Mérési utasítás **ifconfig, ping, WireShark használata**

Az ifconfig parancs

Az **ifconfig** parancs a Linux Ethernet interfészek (hálózati vezérlők) hálózati paramétereinek beállítására, és az aktuális beállítások és állapotok kiíratására szolgál. Paraméterek nélkül használva, csak az éppen aktív hálózati interfészeket sorolja fel, a legfontosabb paraméterekkel.

Például:

```
root@fekete7:~# /sbin/ifconfig
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.117  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::e2d5:5eff:febd:aca  prefixlen 64  scopeid 0x20<link>
        ether e0:d5:5e:bd:0a:ca  txqueuelen 1000  (Ethernet)
        RX packets 20661  bytes 3820748 (3.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10385  bytes 737943 (720.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18  bytes 1976 (1.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 1976 (1.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vmnet1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.65.1  netmask 255.255.255.0  broadcast 192.168.65.255
        inet6 fe80::250:56ff:fec0:1  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:c0:00:01  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 39  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vmnet8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.138.1  netmask 255.255.255.0  broadcast 172.16.138.255
        inet6 fe80::250:56ff:fec0:8  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:c0:00:08  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 38  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Ha a parancs argumentuma egy interfész, akkor csak a megadott interfészt beállításait jeleníti meg.

Például:

```
root@fekete7:~# /sbin/ifconfig enp3s0
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.117  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::e2d5:5eff:febd:aca  prefixlen 64  scopeid 0x20<link>
        ether e0:d5:5e:bd:0a:ca  txqueuelen 1000  (Ethernet)
        RX packets 20825  bytes 3831985 (3.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10413  bytes 739966 (722.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Fontos még a **-a** paraméter is. Segítségével az összes interfész (beleértve a nem aktívakat is) beállításai kiírathatók.

Az **ifconfig** parancs segítségével lehet beállítani is az egyes interfészeket a következő szintaxis alkalmazásával:

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

Amennyiben egy interfész **down**, azaz *lekapcsolt* állapotban van, úgy az **up** kapcsolóval lehet azt aktívvá tenni. A Linux képes kiszámolni a megadott információk alapján a broadcast cím értékét, így azt nem kell megadnunk. Fontos megjegyezni, hogy az **ifconfig** parancs segítségével elvégzett beállítások csak a legközelebbi újraindításig maradnak meg.

Feladatok

1. Indítsa el a terminál (konzole) programot, mely ekvivalens azzal a terminállal, amiben eddig dolgozott.
   K menü -> Alkalmazások -> Rendszer -> Terminál
2. Írassa az összes interfészt, és állapotukat! Hány interfészt lát?
   ```
   ifconfig -a
   ```
3. Állítsa be a fekete gép **enp3s0** (*vagy ahogyan hívják az elsődleges hálózati interfészt!)* interfészének a következő IP-címet: 192.168.100.190+gépszám. Majd adja meg az alapértelmezett hálózati átjárót és a DNS kiszolgálót.
   ```
   ifconfig enp3s0 down
   ifconfig enp3s0 192.168.100.{190+gépszám} netmask 255.255.255.0 up
   route add default gw 192.168.100.1
   echo "nameserver 192.168.100.2" > /etc/resolv.conf
   ```
4. Ellenőrizze, hogy sikerült-e beállítani az IP-címet
   ```
   ifconfig enp3s0
   ```

SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

A ping parancs

A **ping** parancs a hálózati kapcsolat tesztelésére szolgál. Segítségével ICMP echo üzenetet (lásd tankönyv) lehet küldeni a paraméterben megadott hosztnak. Linux esetében a ping parancs a csomagokat addig küldi a megadott hosztnak, míg leállításra nem kerül. Ezt a **<CTRL>+<C>** billentyűkombinációval lehet megtenni.

