
UNIVERSITAS HUNGARIAE OCCIDENTALIS
ANNO 1735

A dinamikus QoS rendszer neve jelenleg intcl (interface control)

- A 2008-as NetworkShop táján kezdem vele foglalkozni
- 2008.05.05-én írtam róla először a HBONE hbone-admin@listserv.niif.hu listára.
- 2008.05.27-ére sikerült olyan állapotba hozni, hogy elküldhettem a fent említett kistára.
- Miért beszélek most róla?

Követelmények a rendszerrel szemben

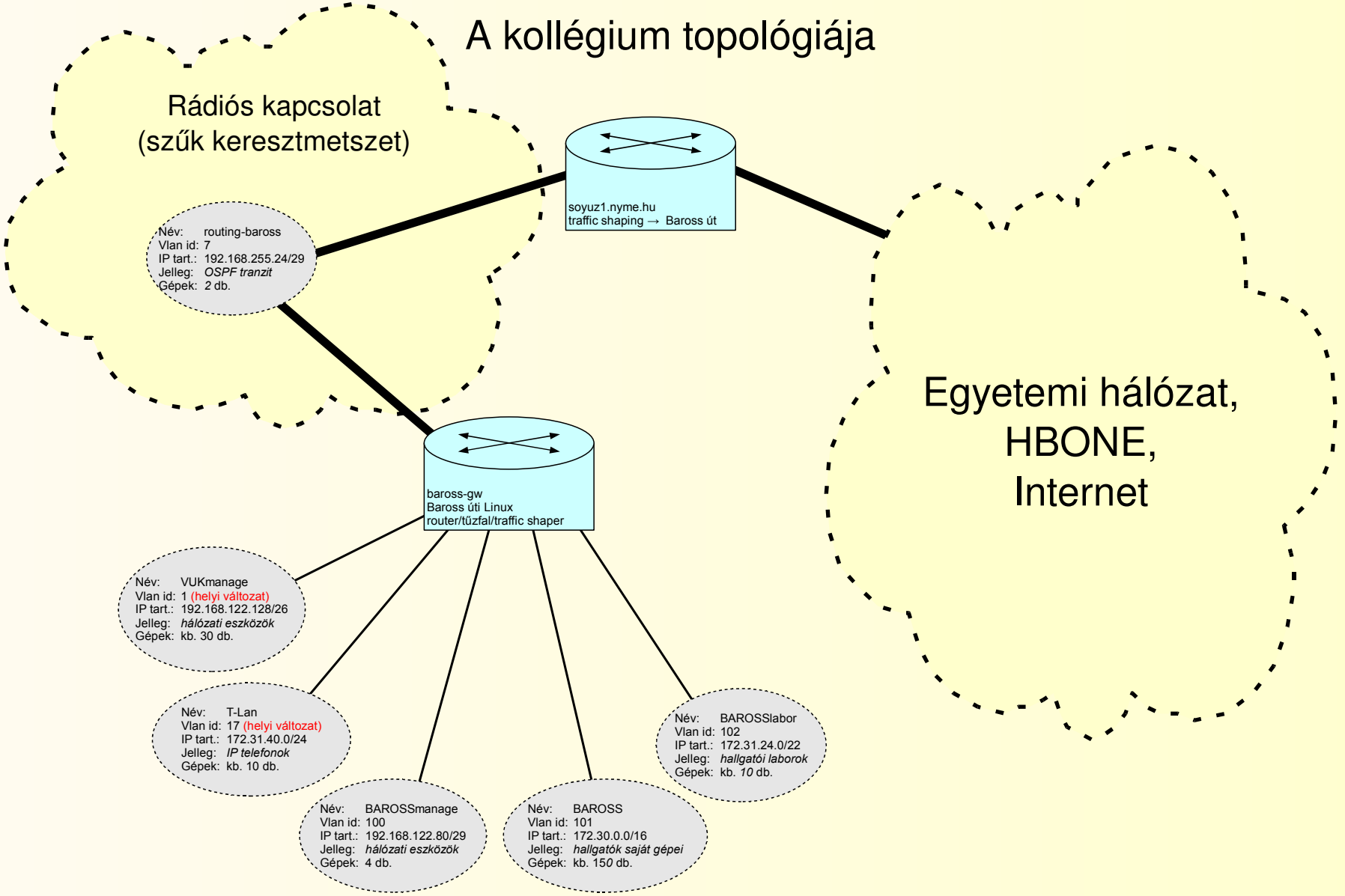
- Kb. 350-400 hallgatói gép ellátása
- A rendelkezésre álló sávszélesség (a legszűkebb ismert keresztmetszeten) 10Mb/s half duplex.
- Ne szűrjön semmit.
- Minden felhasználóval bánjon kesztyűs kézzel. Azokkal is, akik miatt szükség volt a létrehozására.
- Legyen teljesen automatikus.

