

# Számítógépes Hálózatok

## Péntek

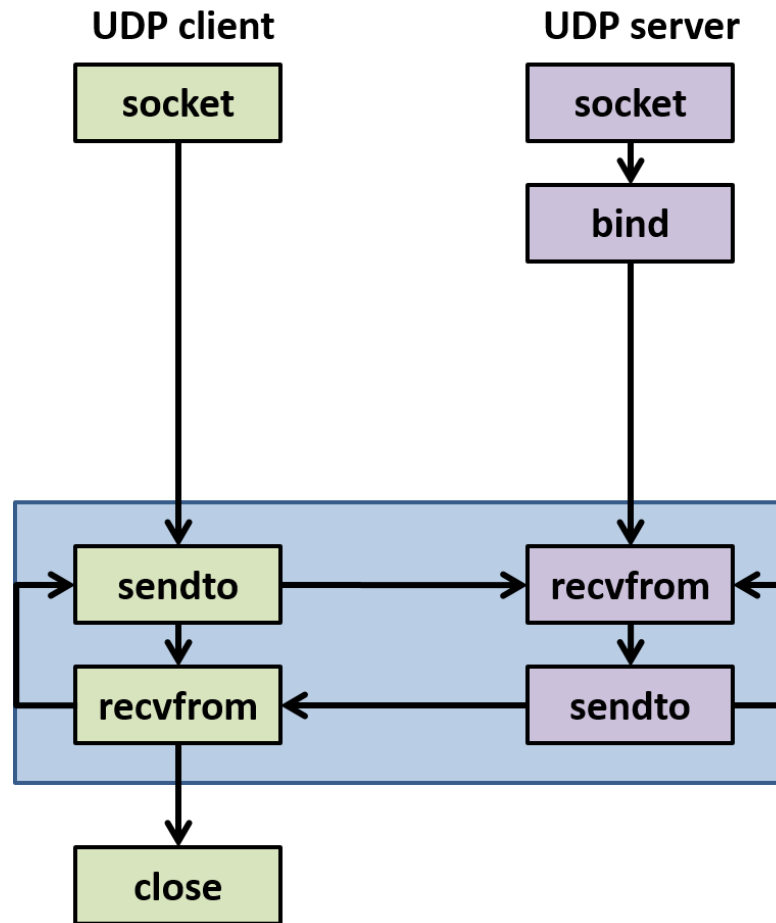
6. gyakorlat

# **PYTHON SOCKET - UDP**

# A kommunikációs csatorna kétféle típusa

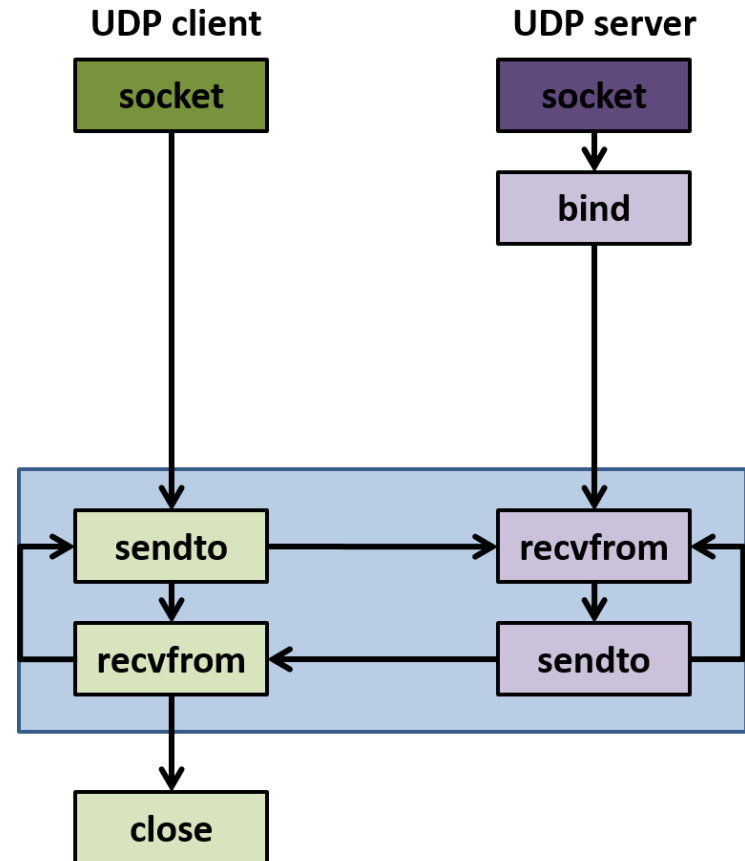
- Kapcsolat-orientált modell (analógia: telefonbeszélgetés)
  - csomagok megérkeznek jó sorrendben
  - ilyen protokoll a TCP
  - kapcsolódó típus: stream socket
- **Kapcsolat-nélküli modell (analógia: postai levelezés)**
  - **csomagok nem biztos, hogy sorrend helyesen érkeznek, sőt el is veszhetnek**
  - **előnye a jobb teljesítmény**
  - **ilyen protokoll a UDP**
  - **kapcsolódó típus: datagram socket**

# UDP



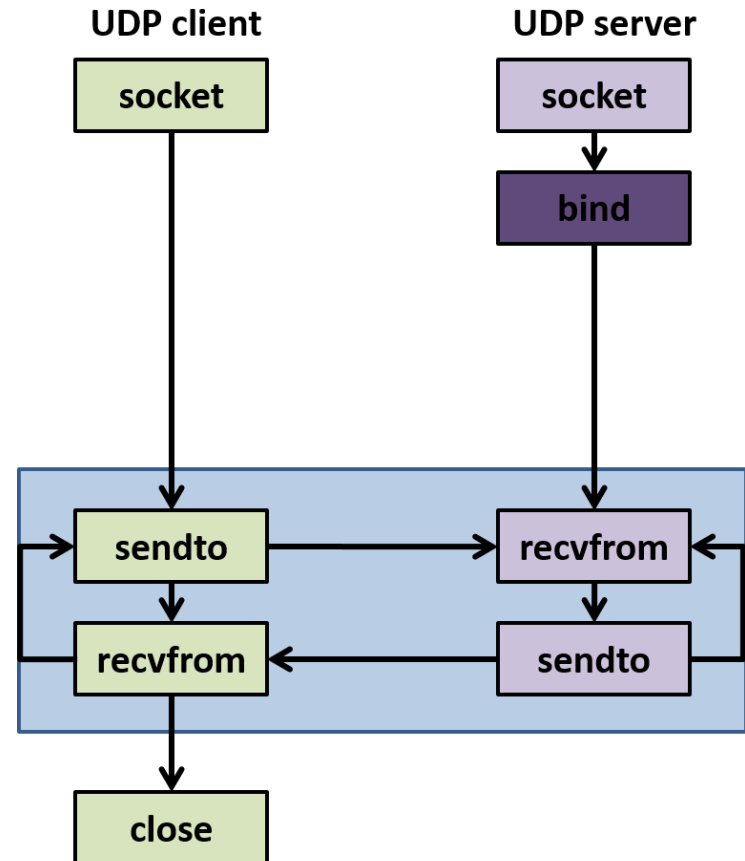
# Socket leíró beállítása

- `socket.socket( [family`  
                  `[, type`  
                  `[, proto]]])`
- `family` : `socket.AF_INET` → IPv4  
          (`AF_INET6` → IPv6)
- **`type` : `socket.SOCK_DGRAM` → UDP**
- `proto` : 0  
(alapértelmezett protokoll lesz)
- visszatérési érték: egy socket objektum, amelynek a metódusai a különböző socket rendszer hívásokat implementálják



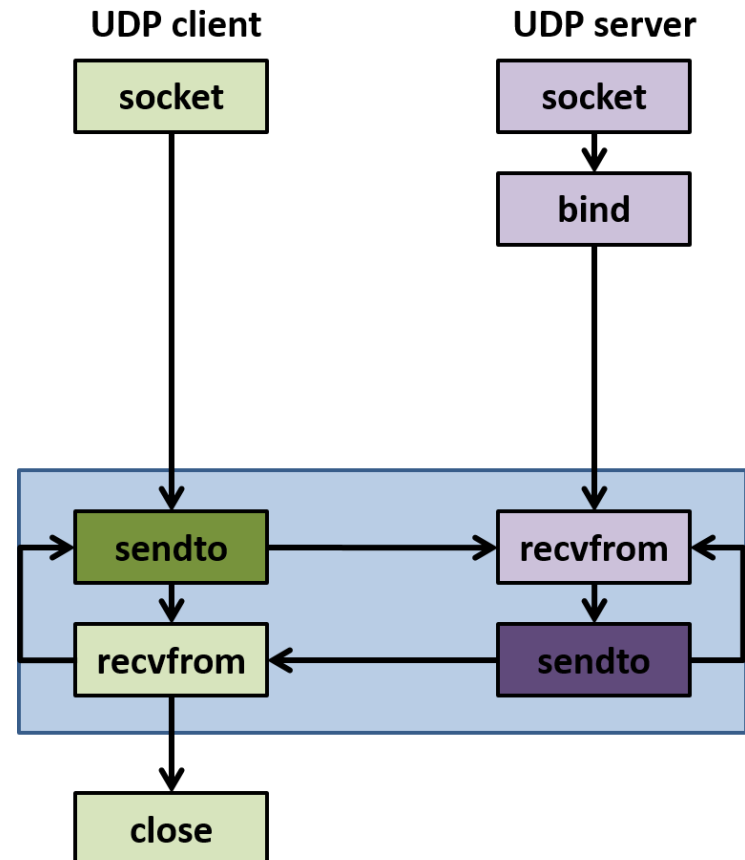
# Bindolás – ismételés

- `socket.socket.bind(address)`
- A socket objektum metódusa
- *address* : egy tuple, amelynek az első eleme egy hosztnév vagy IP cím (sztring reprezentációval), második eleme a portszám



