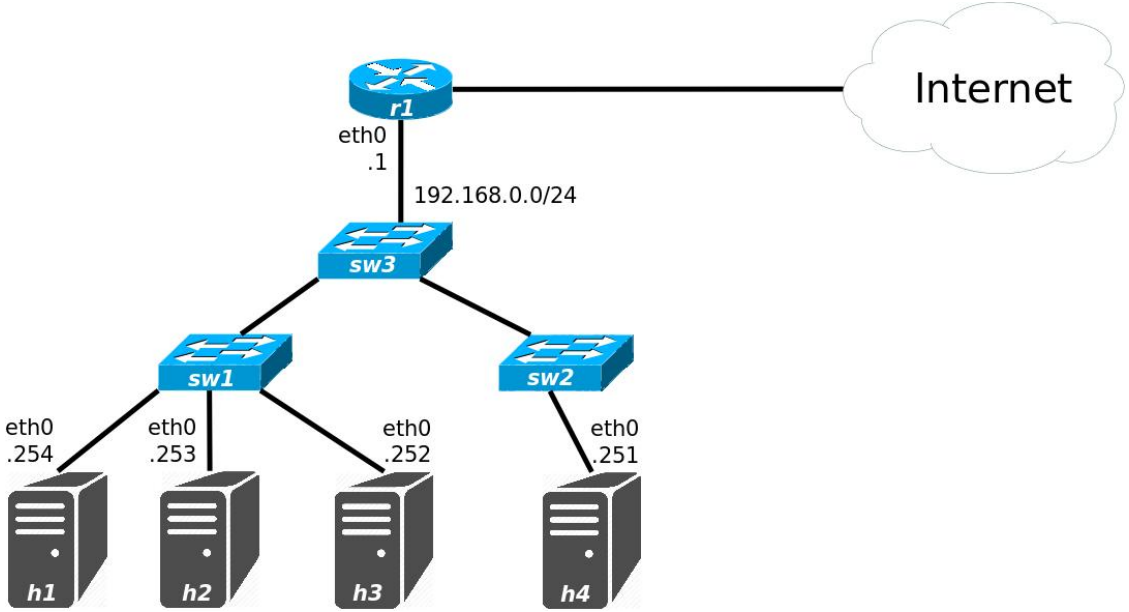
Hálózatok építése és üzemeltetése

Troubleshooting gyakorlat

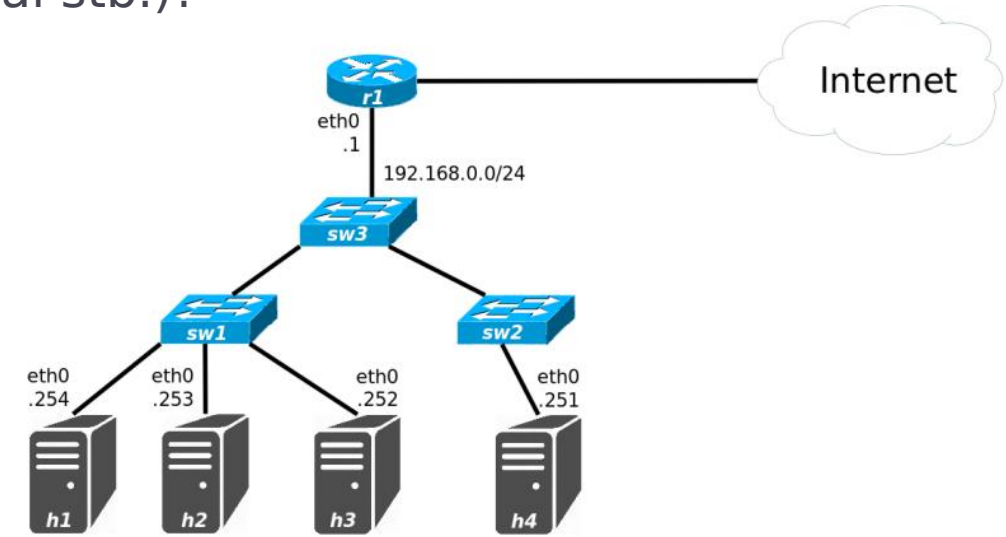
1. feladat

- Az alábbi hálózatban hiba lép fel, melynek eredményeként legalább egy hoszt nem éri el az Internetet.



1. feladat

- Milyen hibák léphetnek fel?
- Mi a legjobb módja a hiba felderítésének? Milyen sorrendben végezne teszteket a hálózat egyes elemein (hosztok, switchek, router, tűzfal stb.)?



