Telekommunikációs Hálózatok

9. gyakorlat

HÁLÓZATI CÍMFORDÍTÁS

NAT, porttovábbítás, SSH Tunnel, iptables

Hálózati címfordítás (NAT)

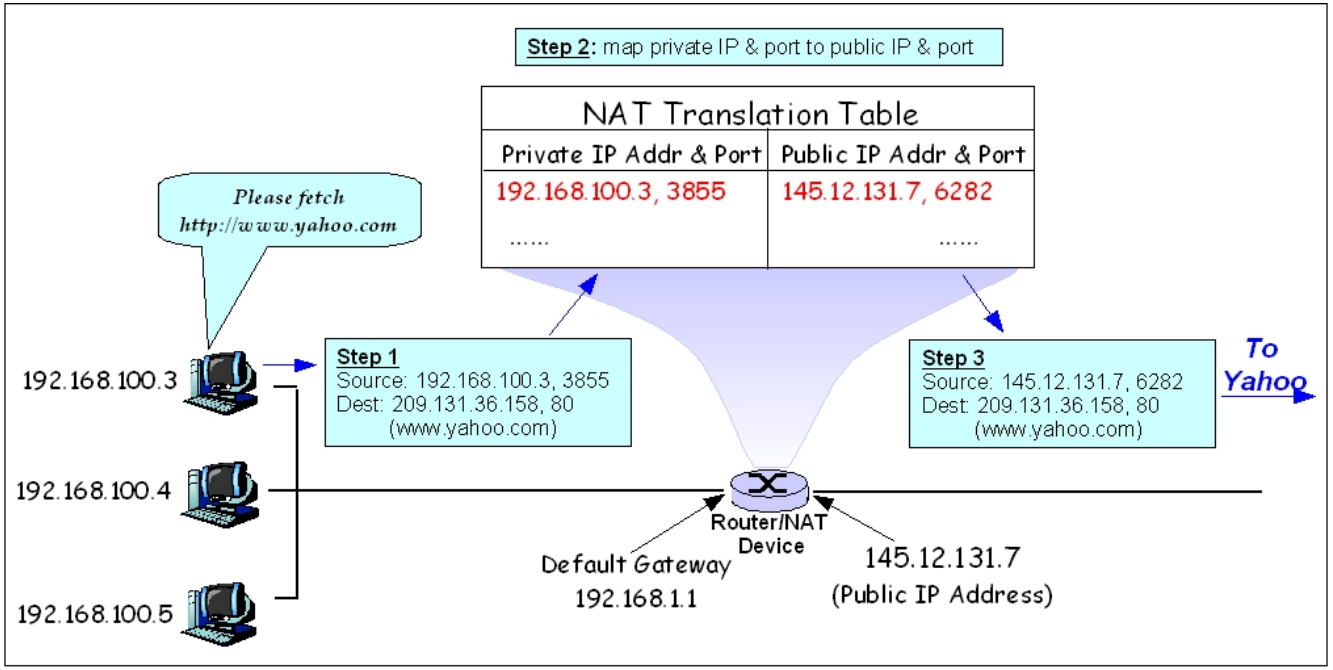
- Gyors javítás az IP címek elfogyásának problémájára.
- Az internet forgalomhoz minden cégnek egy vagy legalábbis kevés IP címet adnak (publikus IP cím(ek))
- A publikus IP cím hozzá van rendelve egy router-hez, a helyi hálózaton (LAN) belül, - amely mögötte van, - minden eszközhöz egy privát IP cím 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 eszközről akarunk egy külső eszközt elérni, mi történik?
- Szükségünk van port mezők használatára, ami TCP-nél vagy UDP-nél van

Hálózati címfordítás (NAT)

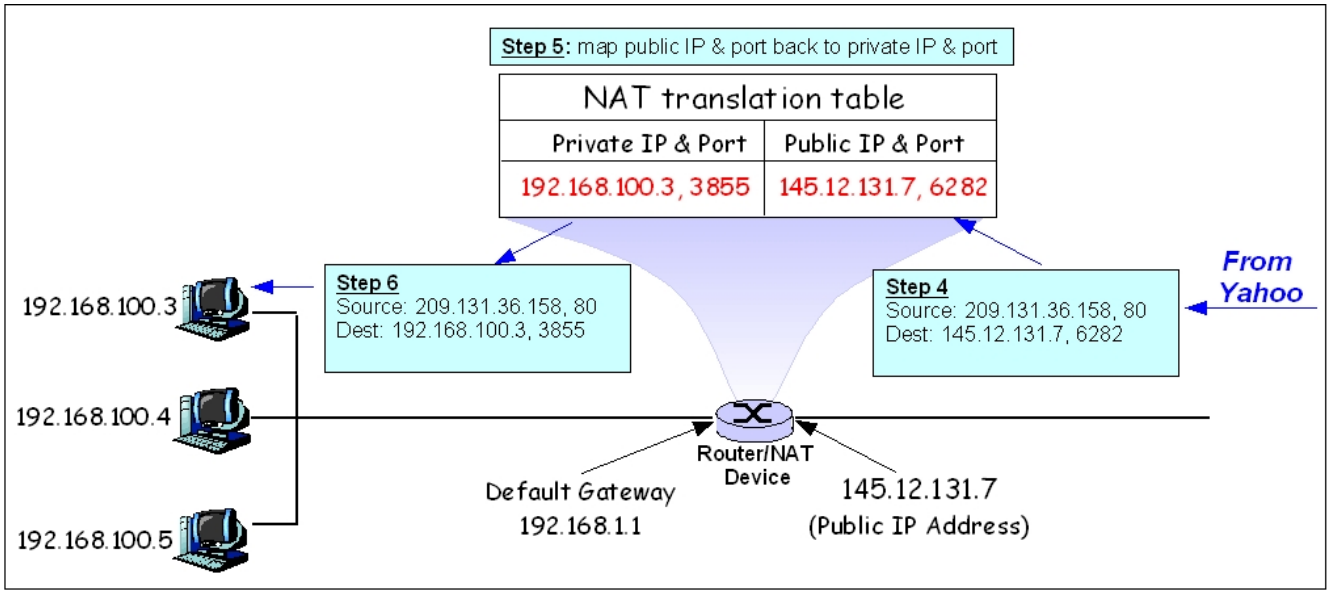
Forrás: https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols



- 192.168.100.3 privát IP című gépről HTTP kérés, 3855 porton → Default gateway (192.168.1.1): megnézi a transzlációs 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)