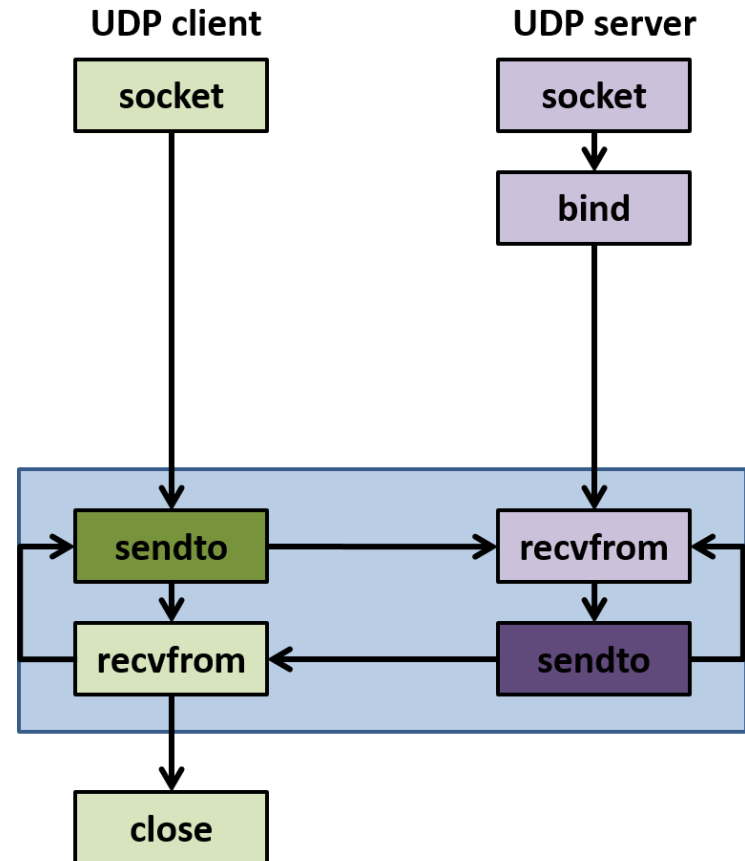
# sendto

- `socket.socket.sendto(bytes, address)`
- `socket.socket.sendto(bytes, flags, address)`
- A socket objektum metódusai
- Adatküldés (*bytes*) a socketnek
- *flags* : 0 (nincs flag meghatározva)
- **A socketnek előtte nem kell csatlakozni a távoli sockethez, mivel azt az *address* meghatározza**



# sendto

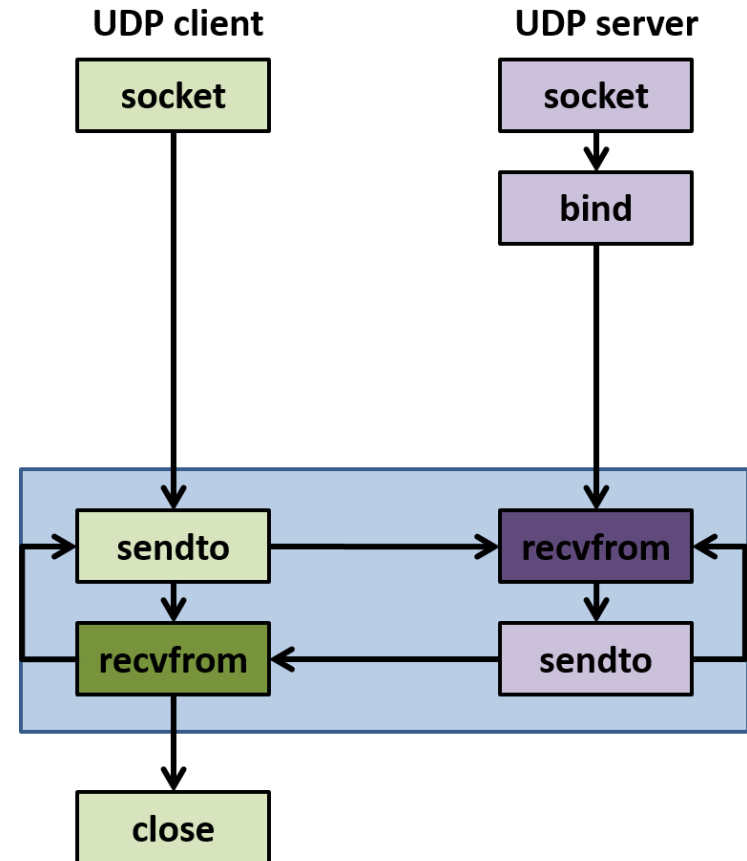
- **Fontos, hogy egy UDP üzenetnek bele kell férni egy egyszerű csomagba (ez IPv4 esetén kb. 65 KB-ot jelent)**
- visszatérési érték: az átküldött bájtok száma
  - az alkalmazásnak kell ellenőrizni, hogy minden adat átment-e
  - ha csak egy része ment át: újra kell küldeni a maradékot





# recvfrom

- `socket.socket.recvfrom( bufsize  
[, flags])`
- A socket objektum metódusa
- Üzenet fogadása
- *bufsize* : a max. adatmennyiség, amelyet egyszerre fogadni fog
- *flags* : 0 (nincs flag meghatározva)
- **visszatérési érték: egy (*bytes*, *address*) tuple, ahol a fogadott adat bytes reprezentációja és az adatküldő socket címe szerepel**



# UDP

- socket

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

- recvfrom()

```
data, address = sock.recvfrom(4096)
```

- sendto()

```
sent = sock.sendto(data, address)
```

# Feladat 1

Készítsünk egy kliens-szerver alkalmazást, amely UDP protokollt használ. A kliens küldje a „Hello Server” üzenetet a szervernek, amely válaszolja a „Hello Kliens” üzenetet.

Nézzük meg a megoldást!

## Feladat 2 - Számológép UDP felett

Készítsünk egy szerver-kliens alkalmazást, ahol a kliens elküld 2 számot és egy operátort a szervernek, amely kiszámolja és visszaküldi az eredményt. A kliens üzenete legyen struktúra. Használjunk UDP protokollt!

Nézzük meg a megoldást!

# Feladat 3 – fájlátvitel UDP felett

Fájlátvitel megvalósítása úgy, hogy a fájl letöltése UDP felett legyen megoldva. Készüljünk fel arra, hogy az átvitel során csomagvesztés, vagy sorrend csere is történhet! Az UDP szerver portját szabadon definiálhatjuk!

A hibakezeléshez egy javaslat:

Max. 1000 bájtanként UDP csomagokban elkezdjük átküldeni a fájl tartalmát. Minden csomag egy pár bájtos fejléccel indul, amiben jelezzük, hogy az utolsó darab-e, amit átküldtünk, továbbá egy másik mező jelzi a byteoffset-et a fájl elejétől. Működés:

- Ha a kliens kapott egy adatcsomagot, akkor egy nyugtacsomagot küld vissza.
- A nyugtacsomag fogadása után a szerver, küldi a következő adatcsomagot.
- Ha nem jön nyugta, akkor T idő után újraküldi a korábbi adatcsomagot. (pl.  $T=200\text{ms}$ )
- Ha nyugta veszik el, akkor a vevő az offset alapján el tudja dönteni, hogy egy új adatcsomag, vagy egy korábbi duplikátuma érkezett-e.
- Ha az utolsó csomag is megérkezett, akkor a kliens nyugtázza azt is és lezárja a fájlba írást. A szerver az utolsó nyugta után befejezi az átvitelt.