Például:

```
root@fekete7:~# ping 193.224.128.1
PING 193.224.128.1 (193.224.128.1) 56(84) bytes of data.
64 bytes from 193.224.128.1: icmp_seq=1 ttl=63 time=0.899 ms
64 bytes from 193.224.128.1: icmp_seq=2 ttl=63 time=0.808 ms
64 bytes from 193.224.128.1: icmp_seq=3 ttl=63 time=0.805 ms
64 bytes from 193.224.128.1: icmp_seq=4 ttl=63 time=0.801 ms
64 bytes from 193.224.128.1: icmp_seq=5 ttl=63 time=0.819 ms
^C
--- 193.224.128.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4071ms
rtt min/avg/max/mdev = 0.801/0.826/0.899/0.036 ms
root@fekete7:~# ping -c 1 193.224.128.1
PING 193.224.128.1 (193.224.128.1) 56(84) bytes of data.
64 bytes from 193.224.128.1: icmp_seq=1 ttl=63 time=0.828 ms

--- 193.224.128.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.828/0.828/0.828/0.000 ms
root@fekete7:~#
```

Lehetőség van megadni, hogy a parancs pontosan hány ICMP echo request üzenetet küldjön a célállomás felé. (Hányszor „pingeljük” meg az adott hosztot.) Ezt a **-c** kapcsoló segítségével lehet megtenni.

Például:

```
root@fekete7:~# ping -c 1 193.224.128.1
PING 193.224.128.1 (193.224.128.1) 56(84) bytes of data.
64 bytes from 193.224.128.1: icmp_seq=1 ttl=63 time=0.828 ms

--- 193.224.128.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.828/0.828/0.828/0.000 ms
```

Feladatok

1. Kérdezze le a ping parancs paramétereit, majd tanulmányozza azokat!
   (ping)

2. A ping parancs segítségével küldjön pontosan 10 ICMP ehco request üzenetet a 192.168.100.1-es IP-címre. Tanulmányozza az eredményeket! Milyen átlagos és maximális válaszidőket tapasztalt?
   `(ping -c 10 192.168.100.1)`

WireShark program alkalmazása

A WireShark (korábbi nevén Ethereal) egy rendkívül fejlett hálózati sniffer és analizátor program. Fejlesztése 1998-óta folyik, jelenleg 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 több mint 81000, de bárki készíthet hozzá újabbakat.

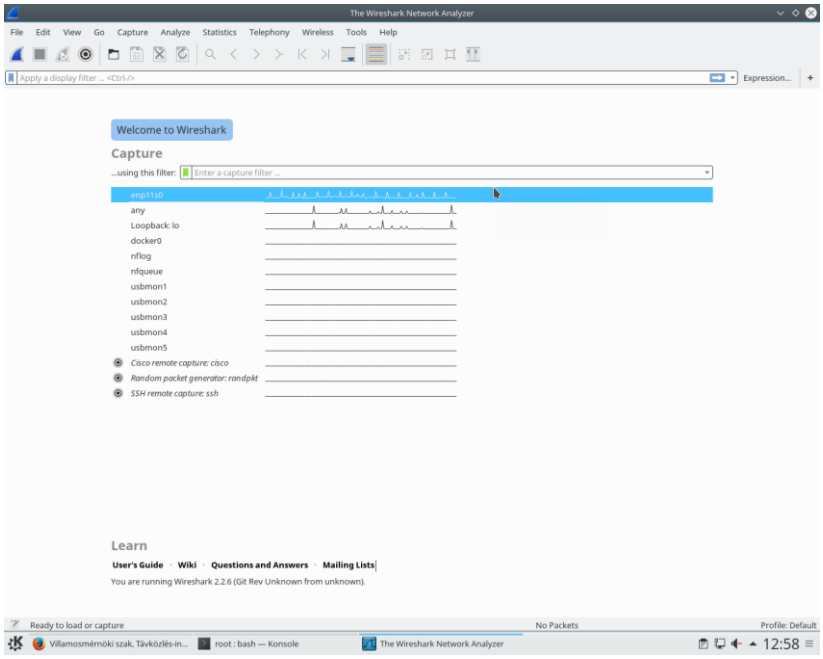
A WireShark analizátor funkcióit több könyv, illetve elektronikus irodalom írja le több száz oldal terjedelemben, így a gyakorlat keretében csak az alap funkciókkal ismerkedünk meg.

