30.50	TCP	http > 24773 [ACK] Seq=1 Ack=529 Win=6432 Len=0 TSV=1

The packet details pane for the second packet (the SYN-ACK) is expanded, showing the IP header and TCP options:

- Ethernet II, Src: aa:aa:03:00:00:00 (aa:aa:03:00:00:00), Dst: Intel-Hf_d1:35:7e (00:a0:c9:d1:35:7e)
- Internet Protocol, Src: 10.152.30.50 (10.152.30.50), Dst: 10.152.30.1 (10.152.30.1)
 - Version: 4
 - Header length: 20 bytes
 - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
 - 0000 00.. = Differentiated Services Codepoint: Default (0x00)
 -0. = ECN-Capable Transport (ECT): 0
 -0 = ECN-CE: 0
 - Total Length: 60
 - Identification: 0xc457 (50263)
 - Flags: 0x04 (Don't Fragment)
 - 0... = Reserved bit: Not set
 - .1.. = Don't fragment: Set
 - ..0. = More fragments: Not set
 - Fragment offset: 0
 - Time to live: 64
 - Protocol: TCP (0x06)
 - Header checksum: 0x2502 [correct]
 - [Good: True]
 - [Bad : False]
 - Source: 10.152.30.50 (10.152.30.50)
 - Destination: 10.152.30.1 (10.152.30.1)
- Transmission Control Protocol, Src Port: 24773 (24773), Dst Port: http (80), Seq: 0, Ack: 0, Len: 0

3.Hlavicka TCP



4.Navazani spojeni TCP

Navazani spojeni - Three way handshake

1.Odeslani synchronizacniho (SYN) paketu a Initial Sequence Number (ISN-A) /ACK number neni definovan/

Host A si preje navazat spojeni s Host B. Host A posle paket na Host B s nastavenym synchronizacnim bitem (SYN),ktery oznamuje vznik noveho spojeni a Initial Sequence Number (ISN), ktery umoznuje rozpoznavat /sledovat, urcovat/ pakety posilane mezi Host A a Host B.

Host A ----- SYN(ISN-A) -----> Host B

Poznamka: ISN jsou nahodna, aby se zabranilo pripadnemu ovlivneni zbloudilymi pakety ze zavreneho a brzy znovu otevreneho spojeni mezi stejnymi Hosty

Host A ----- SYN(ISN-A) -----> Host B

The screenshot shows the Wireshark interface with a capture filter: `(ip.addr eq 10.152.30.50 and ip.addr eq 10.152.30.1) and (tcp.port eq 24773)`. The packet list shows five packets:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.152.30.50	10.152.30.1	TCP	24773 > http [SYN] Seq=590857293 Ack=0 Win=5840 Len=0
2	0.001619	10.152.30.1	10.152.30.50	TCP	http > 24773 [SYN, ACK] Seq=1332285558 Ack=590857293 Win=0 Len=0
3	0.001668	10.152.30.50	10.152.30.1	TCP	24773 > http [ACK] Seq=590857294 Ack=1332285559 Win=0 Len=0
4	0.011116	10.152.30.50	10.152.30.1	HTTP	GET / HTTP/1.1
5	0.012614	10.152.30.1	10.152.30.50	TCP	http > 24773 [ACK] Seq=1332285559 Ack=590857292 Win=0 Len=0

The details pane for the selected packet (No. 1) shows:

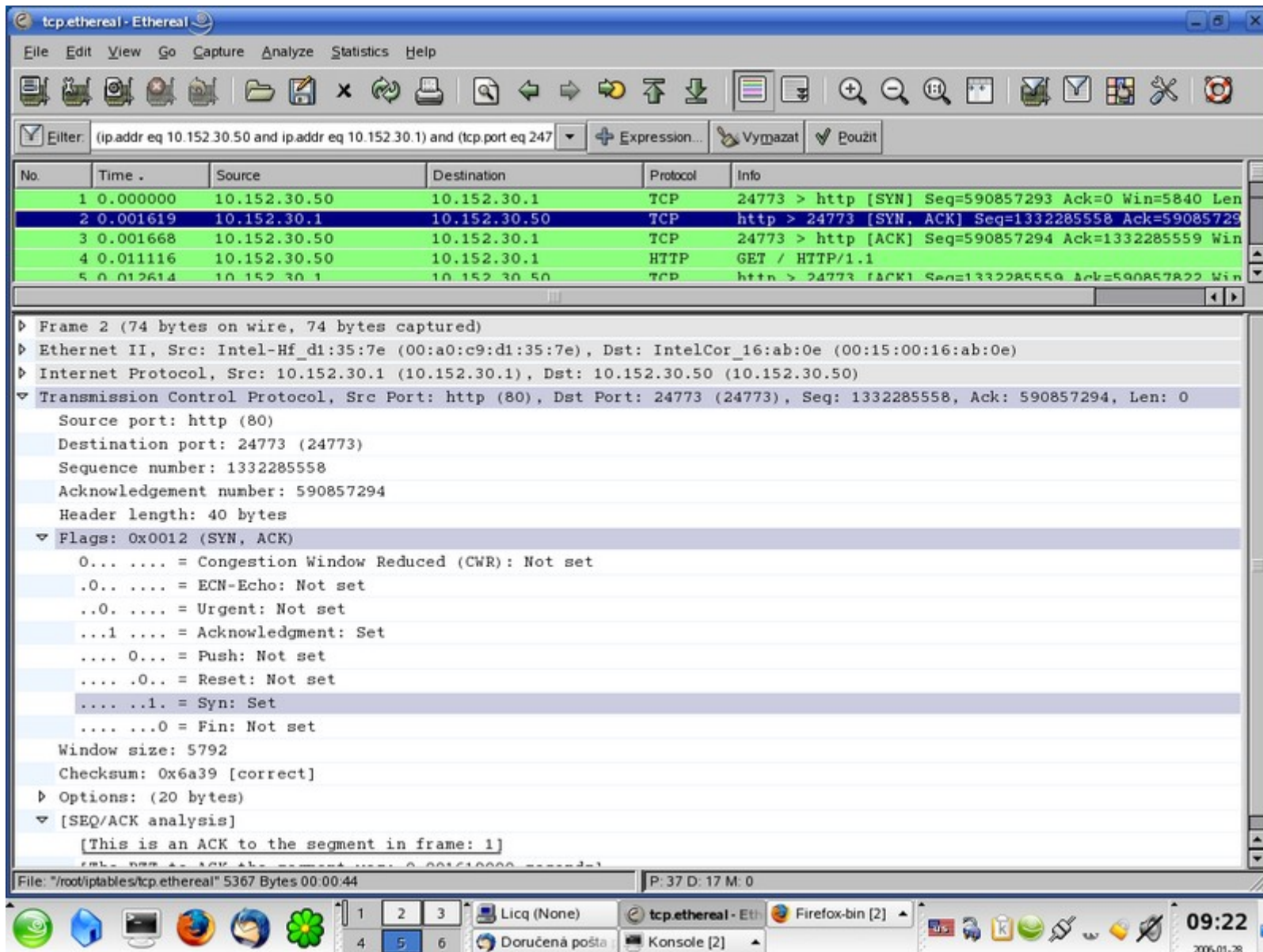
- Frame 1 (74 bytes on wire, 74 bytes captured)
- Ethernet II, Src: aa:aa:03:00:00:00 (aa:aa:03:00:00:00), Dst: Intel-Hf_d1:35:7e (00:a0:c9:d1:35:7e)
- Internet Protocol, Src: 10.152.30.50 (10.152.30.50), Dst: 10.152.30.1 (10.152.30.1)
- Transmission Control Protocol, Src Port: 24773 (24773), Dst Port: http (80), Seq: 590857293, Ack: 0, Len: 0
 - Source port: 24773 (24773)
 - Destination port: http (80)
 - Sequence number: 590857293
 - Header length: 40 bytes
 - Flags: 0x0002 (SYN)
 - 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
 - Window size: 5840
 - Checksum: 0x971b [correct]
 - Options: (20 bytes)

2.Odeslani potvrzovacího (ACK) paketu

Host B na žádost /z Host A/ odpovídá odesláním paketu s nastaveným synchronizačním bitem (SYN) a acknowledgment bitem (ACK) na Host A. Host B zasílá svůj ISN-B a jako ACK number je odeslán Initial Sequence Number /z Host A/ plus 1 (ISN-A+1) které indikuje že paket navazující spojení byl korektně přijat.