# Feladat 4

- Készítsünk egy **kliens-proxy-szerver** alkalmazást, ahol:
  - a **szerver** egy UDP szerver,
  - a **proxy** a **szerver** irányába egy UDP kliens, a **kliens** irányába egy TCP szerver,
  - a **kliens** egy TCP kliens a **proxy** irányába
- Folyamat:
  - a **kliens** küldje a ,Hello Server' üzenetet a **proxy**nak,
  - amely küldje tovább azt a **szerver**nek,
  - amely válaszolja vissza a ,Hello Kliens' üzenetet a **proxy**nak,
  - amely küldje tovább azt a **kliens**nek

# ALAPVETŐ ESZKÖZÖK I.

tcpdump, wireshark

# Hálózati forgalom felvétele/elemzése

tcpdump (Linux):

forgalom figyelő eszköz, a hálózati interfészről  
jövő csomagokat tudja olvasni

```
lakis@dpsdk-switch:~$ sudo tcpdump -i enp8s0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp8s0, link-type EN10MB (Ethernet), capture size 262144 bytes
09:15:26.376139 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 4154664816:4154665024, ack 289117644, win 384, length 208
09:15:26.376403 IP 192.168.0.102.43549 > 192.168.0.192.domain: 52681+ PTR? 35.167.181.157.in-addr.arpa. (45)
09:15:26.376994 IP 192.168.0.192.domain > 192.168.0.102.43549: 52681* 1/0/0 PTR oktnb35.inf.elte.hu. (78)
09:15:26.377100 IP 192.168.0.102.57511 > 192.168.0.192.domain: 64457+ PTR? 102.0.168.192.in-addr.arpa. (44)
09:15:26.377645 IP 192.168.0.192.domain > 192.168.0.102.57511: 64457 NXDomain 0/1/0 (79)
09:15:26.377723 IP 192.168.0.102.49012 > 192.168.0.192.domain: 6981+ PTR? 192.0.168.192.in-addr.arpa. (44)
09:15:26.377851 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 208:400, ack 1, win 384, length 192
09:15:26.378180 IP 192.168.0.192.domain > 192.168.0.102.49012: 6981 NXDomain 0/1/0 (79)
09:15:26.378267 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 400:976, ack 1, win 384, length 576
09:15:26.378291 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 976:1248, ack 1, win 384, length 272
09:15:26.378340 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 1248:1600, ack 1, win 384, length 352
09:15:26.378387 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 1600:1776, ack 1, win 384, length 176
09:15:26.378440 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 1776:1952, ack 1, win 384, length 176
09:15:26.378489 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 1952:2128, ack 1, win 384, length 176
09:15:26.378538 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 2128:2304, ack 1, win 384, length 176
09:15:26.378587 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 2304:2480, ack 1, win 384, length 176
09:15:26.378636 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 2480:2656, ack 1, win 384, length 176
09:15:26.378685 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 2656:2832, ack 1, win 384, length 176
09:15:26.378734 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 2832:3008, ack 1, win 384, length 176
09:15:26.378783 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 3008:3184, ack 1, win 384, length 176
09:15:26.378832 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 3184:3360, ack 1, win 384, length 176
```



# Hálózati forgalom felvétele/elemzése

## tcpdump – protokoll filter

```
lakis@dpsdk-switch:~$ sudo tcpdump -i enp8s0 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp8s0, link-type EN10MB (Ethernet), capture size 262144 bytes
09:16:49.470737 IP dpsdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 1, length 64
09:16:49.470766 IP 192.168.0.102 > dpsdk-pktgen: ICMP echo reply, id 5668, seq 1, length 64
09:16:50.471818 IP dpsdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 2, length 64
09:16:50.471834 IP 192.168.0.102 > dpsdk-pktgen: ICMP echo reply, id 5668, seq 2, length 64
09:16:51.471716 IP dpsdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 3, length 64
09:16:51.471732 IP 192.168.0.102 > dpsdk-pktgen: ICMP echo reply, id 5668, seq 3, length 64
09:16:52.471713 IP dpsdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 4, length 64
09:16:52.471729 IP 192.168.0.102 > dpsdk-pktgen: ICMP echo reply, id 5668, seq 4, length 64
09:16:53.471720 IP dpsdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 5, length 64
09:16:53.471736 IP 192.168.0.102 > dpsdk-pktgen: ICMP echo reply, id 5668, seq 5, length 64
```

# Hálózati forgalom felvétele/elemzése

## tcpdump – további filterek

```
lakis@dpsk-switch:~$ sudo tcpdump -i enp8s0 host 192.168.0.101 and port 1111
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp8s0, link-type EN10MB (Ethernet), capture size 262144 bytes
09:20:23.289035 IP dpsk-pktgen.48524 > 192.168.0.102.1111: Flags [S], seq 1544265047, win 29200, options [mss 1460,sackOK,TS val 409718781 ecr 0,nop,wscale 7],
length 0
09:20:23.289067 IP 192.168.0.102.1111 > dpsk-pktgen.48524: Flags [R.], seq 0, ack 1544265048, win 0, length 0
```

# Hálózati forgalom felvétele/elemzése

tcpdump  
– további  
filterek

```
lakis@dpdk-switch:~$ sudo tcpdump -vvv -A -i enp8s0 host 192.168.0.101 and port 1111
tcpdump: listening on enp8s0, link-type EN10MB (Ethernet), capture size 262144 bytes
09:27:26.361105 IP (tos 0x10, ttl 64, id 14532, offset 0, flags [DF], proto TCP (6), length 60)
    dpdk-pktgen.48546 > 192.168.0.102.1111: Flags [S], cksum 0xelle (correct), seq 3578222049, win 29200, options [mss 1460,sackOK,TS val 409824549 ecr 0,nop,wscale 7], length 0
    E...<8.0.0.....e...f...W.GU.....R.....
    .mm%.....
09:27:26.361137 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    192.168.0.102.1111 > dpdk-pktgen.48546: Flags [S.], cksum 0x824a (incorrect -> 0xdda8), seq 1341274724, ack 3578222050, win 28960, options [mss 1460,sackOK,TS val 835209270 ecr 409824549,nop,wscale 7], length 0
    E...<8.0.0.....e...f...W.GU...O.:e...J.....
    1.H6.mm%.....
09:27:26.361250 IP (tos 0x10, ttl 64, id 14533, offset 0, flags [DF], proto TCP (6), length 52)
    dpdk-pktgen.48546 > 192.168.0.102.1111: Flags [.], cksum 0x7cb0 (correct), seq 1, ack 1, win 229, options [nop,nop,TS val 409824549 ecr 835209270], length 0
    E..48.0.0.....e...f...W.GU.O.:e...J.....
    .mm%1.H6
09:27:31.152091 IP (tos 0x10, ttl 64, id 14534, offset 0, flags [DF], proto TCP (6), length 59)
    dpdk-pktgen.48546 > 192.168.0.102.1111: Flags [P.], cksum 0x4al4 (correct), seq 1:8, ack 1, win 229, options [nop,nop,TS val 409825747 ecr 835209270], length 7
    E...<8.0.0.....e...f...W.GU.O.:e...J.....
    .mq.1.H6Hello
09:27:31.152109 IP (tos 0x0, ttl 64, id 29267, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.102.1111 > dpdk-pktgen.48546: Flags [.], cksum 0x8242 (incorrect -> 0x734f), seq 1, ack 8, win 227, options [nop,nop,TS val 835210468 ecr 409825747], length 0
    E..4rs0.0.FU...f...e.W..O.:e.GU.....B....
    1.L..mq.
09:27:42.531278 IP (tos 0x0, ttl 64, id 29268, offset 0, flags [DF], proto TCP (6), length 55)
    192.168.0.102.1111 > dpdk-pktgen.48546: Flags [P.], cksum 0x8245 (incorrect -> 0x15be), seq 1:4, ack 8, win 227, options [nop,nop,TS val 835213313 ecr 409825747], length 3
    E..7rT0.0.FQ...f...e.W..O.:e.GU.....E....
    1.X..mq.Hi
09:27:42.531425 IP (tos 0x10, ttl 64, id 14535, offset 0, flags [DF], proto TCP (6), length 52)
    dpdk-pktgen.48546 > 192.168.0.102.1111: Flags [.], cksum 0x5d10 (correct), seq 8, ack 4, win 229, options [nop,nop,TS val 409828592 ecr 835213313], length 0
    E..48.0.0.....e...f...W.GU.O.:h.....
    .m|.1.X.
09:27:50.984854 IP (tos 0x10, ttl 64, id 14536, offset 0, flags [DF], proto TCP (6), length 67)
    dpdk-pktgen.48546 > 192.168.0.102.1111: Flags [P.], cksum 0xf203 (correct), seq 8:23, ack 4, win 229, options [nop,nop,TS val 409830705 ecr 835213313], length 15
    E..C8.0.0.....e...f...W.GU.O.:h.....
    .m.11.X.Hogy vagyunk?
09:27:50.984872 IP (tos 0x0, ttl 64, id 29269, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.102.1111 > dpdk-pktgen.48546: Flags [.], cksum 0x8242 (incorrect -> 0x4c81), seq 4, ack 23, win 227, options [nop,nop,TS val 835215426 ecr 409830705], length 0
```