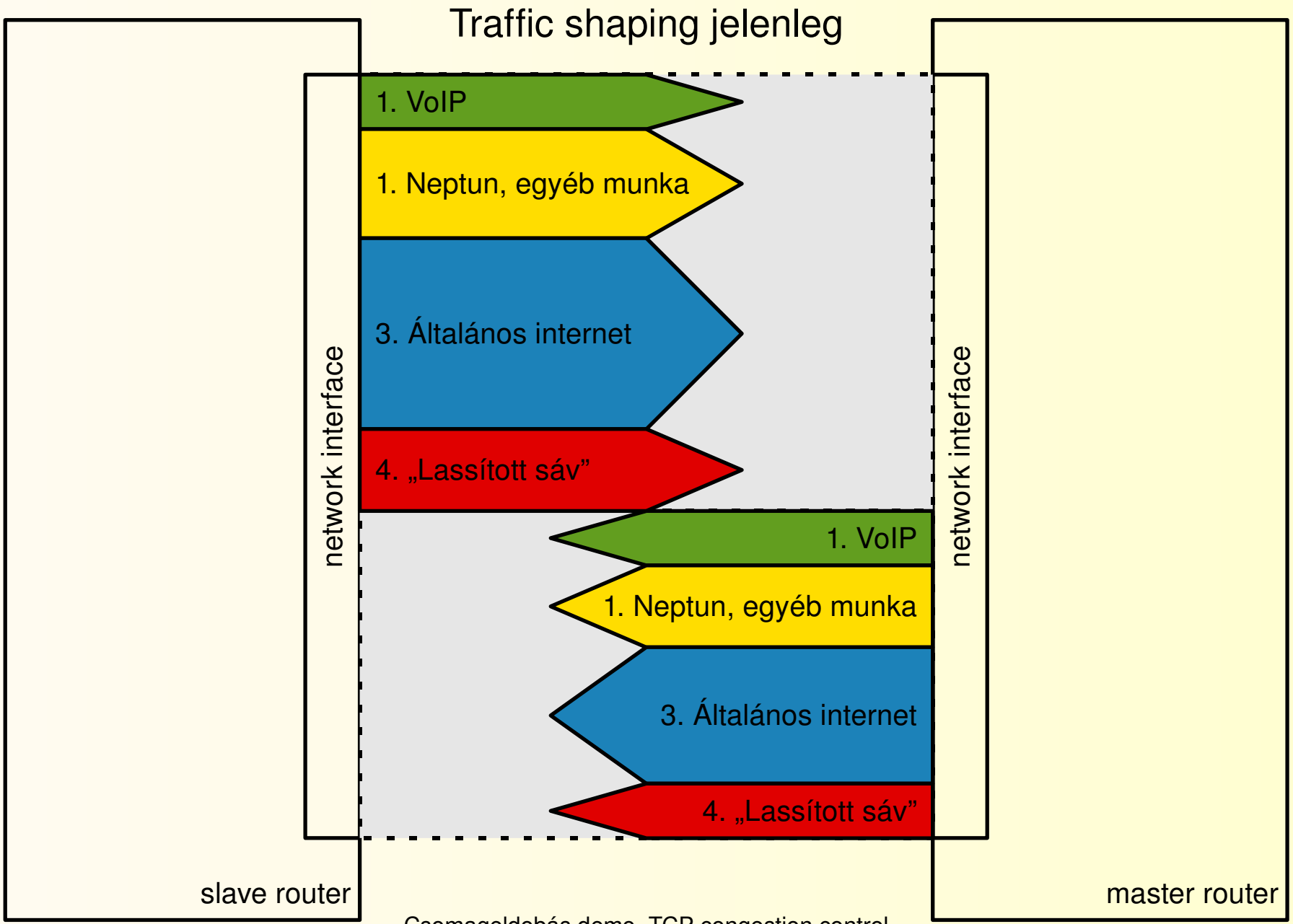
Miért gondoltam, hogy meg lehet csinálni,
és milyen más eszközök használatát vettem fontolóra?