Host A <----- SYN(ISN-B)/ACK(ISN-A+1) ----- Host B

Host A <----- SYN(ISN-B) / ACK(ISN-A+1) ----- Host B



**3. Dokončení negotiation /vyjednávání, dohodnutí/ odesláním
finalního acknowledgment (ACK) paketu na Host B.**

Host A zasílá zpět na Host B finalní ACK paket a sequence number jako potvrzení přijetí předchozího paketu. Spojení je kompletní, data mohou začít tect mezi Host A a Host B.

Host A ----- ACK(ISN-B+1) -----> Host B

Host A ----- ACK(ISN-B+1) -----> Host B

The screenshot shows the Wireshark interface with a packet capture filter: `(ip.addr eq 10.152.30.50 and ip.addr eq 10.152.30.1) and (tcp.port eq 24773)`. The packet list shows five packets:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.152.30.50	10.152.30.1	TCP	24773 > http [SYN] Seq=590857293 Ack=0 Win=5840 Len=
2	0.001619	10.152.30.1	10.152.30.50	TCP	http > 24773 [SYN, ACK] Seq=1332285558 Ack=59085729
3	0.001668	10.152.30.50	10.152.30.1	TCP	24773 > http [ACK] Seq=590857294 Ack=1332285559 Win=
4	0.011116	10.152.30.50	10.152.30.1	HTTP	GET / HTTP/1.1
5	0.012614	10.152.30.1	10.152.30.50	TCP	http > 24773 [ACK] Seq=1332285559 Ack=590857294 Win=

The details pane for Frame 3 (66 bytes on wire, 66 bytes captured) shows the following information:

- Ethernet II, Src: aa:aa:03:00:00:00 (aa:aa:03:00:00:00), Dst: Intel-Hf_d1:35:7e (00:a0:c9:d1:35:7e)
- Internet Protocol, Src: 10.152.30.50 (10.152.30.50), Dst: 10.152.30.1 (10.152.30.1)
- Transmission Control Protocol, Src Port: 24773 (24773), Dst Port: http (80), Seq: 590857294, Ack: 1332285559, Len: 0
 - Source port: 24773 (24773)
 - Destination port: http (80)
 - Sequence number: 590857294
 - Acknowledgement number: 1332285559
 - Header length: 32 bytes
 - Flags: 0x0010 (ACK)
 - 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: 1460
 - Checksum: 0xa9e9 [correct]
 - Options: (12 bytes)
 - [SEQ/ACK analysis]
 - [This is an ACK to the segment in frame: 2]

5. Datova vymena TCP

1. Prenos dat protokolu vyssi vrstvy pomoci paketu s priznakem PSH/ACK

Host A chce prenest data /konkretne protokolu HTTP - GET/ na Host B. Data zabali do TCP paketu a obvykle ho vybavi ho priznakem PSH/ACK. Priznak PSH neni povinny (vubec se na nej neda spolehnout). To ze paket nese nejaka data neni urceno prikazem PSH, ale tim , ze datova delka TCP paketu je vetsi nez nula ! Delka paketu je v nasem pripade 528B.

Host A ----- PSH(SEQ)/ACK(ACK) -----> Host B

Host A ----- PSH(SEQ)/ACK(ACK) -----> Host B

No.	Time .	Source	Destination	Protocol	Info
1	0.000000	10.152.30.50	10.152.30.1	TCP	24773 > http [SYN] Seq=590857293 Ack=0 Win=5840 Len
2	0.001619	10.152.30.1	10.152.30.50	TCP	http > 24773 [SYN, ACK] Seq=1332285558 Ack=59085729
3	0.001668	10.152.30.50	10.152.30.1	TCP	24773 > http [ACK] Seq=590857294 Ack=1332285559 Win
4	0.011116	10.152.30.50	10.152.30.1	HTTP	GET / HTTP/1.1
5	0.012614	10.152.30.1	10.152.30.50	TCP	http > 24773 [ACK] Seq=1332285559 Ack=590857294 Win

