Mérési utasítás — Az ifconfig és a ping parancsok

Az ifconfig parancs

Az ifconfig parancs a Linux hálózati paramétereinek beállítására szolgál. Amennyiben csak önmagában adjuk ki a parancsot, látható, hogy csak az éppen aktív hálózati interfészeket sorolja fel, a legfontosabb paraméterekkel.

```
root@feher4#:ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:ae:00:34
          inet addr:193.224.129.168  Bcast:193.224.129.175  Mask:255.255.255.240
          inet6 addr: 2001:738:2c01:8000:250:56ff:feae:34/64 Scope:Global
          inet6 addr: fe80::250:56ff:feae:34/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17571 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3135 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1863396 (1.7 MiB)  TX bytes:1104162 (1.0 MiB)

eth1      Link encap:Ethernet  HWaddr 00:50:56:ae:00:35
          inet addr:10.9.0.200  Bcast:10.9.0.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:feae:35/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13415 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7904 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1741065 (1.6 MiB)  TX bytes:3278014 (3.1 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:40232 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40232 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4792781 (4.5 MiB)  TX bytes:4792781 (4.5 MiB)
```

Amennyiben argumentumként megadunk egy interfész nevet, úgy csak a megadott interfészt fogja kilistázni.

```
root@feher4#:ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:ae:00:34
          inet addr:193.224.129.168  Bcast:193.224.129.175  Mask:255.255.255.240
          inet6 addr: 2001:738:2c01:8000:250:56ff:feae:34/64 Scope:Global
          inet6 addr: fe80::250:56ff:feae:34/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17571 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3135 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1863396 (1.7 MiB)  TX bytes:1104162 (1.0 MiB)
```

Az ifconfig parancsot használhatjuk beállításra is az alábbi szintaktikával:

```
ifconfig <interfész neve> <ip cím> netmask <netmaszk> up
```

Figyeljük meg az `up` kapcsolót: amennyiben az interfész **down** – azaz lekapcsolt – állapotban volt, úgy ezzel a kapcsolóval tudjuk aktívvá tenni. A Linux képes kiszámolni a megadott információk alapján a broadcast cím értékét, így azt nem kell megadnunk.

1. Feladat.

1. Állítsa be a fekete gép eth0 ás interfészének a következő címet:
   `192.168.100.<190+gépszám> netmaszk: 255.255.255.0`

   (`ifconfig eth0 192.168.100.194 netmask 255.255.255.0 up`)
2. Majd ellenőrizzük sikerült-e átállítani az IP címet! (ifconfig eth0)

A ping parancs

A ping parancs tipikusan hálózati elérhetőség tesztelésére használhatjuk. Segítségével ICMP echo üzenetet (lásd tankönyv) küldhetünk a kapcsolóként megadott hosztnak. Figyelem, a ping parancs kiadásakor a csomagokat addig küldjük a megadott hosztnak, amíg meg nem állítjuk. Ezt a <CTRL>+<C> billentyűkombinációval tehetjük meg.

```
root@cloud:~# ping 10.9.0.1
PING 10.9.0.1 (10.9.0.1) 56(84) bytes of data.
64 bytes from 10.9.0.1: icmp_req=1 ttl=64 time=1.79 ms
64 bytes from 10.9.0.1: icmp_req=2 ttl=64 time=0.292 ms
^C
--- 10.9.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.292/1.042/1.793/0.751 ms
```

Lehetőségünk van megadni pontosan hányszor „pingeljük” meg az adott hosztot, erre a `-c` kapcsolót használjuk.

```
root@cloud:~# ping -c 1 10.9.0.1
PING 10.9.0.1 (10.9.0.1) 56(84) bytes of data.
64 bytes from 10.9.0.1: icmp_req=1 ttl=64 time=0.354 ms

--- 10.9.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.354/0.354/0.354/0.000 ms
```