Feladatok

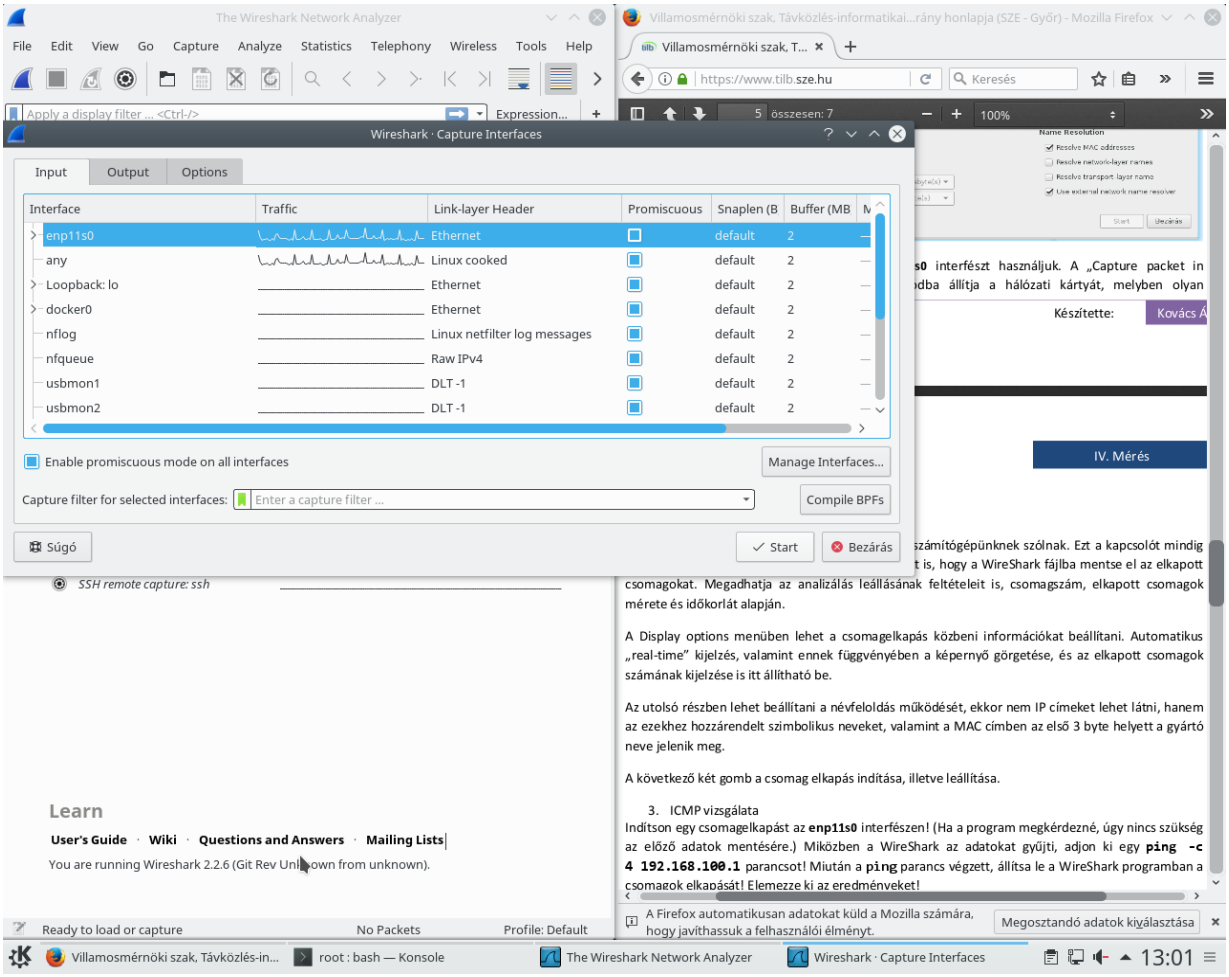
1. Amennyiben nincs telepítve a számítógépre, telepítse a WireSharkot az `apt install wireshark` parancs kiadásával!