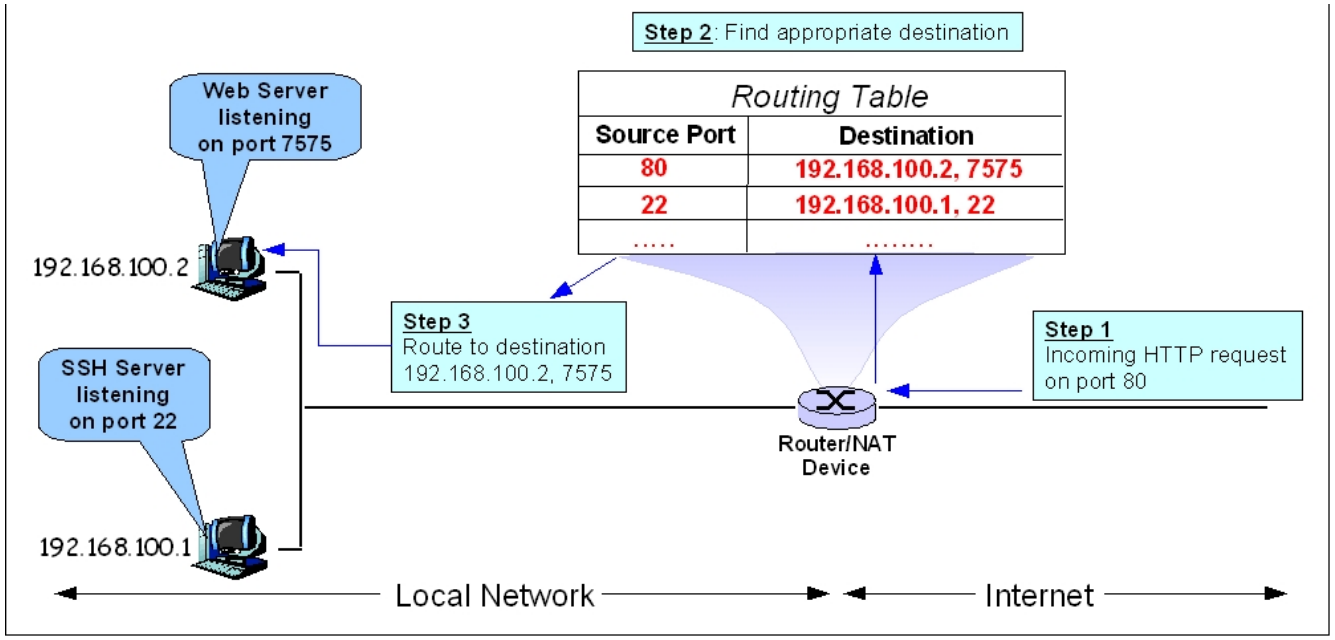
Forrás: https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols



- 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)

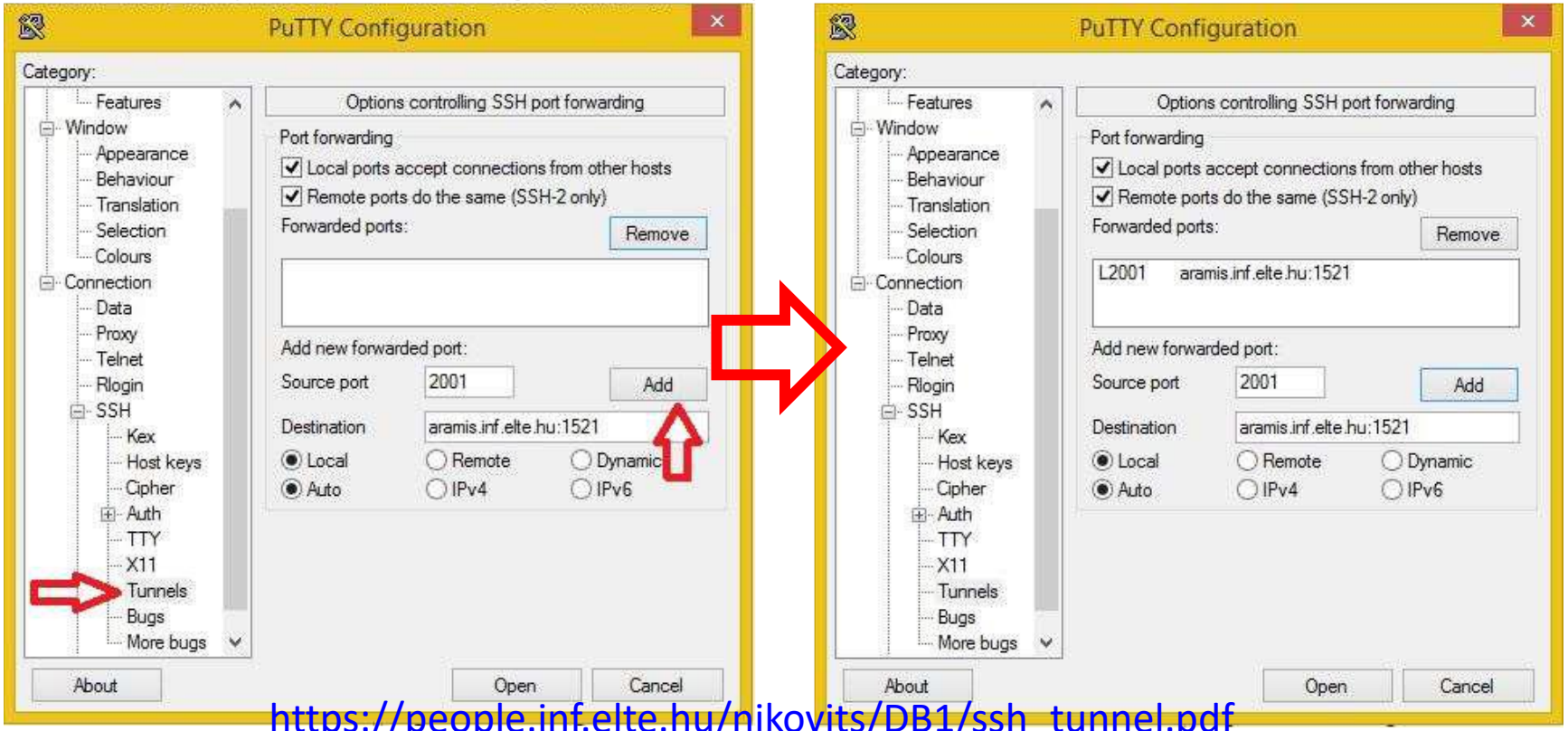
Forrás: https://en.wikibooks.org/wiki/Communication_Networks/NAT_and_PAT_Protocols