2. Feladat.

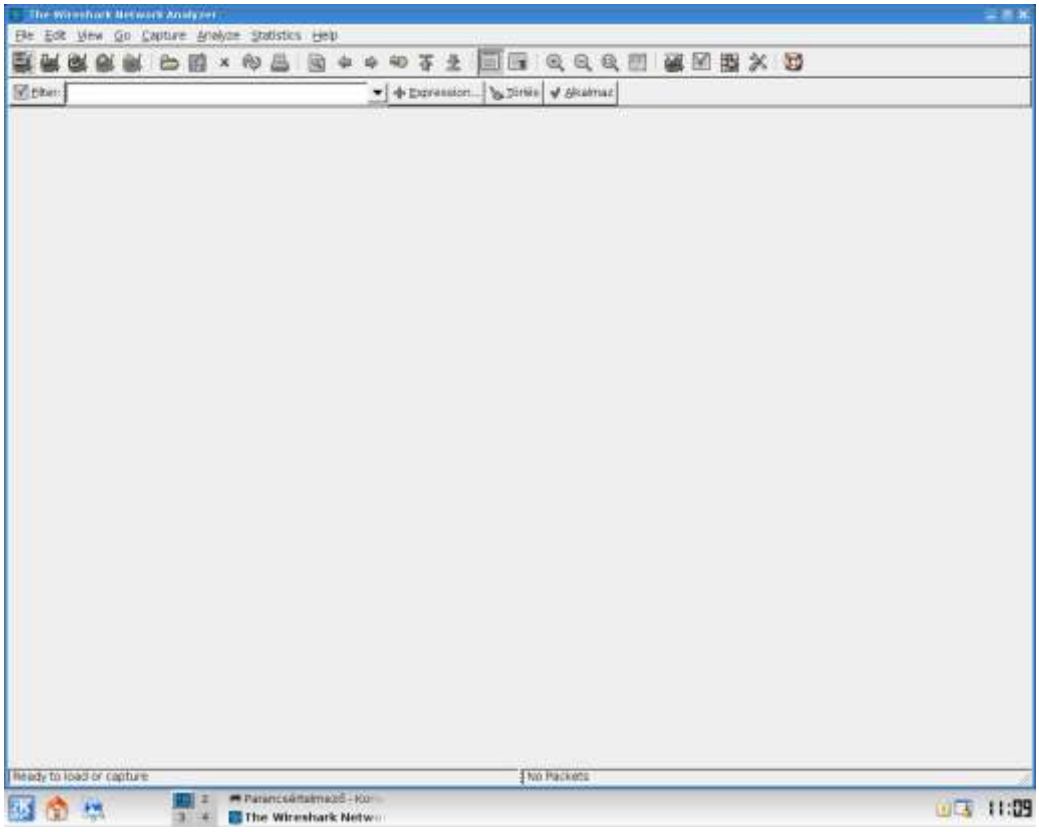
1. Ping parancs segítségével küldjön pontosan **10** ICMP ehco request üzenetet a 192.168.100.1-es IP címre.

SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

Mérési utasítás A WireShark használata: TCP kapcsolatok analizálása

A WireShark (korábbi nevén Ethereal) a legfejlettebb hálózati sniffer és analizátor program. 1998-óta fejlesztik, jelenleg a GPL 2 licensz alatt. Nem igen találni ilyen széleskörű szolgáltatásokkal és ismeretekkel rendelkező hálózati analizátor programot. Támogatott operációs rendszerek: Windows, Linux, OS X, Solaris, FreeBSD, NetBSD és még sok egyéb. Grafikus interaktív interfésszel rendelkezik. Az OSI ISO modell 2-7 rétegének minden implementációját tudja analizálni. A program által jelenleg ismert protokollok száma jelenleg több mint 81000!

A WireShark analizátor funkcióit több könyv, illetve elektronikus irodalom írja le több száz oldal terjedelemben, így gyakorlaton csak az alap funkciókkal ismerkedünk meg.



