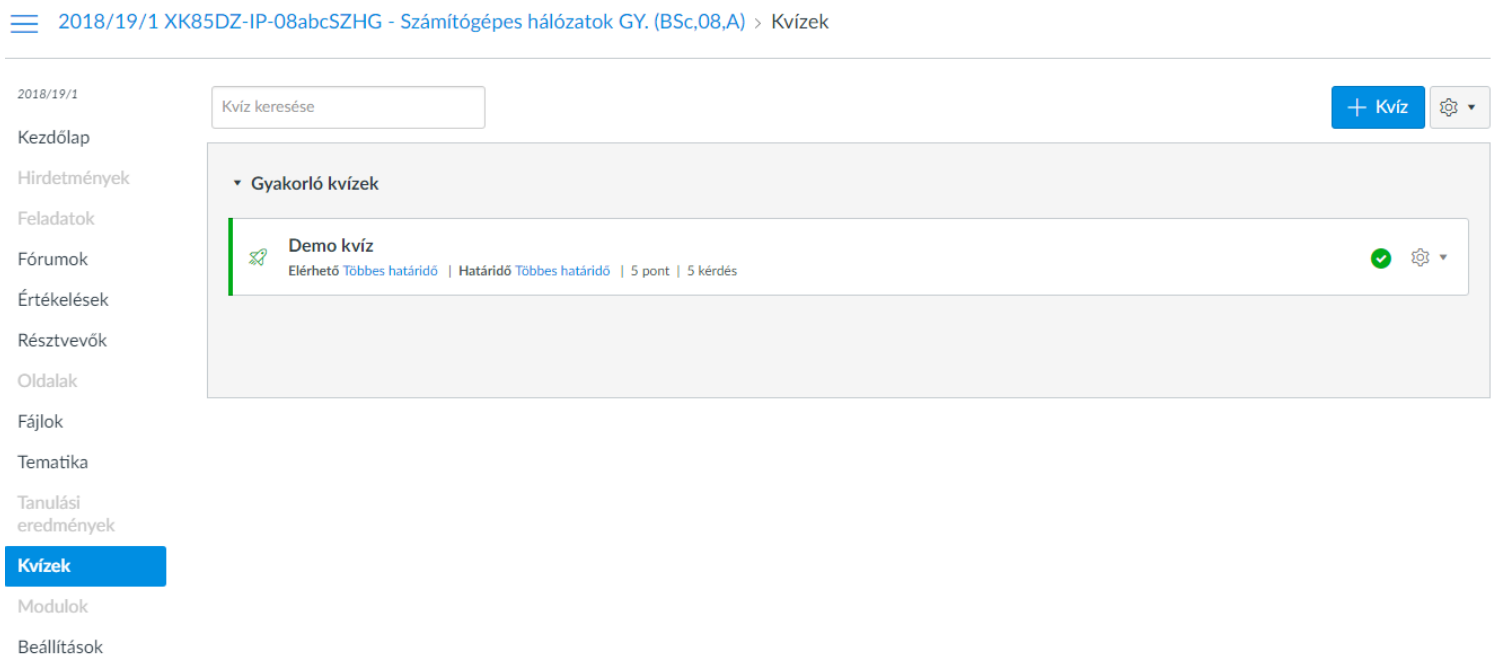
Számítógépes Hálózatok

9. gyakorlat

Óra eleji kisZH

- Elérés:
  - https://canvas.elte.hu



Gyakorlat tematika

- Mininet

Mininet

- Töltsük le a képfájlt: http://bit.ly/2i27mWW
- Nyissuk meg a Hyper-V kezelőjét
- Jobb klikk a számítógépen:
  - Új virtuális gép létrehozása
  - 1. generációs
  - Hálózathoz: NAT switch
  - Meglévő virtuális merevlemez használata: az átmásolt fájlt kiválasztása
- Duplaklikk → indítás

Mininet

- Alternatív megoldás VirtualBox:
- http://bit.ly/2BzivmX (ova fájl)
- Fájl → Gép importálás…

Mininet

- Ubuntu 14 op. rendszer, felhasználó/jelszó: `networks/networks`
- Indítsunk egy terminált, váltsunk `root`-ra:

```
networks@networks:~$ sudo su
```

- Listázzuk az alábbi könyvtárt:

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

  - test1 topológia két fájlból áll:
  - `test1.mn`: meg lehet jeleníteni a miniedit segítségével
  - `test1.py`: egyből elindítja a hálózat emulátort

Mininet

- Indítsuk el a miniedit-et:

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

  - 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 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

- 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

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

```
mininet> h1 ping h2
```

- Kilépés:

```
mininet> exit
```

Mininet

- Indítsuk el a miniedit-et:

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

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

- Indítsuk el:

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

```
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
```

