

Telekommunikációs Hálózatok

11. gyakorlat

MININET

Előfeltétel: *Mininet beállítás.pdf* diasoron végigmenni!

Mininet – következő indítás (ha már egyszer be lett állítva)

- A VM indításakor be kell lépni és az alábbi parancsot kiadni:

```
mininet> sudo dhclient
```

- Utána ellenőrizni, hogy milyen privát IP címet kapott:

- (Nagy valószínűség szerint ez ugyanaz, mint korábban, de ha mégsem, akkor sajnos MobaXterm-nél a session-nél módosítani kell a "Remote host"-nál ugyanarra az IP címre.)

```
mininet> ifconfig
```

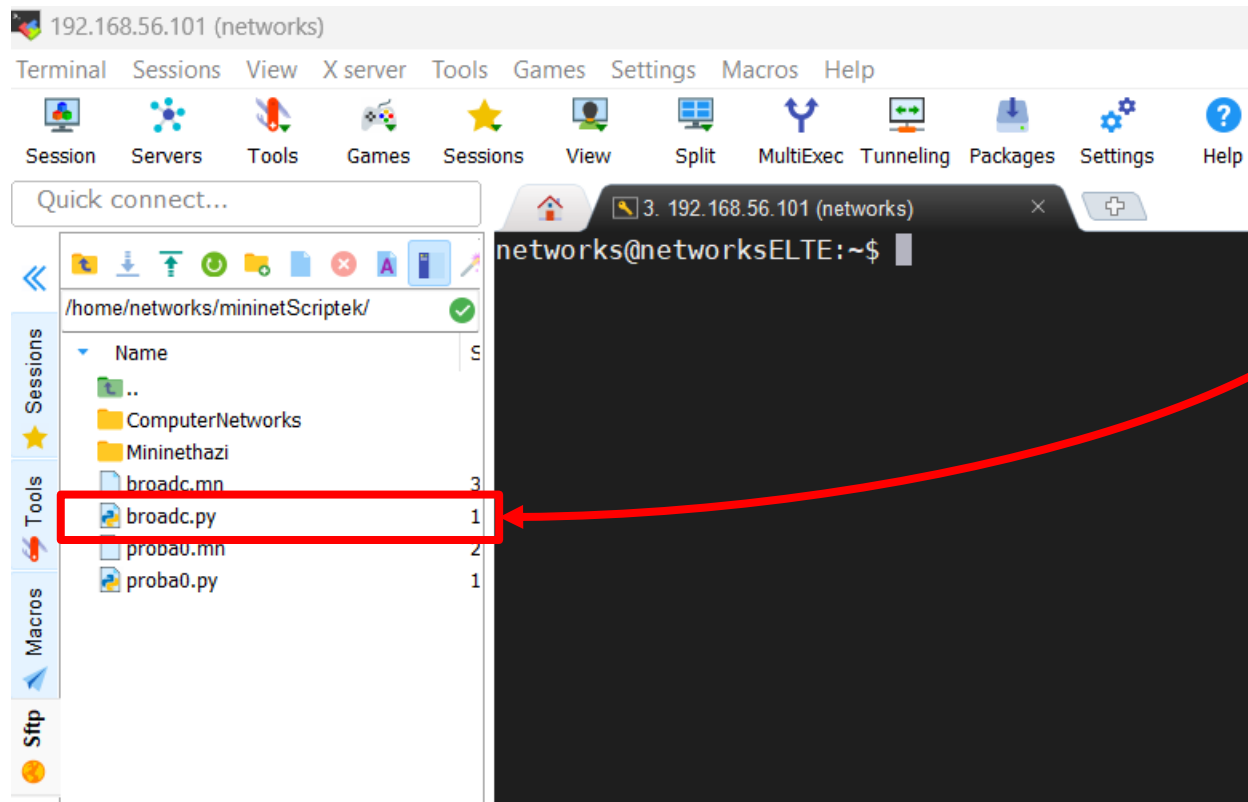
- Ezután ki lehet "exit" paranccsal lépni a VM-ből, de fontos, hogy nem szabad lezárni a gépet, hanem a MobaXterm-nél el kell indítani a megfelelő session-t. Ezután az alábbiakat kell kiadni:

```
mininet> xauth list
```

- Ha itt a "networksELTE/unix:10..." és "networksELTE:10..." soroknál (vagy "networksELTE/unix:11..." és "networksELTE:11..." soroknál stb.) nem egyezik az alfanumerikus karaktersorozat, akkor újra ki kell adni az "xauth add..." parancsot.