1. Feladat.

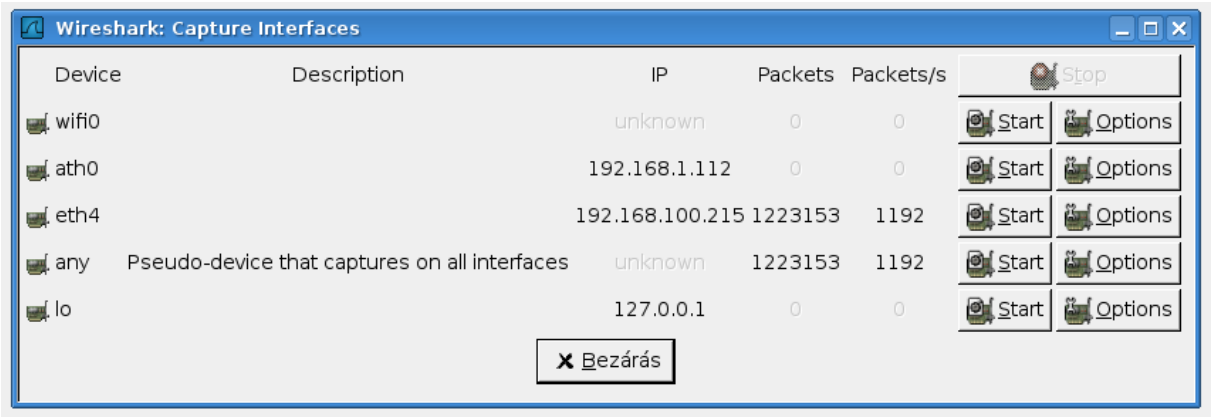
Amennyiben nincs telepítve a számítógépre, telepítse a WireShark-ot.

```
apt-get install WireShark
```

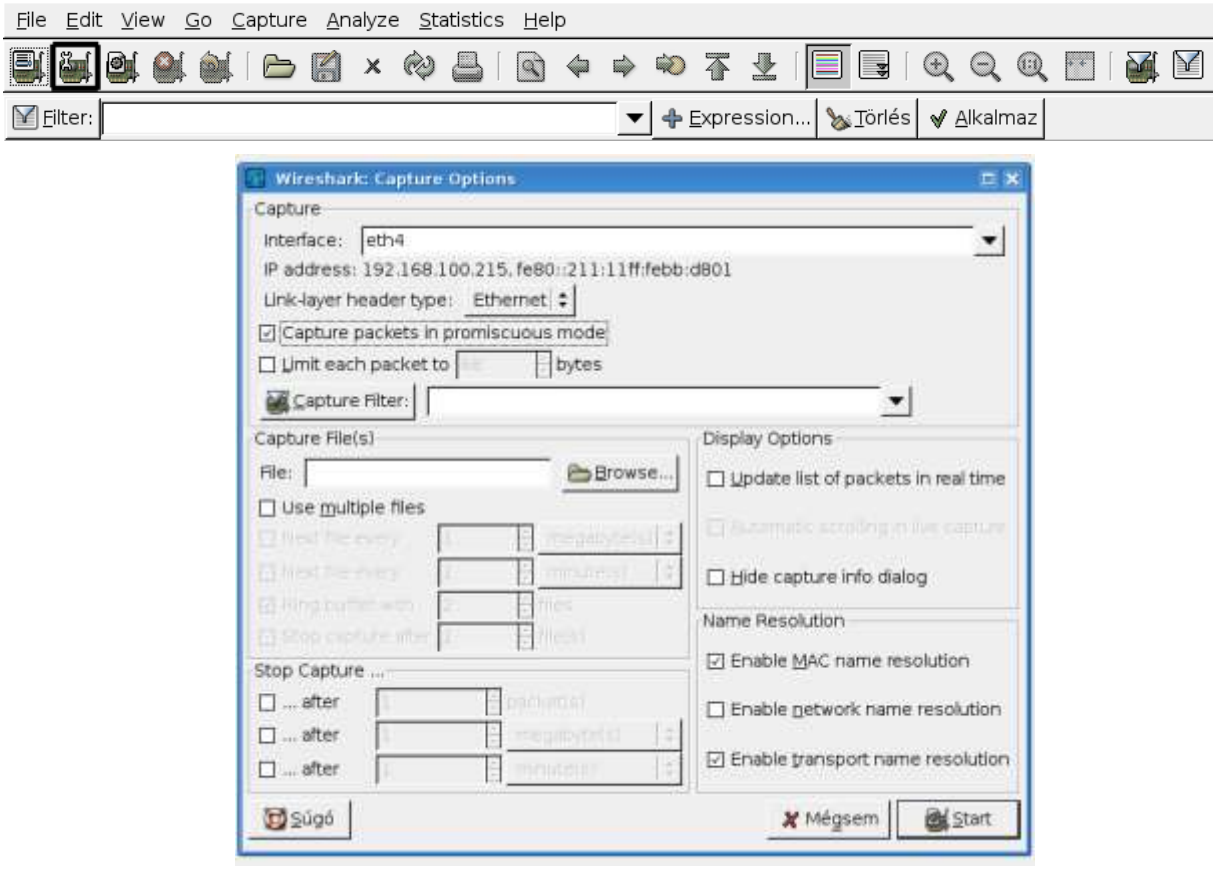
SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