# Hálózati forgalom felvétele/elemzése

tcpdump – mentés pcap fájlba és fájlból elemzés

```
lakis@dppdk-switch:~$ sudo tcpdump -w test.pcap -i enp8s0
tcpdump: listening on enp8s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C4 packets captured
6 packets received by filter
0 packets dropped by kernel
lakis@dppdk-switch:~$ tcpdump -r test.pcap
reading from file test.pcap, link-type EN10MB (Ethernet)
09:31:32.000164 IP 192.168.0.102.ssh > oktnb35.inf.elte.hu.55015: Flags [P.], seq 4154857792:4154857936, ack 289145644, win 384, length 144
09:31:32.060031 IP oktnb35.inf.elte.hu.55015 > 192.168.0.102.ssh: Flags [.], ack 144, win 3542, length 0
09:31:34.354029 IP 192.168.0.192.48309 > 255.255.255.255.7437: UDP, length 173
09:31:37.377992 IP 192.168.0.192.48309 > 255.255.255.255.7437: UDP, length 173
```

Pcap fájl visszajátszására is van lehetőség: tcpreplay

# Wireshark

- Forgalomelemző eszköz: korábban rögzített adatok elemzésére szolgál
- Windows-on és Linux-on is elérhető
- [www.wireshark.org](http://www.wireshark.org)

# Wireshark ablakok

01-centerpoints-0-0.pcap [Wireshark 1.12.1 (v1.12.1-0-g01b65bf from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.1.17	10.1.1.255	OLSR v1	84	OLSR (IPv4) Packet, Length: 20 Bytes
2	0.029669	10.1.1.1	10.1.1.255	OLSR v1	84	OLSR (IPv4) Packet, Length: 20 Bytes
3	0.048943	10.1.1.37	10.1.1.255	OLSR v1	84	OLSR (IPv4) Packet, Length: 20 Bytes
4	0.131429	10.1.1.18	10.1.1.255	OLSR v1	84	OLSR (IPv4) Packet, Length: 20 Bytes
5	0.133386	10.1.1.21	10.1.1.255	OLSR v1	84	OLSR (IPv4) Packet, Length: 20 Bytes
6	0.210913	10.1.1.25	10.1.1.255	OLSR v1	84	OLSR (IPv4) Packet, Length: 20 Bytes
7	0.318007	10.1.1.3	10.1.1.255	OLSR v1	84	OLSR (IPv4) Packet, Length: 20 Bytes
8	1.918582	10.1.1.21	10.1.1.255	OLSR v1	156	OLSR (IPv4) Packet, Length: 92 Bytes
9	1.961619	10.1.1.37	10.1.1.255	OLSR v1	172	OLSR (IPv4) Packet, Length: 108 Bytes
10	2.146770	10.1.1.18	10.1.1.255	OLSR v1	180	OLSR (IPv4) Packet, Length: 116 Bytes
11	2.220068	10.1.1.25	10.1.1.255	OLSR v1	148	OLSR (IPv4) Packet, Length: 84 Bytes
12	2.284430	10.1.1.3	10.1.1.255	OLSR v1	116	OLSR (IPv4) Packet, Length: 52 Bytes
13	2.309605	10.1.1.17	10.1.1.255	OLSR v1	156	OLSR (IPv4) Packet, Length: 92 Bytes
14	2.340700	10.1.1.1	10.1.1.255	OLSR v1	122	OLSR (IPv4) Packet, Length: 68 Bytes

Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits)

IEEE 802.11 Data, Flags: 0.....

Logical-Link control

Internet Protocol Version 4, Src: 10.1.1.17 (10.1.1.17), Dst: 10.1.1.255 (10.1.1.255)

User Datagram Protocol, Src Port: 698 (698), Dst Port: 698 (698)

Optimized Link State Routing Protocol

Offset	Hex	ASCII
0000	08 80 00 00 ff ff ff ff ff 00 00 00 00 00 11	.....
0010	00 00 00 00 00 11 00 00 aa aa 03 00 00 00 08 00	.....
0020	45 00 00 30 00 00 00 00 40 11 00 00 0a 01 01 11	E..0...@.....
0030	0a 01 01 ff 02 ba 02 ba 00 1c 00 00 00 14 00 00	.....
0040	01 86 00 10 0a 01 01 11 01 00 00 00 00 05 03	.....
0050	00 00 00 00	....

File: "C:\Users\ge-sar\Desktop\egyetem\nw... Packets: 218 · Displayed: 218 (100.0%) · Load time: 0:00.004 Profile: Default

Szűrők definiálására  
alkalmas input eszközök

Csomag összefoglaló  
nézete

Kiválasztott csomag  
hierarchikus nézet

Kiválasztott csomag bájtt-  
alapú nézet

Szűrés statisztikái

# Wireshark szűrők



- Operátorok: or, and, xor, not
- protokollok: ip, tcp, http... (teljes listát lásd → Expression gomb)
- Példa: `tcp.flags.ack==1 and tcp.dstport==80` (tcp nyugta flag és fogadó port beállítva)

# Wireshark példa

- A http\_out.pcapng állomány felhasználásával válaszoljuk meg az alábbi kérdéseket:
  1. Milyen oldalakat kértek le a szűrés alapján HTTP GET metódussal? Milyen böngészőt használtak hozzá?
  2. Hány darab képet érintett a böngészés?
  3. Volt-e olyan kérés, amely titkosított kommunikációt takar?



# Wireshark példa megoldás I.

http\_out.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request.method=="GET"

No.	Time	Source	Destination	Protocol	Length	Info
25	0.545934	192.168.1.100	173.194.39.104	HTTP	621	GET / HTTP/1.1
28	0.575342	192.168.1.100	173.194.116.143	HTTP	757	GET /?gfe_rd=cr&ei=1l4dVImtCufb8geysIHoCa HTTP/1.1
145	5.585338	192.168.1.100	157.181.161.79	HTTP	412	GET / HTTP/1.1
155	5.863160	192.168.1.100	157.181.161.79	HTTP	485	GET /Lapok/kezdolap.aspx HTTP/1.1
178	6.015329	192.168.1.100	157.181.161.79	HTTP	568	GET /Style%20Library/hu-HU/Core%20Styles/Band.css HTTP/1.1
182	6.016702	192.168.1.100	157.181.161.79	HTTP	571	GET /Style%20Library/hu-HU/Core%20Styles/controls.css HTTP/1.1
183	6.017730	192.168.1.100	157.181.161.79	HTTP	550	GET /Style%20Library/elteik.css HTTP/1.1
184	6.018474	192.168.1.100	157.181.161.79	HTTP	557	GET /_styles/HtmlEditorCustomStyles.css HTTP/1.1
215	6.073612	192.168.1.100	157.181.161.79	HTTP	565	GET /Style%20Library/ik/ikonok/oldalalterkep.png HTTP/1.1
216	6.075335	192.168.1.100	157.181.161.79	HTTP	561	GET /Style%20Library/ik/ikonok/kereses.png HTTP/1.1
217	6.076699	192.168.1.100	157.181.161.79	HTTP	559	GET /Style%20Library/ik/ikonok/email.png HTTP/1.1
221	6.185875	192.168.1.100	157.181.161.79	HTTP	557	GET /Style%20Library/ik/ikonok/rss.png HTTP/1.1
222	6.191369	192.168.1.100	157.181.161.79	HTTP	561	GET /Style%20Library/ik/ikonok/english.png HTTP/1.1
223	6.195866	192.168.1.100	157.181.161.79	HTTP	569	GET /SiteCollectionImages/terkep_reszlet.png HTTP/1.1
224	6.196355	192.168.1.100	157.181.161.79	HTTP	562	GET /SiteCollectionImages/kari_galeria.png HTTP/1.1
225	6.196788	192.168.1.100	157.181.161.79	HTTP	567	GET /Style%20Library/ik/ik_bg_lighter.gif HTTP/1.1

Frame 155: 485 bytes on wire (3880 bits), 485 bytes captured (3880 bits) on interface 0

Ethernet II, Src: HonHaiPr\_6d:48:8b (cc:af:78:6d:48:8b), Dst: Tp-LinkT\_1c:6b:9a (00:27:19:1c:6b:9a)

Internet Protocol Version 4, Src: 192.168.1.100, Dst: 157.181.161.79

Transmission Control Protocol, Src Port: 36766, Dst Port: 80, Seq: 1, Ack: 1, Len: 431

Hypertext Transfer Protocol

GET /Lapok/kezdolap.aspx HTTP/1.1\r\n

Host: www.inf.elte.hu\r\n

Connection: keep-alive\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8\r\n

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.120 Safari/537.36\r\n

0000 00 27 19 1c 6b 9a cc af 78 6d 48 8b 08 00 45 00 .....k...xmH...E.

0010 01 d7 0f 0b 40 00 80 06 e9 04 c0 a8 01 64 9d b5 .....@... ..d..

0020 a1 4f 8f 9e 00 50 65 03 aa 42 ea 40 8a 0a 50 18 ..O...Pe. .8. @..P.