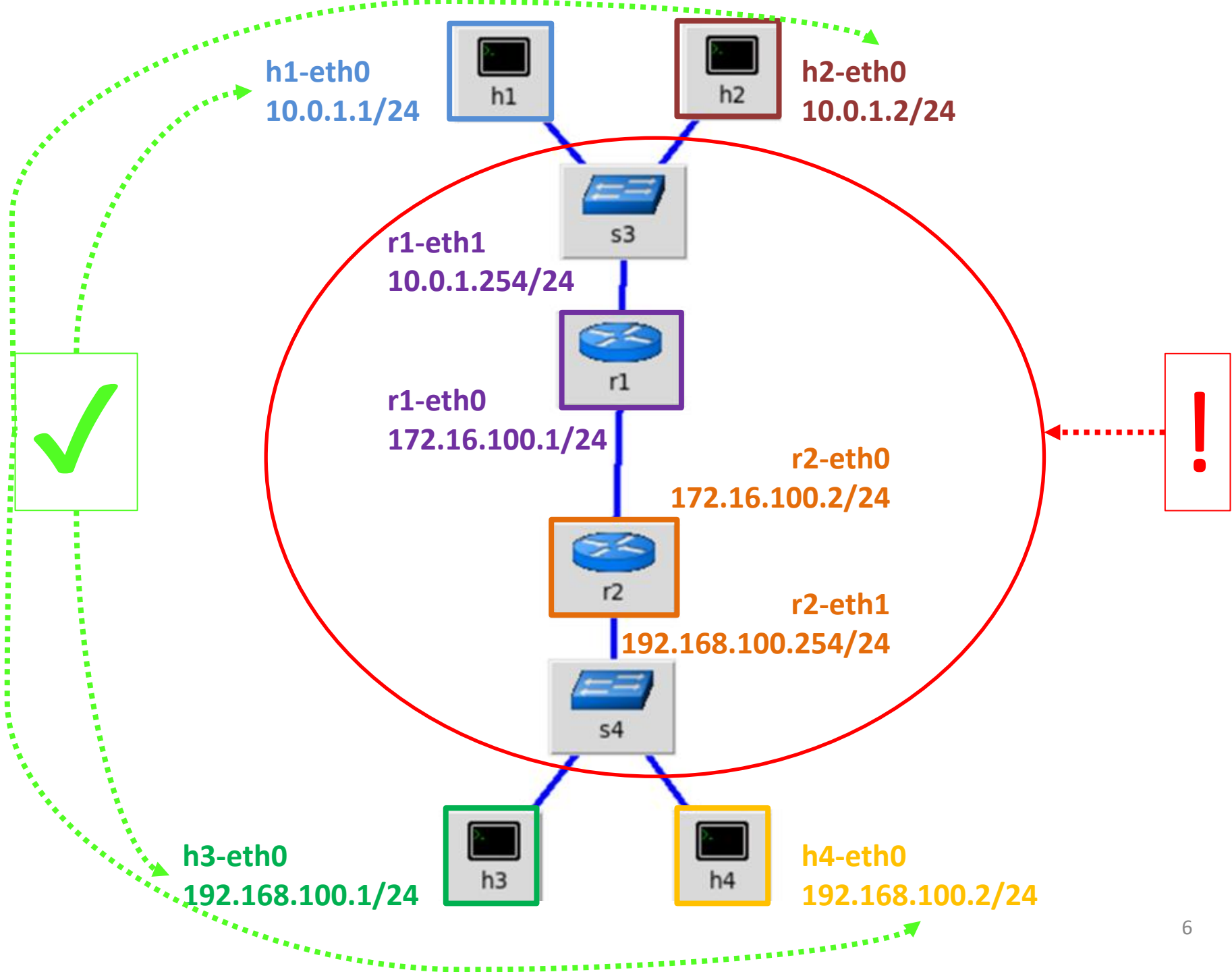
Feladat 1

- Belépés után másoljuk be a MobaXTerm-be a ***broadc.py*** szkriptet (oda kell húzni a fájlt):