Nézze át a WireShark kezelőfelületét!

Az első gombbal hívhatja elő a WireShark által elérhető és használható hálózati interfészeket.



Ezen az ábrán láthatóak a „sniffelhető” interfészek, IP címekkel, és az áthaladt csomagok számával. A második gombbal állíthatja be az analizálás tulajdonságait.



Legfelül látható, hogy jelen esetben az eth0-ás interfészt használjuk. A „Capture packet in promiscuous mode" kapcsolót mindig hagyja bekapcsolva, így ún. monitor módba állítja a hálókártyát. Be lehet állítani, hogy a WireShark fájlba mentse el az elkapott csomagokat. Megadhatja az analizálás leállásának feltételeit is, csomagszám, elkapott csomagok mérete és időkorlát alapján.

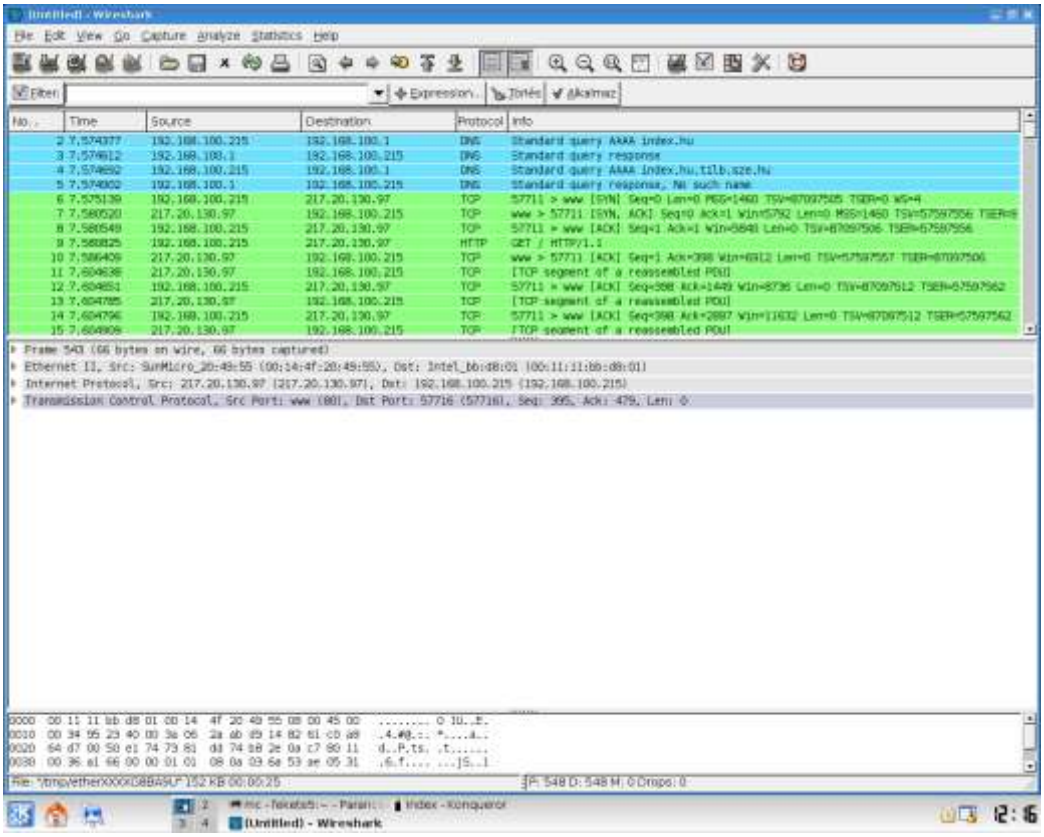
A Display options menüben lehet a csomagelkapás közbeni információkat beállítani. Automatikus „real-time" kijelzés, valamint ennek függvényében a képernyő görgetése, és az elkapott csomagok számának kijelzése.