Frame 4 (594 bytes on wire, 594 bytes captured)

Ethernet II, Src: aa:aa:03:00:00:00 (aa:aa:03:00:00:00), Dst: Intel-Hf_d1:35:7e (00:a0:c9:d1:35:7e)

Internet Protocol, Src: 10.152.30.50 (10.152.30.50), Dst: 10.152.30.1 (10.152.30.1)

Transmission Control Protocol, Src Port: 24773 (24773), Dst Port: http (80), Seq: 590857294, Ack: 1332285559, Len: 528

Source port: 24773 (24773)

Destination port: http (80)

Sequence number: 590857294

[Next sequence number: 590857822]

Acknowledgement number: 1332285559

Header length: 32 bytes

Flags: 0x0018 (PSH, ACK)

- 0... .. = Congestion Window Reduced (CWR): Not set
- .0.. = ECN-Echo: Not set
- ..0. = Urgent: Not set
- ...1 = Acknowledgment: Set
- 1... = Push: Set
-0.. = Reset: Not set
-0. = Syn: Not set
-0 = Fin: Not set

Window size: 1460

Checksum: 0xid85 [correct]

Options: (12 bytes)

File: "/root/.iptables/tcp.etherreal" 5367 Bytes 00.00.44 P: 37 D: 17 M: 0

2. Potvrzeni prijetej paketu s daty ACK

Host B potvrzuje prijetej paketu s daty paketem s priznakem ACK. Delka pakeu je 0B.

Host A <----- ACK(SEQ+528) ----- Host B

Host A <----- ACK(SEQ+528) ----- Host B

The screenshot shows the Wireshark interface with a packet capture filter: `(ip.addr eq 10.152.30.50 and ip.addr eq 10.152.30.1) and (tcp.port eq 24773)`. The packet list shows several packets, with packet 5 selected. The packet details pane shows the following information:

- Frame 5 (66 bytes on wire, 66 bytes captured)
- Ethernet II, Src: Intel-Hf_d1:35:7e (00:a0:c9:d1:35:7e), Dst: IntelCor_16:ab:0e (00:15:00:16:ab:0e)
- Internet Protocol, Src: 10.152.30.1 (10.152.30.1), Dst: 10.152.30.50 (10.152.30.50)
- Transmission Control Protocol, Src Port: http (80), Dst Port: 24773 (24773), Seq: 1332285559, Ack: 590857822, Len: 0
 - Source port: http (80)
 - Destination port: 24773 (24773)
 - Sequence number: 1332285559
 - Acknowledgement number: 590857822
 - Header length: 32 bytes
 - Flags: 0x0010 (ACK)
 - 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: 6432
 - Checksum: 0x946a [correct]
 - Options: (12 bytes)
 - [SEQ/ACK analysis]
 - [This is an ACK to the segment in frame: 4]

6. Ukonceni spojeni TCP

Muze inicionat libovolna strana /jak klient tak server/.

- half closed** - FIN pouze z jedne strany
- closed** - FIN z obou stran

1.Odeslani FIN/ACK paketu

Host B jiz odeslal vsechna data na Host A a chce proto ukoncit spojeni. Posila proto na Host A paket s priznakem FIN/ACK. Host B timto zahajil aktivni ukonceni spojeni a Host A jiz nyní muze provest pouze pasivni ukonceni spojeni. Od teto chvile jiz nemuze Host B odesilat data /pakety s priznakem PSH. Druha strana vsak muze v odesilani dat pokracovat, dokud sama neprovede ukonceni spojeni /tomuto stavu se rika half closed/.Paket se posuzuje jako by mel delku 1B.