- 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

SSH Tunnel

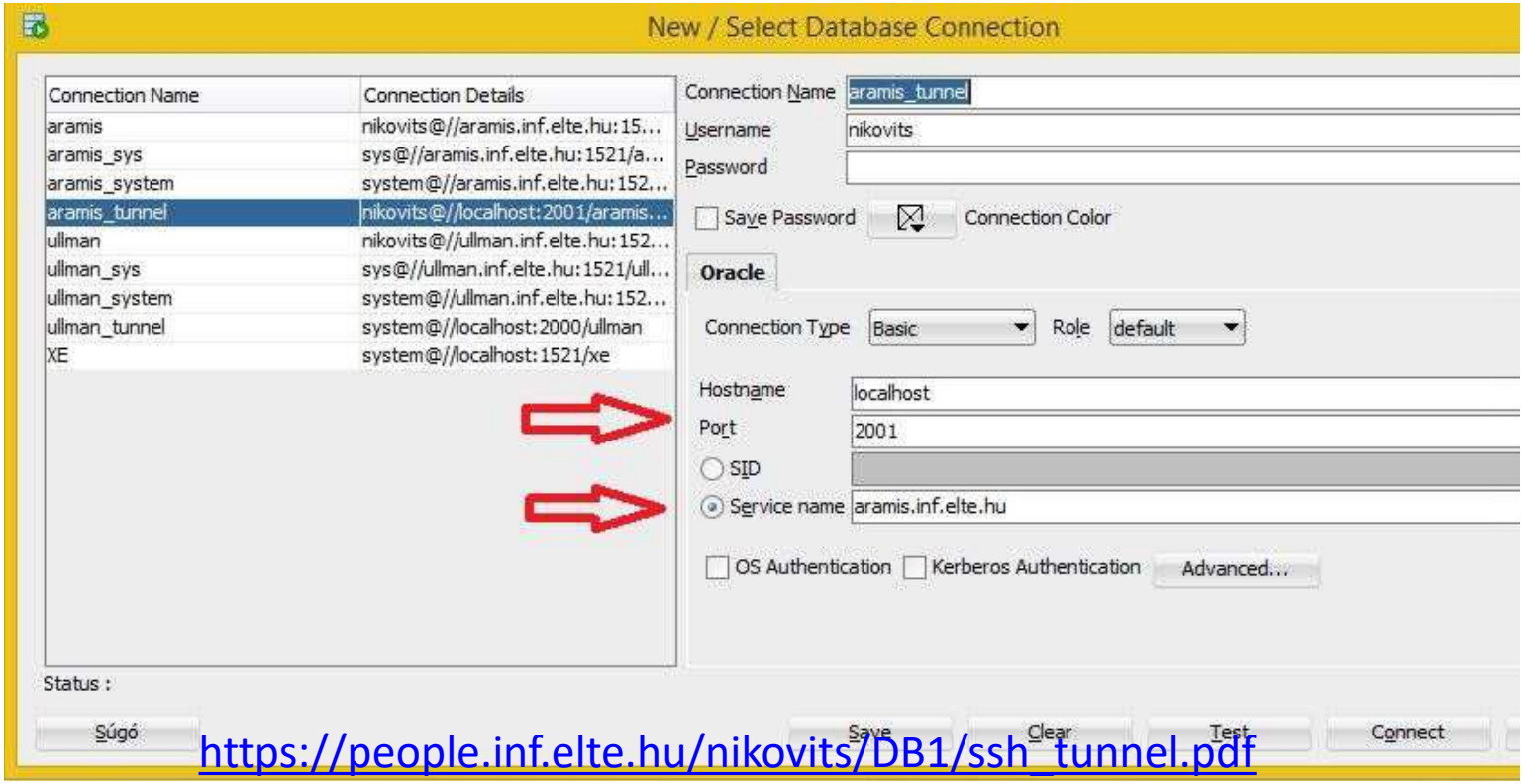
- Linux

```
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 porján 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

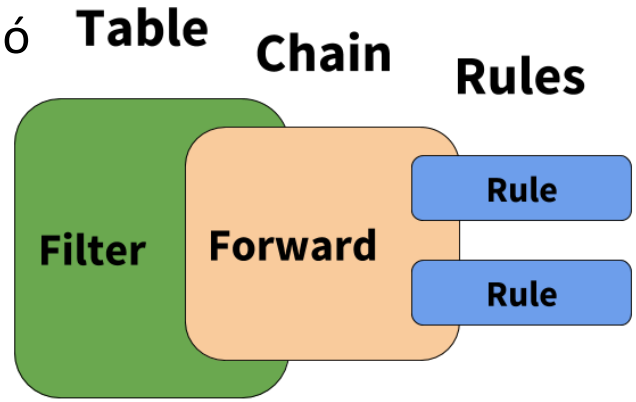
- Használat SqlDeveloper-nél:
  - (ssh kapcsolatnak fenn kell állnia végig az adatbázis kapcsolat ideje alatt)

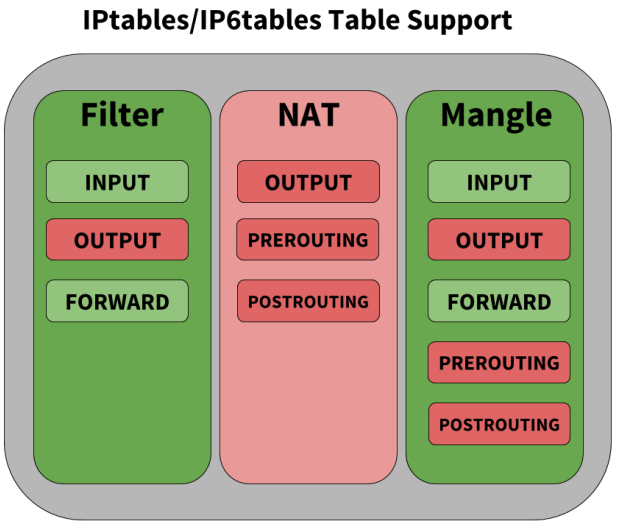


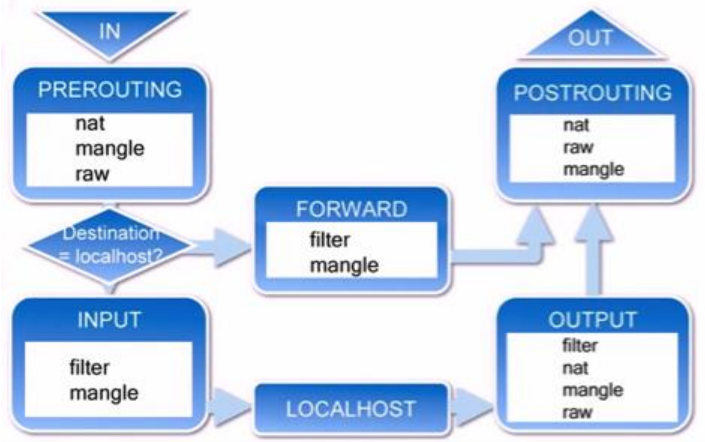
https://people.inf.elte.hu/nikovits/DB1/ssh_tunnel.pdf

iptables

- Az iptables egy Linux alkalmazás, amellyel a felhasználó konfigurálni tud tűzfal funkcionalitást, ill. csomagszűrési/csomagtovábbítási szabályokra, NAT módosítására/lekérdezésére jó







iptables

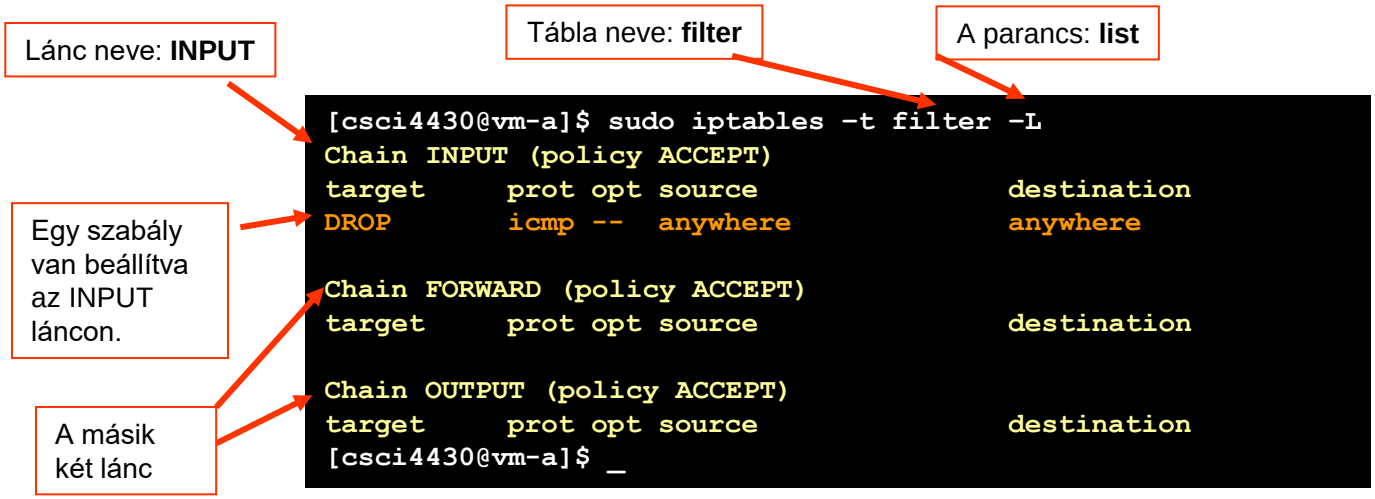
- Alapból három tábla van, amely szabályok halmazait tartalmazza
- A **filter** tábla a csomag szűrésre való
- A **nat** tábla a címfordításra való
- A **mangle** tábla a csomagok speciális célú feldolgozására való (megváltoztatja a csomagok tartalmát)
- Mindegyik táblában szabályok sorozata van, amelyeket láncoknak hívunk

iptables – filter tábla

- Itt három lánc van:
- Az INPUT láncot (az ott megadott szabályok sorozatát) bármely rendszerhez beérkező csomagra használja az alkalmazás
- Az OUTPUT láncot bármely olyan csomagra, amely a rendszerből kilép
- A FORWARD láncot pedig azokra a csomagokra, amelyek továbbítódnak a rendszeren keresztül (tehát ezeket nem a rendszernek szánták)

Forrás: The Chinese University of Hong Kong (CUHK), CSCI4430 Data Communication and Computer Networks