Az utolsó részben va a névfeloldás lehetőségeinek beállítása, vagyis nem IP címeket kell ez esetben keresnie, hanem az ezekhez hozzárendelt szimbolikus neveket, valamint a MAC-ben az első 3 byte helyett a gyártó nevét.

A következő két gomb a csomag elkapás indítása, illetve leállítása.

2. Feladat.

Indítson egy csomagelkapást az eth0-án, úgy hogy a leállítás feltétele legyen 1 perc, valamint a képernyő automatikusan gördüljön a csomagokkal. (Nem kell menteni az előző listát, amennyiben a WireShark megkérdezné.) Ezután a böngészőt elindítva kérje le az *index.hu* honlapot.



A WireShark az elkapott csomagok sorszámát, a forrás és cél IP-t, a protokoll nevét, valamint a csomag részletét jeleníti meg első látásra. Alul látható, hogy a WireShark a különböző protokollokat sorrendbe helyezi. Először a csomag méretét adja meg, majd az Ethernet opciókat. Itt található a forrás és cél MAC cím. Lejjebb az IP protokoll adatai láthatók, mint forrás és cél IP. Majd végezetül a TCP tulajdonságokat nézhetjük meg. (Mint például a forrás és cél port, valamint a különböző TCP bitek értékét (SYN, ACK, FIN, stb.).)

Jól megfigyelhető a képen, hogy először a mi gépünk lekéri a DNS bejegyzést a névkiszolgálótól, majd megkezdi IP cím alapján az index.hu kezdőlapját letölteni.

A hálózatokon sokszor rengeteg „szemét” csomag kering, mint például feszítőfa, illetve más egyéb routing protokoll. Ha ezeket figyelmen kívül szeretnénk hagyni, a csomagszűrőkhöz kell nyúlnunk.