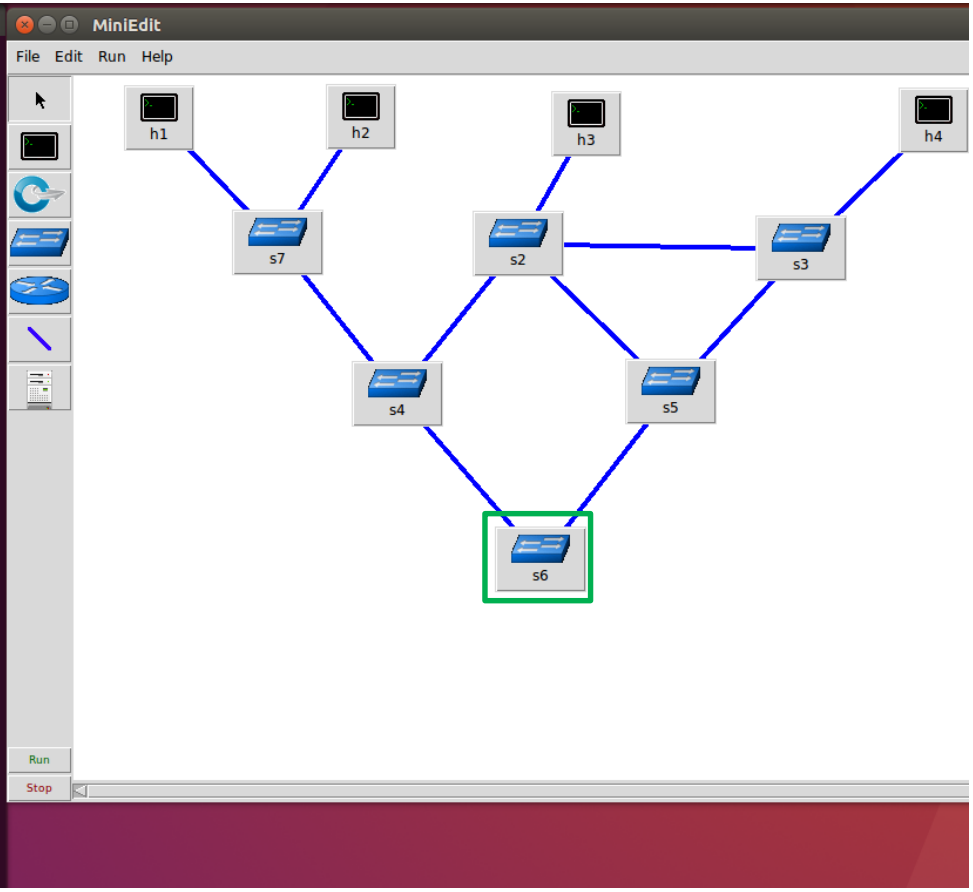
  - Nézzük meg mit ír ki: ki a designated root, ki a designated bridge, mely portok blokkoltak (a körök kiszűrésére)?

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
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.

Mininet

- A következő példában három hoszt lesz összekötve: h1 – h2 – h3, és h2 router-ként lesz konfigurálva
- Ez alapján készült: http://csie.nqu.edu.tw/smallko/sdn/mininet-operations.htm

A `test_router.py` szkript tartalma:

```
#!/usr/bin/python
from mininet.net import Mininet
from mininet.cli import CLI
from mininet.link import Link, TCLink, Intf
from mininet.log import setLogLevel, info
from mininet.node import CPULimitedHost, Host, Node

def myNetwork():
        net = Mininet( topo=None,
                        build=False,
                        ipBase='10.0.0.0/8',
                        link=TCLink)
        info( '*** Adding controller\n' )
        info( '*** Add switches\n')
        info( '*** Add hosts\n')
        h3 = net.addHost('h3', cls=Host, ip='10.0.0.3', defaultRoute=None)
        h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)
        h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)
        info( '*** Add links\n')
        Link(h1, h2)
        Link(h2,h3,intfName1='h2-eth1')
        info( '*** Starting network\n')
        net.build()
        info( '*** Starting controllers\n')
        for controller in net.controllers:
                        controller.start()
        info( '*** Starting switches\n')
        info( '*** Post configure switches and hosts\n')
        CLI(net)
        net.stop()

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

Mininet

- Indítsuk el:

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

- Nem működik a ping:

```
mininet> h1 ping h3
```

- A `h1` `h2` `h3` hostokon elindíthatunk egy-egy terminált:

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

Mininet

- A `h2` terminálján írjuk felül az eredeti IP címét a `h2-eth0` interfésznek:

```
# ifconfig h2-eth0 192.168.10.1 netmask 255.255.255.0
```

- Adjunk IP címet a `h2-eth1` interfésznek:

```
# ifconfig h2-eth1 192.168.20.1 netmask 255.255.255.0
```

- Engedélyezzük az IP forwarding-ot:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Mininet

- A `h1` terminálján töröljük az eredeti IP címet:

```
# ifconfig h1-eth0 0
```

- Adjunk IP címet a `h1-eth0` interfésznek:

```
# ip address add 192.168.10.2/24 dev h1-eth0
```

- Az alapértelmezett útvonalat adjuk meg a `192.168.10.1` lokális átjárón keresztül, amelyet az `h1-eth0` eszközön lehet elérni:

```
# ip route add default via 192.168.10.1 dev h1-eth0
```

Mininet

- A `h3` termináljában töröljük az eredeti IP címet:

```
# ifconfig h3-eth0 0
```

- Adjunk IP címet a `h3-eth0` interfésznek:

```
# ip address add 192.168.20.2/24 dev h3-eth0
```

- Az alapértelmezett útvonalat adjuk meg a `192.168.20.1` lokális átjárón keresztül, amelyet az `h3-eth0` eszközön lehet elérni:

```
# ip route add default via 192.168.20.1 dev h3-eth0
```

Mininet

- Ezután nézzük meg az IP routing táblát terminálokban:

```
# route -n
```

- Most már működni fog a ping:

```
mininet> h1 ping h3
```

- Kilépés:

```
mininet> exit
```

VÉGE