Hálózatok építése és üzemeltetése

Szoftver szerszámok

Feladat 0: man ping

a **ping** program segítségével állapítsuk meg, hogy

- hány hopra van a www.bme.hu (és a www.u-szeged.hu)
- a man page alapján lehet-e következtetni a válaszra a “**ping hostname**” parancsból.

Feladat 1: Traceroute

- Milyen routereken keresztül érhető el a www.elte.hu és a www.u-szeged.hu?
- Hány közös hop van a két útban?

Feladat 0: man ping

- a **ping** program segítségével állapítsuk meg, hogy
  - hány hopra van a www.bme.hu,

```
for i in {1..255}; do
  ping -c 1 -t $i www.bme.hu > /dev/null
  if [ $? != 1 ]; then
    echo $i hop; exit
  fi
done
echo a gep nem elerheto
```

  - a man page alapján lehet-e következtetni a válaszra a "**ping hostname**" parancsból.

Nem, a TTL DETAILS leírja, hogy a célgép tetszőleges értékre állíthatja a válaszában a TTL értéket.

Feladat 1: traceroute

- Milyen routereken keresztül érhető el a www.elte.hu és a www.u-szeged.hu?

```
# traceroute www.elte.hu
traceroute to www.elte.hu (157.181.152.1), 30 hops max, 60 byte packets
 1  244gw.tmit.bme.hu (152.66.244.254)  0.199 ms  0.194 ms  0.189 ms
 2  vl121.ixion.net.bme.hu (152.66.245.254)  0.739 ms  0.828 ms  0.963 ms
 3  xge4-2.styx.net.bme.hu (152.66.0.72)  1.330 ms  1.580 ms  1.754 ms
 4  xge2-2.taz.net.bme.hu (152.66.0.78)  1.095 ms  1.196 ms  1.195 ms
 5  tg0-1-0-1.rtr.bme.hbone.hu (152.66.0.126)  1.186 ms  1.183 ms  1.285 ms
 6  tg0-1-0-2.rtr.elte.hbone.hu (195.111.106.66)  1.541 ms  1.730 ms  1.725 ms
 7  taurus.hbone-elte.elte.hu (157.181.141.13)  0.807 ms  0.796 ms  0.592 ms
 8  fw1.fw.backbone.elte.hu (192.153.18.145)  1.158 ms  1.151 ms  0.890 ms
 9  taurus.firewall.elte.hu (157.181.141.146)  1.329 ms  1.317 ms  1.589 ms
10  leo.taurus-leo.elte.hu (157.181.126.46)  1.318 ms  1.311 ms  1.515 ms
11  crux.crux-leo.elte.hu (157.181.126.9)  2.514 ms  2.503 ms  2.992 ms
12  www2.elte.hu (157.181.152.1)  1.139 ms  1.367 ms  1.360 ms
```

```
# traceroute www.u-szeged.hu
traceroute to www.u-szeged.hu (160.114.8.5), 30 hops max, 60 byte packets
 1  244gw.tmit.bme.hu (152.66.244.254)  0.190 ms  0.187 ms  0.183 ms
 2  vl121.ixion.net.bme.hu (152.66.245.254)  0.822 ms  0.916 ms  1.046 ms
 3  xge4-2.styx.net.bme.hu (152.66.0.72)  7.678 ms  7.743 ms  7.844 ms
 4  xge2-2.taz.net.bme.hu (152.66.0.78)  1.027 ms  1.154 ms  1.151 ms
 5  tg0-1-0-1.rtr.bme.hbone.hu (152.66.0.126)  1.471 ms  1.470 ms  1.466 ms
 6  tg0-0-0-6.rtr1.vh.hbone.hu (195.111.100.43)  5.227 ms  5.028 ms  5.111 ms
 7  be3.rtr.szeged.hbone.hu (195.111.111.191)  5.017 ms  4.919 ms  4.949 ms
 8  c7-c29.net.u-szeged.hu (160.114.120.78)  4.648 ms  4.855 ms  4.930 ms
 9  vb-kul.fw1.u-szeged.hu (160.114.120.177)  5.186 ms  5.346 ms  5.165 ms
10  160.114.120.54 (160.114.120.54)  5.160 ms  5.061 ms  5.236 ms
11  * * *
12  * * *
```

