Számítógépes Hálózatok

10. gyakorlat

péntekieknek 8. gyakorlat

Zárthelyi időpont

- Az utolsó előtti gyakorlat idejében (nov. 30. – dec. 4. héten) zárthelyi írás lesz
- Az utolsó gyakorlaton (dec. 7. – dec. 11. héten) pótzárthelyi írás lesz

Mininet

- Indítsuk el a virtuális gépet!
  - Felhasználó/jelszó: `networks/networks`
- Ha nem kapott IP címet a gép (`ifconfig`), akkor futtassuk a `sudo dhclient` parancsot!
- Utána jegyezzük fel az IP címet (`ifconfig`)!
- Lépjünk ki (de nyilván ne állítsuk le a gépet)
- Indítsunk egy MobaXterm session-t

Mininet

- Belépés után mindig el kell végezni!!!

```
networks@networksELTE:~$ xauth list
networksELTE:10  MIT-MAGIC-COOKIE-1  <egy alfanumerikus karaktersorozat, jelöljük S-sel>
…
networks@networksELTE:~$ xauth add networksELTE/unix:10 MIT-MAGIC-COOKIE-1 S
networks@networksELTE:~$ sudo xhost +
```

- (Függetlenül attól, hogy már az xauth list kimenetében szerepel networksELTE/unix:10 vagy sem…)
- Ez amiatt kell, hogy utána a mininet konzolon keresztül is működjön az xterm parancs…