2. Indítsa el a programot, majd nézze át annak kezelőfelületét!

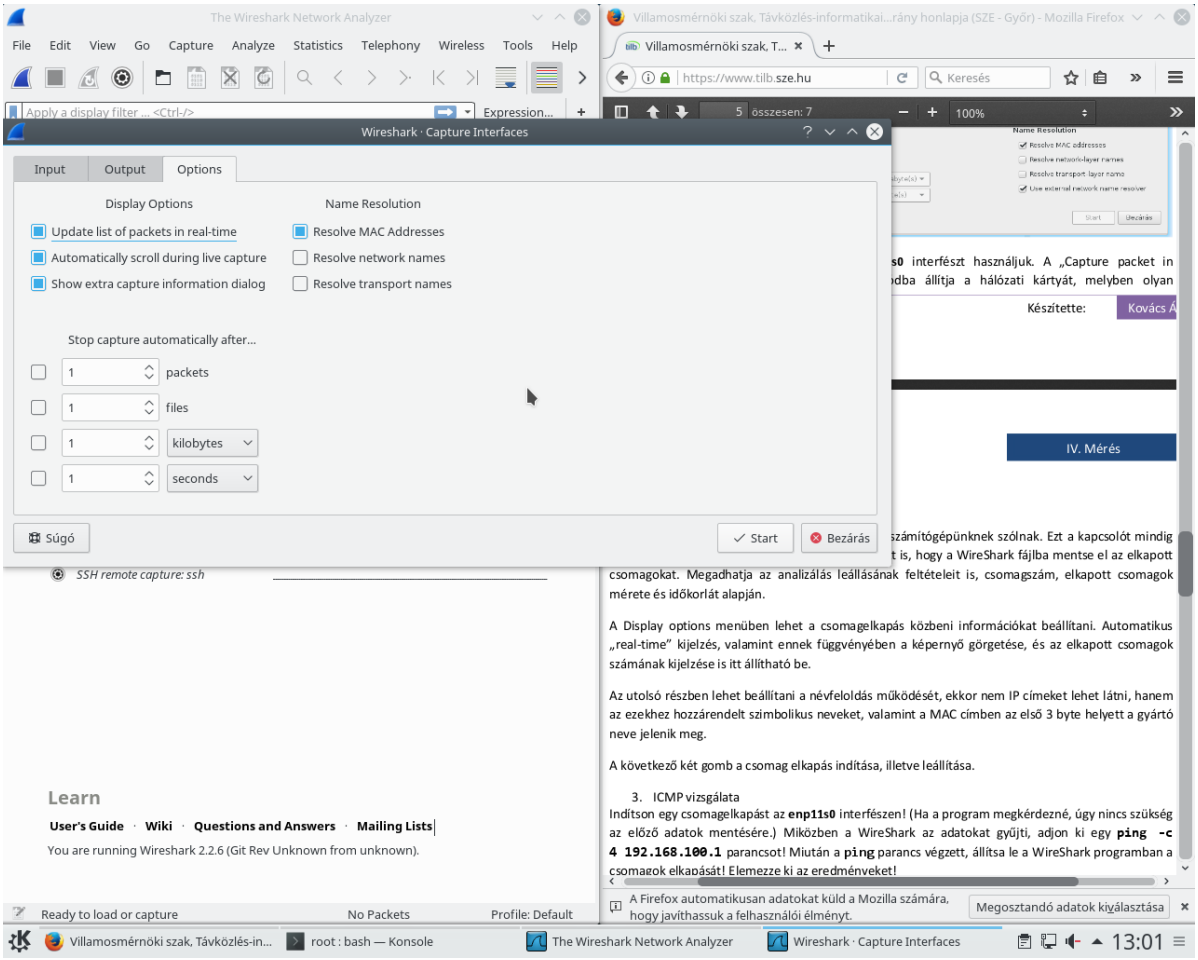
A wireshark indításnál automatikusan láthatók a „sniffelhető” interfészek, IP címeikkel, használati grafikonnal.