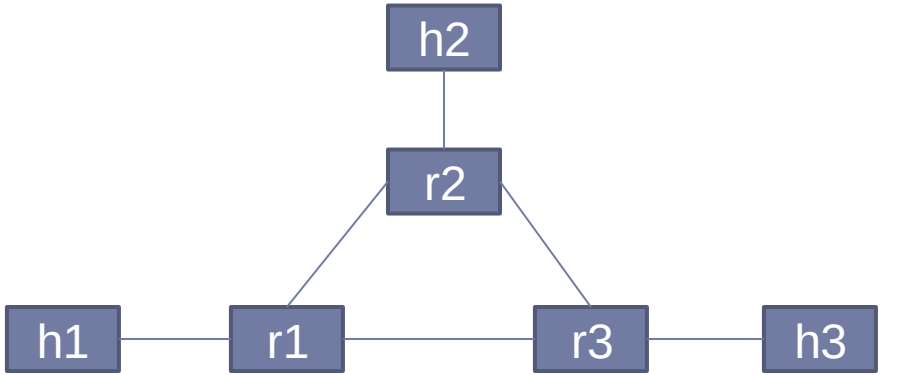
  - A gépek vagy nem generálnak ICMP time exceeded üzenetet vagy valamelyik router kiszűri ezeket. De a pingre válaszol a cél, így ping ttl-el megállapítható, hogy milyen messze van.

- Hány közös hop van a két útban?

5

Teszthálózat (mininetben)

- `$ wget http://hsn.tmit.bme.hu/topo.py`
- `$ sudo -E mn --custom ~/topo.py --topo mytopo --link tc`
- `mininet> xterm h1 h2 h3 r1 r2 r3`