https://millyz.github.io/ta/csci4430_2017/Lab%20on%20iptables.ppt

iptables – filter tábla

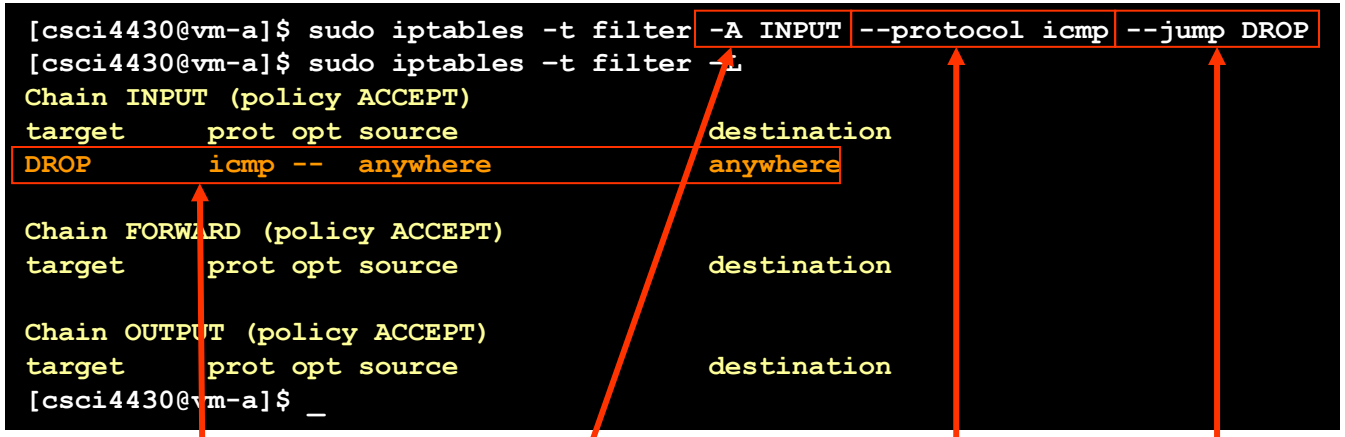


Az INPUT láncban lévő szabály jelentése:

Amikor egy ICMP hasznos teherrel rendelkező csomag áthalad az INPUT-on, DROP-olja ezt a csomagot függetlenül attól, hogy honnan jött, és hova megy.

Forrás: The Chinese University of Hong Kong (CUHK), CSCI4430 Data Communication and Computer Networks
https://millyz.github.io/ta/csci4430_2017/Lab%20on%20iptables.ppt

iptables – filter tábla



Ez a bejegyzés mutatja, hogy egy új szabály sikeresen hozzá lett adva a filter tábla INPUT láncához.

Egy új szabály hozzáadása az INPUT lánchoz.

Az ICMP protokollú csomagok lesznek érdekesek ennél a szabálynál.

Ha egy csomag áthalad az INPUT-on, és egy ICMP csomagról van szó, akkor DROP-olva lesz a csomag.

iptables – nat tábla

- Itt is három lánc van:
- Az OUTPUT lánc itt is van, de kevésbé érdekes
- A PREROUTING lánc még az előtt megváltoztatja a csomagokat mielőtt elérnék az INPUT láncot (pl. porttovábbítást szeretnénk alkalmazni)
- A POSTROUTING lánc pedig azután fogja megváltoztatni a csomagokat miután az OUTPUT láncot elhagyták (pl. a hálózati címfordítás első, egyszerűbb esete)

iptables – nat tábla

- Például szeretnénk a 192.168.1.10 IP címhez és 80-as porthoz jövő csomagot a 192.168.1.20 IP című géphez küldeni a 80-as portjához, akkor az alábbi parancsok (is) kelleni fognak:

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.1.20:80
```

```
iptables -t nat -A POSTROUTING -p tcp -d 192.168.12.20 --dport 80 -j SNAT --to-source 192.168.12.10
```