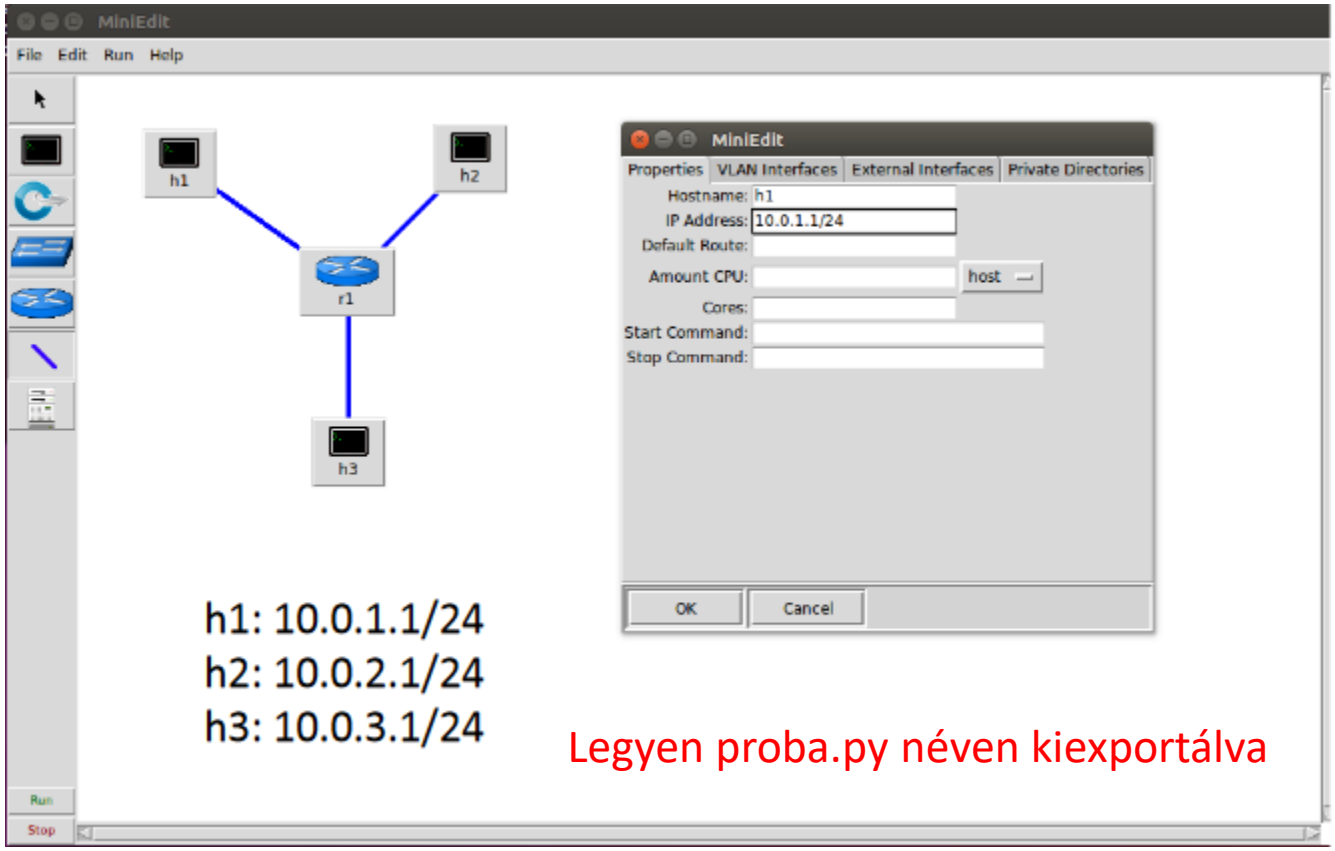
Mininet

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

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

Mininet

- A következő példában létrehozunk miniedit-tel egy kis hálózatot:



Legyen proba.py néven kiexportálva

Mininet

- Indítsuk el:

```
cd /home/networks/mininetScriptek/ComputerNetworks/L2-switching

networks@networksELTE:~/mininetScriptek/ComputerNetworks/L2-switching$sudo 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 networks@10.0.2.1
```

Mininet

- Indítsuk el a miniedit-et:

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

  - Nyissuk meg a `sw-topo.mn` fájlt
  - Hurkot tartalmaz!

- Indítsuk el:

```
networks@networksELTE:~/mininetScriptek/ComputerNetworks/L2-switching$sudo 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:

```
networks@networksELTE:~/mininetScriptek/ComputerNetworks/L2-switching$sudo python sw-topo.py --stp
mininet>
```

- bridge állapot:

```
mininet> sh brctl show
```

- STP információ az `s2` switchhez:

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

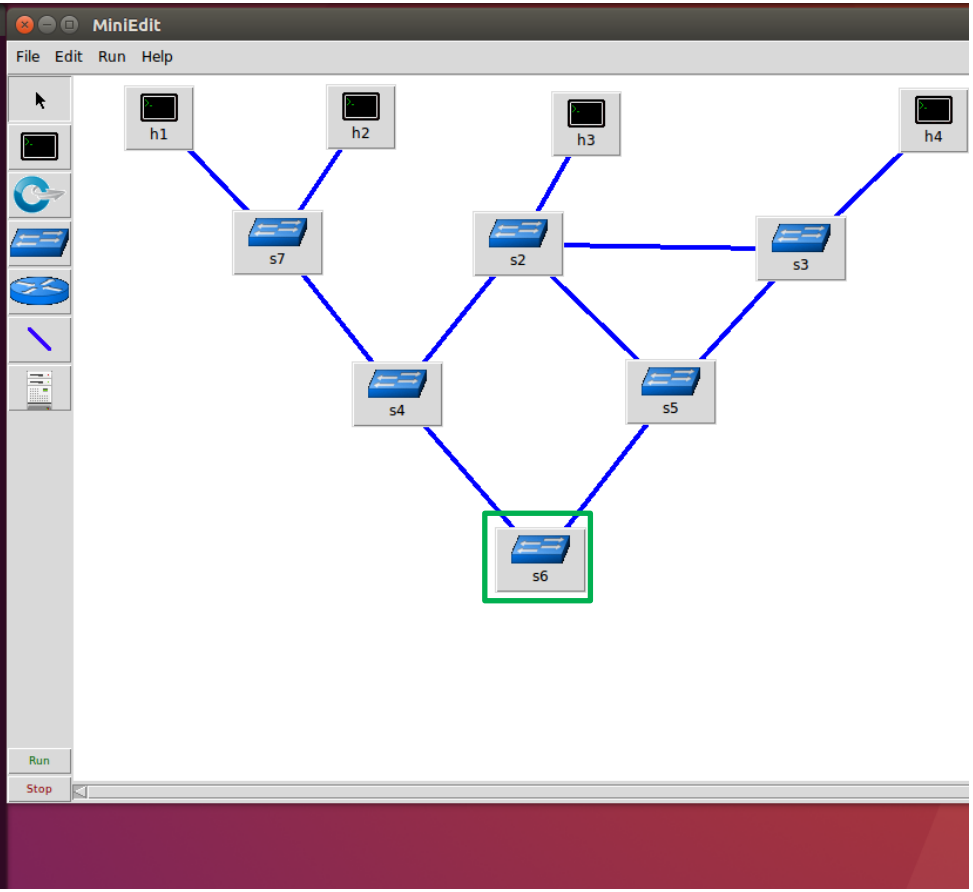
  - Nézzük meg mit ír ki: ki a designated root, ki a designated bridge, mely portok blokkoltak (a körök kiszűrésére)?

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
```



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.

VÉGE
KÖSZÖNÖM A FIGYELMET!