Feladat 1

- A következő dián látható egy ábra a szkript alapjául szolgáló topológiáról
- A h1-eth0, h2-eth0, h3-eth0 és h4-eth0 interfészekhez már beállítottuk az IP címeket
- Állítsuk be az r1-eth0, r1-eth1, r2-eth0 és r2-eth1 interfészekhez kapcsolódó IP címeket az ábra alapján!
- Állítsuk be az alapértelmezett útvonalakat a hosztoknál és router-eknél:
 - h1 és h2 hosztok esetében a 10.0.1.254 lokális átjárón keresztül,
 - h3 és h4 hosztok esetében a 192.168.100.254 lokális átjárón keresztül,
 - r1 router esetében a 172.16.100.2 lokális átjárón keresztül, amelyet az r1-eth0 eszközön lehet elérni,
 - r2 router esetében a 172.16.100.1 lokális átjárón keresztül, amelyet az r2-eth0 eszközön lehet elérni!
- Ellenőrizzük le, hogy a ping működik-e h1 és h4 között!



Mininet

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

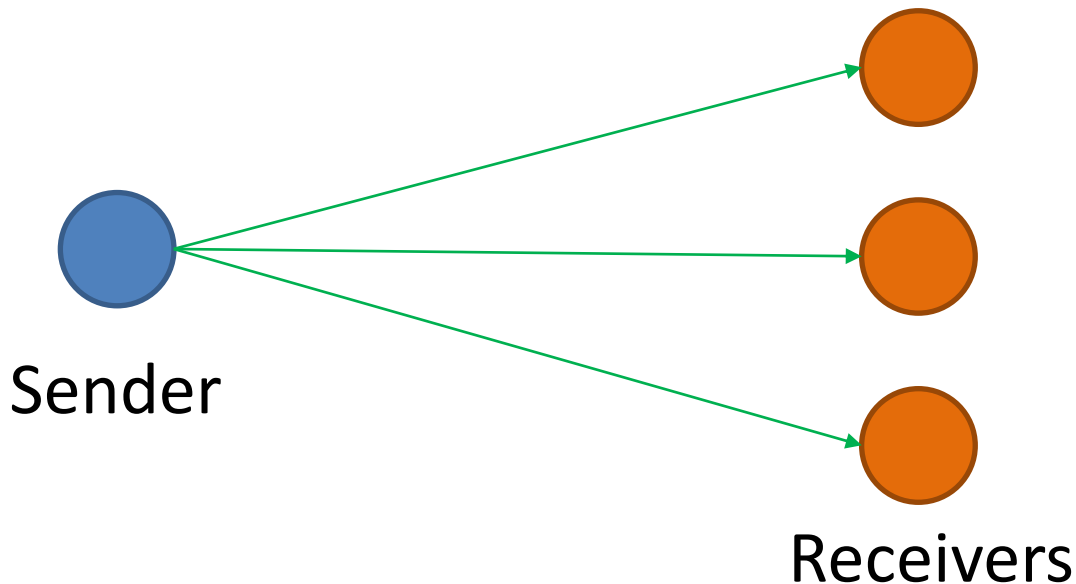
```
r1 ip addr add ...  
...  
h1 ip route add ...
```

- ... akkor utána be tudjuk tölteni a mininet konzolon keresztül

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

BROADCAST

Broadcast



- A broadcast is egy időben több végpontnak is tudja szállítani az üzenetet → jobb hatékonyság

Broadcast

- A broadcast üzenetek küldésénél is **UDP-t** használunk
 - (a TCP végpontok közötti kommunikációs csatornát igényel)
- Egy IPv4 cím van lefoglalva a broadcast-ra:
 - egy adott alhálózatnál a csupa 1-es bitből álló hoszt azonosító
 - Pl. broadcast IP cím a h1, h2 és r1 alhálózatnál: 10.0.1.255

