

Számítógépes Hálózatok

9. gyakorlat

MININET

MobaXterm

- Kényelmes ssh kliens, terminál
- <https://mobaxterm.mobatek.net/download-home-edition.html>
- GUI nélküli virtuális gép esetén szükséges!

GUI-s VM (ajánlott)

VM file: https://ikelte-my.sharepoint.com/:u:/g/personal/cgsmef_inf_elte_hu/EVqWAXnkaOtNkWmztIDRbgoBCh-ZYtMjHeLSGyRdx_1LHQ?e=cuISOD

VirtualBoxban megnyitva mindennek passzolnia kell. Igény esetén állítsd át a RAM, videómemória, használt processzormag mennyiséget.

GUI nélküli VM

Erőforrás spórolósabb megoldás, de felmerülhetnek nehézségek, X11 forwarding stb.

<http://mininet.org/download/#option-1-mininet-vm-installation-easy-recommended>

Az órán használt fájlokat megtaláljátok Teamsen.
(nem a default példákat fogjuk használni)

Mininet ismerkedés 1

- Egy Ubuntu virtuális gépünk van.
Felhasználó/jelszó:mininet
- Az első példánk a
/home/mininet/Desktop/L2-switching
mappában vannak
- Cd-zünk bele és nézzük meg milyen fájljaink
vannak!

```
mininet@mininet-vm:~$ cd Desktop/L2-switching/  
mininet@mininet-vm:~/Desktop/L2-switching$ ls  
sw-no-loop.py  sw-topo.mn  sw-topo.py  test1.mn  test1.py
```

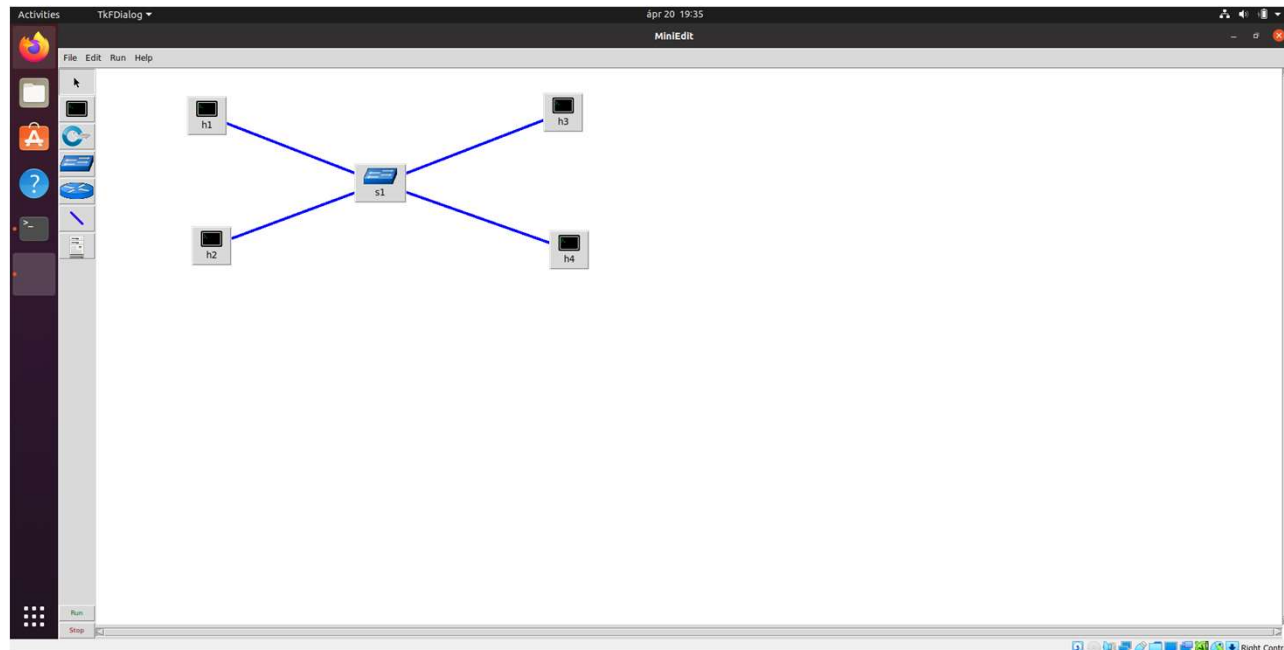
Mininet ismerkedés 2

- Két test1 nevű fájl van:
- test1.mn: Leírja a topológiát
- test1.py: futtatható script
- Indítsuk el a minieditet:

```
mininet@mininet-vm:~/Desktop/L2-switching$  
/home/mininet/mininet/examples/miniedit.py&
```