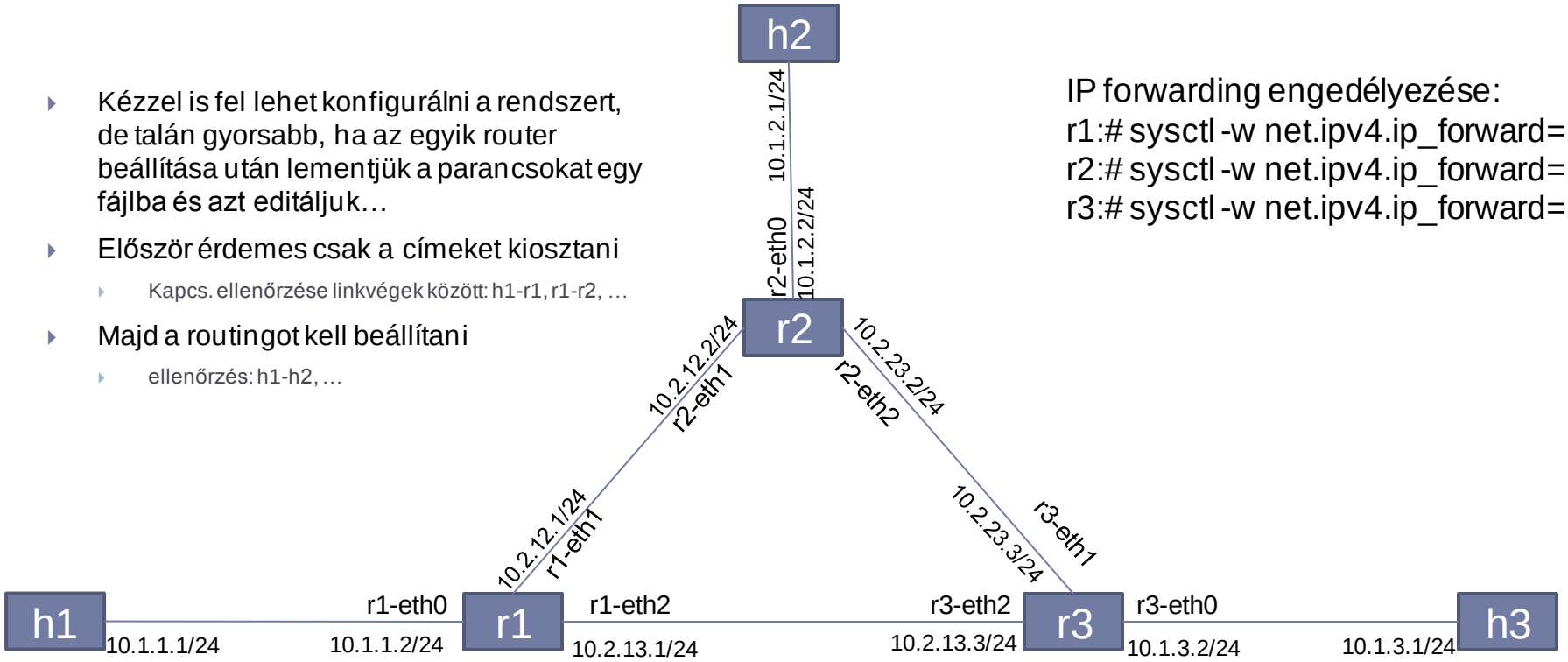


- `h1$ date > test.txt`
- `h2$ cat test.txt`

Feladat 2: interfészek és útvonalválaszás beállítása

- Kézzel is fel lehet konfigurálni a rendszert, de talán gyorsabb, ha az egyik router beállítása után lementjük a parancsokat egy fájlba és azt editáljuk…
- Először érdemes csak a címeket kiosztani
  - Kapcs. ellenőrzése linkvégek között: h1-r1, r1-r2, …
- Majd a routingot kell beállítani
  - ellenőrzés: h1-h2, …

IP forwarding engedélyezése:
r1:# sysctl -w net.ipv4.ip_forward=1
r2:# sysctl -w net.ipv4.ip_forward=1
r3:# sysctl -w net.ipv4.ip_forward=1



Névfeloldás

- A libnss a DNS-en kívül mást is haszálnat névfeloldásra
- Pl. az /etc/hosts fájlt is figyelembe veszi (man hosts):

```
127.0.0.1       localhost
127.0.1.1       p2p-VirtualBox

10.1.1.1 h1
10.1.2.1 h2
10.1.3.1 h3
```

Feladat 2:
interfészek és útvonalválaszás beállítása

```
#!/bin/bash
case $1 in
  h1)
    ip addr del 10.0.0.1/8  dev h1-eth0
    ip addr add 10.1.1.1/24 dev h1-eth0
    ip route add default via 10.1.1.2
    ;;
  h2)
    ip addr del 10.0.0.2/8  dev h2-eth0
    ip addr add 10.1.2.1/24 dev h2-eth0
    ip route add default via 10.1.2.2
    ;;
  h3)
    ip addr del 10.0.0.3/8  dev h3-eth0
    ip addr add 10.1.3.1/24 dev h3-eth0
    ip route add default via 10.1.3.2
    ;;
```

```
  r1)
    sysctl -w net.ipv4.ip_forward=1
    ip addr del 10.0.0.4/8  dev r1-eth0
    ip addr add 10.1.1.2/24 dev r1-eth0
    ip addr add 10.2.12.1/24 dev r1-eth1
    ip addr add 10.2.13.1/24 dev r1-eth2
    ip route add 10.1.2.0/24 via 10.2.12.2
    ip route add 10.1.3.0/24 via 10.2.13.3
    ;;
  r2)
    sysctl -w net.ipv4.ip_forward=1
    ip addr del 10.0.0.5/8  dev r2-eth0
    ip addr add 10.1.2.2/24 dev r2-eth0
    ip addr add 10.2.12.2/24 dev r2-eth1
    ip addr add 10.2.23.2/24 dev r2-eth2
    ip route add 10.1.1.0/24 via 10.2.12.1
    ip route add 10.1.3.0/24 via 10.2.23.3
    ;;
```

```
  r3)
    sysctl -w net.ipv4.ip_forward=1
    ip addr del 10.0.0.6/8  dev r3-eth0
    ip addr add 10.1.3.2/24 dev r3-eth0
    ip addr add 10.2.23.3/24 dev r3-eth1
    ip addr add 10.2.13.3/24 dev r3-eth2
    ip route add 10.1.1.0/24 via 10.2.13.1
    ip route add 10.1.2.0/24 via 10.2.23.2
    ;;
  *)
    echo usage $0 hostname >&2
    ;;
esac
```