A negyedik gombbal állíthatja be az analizálás tulajdonságait.



Legfelül látható, hogy jelen esetben az **enp11s0** interfészt használjuk. A „Capture packet in promiscuous mode” kapcsoló ún. monitor módba állítja a hálózati kártyát, melyben olyan csomagokat is el tudunk kapni, melyek nem a mi számítógépünknek szólnak. Ezt a kapcsolót mindig kapcsoljuk be a mérések során. Az options menüpont alatt be lehet állítani azt is, 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 is itt állítható be.

Az utolsó részben lehet beállítani a névfeloldás működését, ekkor nem IP címeket lehet látni, hanem az ezekhez hozzárendelt szimbolikus neveket, valamint a MAC címben az első 3 byte helyett a gyártó neve jelenik meg.

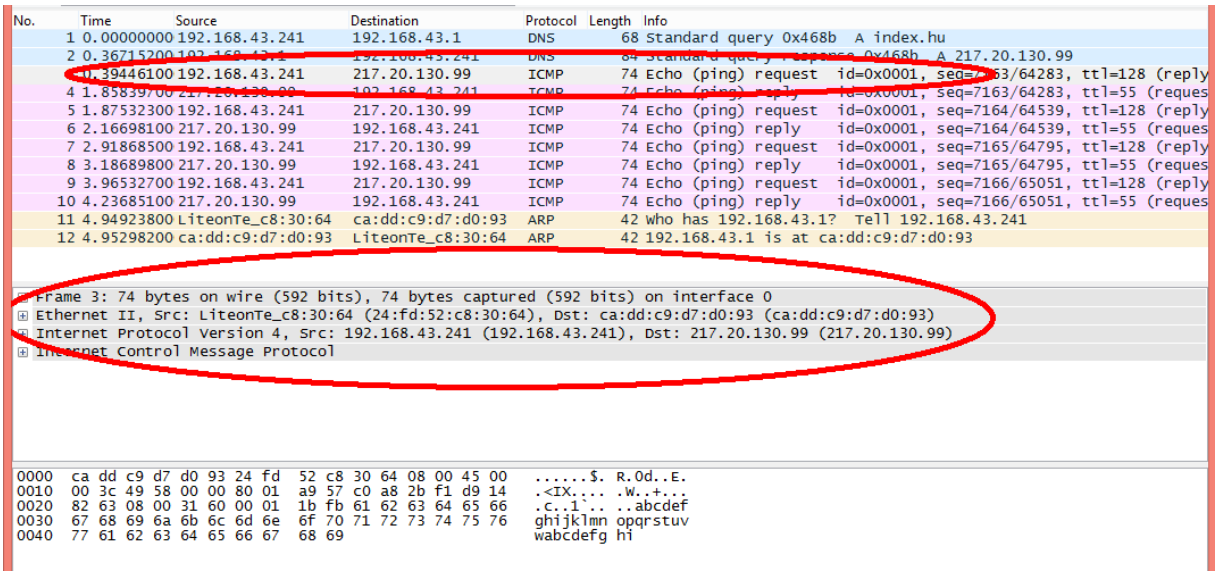
A legelső 2 gomb pedig a csomagelkapás elindítása illetve leállítása.

3. ICMP vizsgálata

Indítson egy csomagelkapást az **enp3s0** (*vagy ahogyan hívják az elsődleges hálózati interfészt!)* interfészen! (Ha a program megkérdezné, úgy nincs szükség az előző adatok mentésére.) Miközben a WireShark az adatokat gyűjti, adjon ki egy **`ping -c 4 index.hu`** parancsot! Miután a `ping` parancs végzett, állítsa le a WireShark programban a csomagok elkapását! Elemezze ki az eredményeket!

A WireShark az elkapott keretek sorszámát, a forrás és cél IP címet, a protokoll nevét valamint a csomag részletét jeleníti meg első látásra. A program felső ablakában keresse meg az első *ICMP Echo request* csomagot, és válassza ki azt! Ekkor a középső ablakban megjelenik a kiválasztott csomaghoz

tartozó keret, ahol részletesebben is elemezheti azt. Vegye észre, hogy az egyes hálózati rétegek jól megfigyelhetőek ebben a középső ablakban! A legelső sor a keret paramétereit tartalmazza, majd alatta az Ethernet keret, az arra épülő IP, legvégül pedig az ICMP.



