

# Számítógépes Hálózatok

Gombos Gergő

# Mininet

- Indítsuk el a miniedit-et:

```
root@networks:/home/networks# mininet/examples/miniedit&
```

- a *File* menüben meg tudjuk nyitni a .mn kiterjesztésű fájlokat
- Nyissuk meg a test.mn fájlt
- A *File* menüben az „Export Level 2 Script”-tel lehet létrehozni python szkriptet

# Mininet

- Nézzük meg a test1.py-t:

```
root@networks:/home/networks/ComputerNetworks/L2-switching# vim 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:

```
root@networks:/home/networks/ComputerNetworks/L2-switching# 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 lekérhetőek 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

- Figyeljük a forgalmat minden interfészen!  
mininet konzolba írva:

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

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

```
# ping 10.0.0.2
```

- Írassuk ki az ARP tábla aktuális tartalmát:

```
# arp
```

# Mininet

The screenshot displays a Mininet environment. At the top, a terminal window shows the command prompt `root@networks: /home/networks` and the execution of `sudo su`, followed by a password prompt and the message "Sorry, try again." Below the terminal is a MiniEdit window showing a network diagram. The diagram consists of a central router node labeled `r1` connected to three host nodes labeled `h1`, `h2`, and `h3`. To the right of the diagram is a configuration window for host `h1`. This window has tabs for "Properties", "VLAN Interfaces", "External Interfaces", and "Private Directories". The "Properties" tab is active, showing fields for "Hostname" (set to `h1`), "IP Address" (set to `10.0.1.1/24`), "Default Route" (empty), "Amount CPU" (empty), "Cores" (empty), "Start Command" (empty), and "Stop Command" (empty). There is a "host" button next to the "Amount CPU" field. At the bottom of the configuration window are "OK" and "Cancel" buttons. Below the network diagram, the following IP addresses are listed:

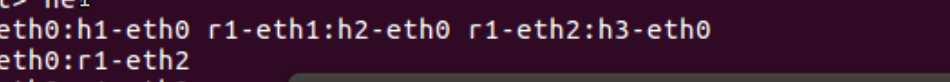
h1: 10.0.1.1/24  
h2: 10.0.2.1/24  
h3: 10.0.3.1/24

At the bottom left of the MiniEdit window, there are "Run" and "Stop" buttons.

# Mininet

- router interface beállítása

```
root@networks:/home/networks# ping 10.0.2.1
connect: Network is unreachable
```



The screenshot shows two terminal windows. The main terminal window on the left has a dark purple background and displays the following commands and output:

```
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
mininet>
```

The secondary terminal window on the right, titled "Node: r1", has a dark gray background and shows the following commands and output:

```
root@networks:/home/networks# ip addr add 10.0.1,254/24 dev r1-eth0
root@networks:/home/networks# ip addr add 10.0.2,254/24 dev r1-eth1
root@networks:/home/networks# ip addr add 10.0.3,254/24 dev r1-eth2
root@networks:/home/networks#
```

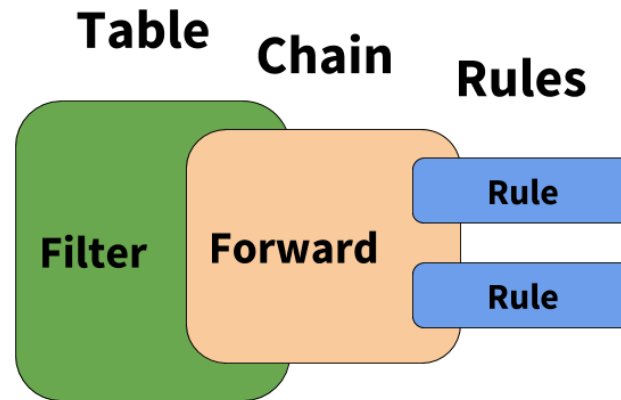
- hostok (h1,h2) default route beállítása:

```
root@networks:/home/networks# ip route add default via 10.0.2,254 dev h2-eth0  
root@networks:/home/networks#  
root@networks:/home/networks#  
root@networks:/home/networks#  
root@networks:/home/networks#  
root@networks:/home/networks#  
root@networks:/home/networks#  
root@networks:/home/networks#  
"Node: h1"  
root@networks:/home/networks# ip route add default via 10.0.1,254 dev h1-eth0  
root@networks:/home/networks# route -n  
Kernel IP routing table  
Destination      Gateway         Genmask          Flags Metric Ref    Use Iface  
0.0.0.0           10.0.1.254     0.0.0.0          UG        0       0       0   h1-eth0  
10.0.1.0          0.0.0.0        255.255.255.0    U         0       0       0   h1-eth0  
root@networks:/home/networks# ping 10.0.2.1  
PING 10.0.2.1 (10.0.2.1) 56(84) bytes of data:  
64 bytes from 10.0.2.1: icmp_seq=1 ttl=63 time=0.101 ms  
64 bytes from 10.0.2.1: icmp_seq=2 ttl=63 time=0.068 ms  
^C
```