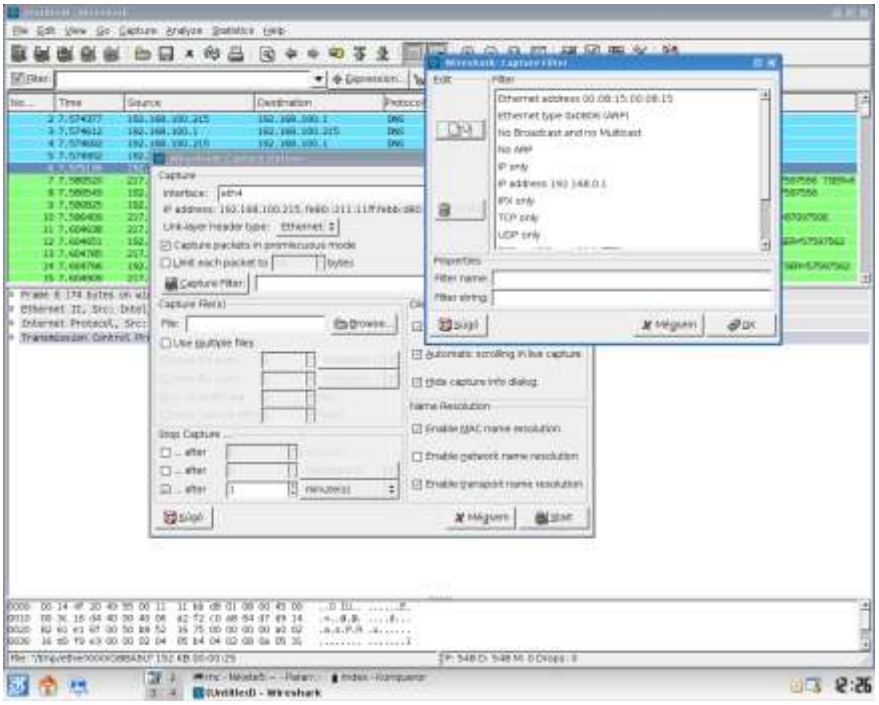
Csomagszűrők két helyen alkalmazhatók:

1. csomagelkapásnál
2. megjelenítésnél

Ha csomagelkapásnál használunk szűrőt, akkor csak a szűrési feltételeknek megfelelő csomagokat fogja a WireShark eltárolni. Az eltárolt csomagok közül megjelenítési szűrővel választhatjuk ki, hogy melyek jelenjenek meg a képernyőn. (A két szűrő szintaxisa különböző!)

A csomagelkapási beállításokon (2. gomb) belül lehet csomagszűrőket alkalmazni.

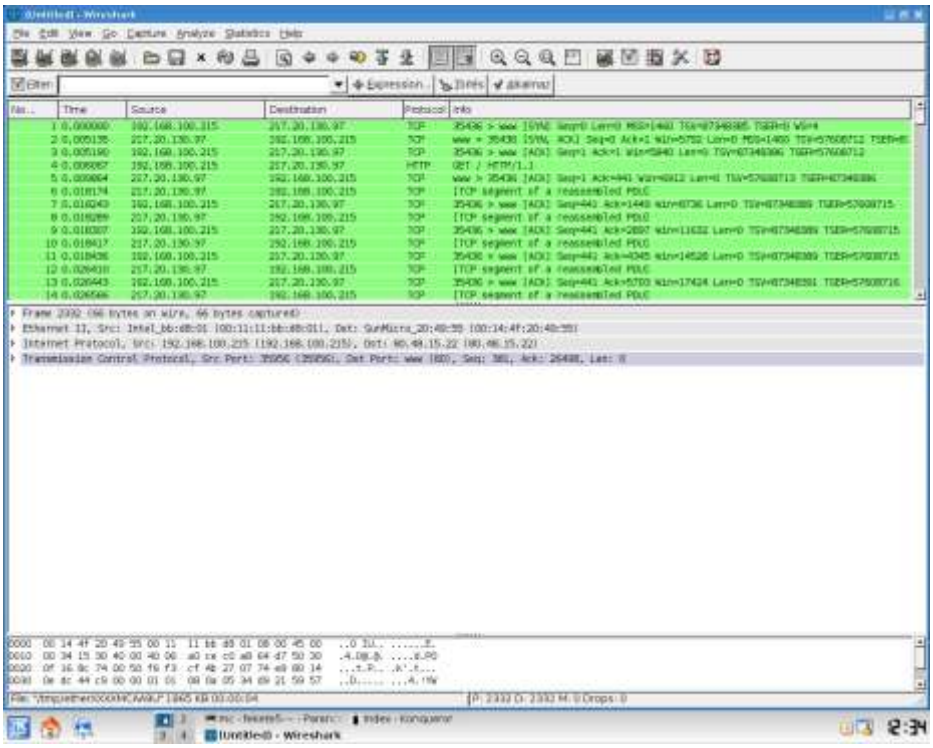
A csomagszűrési beállításokon belül több előre definiált szűrő áll rendelkezésünkre.