0030 44 70 f2 e7 00 00 47 45 54 20 2f 4c 61 70 6f 6b Dp....GE T /Lapok

0040 2f 6b 65 7a 64 6f 6c 61 70 2e 61 73 70 78 20 48 /kezdola p.aspx H

0050 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 TTP/1.1. .Host: w

0060 77 77 2e 69 6e 66 2e 65 6c 74 65 2e 68 75 0d 0a ww.inf.e lte.hu..

0070 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 Connecti on: keep

0080 2d 61 6c 69 76 65 0d 0a 41 63 63 65 70 74 3a 20 -alive.. Accept:

0090 74 65 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 text/htm l,applic

00a0 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c ation/xh tml+xml,

Frame (frame), 485 bytes

Packets: 1453 · Displayed: 70 (4.8%) · Load time: 0:0.68

Profile: Default

# Wireshark példa megoldás I.

- Milyen oldalakat kértek le a szűrés alapján HTTP GET metódussal? Milyen böngészőt használtak hozzá?
- Szűrés: `http.request.method=="GET"`
- User-agent header-ből lehet következtetni a böngésző típusára: User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.120 Safari/537.36
- Ehhez segítségünkre lehet ez a link:  
[http://www.zytrax.com/tech/web/browser\\_ids.htm](http://www.zytrax.com/tech/web/browser_ids.htm)

# Wireshark példa megoldás II.

Wireshark capture of an HTTP GET request for a PNG image. The filter `http.accept == "image/webp,*/*;q=0.8"` is applied. The selected packet shows the full HTTP request with various headers like Host, Connection, Accept, User-Agent, Referer, etc. The packet bytes pane shows the raw data with hex and ASCII views.

No.	Time	Source	Destination	Protocol	Length	Info
215	6.073612	192.168.1.100	157.181.161.79	HTTP	565	GET /Style%20Library/ik/ikonok/oldalterkep.png HTTP/1.1
216	6.075335	192.168.1.100	157.181.161.79	HTTP	561	GET /Style%20Library/ik/ikonok/kereses.png HTTP/1.1
217	6.076699	192.168.1.100	157.181.161.79	HTTP	559	GET /Style%20Library/ik/ikonok/email.png HTTP/1.1
221	6.185875	192.168.1.100	157.181.161.79	HTTP	557	GET /Style%20Library/ik/ikonok/rss.png HTTP/1.1

Frame 215: 565 bytes on wire (4520 bits), 565 bytes captured (4520 bits) on interface 0

Ethernet II, Src: HonHaiPr\_6d:48:8b (cc:af:78:6d:48:8b), Dst: Tp-LinkT\_1c:6b:9a (00:27:19:1c:6b:9a)

Internet Protocol Version 4, Src: 192.168.1.100, Dst: 157.181.161.79

Transmission Control Protocol, Src Port: 36766, Dst Port: 80, Seq: 432, Ack: 33346, Len: 511

Hypertext Transfer Protocol

GET /Style%20Library/ik/ikonok/oldalterkep.png HTTP/1.1\r\n

Host: www.inf.elte.hu\r\n

Connection: keep-alive\r\n

Accept: image/webp,\*/\*;q=0.8\r\n

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.120 Safari/537.36\r\n

Referer: http://www.inf.elte.hu/Lapok/kezdolap.aspx\r\n

Accept-Encoding: gzip,deflate,sdch\r\n

Accept-Language: hu-HU,hu;q=0.8,en-US;q=0.6,en;q=0.4\r\n

If-None-Match: "{02084E23-E4E5-4B60-A388-F77D8FAD8892},2"\r\n

If-Modified-Since: Wed, 16 Jan 2008 11:58:25 GMT\r\n

0090 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 41 63 : keep-a live..Ac

00a0 63 65 70 74 3a 20 69 6d 61 67 65 2f 77 65 62 70 cept: im age/webp

00b0 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 55 73 65 72 ,/\*;q=0 .8..User

00c0 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f -Agent: Mozilla/

00d0 35 2e 30 20 28 57 69 6e 64 6f 77 73 20 4e 54 20 5.0 (Win dows NT

00e0 36 2e 31 3b 20 57 4f 57 36 34 29 20 41 70 70 6c 6.1; WOW 64) Appl

00f0 65 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20 28 eWebKit/ 537.36 (

0100 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 63 6b KHTML, l ike Geck

0110 6f 29 20 43 68 72 6f 6d 65 2f 33 37 2e 30 2e 32 o) Chrom e/37.0.2

0120 30 36 32 2e 31 32 30 20 53 61 66 61 72 69 2f 35 062.120 Safari/5

0130 33 37 2e 33 36 0d 0a 52 65 66 65 72 65 72 3a 20 37.36..R eferer:

0140 68 74 74 70 3a 2f 2f 77 77 2e 69 6e 66 2e 65 http://w ww.inf.e

0150 6c 74 65 2e 68 75 2f 4c 61 70 6f 6b 2f 6b 65 7a lte.hu/L apok/kez

0160 64 6f 6c 61 70 2e 61 73 70 78 0d 0a 41 63 63 65 dolap.as px..Acce

0170 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 pt-Encod ing: gzi

0180 70 2c 64 65 66 6c 61 74 65 2c 73 64 63 68 0d 0a p,deflat e,sdch..

0190 41 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a Accept-L anguage:

01a0 20 68 75 2d 48 55 2c 68 75 3b 71 3d 30 2e 38 2c hu-HU,h u;q=0.8,

HTTP Accept (http.accept), 30 bytes

Packets: 1453 Displayed: 26 (1.8%) Load time: 0:0.63 Profile: Default

# Wireshark példa megoldás II.

- Hány darab képet érintett a böngészés?
- Szűrés: `http.accept == "image/webp,*/*;q=0.8"`
- Accept header: a kérésre adott válasz tartalmának elfogadható típusa
- WebP: új, veszteséges tömörítést alkalmazó képformátum, amelyet a Google fejlesztett ki a web hálózati forgalmának csökkentésére

# Wireshark példa megoldás III.

http\_out.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.dstport==443

No.	Time	Source	Destination	Protocol	Length	Info
4	0.007666	192.168.1.100	173.194.116.143	TCP	66	36763→443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
10	0.043654	192.168.1.100	173.194.116.143	TCP	54	36763→443 [ACK] Seq=1 Ack=1 Win=17160 Len=0
13	0.045430	192.168.1.100	173.194.116.143	TLSv1.2	267	Client Hello
17	0.082913	192.168.1.100	173.194.116.143	TCP	54	36763→443 [ACK] Seq=214 Ack=2861 Win=17160 Len=0
19	0.099481	192.168.1.100	173.194.116.143	TLSv1.2	308	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
22	0.133310	192.168.1.100	173.194.116.143	TCP	54	36763→443 [ACK] Seq=468 Ack=4215 Win=15804 Len=0
24	0.328037	192.168.1.100	173.194.116.143	TCP	54	36763→443 [ACK] Seq=468 Ack=4252 Win=15768 Len=0
31	0.656301	192.168.1.100	173.194.116.143	TLSv1.2	103	Application Data
32	0.656739	192.168.1.100	173.194.116.143	TLSv1.2	91	Application Data
33	0.657355	192.168.1.100	173.194.116.143	TLSv1.2	111	Application Data
34	0.659378	192.168.1.100	173.194.116.143	TLSv1.2	693	Application Data
35	0.671176	192.168.1.100	173.194.39.120	TCP	66	36764→443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
37	0.692679	192.168.1.100	173.194.39.120	TCP	54	36764→443 [ACK] Seq=1 Ack=1 Win=17160 Len=0
39	0.693518	192.168.1.100	173.194.39.120	TLSv1.2	269	Client Hello
43	0.710588	192.168.1.100	173.194.39.120	TCP	54	36764→443 [ACK] Seq=216 Ack=2861 Win=17160 Len=0
45	0.727492	192.168.1.100	173.194.39.120	TLSv1.2	308	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
48	0.745754	192.168.1.100	173.194.39.120	TCP	54	36764→443 [ACK] Seq=470 Ack=4216 Win=15804 Len=0
53	0.794975	192.168.1.100	173.194.116.143	TCP	54	36763→443 [ACK] Seq=1250 Ack=5893 Win=17160 Len=0

Frame 13: 267 bytes on wire (2136 bits), 267 bytes captured (2136 bits) on interface 0

Ethernet II, Src: HonHaiPr\_6d:48:8b (cc:af:78:6d:48:8b), Dst: Tp-LinkT\_1c:6b:9a (00:27:19:1c:6b:9a)

Internet Protocol Version 4, Src: 192.168.1.100, Dst: 173.194.116.143

Transmission Control Protocol, Src Port: 36763, Dst Port: 443, Seq: 1, Ack: 1, Len: 213

Secure Sockets Layer

0030 10 c2 4c d5 00 00 16 03 01 00 d0 01 00 00 cc 03 ..L... ..  
0040 03 79 3e 23 0a c0 08 98 16 d7 3a 0f 4c 50 0b 3d .y>#... ..LP.=  
0050 35 01 f1 22 2f 12 50 7a ad 5c 7b 13 2d a7 5c cd 5.."/.Pz .\{.-.\.  
0060 ea 00 00 28 cc 14 cc 13 c0 2b c0 2f 00 9e c0 0a ...((... +./....  
0070 c0 09 c0 13 c0 14 c0 07 c0 11 00 33 00 32 00 39 .....3.2.9  
0080 00 9c 00 2f 00 35 00 0a 00 05 00 04 01 00 00 7b .../.5. ....{  
0090 00 00 00 12 00 10 00 00 0d 77 77 77 2e 67 6f 6f .....www.goo  
00a0 67 6c 65 2e 68 75 ff 01 00 01 00 00 0a 00 08 00 gle.hu... ..  
00b0 06 00 17 00 18 00 19 00 0b 00 02 01 00 00 23 00 .....#.