Mininet ismerkedés 3

- A file menü open pontjából megtudjuk nyitni a .mn kiterjesztésű fájlokat. Nyissuk is meg a test1-et.



Mininet ismerkedés 4

- A file menü Export Level2 Script pontjával létrehozhatjuk a hálózathoz tartozó python scriptet. Ezt most nem kell megtennünk, hiszen ott a .py fájl (plusz nem olyan switchet használna, mint amelyet szeretnénk).
- Vizsgáljuk meg a test1 scriptet (vim test1.py)

Python script

```
#!/usr/bin/python

from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.node import CPULimitedHost, Host, Node
from mininet.nodelib import LinuxBridge
from mininet.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call

def myNetwork():

    net = Mininet( topo=None,
                  build=False,
                  ipBase='10.0.0.0/8')

    info( '*** Adding controller\n' )
    info( '*** Add switches\n' )
    s1 = net.addSwitch('s1', cls=LinuxBridge, failMode='standalone')

    info( '*** Add hosts\n' )
    h3 = net.addHost('h3', cls=Host, ip='10.0.0.3', defaultRoute=None)
    h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)
    h4 = net.addHost('h4', cls=Host, ip='10.0.0.4', defaultRoute=None)
    h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)

    info( '*** Add links\n' )
    net.addLink(h2, s1)
    net.addLink(s1, h3)
    net.addLink(s1, h4)
    h1s1 = {'bw':10}
    net.addLink(h1, s1, cls=TCLink, **h1s1)

    info( '*** Starting network\n' )
    net.build()
    info( '*** Starting controllers\n' )
    for controller in net.controllers:
        controller.start()

    info( '*** Starting switches\n' )
    net.get('s1').start([1])

    info( '*** Post configure switches and hosts\n' )

    CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    myNetwork()
```

Python script

- Egy LinuxBridge-t hozunk létre ami a switch szerepét fogja betölteni a virtuális hálózatunkban.
- Hozzáadunk hosztokat is, privát IP címekkel
- Végül összekötjük ezeket a topológia alapján
- A h1-s1 kapcsolat sávszélessége: 10 Mbps(alapból nem limitált, a TCLink osztály azért kell, hogy limitálni tudjuk)

Linkek tulajdonságai

- Hálózati linkek tulajdonságai:
- késleltetés (delay), csomagvesztés (loss), sávszélesség (bw)
- A mininetes pythonscriptben:
- linkopts= {'bw':10, 'delay':'5ms', loss=10 }
- # bw: sávszélesség Mbps-ben
- # delay: késleltetés mértékegységgel: ms, s, us, stb.
- # loss: 0-100 közötti egész, csomagvesztés százalék
- ```
net.addLink(h1, r1, cls=TCLink, **linkopts)
```

# Elérhető eszközök

- Indítsuk el a hálózatunkat:

```
mininet@mininet-vm:~/Desktop/L2-switching$ sudo python
test1.py
```

- Létrehozott csomópontok:

```
mininet> nodes
```

- Csomópontokat összekötő linkek:

```
mininet> links
```

- A switch beállításai:

```
mininet> sh brctl show
```

```
networks@networks: ~/ComputerNetworks/L2-switching
*** Mininet must run as root.
networks@networks:~/ComputerNetworks/L2-switching$ sudo python test1.py
[sudo] password for networks:
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
(10.00Mbit) (10.00Mbit) *** Starting network
*** Configuring hosts
h3 h2 h4 h1
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> nodes
available nodes are:
h1 h2 h3 h4 s1
mininet> sh brctl show
bridge name bridge id STP enabled interfaces
s1 8000.124ca2c211d7 no s1-eth1
 s1-eth2
 s1-eth3
 s1-eth4
mininet>
```

