

Telekommunikációs Hálózatok

12. 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.

Mininet – source ...

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

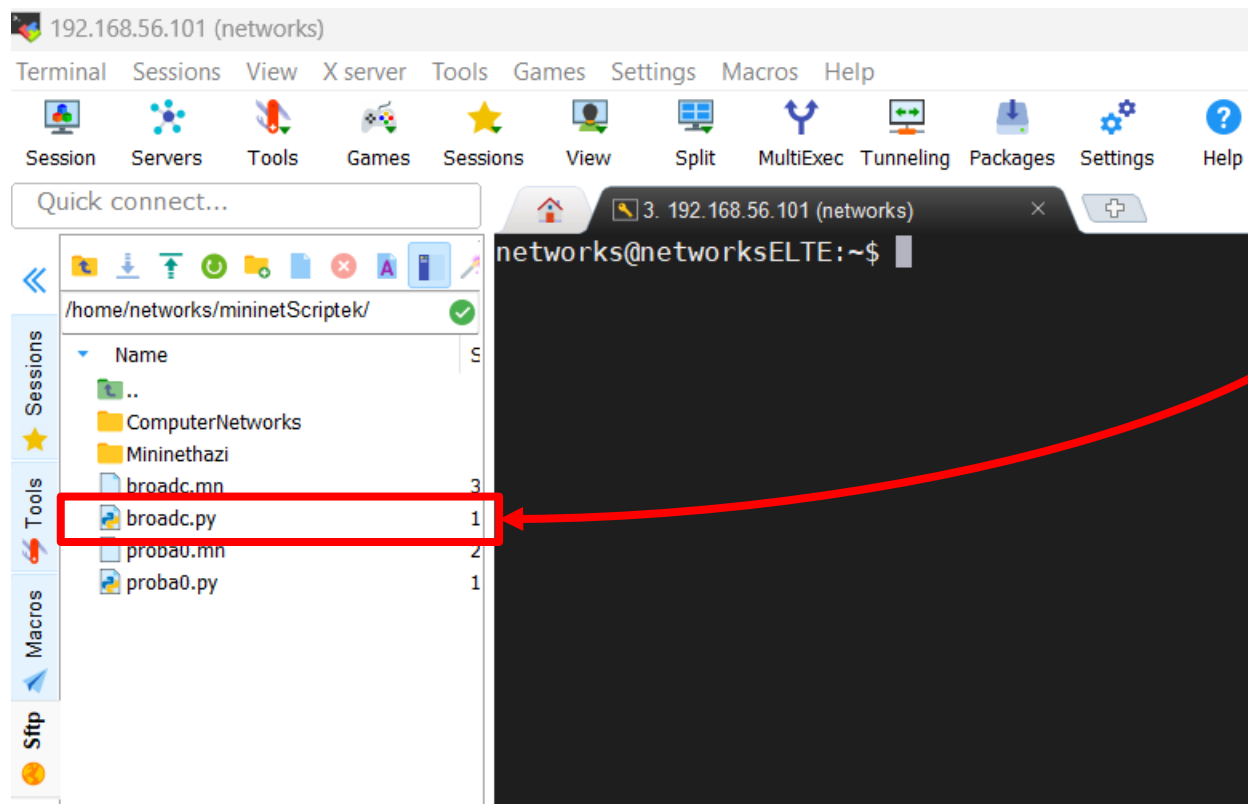
```
r1 ip addr add ...  
...  
h1 ip route add ...
```