- (-t kapcsolóval a táblát határozzuk meg, -A PREROUTING : a szabályt a PREROUTING lánc végére szúrja be, -j a csomagcél megadására (SNAT: Source NAT, DNAT: Destination NAT))

iptables

- További példák itt:
- http://linux-training.be/networking/ch14.html
- (a Fájlok között is megvan: Chapter%A014.%A0iptables firewall.pdf)

HÁLÓZATI FORGALOM

tcpdump

Hálózati forgalom felvétele/elemzése

tcpdump (Linux):

forgalom figyelő eszköz, a hálózati interfészről jövő csomagokat tudja olvasni

```
lakis@dpdk-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@dpdk-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 dpdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 1, length 64
09:16:49.470766 IP 192.168.0.102 > dpdk-pktgen: ICMP echo reply, id 5668, seq 1, length 64
09:16:50.471818 IP dpdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 2, length 64
09:16:50.471834 IP 192.168.0.102 > dpdk-pktgen: ICMP echo reply, id 5668, seq 2, length 64
09:16:51.471716 IP dpdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 3, length 64
09:16:51.471732 IP 192.168.0.102 > dpdk-pktgen: ICMP echo reply, id 5668, seq 3, length 64
09:16:52.471713 IP dpdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 4, length 64
09:16:52.471729 IP 192.168.0.102 > dpdk-pktgen: ICMP echo reply, id 5668, seq 4, length 64
09:16:53.471720 IP dpdk-pktgen > 192.168.0.102: ICMP echo request, id 5668, seq 5, length 64
09:16:53.471736 IP 192.168.0.102 > dpdk-pktgen: ICMP echo reply, id 5668, seq 5, length 64
```

Hálózati forgalom felvétele/elemzése

tcpdump – további filterek

```
lakis@dpdk-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 dpdk-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 > dpdk-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,ws
cale 7], length 0
E..<8.@.@......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..<..@.@......f...e.W..O.:d.GU...q .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.@.@......e...f...W.GU.O.:e....|......
.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 0x4a14 (correct), seq 1:8, ack 1, win 229, options [nop,nop,TS val 409825747 ecr 835209270], lengt
h 7
E..;8.@.@......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 40982574
7], length 0
E..4rS@.@.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 40982
5747], length 3
E..7rT@.@.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.@.@......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], leng
th 15
E..C8.@.@......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 4098307
05], length 0
```

Hálózati forgalom felvétele/elemzése

tcpdump – mentés pcap fájlba és fájlból elemzés

```
lakis@dpdk-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@dpdk-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

MININET

Előfeltétel: *Mininet beállítás.pdf* diasoron végigmenni!

Mininet

- Nézzük meg a `test1.py`-t:

```
mininet@mininet-vm:~/gyakorlat$ vi test1.py
```

  - Egy `LinuxBridge`-et definiálunk, amellyel futtatni tudjuk a feszítőfa algoritmust (Spanning Tree Protocol, STP) hurkok kezelésére
  - Hozzáadunk hosztokat is, privát IP címekkel
  - Végül összekötjük ezeket a topológia alapján
  - A `h1` és `s1` kapcsolat sávszélessége: 10 Mbps (alapból elvileg nem limitált, a TCLink osztály azért kell, hogy limitálni tudjuk)
- Indítsuk el:

```
$ sudo -E python test1.py
mininet>
```

Mininet

- Elérhető csomópontok:

```
mininet> nodes
```

- Az s1 switchről infót kaphatunk
  - (brctl: ethernet bridge adminisztráció)

```
mininet> sh brctl show
```

  - Látszik, hogy nincs engedélyezve az STP
- A h1 h2 hostokon elindíthatunk egy-egy terminált:

```
mininet> xterm h1 h2
```

Mininet

- Itt lekérhetők az interface adatok, érdemes a mac címet megnézni!

```
# ifconfig
```

- Írassuk ki az ARP tábla aktuális tartalmát:

```
# arp
```

- Az s1 switch forwarding tábláját lekérdezhetjük a mininet konzolban:

```
mininet> sh brctl showmacs s1
```

Mininet

- Derítsük ki, hogy melyik interfésze van s1-nek a h2-vel összekötve (mininet konzol):

```
# mininet> links
h2-eth0<->s1-eth1 (OK OK)
...
```

- Figyeljük a forgalmat az „s1-eth1” interfészen! mininet konzolba írva:

```
mininet> s1 tcpdump -n -i s1-eth1
```

Mininet

- Pingetés xterm ablakból: `h2` termináljából: (a `h1` `h2` nevek itt nem használhatók!)