•Úgy tűnik, hogy a P2P fájlátvitel részesedése a sávszélességből olyan nagy, hogy ha ez nem lenne, azt lehetne mondani, hogy az előbb említett sávszélesség elegendő a szóban forgó mennyiségű gép ellátására, főként mivel a P2P állandó okoz csak állandó terhelést, a közönséges internetezés pedig löketszerűt. A löketek időben fésűfogszerűen eltolva jól megférnek a szűk keresztmetszeten, főleg valami bufferelő traffic shaping-gel megtámogatva.

•Amivel még (legjobb tudomásom szerint) próbálkozni lehetne Linux alapon: http://www.ipp2p.org/, http://l7-filter.sourceforge.net/. De ezek leginkább a P2P felismerését teszik lehetővé. Mi van, ha más alkalmazás generál elfogadhatatlanul nagy forgalamat? Ráadásul úgy tűnik, nem ismerik fel a titkosított P2P forgalmat (demo). Talán éppen ezért állt le az ipp2p fejlesztése 2006 szeptemberében, ezáltal nagyon megnehezítve a legújabb kernelekbe történő integrálást?

A kollégium topológiája





Csomageldobás demo, TCP congestion control

Forgalomszabályozás az iproute2 segítségével:

```
tc qdisc del dev eth0.7 root
tc qdisc add dev eth0.7 root        handle   1:   htb default 20
tc class add dev eth0.7 parent 1:   classid  1:1  htb rate 2100kbit ceil 2100kbit
tc class add dev eth0.7 parent 1:1  classid  1:5  htb rate 300kbit  ceil 2100kbit prio 0
tc class add dev eth0.7 parent 1:1  classid  1:10 htb rate 400kbit  ceil 2100kbit prio 1
tc class add dev eth0.7 parent 1:1  classid  1:20 htb rate 1200kbit ceil 2100kbit prio 2
tc class add dev eth0.7 parent 1:1  classid  1:30 htb rate 200kbit  ceil 2100kbit prio 3
tc qdisc add dev eth0.7 parent 1:5  handle   5: bfifo limit 65536
tc qdisc add dev eth0.7 parent 1:10 handle  10: sfq perturb 10
tc qdisc add dev eth0.7 parent 1:20 handle  20: sfq perturb 10

iptables -t mangle -F POSTROUTING
iptables -t mangle -A POSTROUTING -o eth0.7                     -j CLASSIFY --set-class 1:20
iptables -t mangle -A POSTROUTING -o eth0.7                     -j DSCP --set-dscp-class AF12
iptables -t mangle -A POSTROUTING -o eth0.7 -m connmark --mark 3 -j CLASSIFY --set-class 1:30
iptables -t mangle -A POSTROUTING -o eth0.7 -m connmark --mark 3 -j DSCP --set-dscp-class AF13
iptables -t mangle -A POSTROUTING -o eth0.7 -m connmark --mark 1 -j CLASSIFY --set-class 1:10
iptables -t mangle -A POSTROUTING -o eth0.7 -m connmark --mark 1 -j DSCP --set-dscp-class AF11
iptables -t mangle -A POSTROUTING -o eth0.7 -m connmark --mark 5 -j CLASSIFY --set-class 1:5
iptables -t mangle -A POSTROUTING -o eth0.7 -m connmark --mark 5 -j DSCP --set-dscp-class EF
```