Meg lehet adni protokollszűrést, IP cím-szűrést, forrás és célport szűrést.

3. Feladat.

Hajtsa végre az előző feladatot, úgy hogy most filterként beállítva, csak a 80-as portot érintő kommunikációt vizsgálja. (*Capture Filter port 80*).

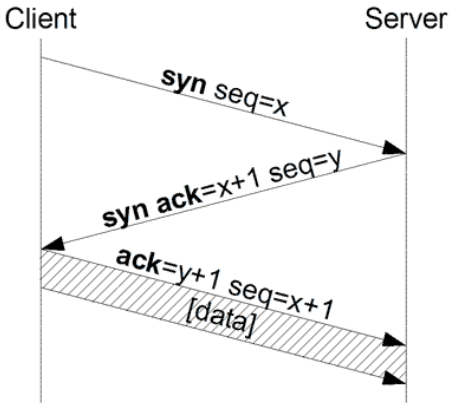


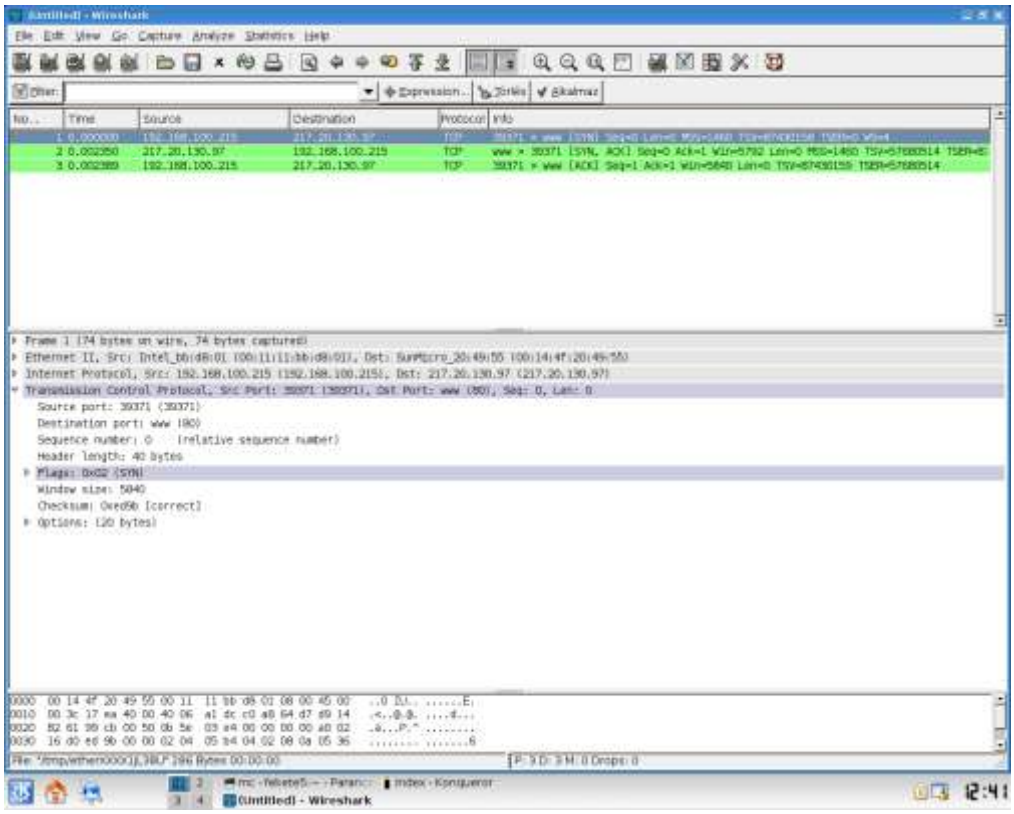
Most csak a 80-as portot érintő kommunikációt jelenítjük meg.