wget http://hsn.tmit.bme.hu/c.sh

Feladat 3:
alternatív útvonalak eltérő költséggel

Példa:

```
ip route add 1.1.1.1/1 via 2.2.2.2 metric 3
```

több út esetén a kisebb költségűt választja a router

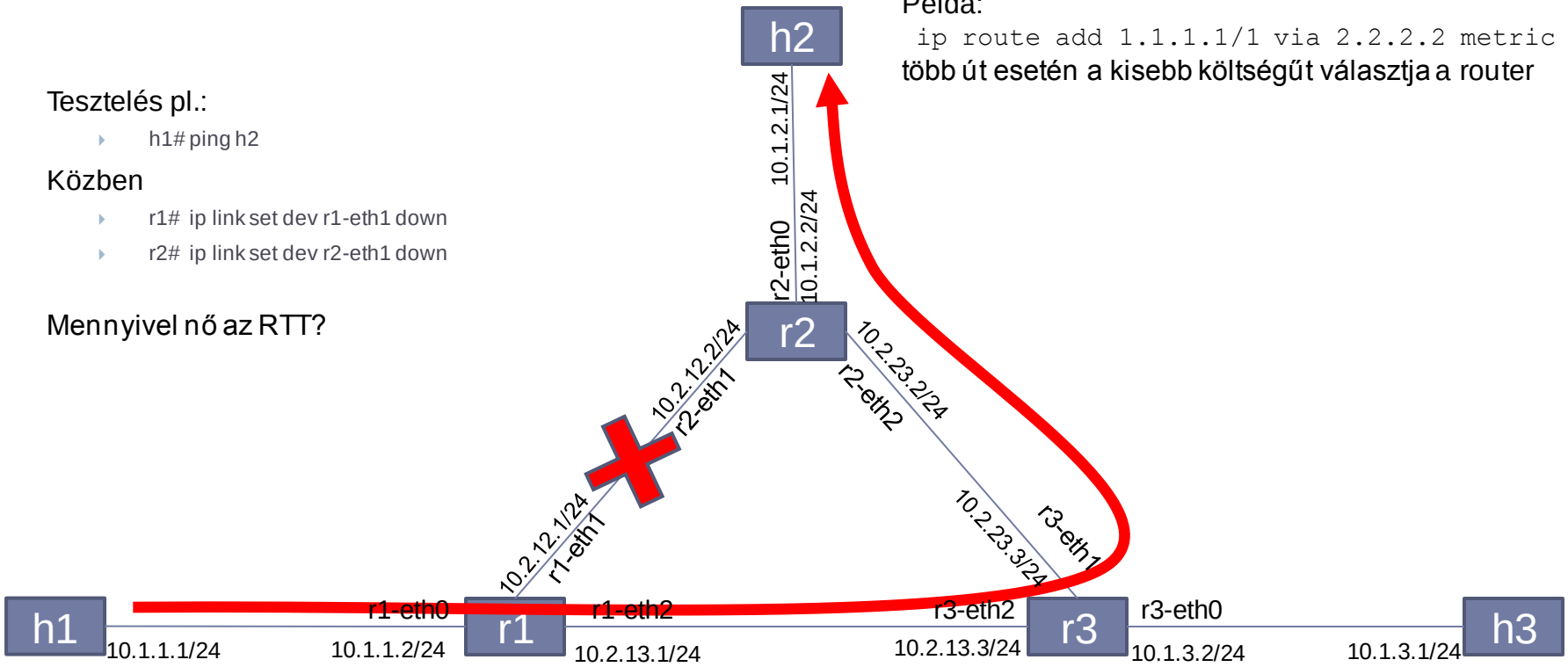
Tesztelés pl.:

- h1# ping h2

Közben

- r1# ip link set dev r1-eth1 down
- r2# ip link set dev r2-eth1 down

Mennyivel nő az RTT?



Feladat 3:
alternatív útvonalak eltérő költséggel

```
#!/bin/bash
case $1 in
  h1)
    ip addr del 10.0.0.1/8  dev h1-eth0
    ip addr add 10.1.1.1/24 dev h1-eth0
    ip route add default via 10.1.1.2
    ;;
  h2)
    ip addr del 10.0.0.2/8  dev h2-eth0
    ip addr add 10.1.2.1/24 dev h2-eth0
    ip route add default via 10.1.2.2
    ;;
  h3)
    ip addr del 10.0.0.3/8  dev h3-eth0
    ip addr add 10.1.3.1/24 dev h3-eth0
    ip route add default via 10.1.3.2
    ;;
```

```
  r1)
    sysctl -w net.ipv4.ip_forward=1
    ip addr del 10.0.0.4/8  dev r1-eth0
    ip addr add 10.1.1.2/24 dev r1-eth0
    ip addr add 10.2.12.1/24 dev r1-eth1
    ip addr add 10.2.13.1/24 dev r1-eth2
    ip route add 10.1.2.0/24 via 10.2.12.2 metric 10
    ip route add 10.1.2.0/24 via 10.2.13.3 metric 20
    ip route add 10.1.3.0/24 via 10.2.13.3
    ;;
  r2)
    sysctl -w net.ipv4.ip_forward=1
    ip addr del 10.0.0.5/8  dev r2-eth0
    ip addr add 10.1.2.2/24 dev r2-eth0
    ip addr add 10.2.12.2/24 dev r2-eth1
    ip addr add 10.2.23.2/24 dev r2-eth2
    ip route add 10.1.1.0/24 via 10.2.12.1 metric 10
    ip route add 10.1.1.0/24 via 10.2.23.3 metric 20
    ip route add 10.1.3.0/24 via 10.2.23.3
    ;;
```

```
  r3)
    sysctl -w net.ipv4.ip_forward=1
    ip addr del 10.0.0.6/8  dev r3-eth0
    ip addr add 10.1.3.2/24 dev r3-eth0
    ip addr add 10.2.23.3/24 dev r3-eth1
    ip addr add 10.2.13.3/24 dev r3-eth2
    ip route add 10.1.1.0/24 via 10.2.13.1
    ip route add 10.1.2.0/24 via 10.2.23.2
    ;;
  *)
    echo usage $0 hostname >&2
    ;;
esac
```

wget http://hsn.tmit.bme.hu/d.sh

Feladat 4:

- Mi történik, ha h1-eth0 10.1.1.1/24 címét 10.1.1.1/8 –ra cseréljük és h1-ről pingeljük h2-t?
  - Nézzük meg tcpdumppal, vagy wiresharkkal.

Feladat 4:

- Mi történik, ha h1-eth0 10.1.1.1/24 címét 10.1.1.1/8 –ra cseréljük és pingeljük h1-ről h2-t?
  - Nézzük meg tcpdumppal, vagy wiresharkkal.

  h1 szerint h2 egy alhálózatban van vele, ezért a MAC címét próbálja lekérni (sikertelenül).

Feladat 5:

- Adjuk h1 ARP táblájához hozzá h2-t 66:55:44:33:22:11 mac címmel. Mi történik, ha h1 pingeli h2-t?