Forgalomszabályozás az iproute2 segítségével 2.:

```
iptables -t mangle -N traffic_prio
iptables -t mangle -F traffic_prio
iptables -t mangle -N intcl-eth0.101
iptables -t mangle -F FORWARD
iptables -t mangle -A FORWARD -o eth0.7 -m connmark --mark 0 -j traffic_prio
##########################################
iptables -t mangle -A traffic_prio -i eth0              -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -i eth0.100          -j CONNMARK --set-mark 1
#
iptables -t mangle -A traffic_prio -d 193.225.93.1      -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -d 193.225.93.200    -j CONNMARK --set-mark 1
#
iptables -t mangle -A traffic_prio -p icmp              -j CONNMARK --set-mark 1
#
iptables -t mangle -A traffic_prio -p udp --dport 123   -j CONNMARK --set-mark 1
#
iptables -t mangle -A traffic_prio -p tcp --dport 465   -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -p tcp --dport 110   -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -p tcp --dport 995   -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -p tcp --dport 139   -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -p tcp --dport 445   -j CONNMARK --set-mark 1
###
iptables -t mangle -A traffic_prio -i eth0.101          -j intcl-eth0.101
###
iptables -t mangle -A traffic_prio -d 193.224.61.224/27 -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -d 193.225.93.11     -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -d 193.225.93.77     -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -d 193.225.93.72     -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -d 172.16.1.6        -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -d 172.16.1.7        -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -d 172.16.1.21       -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -d 172.16.4.223      -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -p tcp --sport 5800  -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -p tcp --sport 5900  -j CONNMARK --set-mark 1
###
iptables -t mangle -A traffic_prio -s 172.23.40.0/24 -d 172.22.0.0/16 -j CONNMARK --set-mark 5
```

Forgalomszabályozás az iproute2 segítségével 3.:
(Egy kicsit nem illik ide.)

```
Chain intcl-eth0.101 (1 references)
 pkts bytes target     prot opt in     out     source               destination
   10   480 CONNMARK   all  --  *      *       172.21.255.248       0.0.0.0/0           CONNMARK set 0x3
    0     0 CONNMARK   all  --  *      *       172.21.255.216       0.0.0.0/0           CONNMARK set 0x3
  147  8278 CONNMARK   all  --  *      *       172.21.255.203       0.0.0.0/0           CONNMARK set 0x3
   51  2683 CONNMARK   all  --  *      *       172.21.253.162       0.0.0.0/0           CONNMARK set 0x3
    7   365 CONNMARK   all  --  *      *       172.21.254.128       0.0.0.0/0           CONNMARK set 0x3
```

Alapötlet

•Az adott interfész arp táblájából megtudhatjuk, milyen gépek aktívak jelenleg, mi a MAC- és IP-címük.

•Az iptables-be megfelelő módon betöltve ezeket az adatokat, minden gépről külön forgalmi statisztikát vezethetünk feltöltési és letöltési irányban egyaránt.

•A túlságosan nagy terhelést okozó gépek forgalmát a „Lassított sávba” irányítjuk. MAC-cím alapján tartjuk nyilván a gépeket, hogy IP-cím változtatással ne lehessen megkerülni a rendszert (demo).

•A fontos munkához (pl. Neptun) kapcsolódó forgalmat még a lassított sávban levő gépek esetében se korlátozzuk.

•A lassított sávban levő gépek közül a „megbánást tanúsítók” egy rövid büntetési periódus után visszatérhetnek az általános internet gyors sávjába.