Secure Sockets Layer (ssl), 213 bytes

Packets: 1453 · Displayed: 428 (29.5%) · Load time: 0:0.67 · Profile: Default

# Wireshark példa megoldás III.

- Volt-e olyan kérés, amely titkosított kommunikációt takar?
- Szűrés: `tcp.dstport==443`
- Transport Layer Security (TLS) titkosító protokoll feletti HTTP kommunikációra utal a 443-as port
- Elektronikus levelezéshez, banki szolgáltatásokhoz stb. elengedhetetlen
- Nélküle le lehet hallgatni a kommunikációt, lásd a következő diát (`sample3.pcapng` felhasználásával)

# Wireshark – „leleplezés”

sample3.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
12	4.106306	192.168.2.101	84.2.36.197	HTTP	131	POST /pages/user/login.jsp?method=Login HTTP/1.1 (application/x-www-form-urlencoded)
13	4.121234	84.2.36.197	192.168.2.101	TCP	60	80→65533 [ACK] Seq=1 Ack=1393 Win=7890 Len=0
14	6.160352	84.2.36.197	192.168.2.101	HTTP	594	HTTP/1.1 302 Moved Temporarily

Internet Protocol Version 4, Src: 192.168.2.101, Dst: 84.2.36.197

Transmission Control Protocol, Src Port: 65533, Dst Port: 80, Seq: 1316, Ack: 1, Len: 77

[2 Reassembled TCP Segments (1392 bytes): #11(1315), #12(77)]

Hypertext Transfer Protocol

POST /pages/user/login.jsp?method=Login HTTP/1.1\r\n

Host: **iiw.hu\r\n**

Connection: keep-alive\r\n

Referer: http://iiw.hu/pages/user/login.jsp\r\n

Content-Length: 77\r\n

Cache-Control: max-age=0\r\n

Origin: http://iiw.hu\r\n

Content-Type: application/x-www-form-urlencoded\r\n

Accept: application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5\r\n

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US) AppleWebKit/534.3 (KHTML, like Gecko) Chrome/6.0.472.59 Safari/534.3\r\n

Accept-Encoding: gzip,deflate,sdch\r\n

Accept-Language: hu-HU,hu;q=0.8,en-US;q=0.6,en;q=0.4\r\n

Accept-Charset: ISO-8859-2,utf-8;q=0.7,\*;q=0.3\r\n

[truncated]Cookie: ajaxable=1; forgetEmail=0; email=bGFRaXNAaW5mLmVsdGUuaHU\$; httpslogin=0; \_\_utma=114476831.2067882217.1284887869.1284887869.1284887869.1; \_\_utmc=114476831;\r\n

[Full request URI: <http://iiw.hu/pages/user/login.jsp?method=Login>]

[HTTP request 1/1]

[Response in frame: 14]

File Data: 77 bytes

HTML Form URL Encoded: application/x-www-form-urlencoded

Form item: "email" = "kulimasz@perec.hu"

Form item: "password" = "kistraktor53"

0020 24 c5 ff fd 00 50 e6 41 31 0f 9a 6f 83 0a 50 18 \$....P.A 1..o..P.

0030 fa f0 7c ef 00 00 65 6d 61 69 6c 3d 6b 75 6c 69 ..|...em ail=kuli

0040 6d 61 73 7a 25 34 30 70 65 72 65 63 2e 68 75 26 masz%40p erec.hu&

0050 70 61 73 73 77 6f 72 64 3d 6b 69 73 74 72 61 6b password =kistrak

0060 74 6f 72 35 33 26 68 74 74 70 73 6c 6f 67 69 6e tor53&ht tpslogin

Frame (131 bytes) Reassembled TCP (1392 bytes)

sample3 Packets: 381 · Displayed: 381 (100.0%) · Load time: 0:0.16 Profile: Default

# Alhálózati maszk

- Az alhálózat egy logikai felosztása egy IP hálózatnak. Az IP cím ezért két részből áll: hálózatszámból és hoszt azonosítóból.
- A szétválasztás a 32 bites alhálózati maszk segítségével történik, amellyel bitenkénti ÉS-t alkalmazva az IP címre megkapjuk a hálózat-, komplementerével pedig a hoszt azonosítót.
- Ez arra jó, hogy meg tudjuk határozni, hogy a címzett állomás a helyi alhálózaton van-e, vagy sem.
- Az utóbbi esetben az alapértelmezett router felé továbbítják a csomagot (default gateway).



# Alhálózati maszk

- CIDR jelölés: kompakt reprezentációja egy IP címnek és a hozzá tartozó hálózatszámnak
- → IP cím + '/' + decimális szám.
- Pl.: 135.46.57.14/24 esetben 135.46.57.14 az IP cím,
- 255.255.255.0 a hálózati maszk (24 db. 1-es bit az elejétől),
- így 135.46.57.0 a hálózat azonosító.

# Alhálózati maszk – példa

	10000111	00101110	00111001	00001110	135.46.57.14
AND	11111111	11111111	11111111	00000000	255.255.255.0
<hr/>					
	10000111	00101110	00111001	00000000	135.46.57.0

**135.46.57.14/24 → 135.46.57.0**

# Számolós feladat 1

- Hány cím érhető el a következő alhálózati maszkokkal? Adjuk meg a minimális és maximális címet is!
- 188.100.22.12/32
- 188.100.22.12/20
- 188.100.22.12/10

# Számológép feladat 1 megoldása

**188.100.22.12/32**

	10111100	01100100	00010110	00001100	188.100.22.12
AND	11111111	11111111	11111111	11111111	255.255.255.255
<hr/>					
	10111100	01100100	00010110	00001100	188.100.22.12

egy darab a 188.100.22.12

# Számológép feladat 1 megoldása

**188.100.22.12/20**

	10111100	01100100	00010110	00001100	188.100.22.12
AND	11111111	11111111	11110000	00000000	255.255.240.0
<hr/>					
	10111100	01100100	00010000	00000000	188.100.16.0

$2^{32-20} = 2^{12} = 4096$  darab lenne, de valójában ebből még kettőt le kell vonni, mert speciális jelentéssel bírnak:

- csupa 0: az alhálózat hálózati címe (magára az alhálózatra vonatkozik)
- csupa 1-es: broadcast a helyi hálózaton

Min.	10111100	01100100	00010000	00000001	188.100.16.1
Max.	10111100	01100100	00011111	11111110	188.100.31.254

# Számológép feladat 1 megoldása

**188.100.22.12/10**

	10111100	01100100	00010110	00001100	188.100.22.12
AND	11111111	11000000	00000000	00000000	255.255.240.0
<hr/>					
	10111100	01000000	00000000	00000000	188.64.0.0

$$2^{32-10} = 2^{22} - 2 = 4194302 \text{ darab}$$

Min.	10111100	01000000	00000000	00000001	188.64.0.1
Max.	10111100	01111111	11111111	11111110	188.127.255.254

# Hálózati címfordítás (NAT)

- Gyors javítás az IP címek elfogyásának problémájára.
- Az internet forgalomhoz minden cégnek egy vagy legalábbis kevés IP címet adnak (publikus IP cím(ek))
- A publikus IP cím hozzá van rendelve egy router-hez, a helyi hálózaton (LAN) belül, - amely mögötte van, - minden eszközhöz egy privát IP cím van rendelve
- A privát IP címek csak a LAN-on belül érvényesek (vannak IP cím tartományok erre a célra foglalva)