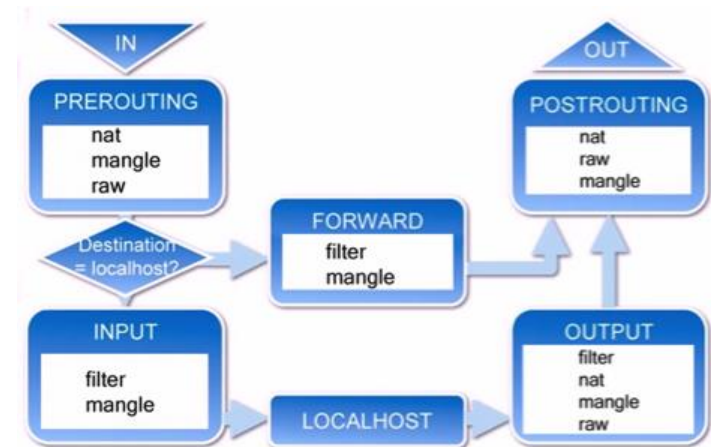
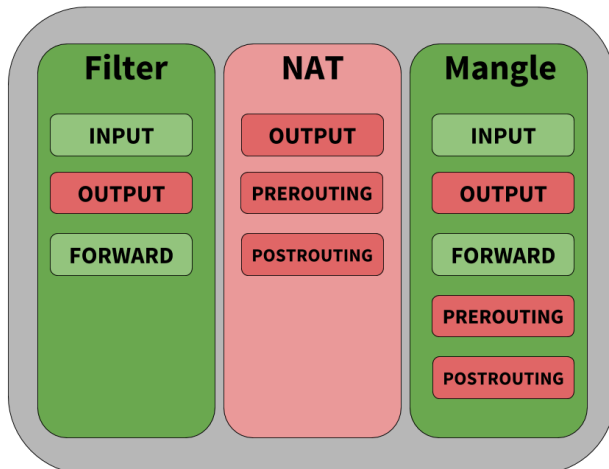


# Iptables

- <http://linux-training.be/networking/ch14.html>



**IPTables/IP6tables Table Support**



# Mininet

- Iptables szabályok kiírása:

```
# sudo iptables-save
```

- Ping tiltás szabály felvétele a INPUT lánc elejére:

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

- Ping tiltás szabály felvétele az OUTPUT lánc elejére :

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

- Ping tiltás szabály törlése:

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

# Mininet

- Iptables port forwarding:
- h3 node-on inditsunk el egy ssh deamont

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

- Állítsuk be a r1-es routern a forwarding szabályt:

```
# iptables -t nat -A PREROUTING -i h1-eth0 -p tcp -d 10.0.2.1 --dport 2222 -j DNAT --to-destination 10.0.3.1:22
```

- 

- SSH-zunk be h1-ről a h3-ra a port forwardinggal:

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

# Mininet

- Indítsuk el a miniedit-et:

```
root@networks:/home/networks# mininet/examples/miniedit&
```

- Nyissuk meg a `sw-topo.mn` fájlt

- Hurkot tartalmaz!

- Indítsuk el:

```
root@networks:/home/networks/ComputerNetworks/L2-switching# python sw-topo.py  
mininet>
```

# 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                8000.32c7c790adac  yes          s2-eth2
s2                8000.32c7c790adac  yes          s2-eth3
s3                8000.369e11b8a7b3  yes          s2-eth4
s3                8000.369e11b8a7b3  yes          s3-eth1
s3                8000.369e11b8a7b3  yes          s3-eth2
s4                8000.4a9490f7e79c  yes          s3-eth3
s4                8000.4a9490f7e79c  yes          s4-eth1
s4                8000.4a9490f7e79c  yes          s4-eth2
s5                8000.2e073f193228  yes          s4-eth3
s5                8000.2e073f193228  yes          s5-eth1
s5                8000.2e073f193228  yes          s5-eth2
s6                8000.1ea24d709a2f  yes          s5-eth3
s6                8000.1ea24d709a2f  yes          s6-eth1
s7                8000.2a410c04c349  yes          s6-eth2
s7                8000.2a410c04c349  yes          s7-eth1
s7                8000.2a410c04c349  yes          s7-eth2
s7                8000.2a410c04c349  yes          s7-eth3

mininet> sh brctl showstp s2
s2
bridge id        8000.32c7c790adac
designated root   8000.1ea24d709a2f
root port        2
max age          20.00
hello time       2.00
forward delay    15.00
ageing time      300.00
hello timer      0.00
topology change timer 0.00
flags

s2-eth1 (1)
port id          8001
designated root   8000.1ea24d709a2f
designated bridge 8000.32c7c790adac
designated port    8001
designated cost    4
flags

state            forwarding
path cost        2
message age timer 0.00
forward delay timer 0.00
hold timer       0.38
```

# 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

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

```
mininet> h1 ping h2
```

- Kilépés:

```
mininet> exit
```

# 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:

```
root@networks:/home/networks/ComputerNetworks/L2-switching# 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
```

bridge id	8000.32c7c790adac
designated root	8000.1ea24d709a2f
root port	2
max age	20.00
hello time	2.00
forward delay	15.00
ageing time	300.00
hello timer	0.00
topology change timer	0.00
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

Run

Stop

**VÉGE**