Host A <----- FIN(SEQ-B)/ACK(ACK) ----- Host B

Host A <----- FIN(SEQ-B) /ACK(ACK) ----- Host B

The screenshot displays the Wireshark interface with a network capture. The packet list pane shows several packets, with packet 26 highlighted. The details pane for packet 26 shows the following information:

- Frame 26 (66 bytes on wire, 66 bytes captured)
- Ethernet II, Src: Intel-Hf_d1:35:7e (00:a0:c9:d1:35:7e), Dst: IntelCor_16:ab:0e (00:15:00:16:ab:0e)
- Internet Protocol, Src: 10.152.30.1 (10.152.30.1), Dst: 10.152.30.50 (10.152.30.50)
- Transmission Control Protocol, Src Port: http (80), Dst Port: 24773 (24773), Seq: 1332286340, Ack: 590858661, Len: 0
 - Source port: http (80)
 - Destination port: 24773 (24773)
 - Sequence number: 1332286340
 - Acknowledgement number: 590858661
 - Header length: 32 bytes
 - Flags: 0x0011 (FIN, ACK)
 - 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: 8576
 - Checksum: 0x7eee [correct]
 - Options: (12 bytes)

2. Potvrzeni prijati FIN paketu

Host A potvrzuje uzavreni spojeni druhou stranou /provadi pasivni ukonceni spojeni/. Je mozne v tomto paketu odeslat i priznak FIN a tim provest uzavreni celeho spojeni. Paket se posuzuje jako by mel delku 0B.

Host A ----- ACK(SEQ-B+1) -----> Host B

Nyni muze Host A stale posilat data na Host B a ten mu je bude potvrzovat pakety s priznakem ACK. Pokud jiz Host A nema co odesilat a preje si celkove ukoncit spojeni nasleduje dalsi krok.

Host A ----- ACK(SEQ-B+1) -----> Host B

The screenshot shows the Wireshark interface with a packet capture filter: `(ip.addr eq 10.152.30.50 and ip.addr eq 10.152.30.1) and (tcp.port eq 24773)`. The packet list shows several packets, with packet 27 selected. The packet details pane shows the following information:

- Frame 27 (66 bytes on wire, 66 bytes captured)
- Ethernet II, Src: aa:aa:03:00:00:00 (aa:aa:03:00:00:00), Dst: Intel-Hf_d1:35:7e (00:a0:c9:d1:35:7e)
- Internet Protocol, Src: 10.152.30.50 (10.152.30.50), Dst: 10.152.30.1 (10.152.30.1)
- Transmission Control Protocol, Src Port: 24773 (24773), Dst Port: http (80), Seq: 590858661, Ack: 1332286341, Len: 0
 - Source port: 24773 (24773)
 - Destination port: http (80)
 - Sequence number: 590858661
 - Acknowledgement number: 1332286341
 - Header length: 32 bytes
 - Flags: 0x0010 (ACK)
 - 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: 1728
 - Checksum: 0x8970 [correct]
 - Options: (12 bytes)
 - [SEQ/ACK analysis]
 - [This is an ACK to the segment in frame: 26]

3.Ukonceni spojeni i z druhe strany

Host A jiz nema co odesilat a preje si ukoncit spojeni. Odesle na Host B paket s priznakem FIN/ACK a tim ukonci spojeni i z druhe strany. Paket se posuzuje jako by mel delku 1B.

Host A ----- FIN(SEQ-A)/ACK(SEQ-B+1) -----> Host B

Host A ----- FIN(SEQ-A) / ACK(SEQ-B+1) -----> Host B

The screenshot shows the Wireshark interface with a capture filter: `(ip.addr eq 10.152.30.50 and ip.addr eq 10.152.30.1) and (tcp.port eq 247)`. The packet list shows several packets, with packet 28 highlighted in blue, indicating it is selected. Packet 28 is a TCP packet from 10.152.30.50 to 10.152.30.1, with the following details:

- Frame 28 (66 bytes on wire, 66 bytes captured)
- Ethernet II, Src: aa:aa:03:00:00:00 (aa:aa:03:00:00:00), Dst: Intel-Hf_d1:35:7e (00:a0:c9:d1:35:7e)
- Internet Protocol, Src: 10.152.30.50 (10.152.30.50), Dst: 10.152.30.1 (10.152.30.1)
- Transmission Control Protocol, Src Port: 24773 (24773), Dst Port: http (80), Seq: 590858661, Ack: 1332286341, Len: 0
 - Source port: 24773 (24773)
 - Destination port: http (80)
 - Sequence number: 590858661
 - Acknowledgement number: 1332286341
 - Header length: 32 bytes
 - Flags: 0x0011 (FIN, ACK)
 - 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: 1728
 - Checksum: 0x885e [correct]
 - Options: (12 bytes)

4. Potvrzení celkového ukončení spojení ACK

Host B provede potvrzení celkového ukončení spojení paketem s příznakem ACK. Po tomto paketu je spojení ukončeno a data nemohou tect ani jedním směrem.