4. Feladat.

Hajtsa végre az előző feladatot úgy, hogy a csomagelkapás leállításának feltétele 3 csomag elkapása legyen. Ezzel az előző feladatból csak a „three-way-handshake”, vagyis a 3 utas kézfogást kapjuk meg.

Ez a TCP protokoll kapcsolat-felépítési fázisa.





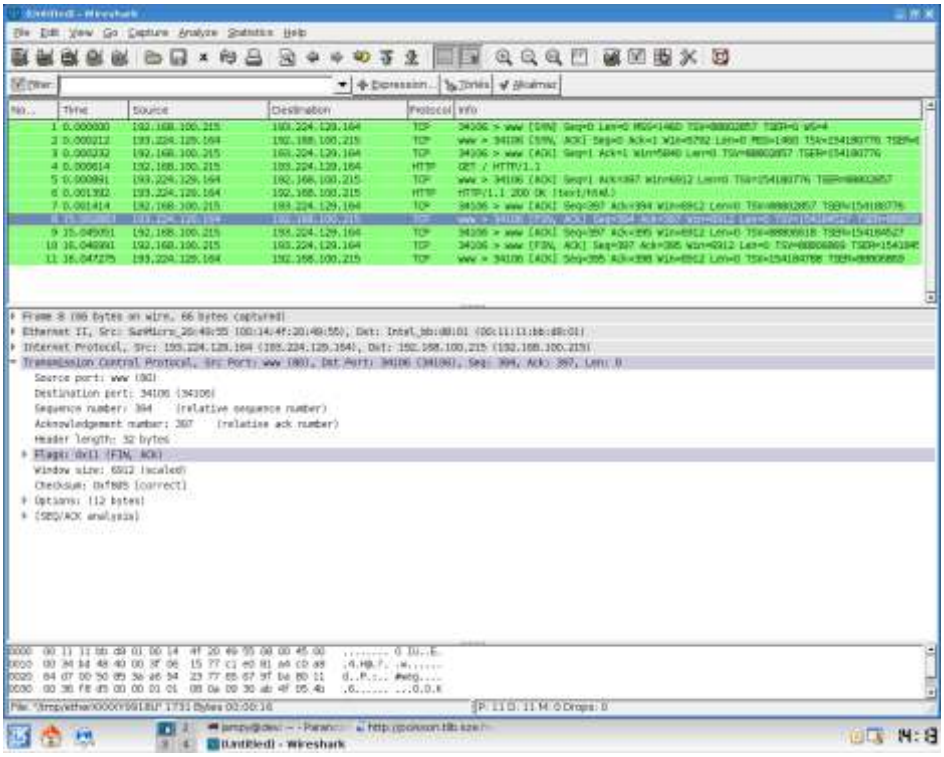
A csomagokat „kibontva” látható, hogy a 3 utas kézfogás egy TCP SYN-nel kezdődik, sequence number=0-val, majd a szerver visszaküldi a TCP SYN,ACK-t , egy sequence number=0 és Acknowledge number=1-el, amelyre a válasz egy TCP ACK, ahol mind a sequence number mind az acknowledge number 1-re van állítva.

Természetesen ezek csak jelen helyzetben ilyen értékűek a könnyebb megértés érdekében.

Ezzel létrejött a TCP kapcsolat.

5. Feladat.

Hajtsa végre az előző feladatot úgy, hogy vegye ki a csomagelkapás leállítási feltételt, és most – az egyszerűség kedvéért – a http://dev.tilb.sze.hu lapot kérje le!



Itt az utolsó négy csomagban megfigyelhető a 4 utas kézfogás, mely a TCP kapcsolat lebontását jelenti. Először a szerver küld egy FIN bitet, amelyre ACK a válasz, majd megy egy FIN, amelyre a szerver válaszol ACK-val.