# Hostok

- A h1 h2 hostokon elindíthatunk egy-egy terminált:

```
mininet> xterm h1 h2
```

- Itt lekérhetőek az interface adatok:

```
ip a
```

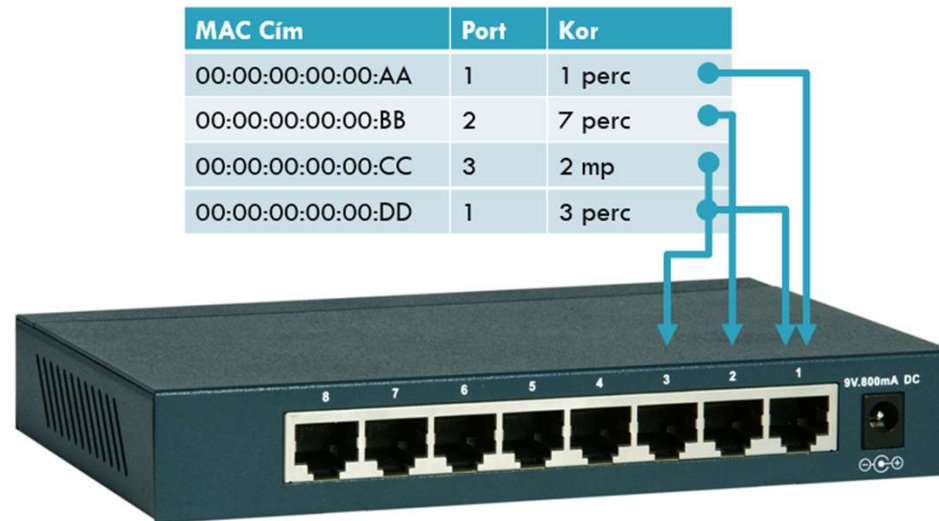
- Illetve a host ARP táblája is:

```
arp
```

- (Ez most üres lesz)

# Hogy fog ez működni?

- Mi az, hogy interfész/port?
- Mi az, hogy MAC cím?
- Honnan tudja a switch, hogy ki merre van?





# Hálózati forgalom generálás

- Az s1 switch forwarding tábláját lekérdezhetjük a mininetkonzolban:

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

- Figyeljük a forgalmat minden interfészen:

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

- Pingessük meg h1-ről h2-öt:

```
ping 10.0.0.1
```

- Nézzük meg újra az arp táblákat és a switch forwarding tábláját!

# Hálózati forgalom generálás

- Közvetlenül a mininetből is pingethetünk:

```
mininet> h1 ping h2
```

- Kilépés:

```
mininet> exit
```

# Sw-topo

- Nyissuk meg a minieditben az sw-topo.mn fájlt
- Nézzük meg alaposan a topológiát! Mit veszünk észre?

# STP

- Hurkokat tartalmaz

```
mininet> sh brctl show
```

- Az STP mindenhol ki van kapcsolva
- h1 és h2 szomszédok, működni kell a pingnek igaz?

```
mininet> h1 ping h2
```

- Nem, sok csomag elveszik és óriási a késés

# STP

- h1 és h4 távol vannak egymástól, ha ők pingelik egymást mi lesz?

- ```
mininet> h1 ping h4
```

- Csak sikertelen próbálkozás lesz
- Tcpcmdump-pal megfigyelhetjük, hogy broadcast üzenetek próbálják folyamatosan felderíteni a hálózatot:

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

- Valahogy fel kell oldanunk a logaikai hurkokat

STP

- Indítsuk újra az stp kapcsolóval:

```
# python sw-topo.py --stp
```

- Switch állapot:

```
mininet> sh brctl show
```

- STP információ az s2switchhez:

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

- Ki a designated root? Ki a designated bridge?
Mely portok blokkoltak?
- Designated root: a fa gyökere, designated bridge:
a roothoz „vezető” switch

VÉGE
KÖSZÖNÖM A FIGYELMET!