arp tábla:

```
baross-gw:~ # arp -n -i eth0.101
Address                  HWtype  HWaddress           Flags Mask            Iface
172.21.255.203           ether   00:16:36:37:E3:39   C                     eth0.101
172.21.254.161           ether   00:90:F5:53:2D:BD   C                     eth0.101
172.21.255.86            ether   00:1D:60:50:F5:F2   C                     eth0.101
172.21.254.58            ether   00:1D:60:B0:83:72   C                     eth0.101
172.21.100.2             ether   00:15:F2:01:98:DF   C                     eth0.101
172.21.255.173           ether   00:00:E2:6E:48:77   C                     eth0.101
172.21.252.245           ether   00:90:F5:55:4C:3E   C                     eth0.101
172.21.252.103           ether   00:1A:4B:76:5C:43   C                     eth0.101
172.21.253.253           ether   00:13:D3:C4:C5:66   C                     eth0.101
172.21.255.55            ether   00:19:DB:F0:79:CF   C                     eth0.101
172.21.255.10            ether   00:06:29:99:F1:99   C                     eth0.101
```

Az iptables adatgyűjtő része:

```
baross-gw:~ # iptables -t filter -L -nv
Chain ACCT_IN_eth0.101 (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 RETURN     all  --  *      *       172.21.255.173       0.0.0.0/0           /* 00:00:E2:6E:48:77 */
    0     0 RETURN     all  --  *      *       172.21.252.67        0.0.0.0/0           /* 00:03:0D:47:CE:05 */
  155  127K RETURN     all  --  *      *       172.21.254.68        0.0.0.0/0           /* 00:03:25:10:4E:9A */
    0     0 RETURN     all  --  *      *       172.21.255.10        0.0.0.0/0           /* 00:06:29:99:F1:99 */
    0     0 RETURN     all  --  *      *       172.21.254.17        0.0.0.0/0           /* 00:0F:3D:32:A7:8C */
   84 11004 RETURN     all  --  *      *       172.21.255.147       0.0.0.0/0           /* 00:0F:EA:1C:74:D3 */
    1    44 RETURN     all  --  *      *       172.21.253.99        0.0.0.0/0           /* 00:11:2F:A6:F8:AC */
   87  5268 RETURN     all  --  *      *       172.21.253.51        0.0.0.0/0           /* 00:11:D8:CD:70:E5 */
  116  8091 RETURN     all  --  *      *       172.21.253.253       0.0.0.0/0           /* 00:13:D3:C4:C5:66 */

Chain ACCT_OUT_eth0.101 (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 RETURN     all  --  *      *       0.0.0.0/0            172.21.255.173      /* 00:00:E2:6E:48:77 */
    0     0 RETURN     all  --  *      *       0.0.0.0/0            172.21.252.67       /* 00:03:0D:47:CE:05 */
   65  3126 RETURN     all  --  *      *       0.0.0.0/0            172.21.254.68       /* 00:03:25:10:4E:9A */
    0     0 RETURN     all  --  *      *       0.0.0.0/0            172.21.255.10       /* 00:06:29:99:F1:99 */
    0     0 RETURN     all  --  *      *       0.0.0.0/0            172.21.254.17       /* 00:0F:3D:32:A7:8C */
   38 11542 RETURN     all  --  *      *       0.0.0.0/0            172.21.255.147      /* 00:0F:EA:1C:74:D3 */
    0     0 RETURN     all  --  *      *       0.0.0.0/0            172.21.253.99       /* 00:11:2F:A6:F8:AC */
  151  190K RETURN     all  --  *      *       0.0.0.0/0            172.21.253.51       /* 00:11:D8:CD:70:E5 */
  142  134K RETURN     all  --  *      *       0.0.0.0/0            172.21.253.253      /* 00:13:D3:C4:C5:66 */
```