1. feladat – megoldás

- Milyen hibák léphetnek fel?
  - Tápellátás; link/interfész hibák a hosztokon, switcheken, routeren; rosszul konfigurált hosztok: rossz IP cím, netmask, gateway; rosszul konfigurált switch portok; rosszul konfigurált router: nincs route az Internet felé, rosszul beállított NAT, tűzfal; DNS hiba stb.
- Milyen sorrendben végezne teszteket a hálózat egyes elemein (hosztok, switchek, router, tűzfal stb.)?
  - Az egyik hosztról kiindulva feltérképezzük, hogy meddig vagyunk képesek elérni a hálózatot: kapcsolatfelvétel a routerrel, más hosztokkal a hálózaton. A router Internet elérésének tesztelése: kapcsolat felvétele egy távoli géppel, kapcsolat felvétele egy távoli webszerverrel.
- Mi a legjobb módja a hiba felderítésének?

Hibaelhárítás

- Dokumentáljunk
  - Egy jól dokumentált hálózatban sokkal könnyebb hibát keresni
- Gyűjtsünk információt és azonosítsuk a tüneteket
  - Szűrjük ki azokat, amelyeknek közük lehet a hibához
  - Szükséges a normális viselkedés ismerete
  - Ha nem személyesen tapasztaljuk a hibát, próbáljuk meg reprodukálni
- Ismerjük meg a problémát
  - A gyűjtött adatok alapján keressünk okokat, melyek az adott tüneteket okozhatják
- Azonosítsuk azokat a rendszerelemeket, melyek részt vehetnek a probléma kialakulásában

Hibaelhárítás

- Állítsunk fel tesztelhető hipotéziseket az eddig begyűjtött ismeretekre alapozva
  - Állapítsuk meg, mely tesztekkel lehet ezeket a leghatékonyabban elkülöníteni egymástól
- Válasszunk és alkalmazzunk teszteket
  - Szempontok: erőforrásigény, komplexitás és információtartalom
  - Adott esetben egy egyszerű teszt jelentős információval szolgálhat, míg a komplex teszt nem feltétlenül hoz a bonyolultságával arányos többlet információt
- Értékeljük az eredményeket
  - Ezek alapján egyre hatékonyabb teszteket végezhetünk és finomíthatjuk a hipotézisünket

Hibaelhárítás

- Készítsünk megoldási javaslatokat és elemezzük, értékeljük őket
  - Több megoldás is lehet egy problémára, eltérő hatékonysággal
  - A jelenlegi helyzetnek legmegfelelőbbet válasszuk
  - Lehet, hogy a legjobb megoldás jelenleg nem kivitelezhető, viszont egy nem optimális ideiglenes megoldás már elfogadható eredményeket hozna: értékeljük, milyen pozitív illetve negatív hatásai lennének az ideiglenes megoldásnak az optimálissal szemben
- Alkalmazzuk a választott megoldást és értékeljük az eredményeket
  - Definiáljuk, mit várunk el a választott megoldástól
  - Ellenőrizzük, hogy a választott megoldás ténylegesen a várt eredményeket hozza-e

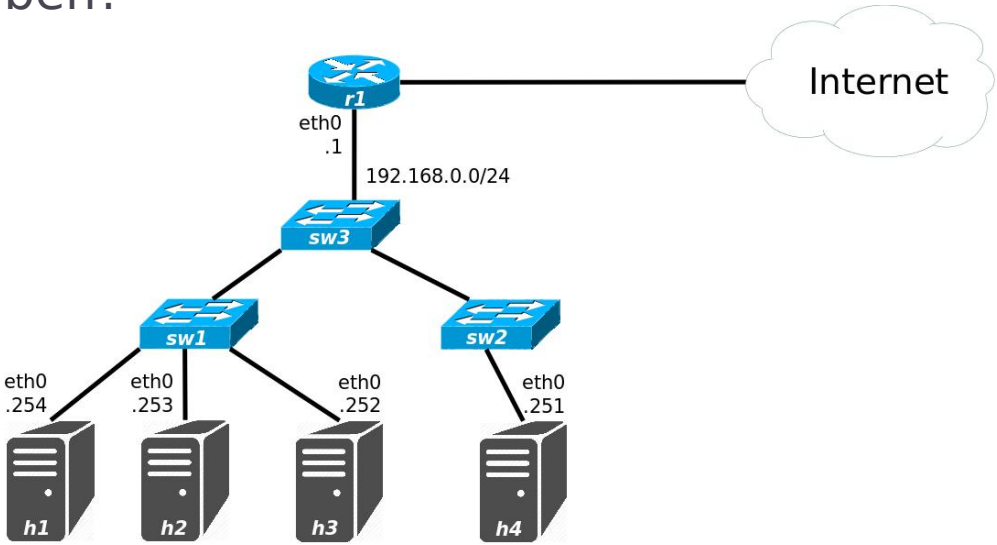
Hibaelhárítás

- A hibajelenségek általában elég bonyolultaknak tűnnek, de ez nem feltétlenül jelenti azt, hogy valami rendkívül összetett probléma az okozójuk
- Gyakran egy egyszerű hiba is szerteágazó problémákhoz vezethet
- Érdemes ezért minidig a legegyszerűbb hibákat feltételezni, és első körben azokat kizárni

Eszközök

1. feladat

- Az alábbi hálózatban hiba lép fel, melynek eredményeként legalább egy hoszt nem éri el az Internetet.
  - Milyen hibakereső eszközöket használna a hiba feltárásához, milyen sorrendben?