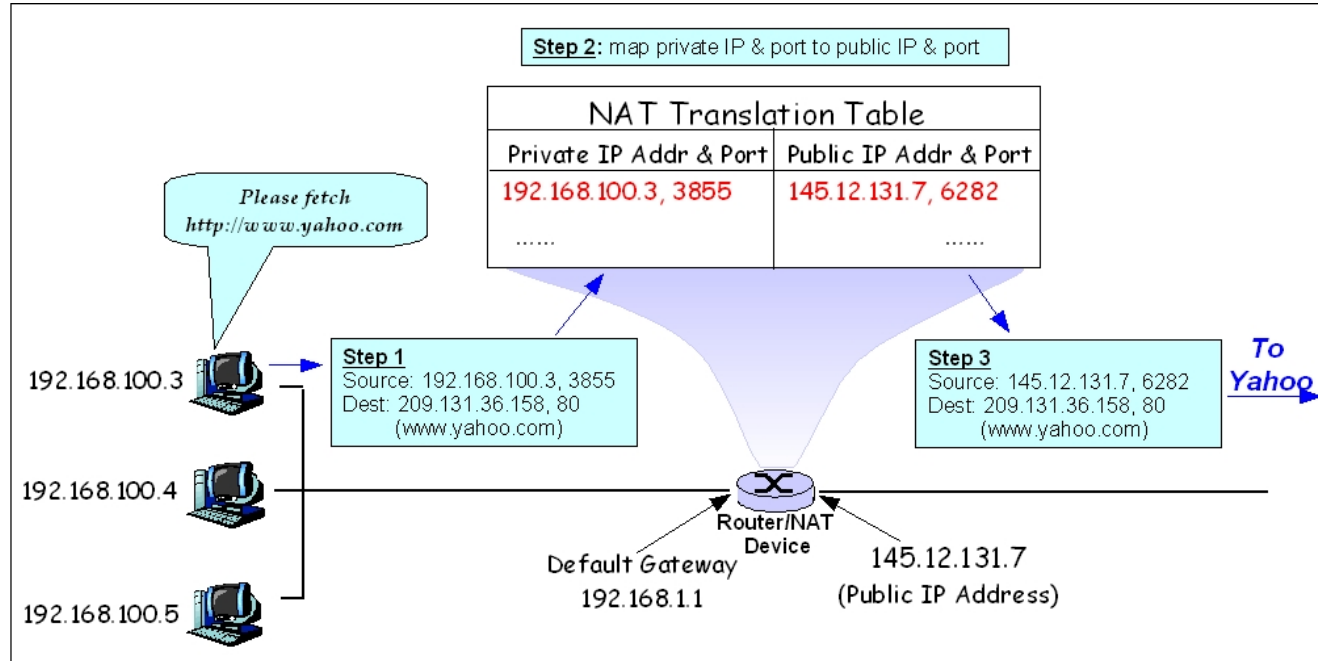
# Hálózati címfordítás (NAT)

- Ha a helyi hálózaton lévő másik géppel akarunk kapcsolatot létesíteni → közvetlenül el tudjuk érni
- Amikor helyi eszkösről akarunk egy külső eszközt elérni, mi történik?
- Szükségünk van port mezők használatára, ami TCP-nél vagy UDP-nél van



# Hálózati címfordítás (NAT)

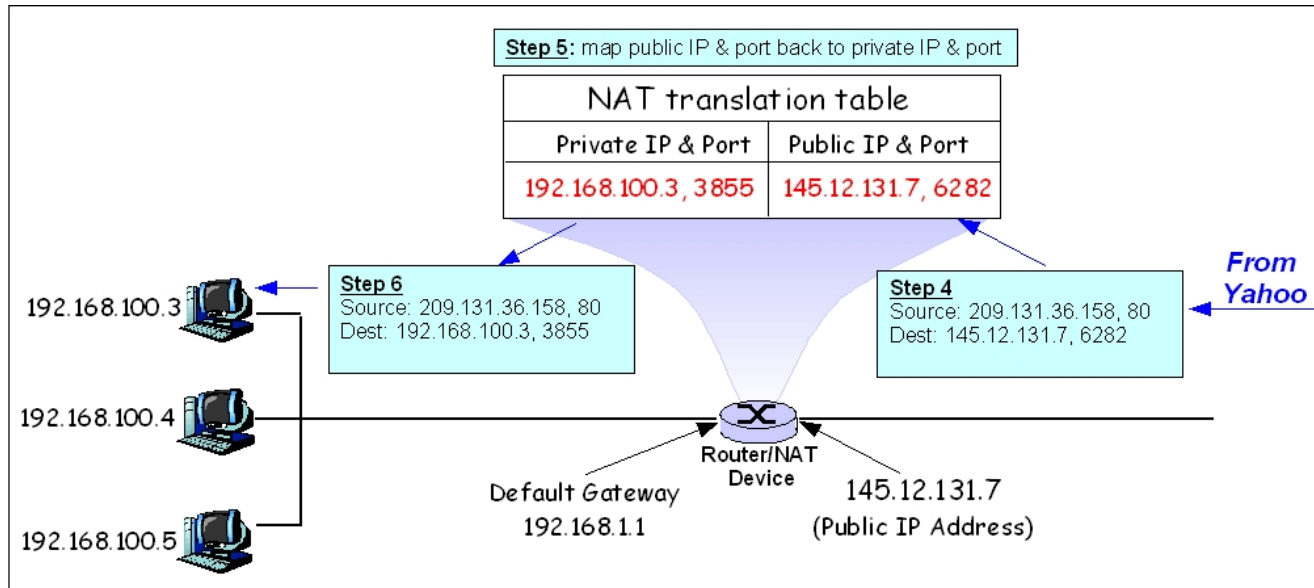
Forrás: [https://en.wikibooks.org/wiki/Communication\\_Networks/NAT\\_and\\_PAT\\_Protocols](https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols)



- 192.168.100.3 privát IP című gépről HTTP kérés, 3855 porton → Default gateway (192.168.1.1): megnézi a translációs tábláját:
  - Ha létezik már a (192.168.100.3, 3855) párhoz (publikus IP cím, port) bejegyzés → lecseréli a küldő forrását arra
  - Ha nincs létrehoz egy új bejegyzést (egyedi lesz!), és azt használja fel a cseréhez

# Hálózati címfordítás (NAT)

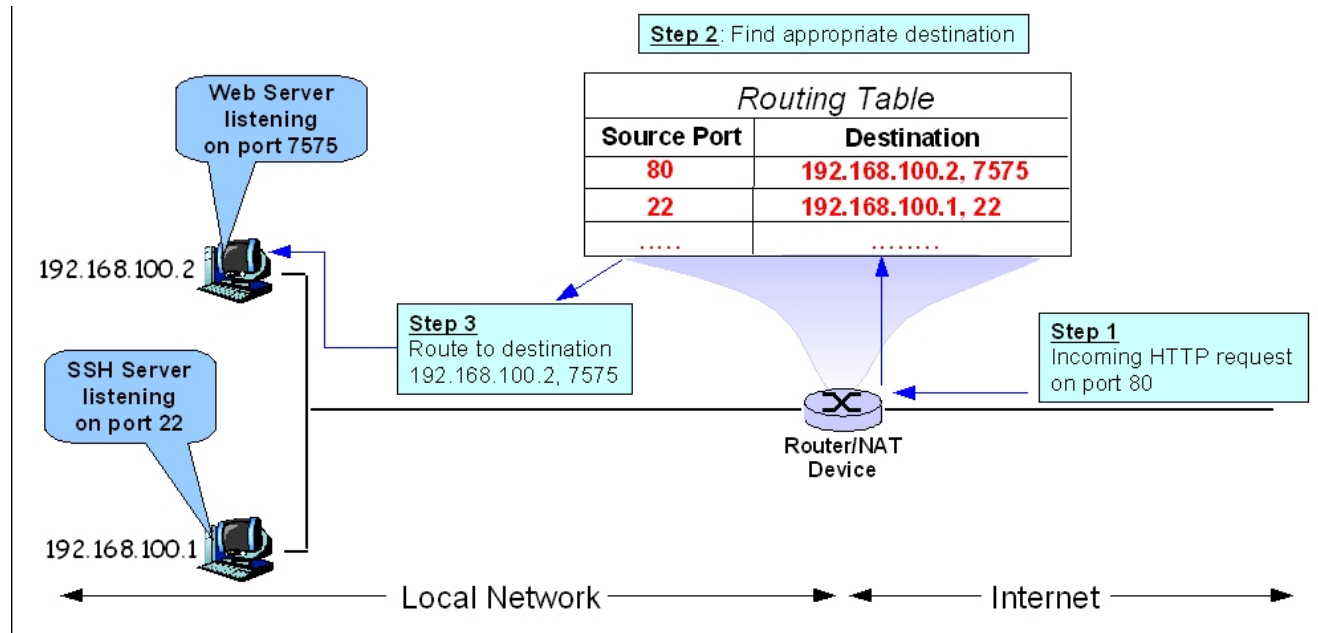
Forrás: [https://en.wikibooks.org/wiki/Communication\\_Networks/NAT\\_and\\_PAT\\_Protocols](https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols)



- A HTTP válasz a yahoo-tól ugyanúgy a router transzlációs tábláján keresztül megy végbe, csak fordított irányban
- Egy különbség: hiányzó bejegyzés esetén a csomagot eldobja a router