Host A <----- ACK(SEQ-A+1) ----- Host B

Host A <----- ACK(SEQ-A+1) ----- Host B

The screenshot shows the Wireshark interface with a network capture. The filter is set to `(ip.addr eq 10.152.30.50 and ip.addr eq 10.152.30.1) and (tcp.port eq 24773)`. The packet list shows several packets, with packet 29 selected. The packet details pane shows the following information:

- Frame 29 (66 bytes on wire, 66 bytes captured)
- Ethernet II, Src: Intel-Hf_di:35:7e (00:a0:c9:d1:35:7e), Dst: IntelCor_16:ab:0e (00:15:00:16:ab:0e)
- Internet Protocol, Src: 10.152.30.1 (10.152.30.1), Dst: 10.152.30.50 (10.152.30.50)
- Transmission Control Protocol, Src Port: http (80), Dst Port: 24773 (24773), Seq: 1332286341, Ack: 590858662, Len: 0
 - Source port: http (80)
 - Destination port: 24773 (24773)
 - Sequence number: 1332286341
 - Acknowledgement number: 590858662
 - Header length: 32 bytes
 - Flags: 0x0010 (ACK)
 - 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: 8576
 - Checksum: 0x6d2c [correct]
 - Options: (12 bytes)
 - [SEQ/ACK analysis]
 - [This is an ACK to the segment in frame: 28]

6.Reset spojeni TCP

Spojeni se odmítá nastavením příznaku RST v TCP paketu. Toto se děje obvykle ve dvou případech:

- Klient požaduje spojení na TCP portu na kterém server neposlouchá
- Aplikace odmítá dále pokračovat v již navázaném spojení

Reset spojeni TCP

The screenshot shows the Wireshark interface with a packet capture of a TCP Reset (RST) packet. The packet list pane shows two packets:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.152.30.50	10.152.30.50	TCP	21155 > http [SYN] Seq=126786872 Ack=0 Win=32767 Len=0
2	0.000101	10.152.30.50	10.152.30.50	TCP	http > 21155 [RST, ACK] Seq=0 Ack=126786873 Win=0 Len=0

The packet details pane for the selected packet (Frame 2) shows the following information:

- Frame 2 (54 bytes on wire, 54 bytes captured)
- Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
- Internet Protocol, Src: 10.152.30.50 (10.152.30.50), Dst: 10.152.30.50 (10.152.30.50)
- Transmission Control Protocol, Src Port: http (80), Dst Port: 21155 (21155), Seq: 0, Ack: 126786873, Len: 0
 - Source port: http (80)
 - Destination port: 21155 (21155)
 - Sequence number: 0
 - Acknowledgement number: 126786873
 - Header length: 20 bytes
 - Flags: 0x0014 (RST, ACK)
 - Window size: 0
 - Checksum: 0x6682 [correct]
 - [SEQ/ACK analysis]
 - [This is an ACK to the segment in frame: 1]
 - [The RTT to ACK the segment was: 0.000101000 seconds]

7. Pouzite zdroje a nastroje

Zdroje

- Velky pruvodce protokoly TCP/IP a systemem DNS - L.Dostalek A.Kabelova
- V prezentaci jsou pouzity obrazky z vyse uvedene knihy.
- V prezentaci byly pouzity obrazky z knihy A.S.Tanenbaum: Computer Networks. Fourth edition. Pearson Education (Prentice Hall) 2003. ISBN 0-13-038488-7

Nastroje

- ethereal
- tcpdump
- OpenOffice

DeKuji za pozornost