A sorok elején lévő + jel segítségével az egyes szinteket is alaposabban kielemezheti! (Miközben a program középső részében kijelöl valamit, az a program alsó részében is kiemelten jelenik meg.)

Milyen hosszú egy `ping` parancshoz tartozó keret?

Keresse meg, a vizsgált csomag feladójának fizikai címét! Melyik szinten fogja ezt keresni? Milyen címzési módot talált?

Keresse meg, hogy mi adja meg, hogy IP datagram van a vizsgált Frame-ben! Milyen értéket talált? Hány byte ez az érték?

Keresse meg, hogy milyen forrás IP címről, milyen IP célcímre ment az IP datagram. Melyik szinten fogja ezt keresni? Hány byte hosszú egy cím? Milyen TTL értéket talált?

Keresse meg, hogy mi adja meg, hogy ICMP protokoll van a vizsgált IP datagramban! Milyen értéket talált? Hány byte ez az érték?

Keresse meg, hogy mi adja meg, hogy az ICMP-ben Echo Request van! Milyen értéket talált? Hány byte ez az érték? Melyik szinten fogja ezt keresni?

Hány byte a payload az elfogott ICMP kérésben? Milyen értékűek ezek a byte-ok? Hol találta meg ezt a payloadot?

Most hasonlóan vizsgálja meg az ICMP Echo Replay-t is!

4. UDP vizsgálata

Vizsgálja meg az előző feladatban végrehajtott csomagelkapásban még az ICMP üzenetek előtt elkapott DNS Standard Query üzenetet! Milyen forrás fizikai címről ment a kérés? Milyen forrás IP címről, milyen cél IP címre ment a kérés?

Milyen szállítási szintű protokollt használt az DNS? Milyen érték azonosítja az UDP-t?

Hány byte egy UDP checksum? Hol találta meg?

Milyen forrásportról ment a kérés milyen célportra? Melyik szinten, hol találta meg az értékeket? Mi a DNS kérések portszáma?

Valahol megtalálta a kért nevet? (index.hu) Hol? Melyik szinten?

5. TCP vizsgálata és szűrők

Indítson egy csomagelkapást az **enp3s0** (*vagy ahogyan hívják az elsődleges hálózati interfészt!)* interfészen úgy, hogy a leállítás feltétele legyen 1 perc, valamint a képernyő automatikusan gördüljön a csomagokkal. (Nem kell menteni az előző listát.) Ezután a böngészőt elindítva kérje le az *index.hu* honlapot.

A TCP protokollt a következő gyakorlaton fogjuk részletesen megismerni, most csak az első TCP szegmens tartalmát vizsgáljuk meg.

Az TCP adategységet szállító IP datagram mely mezőjének milyen értékéből derül ki, hogy a TCP protokoll adategysége utazik fölötte?

A TCP protokoll fejrészében keresse meg a cél port számát!

A hálózatokon sokszor rengeteg „minket nem érdeklő” forgalom van. Ha ezeket figyelmen kívül szeretnénk hagyni, a csomagszűrőkhöz kell nyúlnunk.

Csomagszűrők két helyen alkalmazhatók:

1. csomagelkapásnál
2. megjelenítésnél

Ha csomagelkapásánál 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 fajta szűrő szintaxisa sajnos 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. Bővebben majd a következő gyakorlaton foglalkozunk vele.)

SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

Most csak a megjelenítésnél használható szűrőket (Display Filter) ismerjük meg. A korábban megismert gombsor alatt a „Filter:” mezőben adjuk meg a következőt: **tcp.port == 80**

Vegyük észre, hogy gépelés közben a mező háttere zöldre vált, amikor az addig begépelt szöveg szintaktikailag helyes!

Érvényesítsük a szűrőt az Apply feliratra való kattintással.

Mit tapasztalunk?