Az adatgyűjtő rész működéséhez arra van szükség, hogy az intcl szűrőlistáin minden engedélyezett forgalom áthaladhasson, de csak az engedélyezett:

```
modprobe nf_conntrack hashsize=65535
modprobe nf_conntrack_ftp
################################################
iptables -t filter -N rejecter
iptables -t filter -F rejecter
iptables -t filter -A rejecter        -j LOG
iptables -t filter -A rejecter -p tcp -j REJECT --reject-with tcp-reset
iptables -t filter -A rejecter -p udp -j REJECT --reject-with icmp-port-unreachable
iptables -t filter -A rejecter        -j REJECT --reject-with icmp-proto-unreachable
###
iptables -t filter -F FORWARD
iptables -t filter -P FORWARD ACCEPT
iptables -t filter -A FORWARD -m state --state INVALID -j rejecter
###
iptables -t filter -N to_T-Lan
iptables -t filter -F to_T-Lan
iptables -t filter -A FORWARD -o eth0.17 -m state --state NEW -j to_T-Lan
#
IFS=$'\n'
for EXPR in $(</usr/local/LOCALHOST/network/to_T-Lan)
do
 IFS=$' \t\n'
 iptables -t filter -A to_T-Lan $EXPR
done
################################################
iptables -t filter -N ACCT_IN_eth0.101
iptables -t filter -A FORWARD -i eth0.101 -o eth0.7 -j ACCT_IN_eth0.101
iptables -t filter -N ACCT_OUT_eth0.101
iptables -t filter -A FORWARD -o eth0.101 -i eth0.7 -j ACCT_OUT_eth0.101
```