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

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

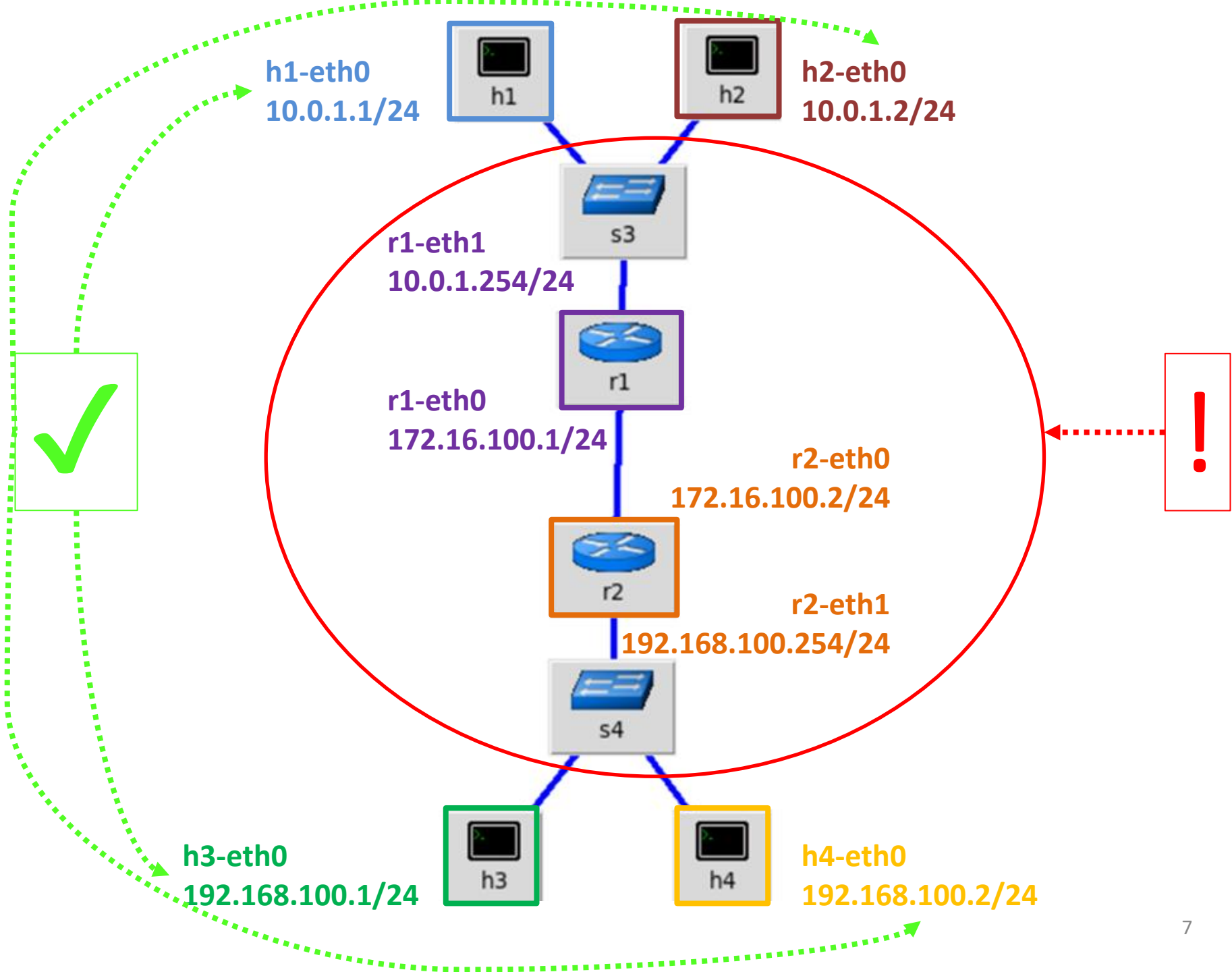
Korábbi Feladat 1

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



Korábbi 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!



Korábbi 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
broadcast.mn  ComputerNetworks  feladat3_sender.py  proba0.mn
broadcast.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# █
```

Korábbi 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>	

Feladat 1

- Másoljuk be a **db_server.py** és **db_client.py** szkripteket a VM-be!
- Az előbbi mininet-es hálózatot elindítva és „xterm” paranccsal a h1, h3 terminálokat nyissunk!
- A következő alfeladatok lesznek (következő dia)

Feladat 1

- a) Állítsunk be az r1 router-en egy olyan forwarding szabályt, amely a 172.16.100.1 IP címhez és 20000-es porthoz jövő TCP csomagokat a 10.0.1.1 IP című hoszthoz küldi a 10000-es portjához!

– Próbáljuk ki! A h1 termináljában:

```
# python3 db_server.py 10.0.1.1 10000  
Incoming query: SELECT * FROM DOGS WHERE age >= 5
```

– A h3 termináljában:

```
# python3 db_client.py 172.16.100.1 20000 192.168.100.1 10001  
query: SELECT * FROM DOGS WHERE age >= 5  
( 'Bailey', 5)  
( 'Coco', 10)
```

Feladat 1

- b) A h1 hoszton tiltsuk le a 192.168.100.0/24 alhálózatról/-ra bejövő/kimenő TCP csomagokat!
- c) Állítsunk be egy olyan szabályt az r1 router-en, amely a tőle kimenő, 10.0.1.1 IP című géphez, annak 10000-es portjához továbbküldött csomagoknál átállítja a forrás IP címet és portszámot 172.16.100.1:20000 – re.

VÉGE
KÖSZÖNÖM A FIGYELMET!