```
# ping 10.0.0.1
```

- Írassuk ki az ARP tábla aktuális tartalmát:

```
# arp
```

Mininet

- Közben látjuk a mininet konzolban, hogy mentek ARP üzenetek
- Pingetés mininet konzolból, pl.:

```
mininet> h1 ping h2
```

- Kilépés:

```
mininet> exit
```

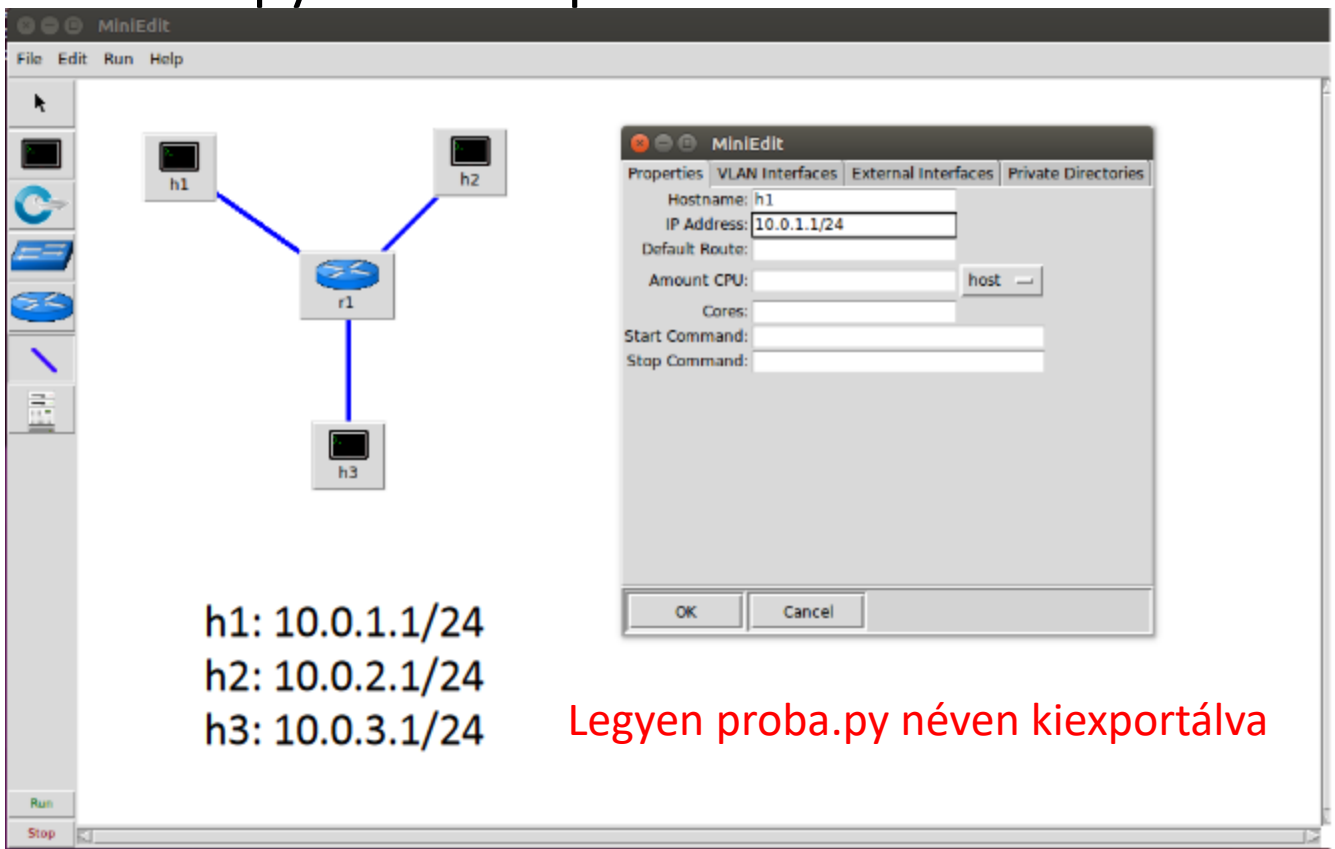
Mininet

- (Átváltás arra a könyvtárra, ahol a miniedit van:)

```
$ cd /home/mininet/mininet/examples/
python miniedit.py&
```

Mininet

- A következő példában létrehozunk miniedit-tel egy kis hálózatot:
  - A *File* menüben az „Export Level 2 Script”-tel lehet létrehozni python szkriptet



Legyen proba.py néven kiexportálva

Mininet

- Indítsuk el (feltéve, hogy „gyakorlat” könyvtárban van):

```
cd /home/mininet/gyakorlat

mininet@mininet-vm:~/gyakorlat$ sudo -E python proba.py
mininet>
```

- A h1 h2 h3 hosztokon és a routeren elindíthatunk egy-egy terminált:

```
mininet> xterm h1 h2 h3 r1
```

Mininet

- A `h1` terminálján próbáljuk ki a ping-et a h2 hoszthoz:

```
# ping 10.0.2.1
connect: Network is unreachable
```

- Router interfész beállítása:

```
mininet> net
r1 r1-eth0:h1-eth0 r1-eth1:h2-eth0 r1-eth2:h3-eth0
h3 h3-eth0:r1-eth2
h1 h1-eth0:r1-eth0
h2 h2-eth0:r1-eth1
```

- Az r1 terminálján adjunk IP címeket az `r1-eth0`, `r1-eth1`, `r1-eth2` interfészeknek:

```
# ip addr add 10.0.1.254/24 dev r1-eth0
# ip addr add 10.0.2.254/24 dev r1-eth1
# ip addr add 10.0.3.254/24 dev r1-eth2
```

Mininet

- A `h2` terminálján az alapértelmezett útvonalat adjuk meg a `10.0.2.254` lokális átjárón keresztül, amelyet az `h2-eth0` eszközön lehet elérni:

```
# ip route add default via 10.0.2.254 dev h2-eth0
```

Mininet

- A `h3` terminálján az alapértelmezett útvonalat adjuk meg a `10.0.3.254` lokális átjárón keresztül, amelyet az `h3-eth0` eszközön lehet elérni:

```
# ip route add default via 10.0.3.254 dev h3-eth0
```

Mininet

- A `h1` terminálján az alapértelmezett útvonalat adjuk meg a `10.0.1.254` lokális átjárón keresztül, amelyet az `h1-eth0` eszközön lehet elérni:

```
# ip route add default via 10.0.1.254 dev h1-eth0
```

- Ezután nézzük meg az IP routing táblát:

```
# route -n
```

- Most már működni fog a ping:

```
# ping 10.0.2.1
```

Mininet

- A h2 terminálját nyissuk meg!
- iptables szabályok kiíratása:

```
# iptables-save
```

- vagy

```
# iptables -L
```

- Ping tiltás szabály felvétele a INPUT lánc elejére:

```
# iptables -I INPUT -p icmp --icmp-type echo-request -j DROP
```

Mininet

- Ping tiltás szabály hozzáfűzése az OUTPUT lánc végére:

```
# iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP
```

- Próba:

```
# ping 10.0.1.1
```

- Ping tiltás szabály törlése:

```
# iptables -D OUTPUT -p icmp --icmp-type echo-request -j DROP
```

Mininet

- iptables port forwarding
- A `h3` terminálját nyissuk meg!
- h3 hoszton inditsunk el egy ssh deamon-t

```
# /usr/sbin/sshd
```

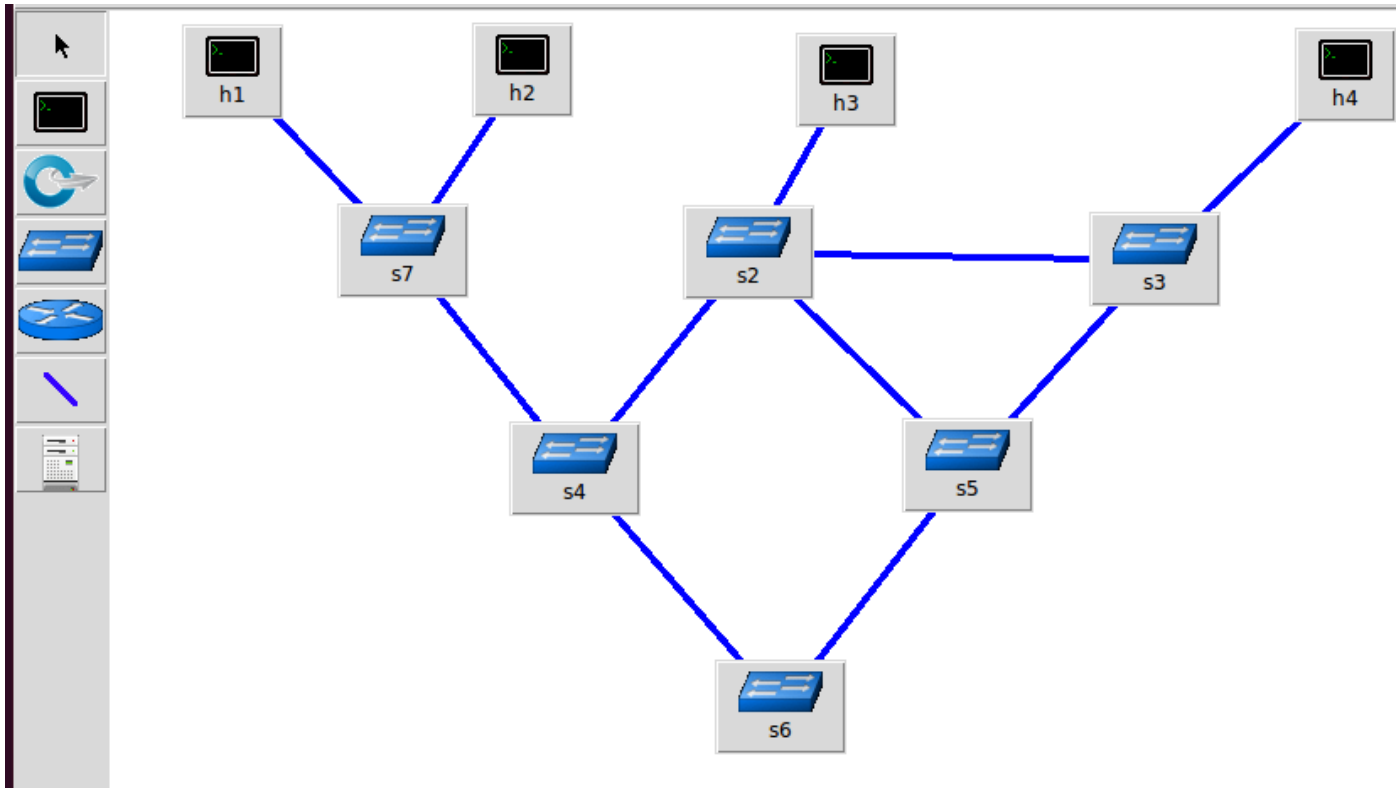
- Az `r1` terminálját nyissuk meg!
- Állítsuk be a r1-es routeren a forwarding szabályt:

```
# iptables -t nat -A PREROUTING -i r1-eth0 -p tcp -d 10.0.2.1 --dport 2222 -j DNAT \
--to-destination 10.0.3.1:22
```

- SSH-zunkbe h1-ről a h3-ra a port forwardinggal:

```
# ssh -p 2222 mininet@10.0.2.1
```

Mininet



- Indítsuk el:

```
mininet@mininet-vm:~/gyakorlat$ sudo -E python sw-topo.py
mininet>
```

Mininet

- Nézzük meg a switcheket a mininet konzolban:

```
mininet> sh brctl show
```

- STP mindenhol ki van kapcsolva!
- h1 és h2 szomszédok

```
mininet> h1 ping h2
```

- Azt tapasztaljuk, hogy nagy a késés és csak néhány csomag megy át
- h1 és h4 távol vannak egymástól

```
mininet> h1 ping h4
```

- Csak sikertelen próbálkozás lesz, semmi se megy át

Mininet

- `tcpdump`-pal érdekes jelenség látható:

```
mininet> sh tcpdump -n -i any
```

- Multicast üzenetek próbálják a hálózatot felderíteni
- Konklúzió: hurok van a hálózatban, nem igazán működik semmi
- Kilépés:

```
mininet> exit
```

Mininet

- Indítsuk el újra `--stp` kapcsolóval:

```
mininet@mininet-vm:~/gyakorlat$sudo -E python sw-topo.py --stp
mininet>
```

- bridge állapot:

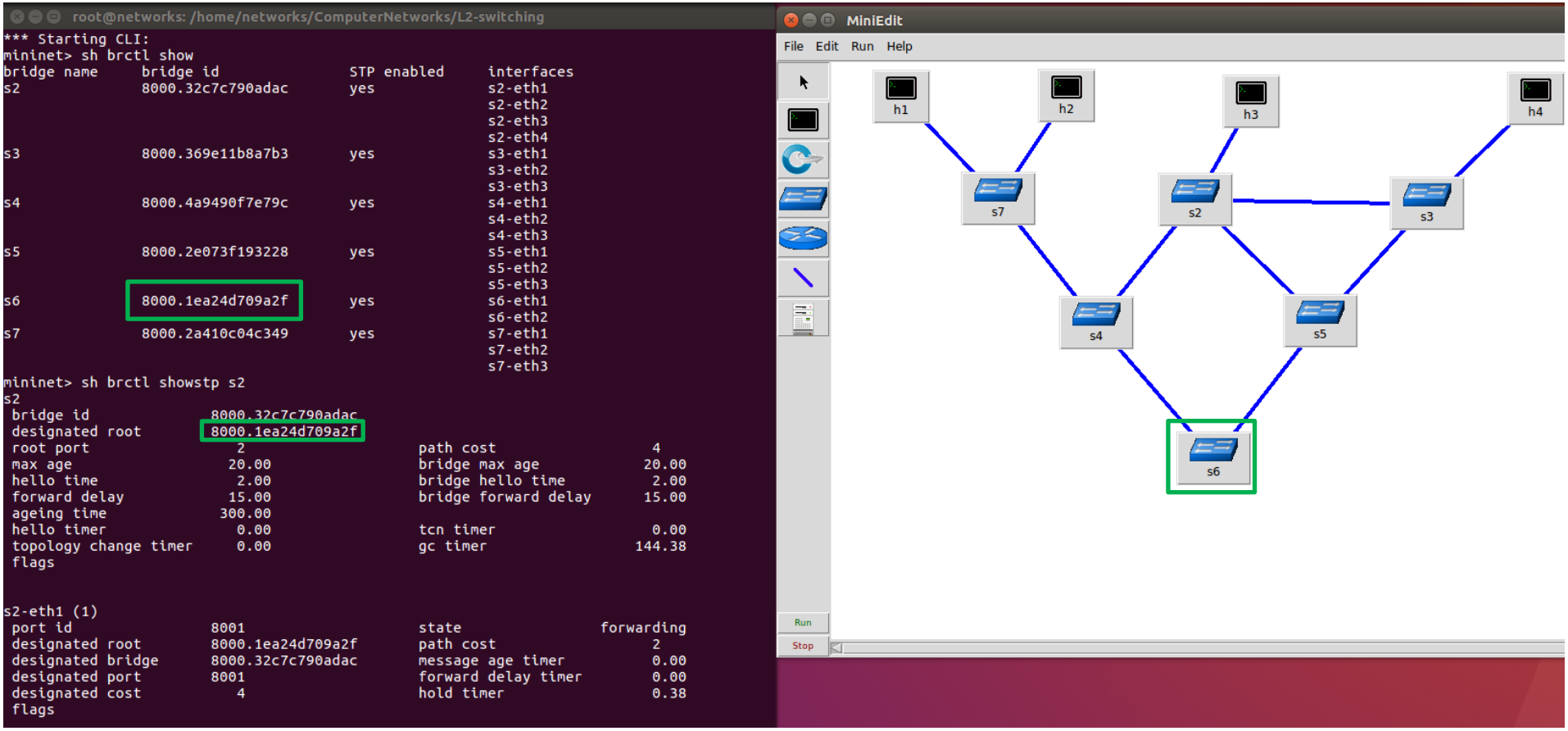
```
mininet> sh brctl show
```

- STP információ az `s2` switchhez:

```
mininet> sh brctl showstp s2
```

  - Nézzük meg mit ír ki: ki a designated root, ki a designated bridge, mely portok blokkoltak (a körök kiszűrésére)?

Mininet


root@networks: /home/networks/ComputerNetworks/L2-switching
*** Starting CLI:
mininet> sh brctl show
bridge name bridge id STP enabled interfaces
s2 8000.32c7c790adac yes s2-eth1
s2-eth2
s2-eth3
s2-eth4
s3 8000.369e11b8a7b3 yes s3-eth1
s3-eth2
s3-eth3
s4 8000.4a9490f7e79c yes s4-eth1
s4-eth2
s4-eth3
s5 8000.2e073f193228 yes s5-eth1
s5-eth2
s5-eth3
s6 8000.1ea24d709a2f yes s6-eth1
s6-eth2
s7 8000.2a410c04c349 yes s7-eth1
s7-eth2
s7-eth3
mininet> sh brctl showstp s2
s2
bridge id 8000.32c7c790adac
designated root 8000.1ea24d709a2f
root port 2 path cost 4
max age 20.00 bridge max age 20.00
hello time 2.00 bridge hello time 2.00
forward delay 15.00 bridge forward delay 15.00
ageing time 300.00
hello timer 0.00 tcn timer 0.00
topology change timer 0.00 gc timer 144.38
flags
s2-eth1 (1)
port id 8001 state forwarding
designated root 8000.1ea24d709a2f path cost 2
designated bridge 8000.32c7c790adac message age timer 0.00
designated port 8001 forward delay timer 0.00
designated cost 4 hold timer 0.38
flags
MiniEdit
File Edit Run Help
h1
h2
h3
h4
s7
s2
s3
s4
s5
s6
Run
Stop

Mininet

- Működik-e most a hálózat???

```
mininet> h1 ping h2
```

```
mininet> h1 ping h4
```

- és megy minden... érdemes még a tcpdumpot is futtatni:

```
mininet> s2 tcpdump -n -i any
```

- látjuk, ahogy az STP üzenetek mennek a szomszédok között.

Mininet – **source …**

- Ha kimentettük a mininet konzolban kiadandó parancsokat egy fájlba (***input.txt***) …:

```
r1 ip addr add …
…
h1 ip route add …
```

- … akkor utána be tudjuk tölteni a mininet konzolon keresztül

```
mininet> source input.txt
```

Házi feladat
mininet beadandó

- **A feladat mindenkinek egyedi, emiatt le kell tölteni a megfelelő topológia fájlt, az alábbi linken keresztül!**
- Feladat leírása és topológiafájl letöltés:

https://ggombos.web.elte.hu/halobeadando/6-mininet/

Házi feladat
mininet beadandó

- **Leadás**: A programot **zip** formátumban kell leadni! A .zip fájlban **EGY** darab ***input.txt*** fájl legyen! A fájlban parancsok szerepeljenek olyan formában ahogy a mininet console-n meg lehet adni.
- Példa parancsok:

```
h1 ping 10.0.0.2
r3 ifconfig
```

- A TMS tesztelni fogja a beadott házifeladatot!
- Beadási határidő: **TMS rendszerben**

VÉGE
KÖSZÖNÖM A FIGYELMET!