Az adatgyűjtő rész működéséhez arra van szükség,
hogy az intcl szűrőlistáin minden engedélyezett forgalom
áthaladhasson 2.:

```
elisabeth-gw:~ # iptables -t filter -L to_T-Lan -nv
Chain to_T-Lan (1 references)
 pkts bytes target     prot opt in     out     source               destination
   53  3168 RETURN     all  --  *      *       172.22.0.0/16        192.168.19.128/26
    0     0 RETURN     all  --  *      *       172.23.40.0/24       192.168.19.128/26
    0     0 rejecter   all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

Konfigurációs paraméterek 1.:

•MASTER_INTERFACE: Ha egy adott interfészen túl kevés gép van jelen ahhoz, hogy megbízhatóan lehessen átlagot számolni, akkor az interfész konfigurációjában, hivatkozhatunk egy másik interfészre, mint MASTER-re. Ezzel az aktuális interfész SLAVE-vé válik, és nem történik rajta átlagszámítás, hanem a MASTER interfészen kiszámított forgalmi átlagértékekhez viszonyítva állapítja meg, hogy mely gépek milyen forgalmi sávba kerüljenek. Csak akkor van értelme, ha a két interfész osztozik a szűk keresztmetszet sávszélességén.

•GOOD_PROPORTION: Ha egy nem lassított sávban (ez az ún. GOOD kategória) levő gép túllépi a saját kategóriája átlagos forgalmi értékének ennyi ezrelékét, akkor a lassított sávba (EVIL kategória) kerül.

•EVIL_PROPORTION: Ha egy lassított sávban (ez az ún. EVIL kategória) levő gép nem lépi túl a saját kategóriája átlagos forgalmi értékének ennyi ezrelékét, akkor a büntetési periódus (PENALTY_MINUTES) letelte után visszakerülhet a nem lassított sávba (GOOD kategória).

•PENALTY_MINUTES: Ld. fent.

•INITIAL_PENALTY_MINUTES: Az újonnan megjelenő gépek (MAC címek) az első néhány percben mindenképpen a lassított sávba kerülnek. Ez a MAC-cím változtatással történő büntetésmegkerülés ellen nyújt némi védelmet. Cisco port security-vel kiegészítve jobb lenne, de az itt nincs.

Konfigurációs paraméterek 2.:

•MAC_EXPIRY_MINUTES: Ha ennyi percig nem kommunikál egy hoszt, akkor kikerül a szkript adatbázisából, és legközelebb új hosztként jelenik meg (ld.: INITIAL_PENALTY_MINUTES).

•UPLOAD_PENALTY_MULTIPLIER: A feltöltött bájtokat többszörösen vesszük figyelembe, így hatékonyabb a P2P kategorizálása.

•SUMMARY_ITERATIONS: A szkriptet a cron démon futtatja percenként. Az utolsó SUMMARY_ITERATIONS perc forgalmi adatait használja fel a szkript a forgalmi átlagadatok kiszámolásához.

•SANCTIONING_SCRIPT: Minden futási periódusben meghívjuk ezt a szkriptet, és átadjuk neki az EVIL kategóriábn levő gépek IP- és MAC címeit E szkript feladata az EVIL gépek lassított forgalmi sávba helyezése..

•AVG_BOUND_UP és AVG_BOUND_DOWN: Az átlagszámítás két körben történik, Az első körben a kategóriák összes gépépnek forgalmi adatait figyelembe veszük, a másodikban viszont kihagyjuk a túlságosan szélsőséges forgalmat produkáló gépeket, hogy az átlagszámítás egyenletesebb, megbízhatóbb eredményt adjon.