# Porttovábbítás (port forwarding)

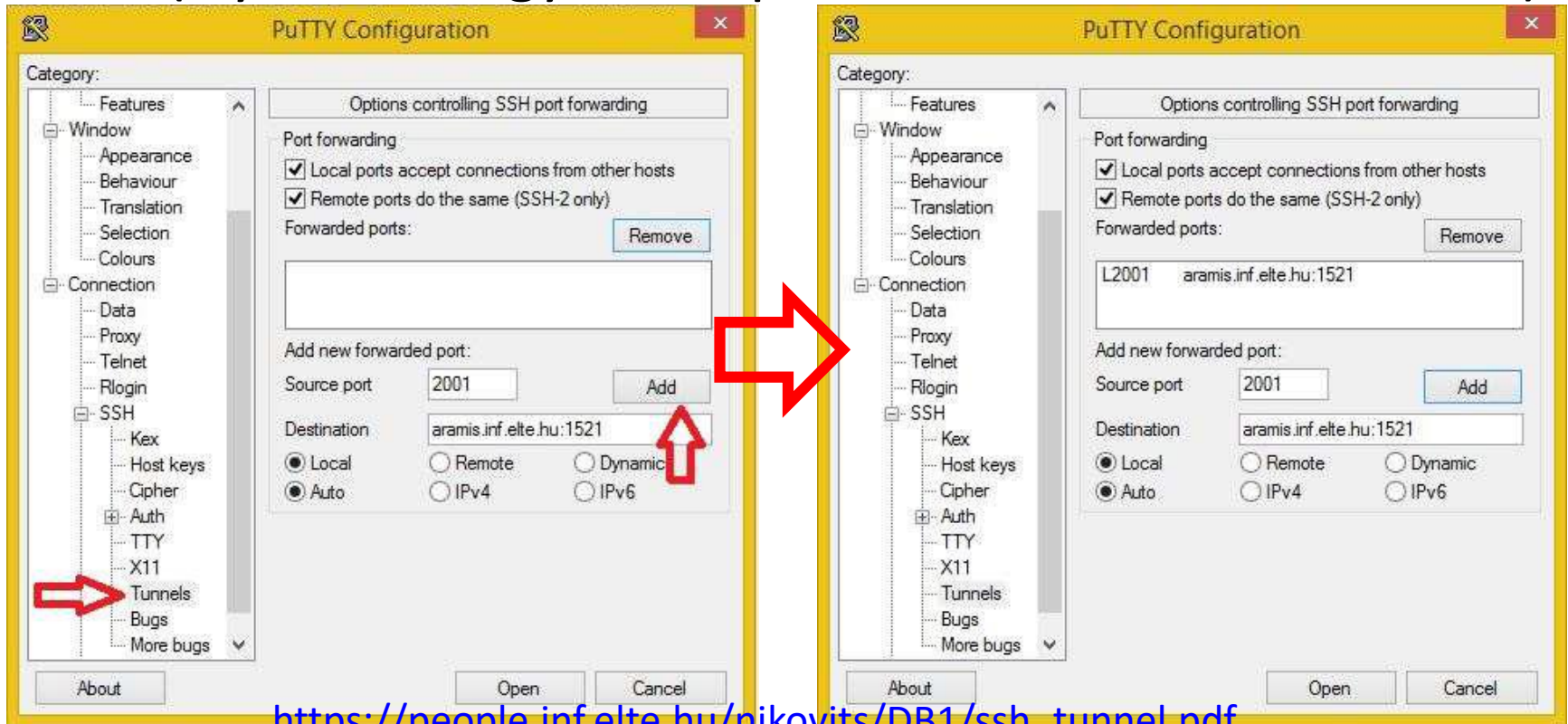
Forrás: [https://en.wikibooks.org/wiki/Communication\\_Networks/NAT\\_and\\_PAT\\_Protocols](https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols)



- Az előző példánál a címfordítás transzparens volt (csak a router tudott arról, hogy IP konverzió zajlik). Mit lehet tenni, ha pl. egy belső hálózaton lévő HTTP szervert akarunk elérni kívülről?
- **Porttovábbítás** lehetővé teszi adott lokális hálózaton (LAN) lévő privát IP címek külső elérését egy megadott porton keresztül
- Gyakorlatilag ez a *statikus* NAT alkalmazása

# SSH Tunnel

- A porttovábbítás egyik tipikus alkalmazása
- Windows (putty) beállítások
  - (Nyitni kell egy ssh kapcsolatot a caesar.elte.hu-ra)



[https://people.inf.elte.hu/nikovits/DB1/ssh\\_tunnel.pdf](https://people.inf.elte.hu/nikovits/DB1/ssh_tunnel.pdf)

# SSH Tunnel

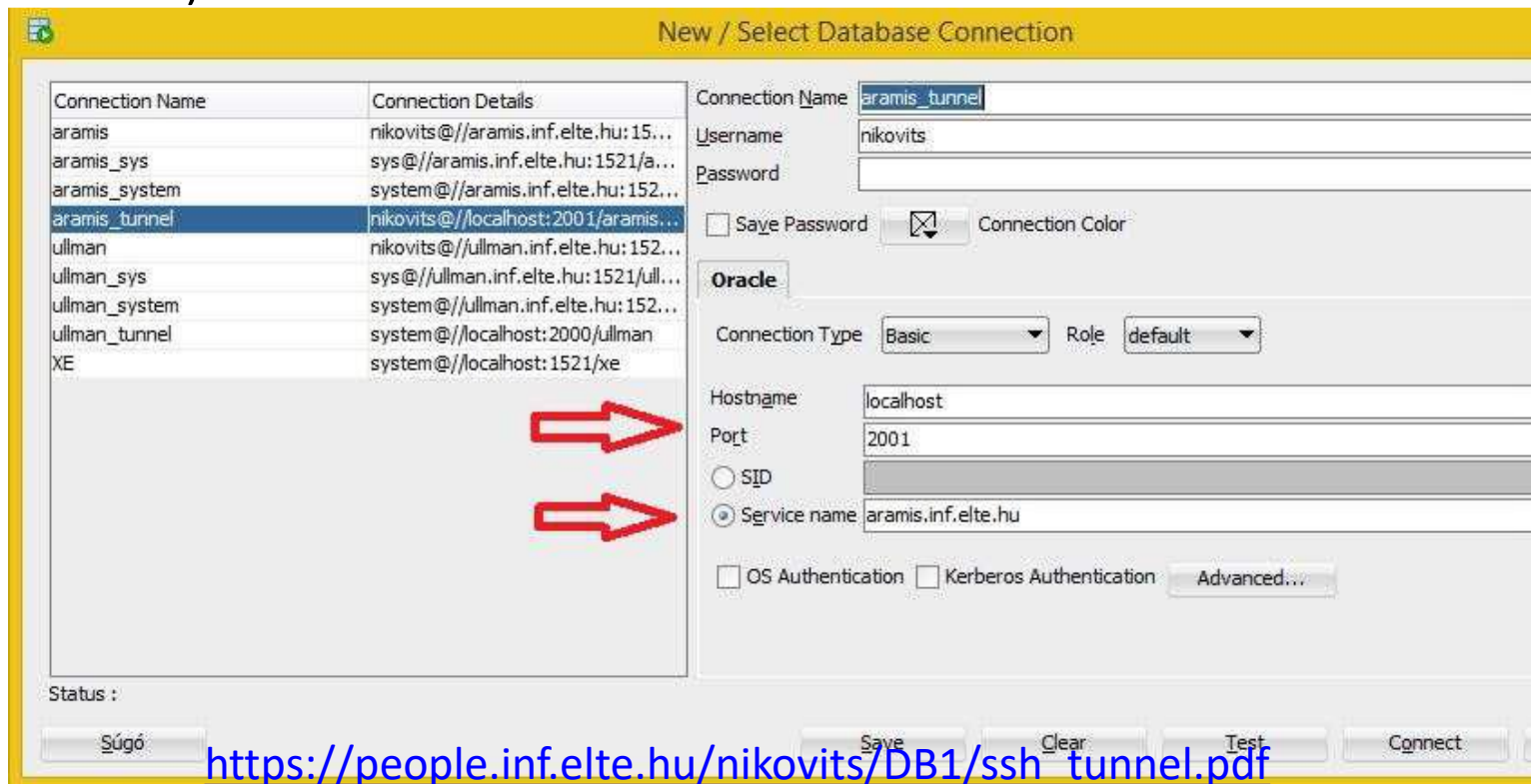
- Linux

```
ssh -L 2001:aramis.inf.elte.hu:1521 user@hostname
```

- `ssh -L <localport>:<remote host>:<remote port> <gateway you can ssh in>`
  - localport: a localhost ezen portján lesz elérhető a távoli szerver/szolgáltatás
  - remote host:remote port: ide csatlakozik a tunnel végpont, minden, amit a localportra küldünk ide fog továbbítódni és vissza. A gateway-ről elérhetőnek kell lennie!
  - gateway: a gép, amire be tudunk sshval lépni!

# SSH Tunnel

- Használat SqlDeveloper-nél:
  - (ssh kapcsolatnak fenn kell állnia végig az adatbázis kapcsolat ideje alatt)



**VÉGE**  
**KÖSZÖNÖM A FIGYELMET!**