Feladat 5: arp

- h1# arp -s 10.1.2.1 66:55:44:33:22:11
- h1# ping h2

- h1# tcpdump -e -i h1-eth0

Python - scapy

```
$ sudo apt install python-scapy
$ wget http://hsn.tmit.bme.hu/smallFlows.pcap
```

(https://s3.amazonaws.com/tcpreplay-pcap-files/smallFlows.pcap)

```
p2p@p2p-VirtualBox:~$ scapy
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> cap=rdpcap("smallFlows.pcap")
>>> cap
<smallFlows.pcap: TCP:13708 UDP:501 ICMP:34 Other:18>
>>> cap[0].show()
###[ Ethernet ]###
  dst= 00:1a:8c:15:f9:80
  src= 40:61:86:9a:f1:f5
  type= 0x800
###[ IP ]###
     version= 4L
     ihl= 5L
     tos= 0x0
     len= 983
     id= 30432
     flags= DF
     frag= 0L
     ttl= 128
     proto= tcp
     chksum= 0x9e7c
     src= 192.168.3.131
     dst= 72.14.213.138
     \options\
###[ TCP ]###
        sport= 57011
        dport= http
        seq= 2542412440
        ack= 2428019935
        dataofs= 5L
        reserved= 0L
        flags= PA
        window= 16288
        chksum= 0x5df8
        urgptr= 0
        options= []
###[ Raw ]###
           load= 'GET /complete/search?client=chrome&hl=en-US&q=cr HTTP/1.1
ost: clients1.google.ca\r\nConnection: keep-alive\r\nUser-Agent: Mozilla/5.
ndows; U; Windows NT 6.1; en-US) AppleWebKit/534.10 (KHTML, like Gecko) Chr
.0.552.237 Safari/534.10\r\nAccept-Encoding: gzip,deflate,sdch\r\nAccept-La
e: en-US,en;q=0.8\r\nAccept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3\r\nCook
REF=ID=c2e350012258df1c:U=386a6ebef0db287c:FF=0:TM=1294164294:LM=1294164294
uwM6Vh5ecKxqmk; SID=DQAAAN4AAAB3Mw7hSAXm29svfZQxRhaEVL5x_7JEyWEywPtfIKmV2QM
fSvGxg-WCwS7OYnEonavdReiTgZ_3JalcPyInxYbHG668hbhfVRxCHWraC8lNhhhZvC45L32WDj
0qmoz_3SGzIDgumB2mgyjTHiqRdgEmopsEvouobSZDRxixXdANvTHyq85PmVnzKHK_-x7hVdYhU
_oI4bZWnHA966Qna73q5YOPPevvZQVX8F71nVjDk4aJM5KhlAQwBDx5fzrV9Wk_R_Y-egz0sDLS
URGWup4ywQ; HSID=AqgM3JlzrVA3Qkiyz; NID=43=F_oSZWyo9Niadk17m65QtM9alBJQ4YLC
71ruNqa5sV4JORImsQovU1PW2PEII7a-5KUi4YCRMCeytuhiwkWgSlWtHEAj_nt_EF8y84MNmrt
Kth96N4-bPZ\r\n\r\n'
>>> print cap[0][TCP].dport
80
>>>
```

Feladat N+1: melyik TOS értékből hány darab van a pcap fájlban?

- Például:

TOS	Darab
0x0	123
0x2	44
0x10	2

- a gépen egy régi scapy változat van előtelepítve,
  - python2-re épül
  - a feladat szempontjából egyetlen fontos különbség:
    - **`print(a)`** helyett **`print a`**

Feladat N+1:
melyik TOS értékből hány darab van a pcap fájlban?

```
cap=rdpcap("smallFlows.pcap")

tos = {}
for p in cap:
  if IP in p:
    t = p[IP].tos
    tos[t] = tos.get(t, 0) + 1

for t in tos:
  print t, tos[t]
```

Kimenet:

```
0 13787
8 428
3 8
192 20
```

Kiugró

https://bit.ly/haepuz-gy2