Az IP fejléc DSCP mezőjének használata a slave router vezérlésére:

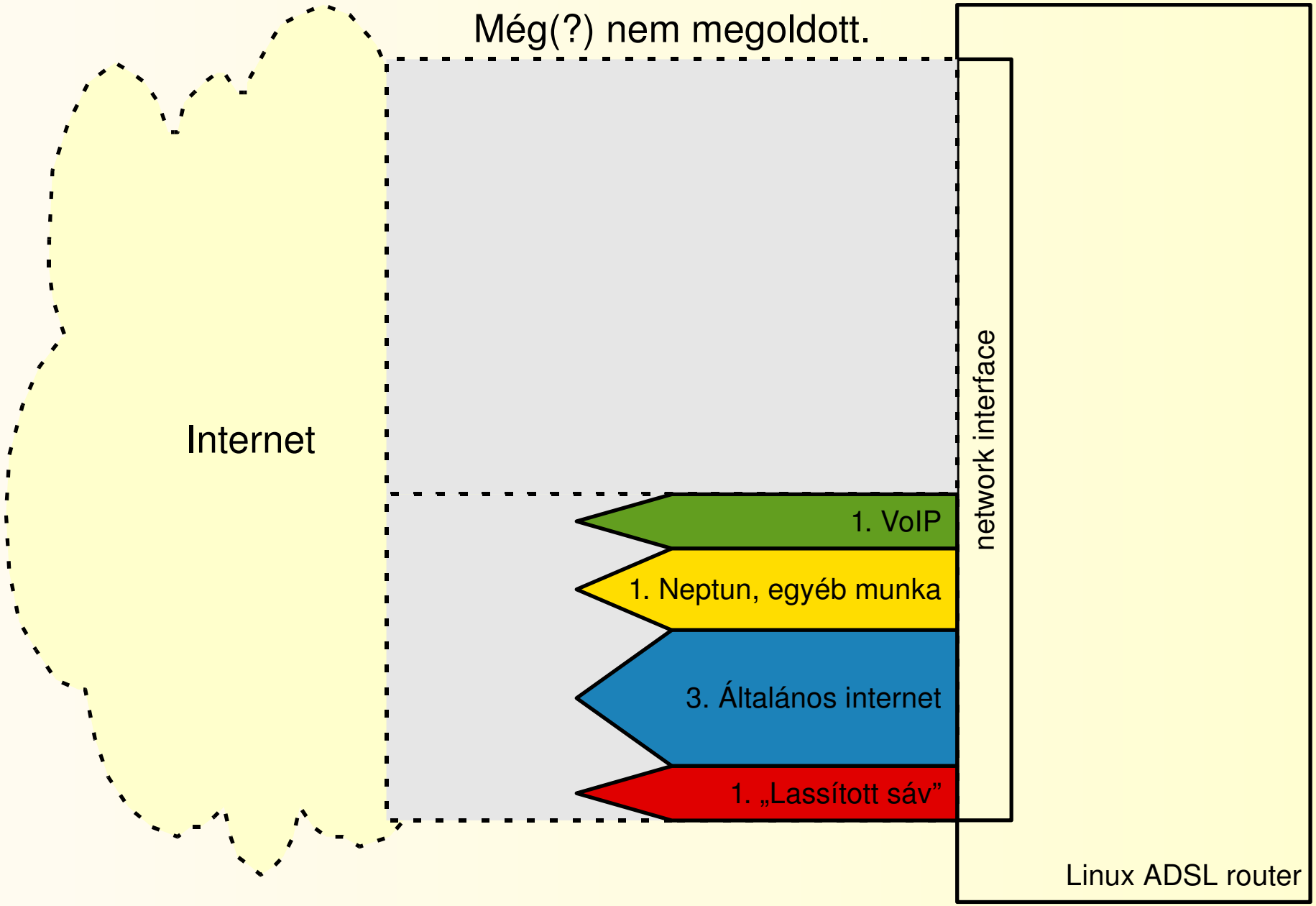
```
tc qdisc del dev eth1 root
tc qdisc add dev eth1 root        handle   1:   htb default 20
tc class add dev eth1 parent 1:   classid  1:1  htb rate 7900kbit ceil 7900kbit
tc class add dev eth1 parent 1:1  classid  1:5  htb rate 300kbit  ceil 7900kbit prio 0
tc class add dev eth1 parent 1:1  classid  1:10 htb rate 2400kbit ceil 7900kbit prio 1
tc class add dev eth1 parent 1:1  classid  1:20 htb rate 4500kbit ceil 7900kbit prio 2
tc class add dev eth1 parent 1:1  classid  1:30 htb rate 700kbit  ceil 7900kbit prio 3
tc qdisc add dev eth1 parent 1:5  handle   5: bfifo limit 65536
tc qdisc add dev eth1 parent 1:10 handle  10: sfq perturb 10
tc qdisc add dev eth1 parent 1:20 handle  20: sfq perturb 10
tc qdisc add dev eth1 parent 1:30 handle  30: sfq perturb 10
##########################################
iptables -t nat    -F PREROUTING
iptables -t nat    -F POSTROUTING
##########################################
iptables -t mangle -F OUTPUT
iptables -t mangle -A OUTPUT -j MARK --set-mark 1
##########################################
iptables -t mangle -N traffic_prio
iptables -t mangle -F traffic_prio
iptables -t mangle -F FORWARD
iptables -t mangle -A FORWARD -i eth1 -m connmark --mark 0 -j traffic_prio
##########################################
iptables -t mangle -A traffic_prio -m dscp --dscp-class AF11 -j CONNMARK --set-mark 1
iptables -t mangle -A traffic_prio -m dscp --dscp-class AF13 -j CONNMARK --set-mark 3
iptables -t mangle -A traffic_prio -m dscp --dscp-class EF   -j CONNMARK --set-mark 5
##########################################
iptables -t mangle -F POSTROUTING
iptables -t mangle -A POSTROUTING -o eth1 -m connmark --mark 0 -j CLASSIFY --set-class 1:20
iptables -t mangle -A POSTROUTING -o eth1 -m connmark --mark 0 -j DSCP --set-dscp-class AF12
iptables -t mangle -A POSTROUTING -o eth1 -m connmark --mark 3 -j CLASSIFY --set-class 1:30
iptables -t mangle -A POSTROUTING -o eth1 -m connmark --mark 3 -j DSCP --set-dscp-class AF13
iptables -t mangle -A POSTROUTING -o eth1 -m connmark --mark 1 -j CLASSIFY --set-class 1:10
iptables -t mangle -A POSTROUTING -o eth1 -m connmark --mark 1 -j DSCP --set-dscp-class AF11
iptables -t mangle -A POSTROUTING -o eth1 -m connmark --mark 5 -j CLASSIFY --set-class 1:5
iptables -t mangle -A POSTROUTING -o eth1 -m connmark --mark 5 -j DSCP --set-dscp-class EF
```

Algoritmus és adatbázis demo
(ha még eddig nem lett volna).

Jövőbeni fejlesztési lehetőségek:

- A slave router Xen virtuális gépre történő áthelyezése
- A tűzfalszkriptek tisztítása, fejlesztése, hordozhatóbbá tétele
- Anomáliák (demo) kezelése, esetleg jobb algoritmussal.
- ADSL-en is működőképes rendszer kifejlesztése (aszimmetrikusság, Intermediate Functional Block, ld. következő dia).

ADLS elrendezés
Még(?) nem megoldott.



http://titanic.nyme.hu/~nice/intcl/