Broadcast – python3 socket

- A broadcast üzenetekhez a **setsockopt** függvény segítségével be kell állítani az alábbi (a küldő oldalon mindenképpen):

```
sock = socket.socket(type=socket.SOCK_DGRAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
```

- A küldő az alhálózatnak megfelelő broadcast IP címre fogja küldeni a csomagját egy olyan port-ra (pl. 5005), amelyhez a fogadók előzetesen hozzákötötték (**bind** fv.) magukat
- Tehát fontos, hogy a fogadók socket-jénél az alábbi legyen (feltéve, hogy a port 5005):

```
sock.bind('10.0.1.255', 5005)
```

Feladat 2

- Készítsünk egy broadcast küldő és fogadó alkalmazást és próbáljuk is ki az előző mininet-es hálózaton, pontosabban annak h1, h2 és r1 alhálózatán!
- Egyszerű „Hello world!” legyen az üzenetszórásnál
- Lehet úgy is, hogy az elkészített szkripteket egy szövegszerkesztőben vagy fejlesztői környezetben hozzuk létre a „gazda” gépünkön, majd bemásoljuk a MobaXTerm-be
- A mininet-es hálózatot elindítva és „xterm” paranccsal a h1, h2 és r1-en terminálokat nyitva a „python3 feladat3_sender.py” és „python3 feladat3_receiver.py” parancsokkal tudjuk futtatni

```
"Node: r1"@networksELTE
root@networksELTE:~/mininetScriptek# ls
broadc.mn  ComputerNetworks  feladat3_sender.py  proba0.mn
broadc.py  feladat3_receiver.py  Mininethazi  proba0.py
root@networksELTE:~/mininetScriptek# python3 feladat3_sender.py
root@networksELTE:~/mininetScriptek#

"Node: h1"@networksELTE
root@networksELTE:~/mininetScriptek# python3 feladat3_receiver.py
b'Hello world' ('10.0.1.254', 56945)
root@networksELTE:~/mininetScriptek#

"Node: h2"@networksELTE
root@networksELTE:~/mininetScriptek# python3 feladat3_receiver.py
b'Hello world' ('10.0.1.254', 56945)
root@networksELTE:~/mininetScriptek#
```

Feladat 3: Chat UDP-vel, broadcast-tal

- Készítsünk egy chat alkalmazást, amelynél nincsen szerver, hanem a kliensek broadcast üzenetküldéssel kommunikálnak egymással UDP protokollt használva.
- Tehát a kliensek üzenetszórással küldik az adatot a broadcast IP címre:
[<név>] <üzenet> ; pl. [Józsi] Kék az ég!
- A kliensek a szervertől jövő üzeneteket kiírják a képernyőre.
- A mininet-es hálózatot elindítva és „xterm” paranccsal a h1, h2 és r1-en terminálokat nyitva, pl. „python3 feladat4.py h1” paranccsal tudjuk futtatni.

<pre> ❏ "Node: h1"@networksELTE root@networksELTE:~/mininetScriptek# python3 feladat4.py h1 <h1> [r1] Hello <h1> [h2] Hi <h1> Exit <h1> ^C Close the client root@networksELTE:~/mininetScriptek# █ </pre>	<pre> ❏ "Node: h2"@networksELTE root@networksELTE:~/mininetScriptek# python3 feladat4.py h2 <h2> [r1] Hello <h2> Hi <h2> [h1] Exit <h2> [h1] is LOGOUT <h2> [r1] Bye <h2> [r1] is LOGOUT <h2> ^C Close the client root@networksELTE:~/mininetScriptek# □ </pre>
<pre> ❏ "Node: r1"@networksELTE root@networksELTE:~/mininetScriptek# python3 feladat4.py r1 <r1> Hello <r1> [h2] Hi <r1> [h1] Exit <r1> [h1] is LOGOUT <r1> Bye <r1> ^C Close the client root@networksELTE:~/mininetScriptek# █ </pre>	

VÉGE
KÖSZÖNÖM A FIGYELMET!