ARP cache

- ARP cache: az eszköz által rögzített IP cím–fizikai cím összerendeléseket rögzíti
- Az `arp` parancs használatával az ARP cache kérdezhető le vagy állítható be
  - A `-n` kapcsoló használatával elkerülhető a kapott IP címek feloldása
  - A `-i` kapcsoló használatával szűkíthető az információ egy adott interfészre
  - Pl.: `$ arp -ni eth0`
- A lekérdezéshez használható az `ip neighbor` (`ip neighbour`) parancs is
  - Pl. összes interfész lekérdezése: `$ ip n`
  - Pl. csak az *eth0* lekérdezése: `$ ip n show dev eth0`

Interfész beállítások

- Az `ifconfig` parancs az interfészek IP paramétereinek beállítására, lekérdezésére használható
- Alapértelmezetten azt adja meg, hogy mely interfészek működnek jelenleg,
  és ezekhez milyen konfiguráció tartozik:
  - A jelenleg használt IP cím, netmask
  - Statisztikát készít a fogadott és küldött csomagokról és az ezekkel kapcsolatos hibákról
- A `-a` kapcsoló segítségével megnézhetjük az összes (működő és nem működő) interfészt
- Interfész konfigurálása:
  - `$ sudo ifconfig <interfész név> <IP cím> netmask <netmask> [up]`
- Pl.: `$ ifconfig eth0`

Interfész beállítások

- Az `ip address/ip link` parancsok is használhatók
  - Pl.: `$ ip a`
  - Pl.: `$ ip l`
- Interfész konfigurálása:
  - Interfész bekapcsolása:
    `$ sudo ip link set <interfész név> up`
  - IP cím hozzárendelése:
    `$ sudo ip address add <IP cím>[/<netmask>] dev  <interfész név>`
  - Törlés:
    `$ sudo ip address del ...`

netcat

- A `netcat` egy egyszerű szoftveres eszköz, amivel TCP vagy UDP hálózati kapcsolatokat használva írhatunk vagy olvashatunk adatokat
- `-z` kapcsoló: ellenőrzi, hogy a célon hallgat-e megfelelő folyamat
- `-v` kapcsoló: bővebb információt ad
- Pl. `$ nc google.com 80`
  `HEAD / HTTP/1.0`

telnet

- A `telnet` alapvetően távoli bejelentkezéshez volt használható, mely helyett biztonsági okokból az `ssh` használatos
- Használható hibakereséshez: nyílt szöveget küld a fogadó félnek, így szöveges alapú kiszolgálóhoz csatlakozhat, és ellenőrizheti a működésüket
- Pl. `$ telnet google.com 80`
  `HEAD / HTTP/1.0`

ssh

- Az `ssh` titkosított kommunikációt biztosít hosztok között nem megbízható hálózat felett
- Lehetővé teszi a bejelentkezést távoli gépekre, valamint utasítások végrehajtását
- A `-X` kapcsoló használatával X11 kapcsolatok felépítését is biztosítja

nslookup

- Az `nslookup` segítségével a címfeloldás ellenőrizhető
- Egy név alapján a DNS kiszolgálók segítségével megkísérli kikeresni a névhez tartozó IP címet vagy címeket
- Paraméterként megadhatjuk azt a DNS szervert, melytől a lekérdezés eredményét várjuk:

```
$ nslookup <keresett név> [<DNS szerver IP címe>]
```

- Pl. `$ nslookup google.com`

ping

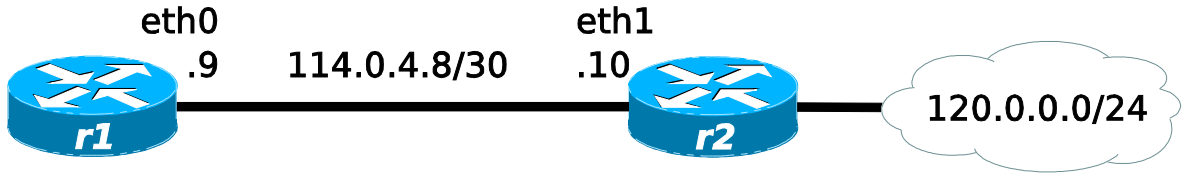
- A `ping` révén egyszerű kapcsolat felvételi teszt végezhető
- ICMP echo requesteket küld és az ezekre érkezett replyokból kiszámolja a körülfordulási időt
- Pl. `$ ping google.com`
  - Miért szakad meg a `ping`?

Útválasztás

- A `route` parancs használható útválasztási szabályok felvételére, törlésére és lekérdezésére
- Lekérdezéskor megkapjuk a cél hálózat címét és maszkját, azt az átjárót és interfészt, amin keresztül a hálózat elérhető
- A `-n` kapcsoló használatával gyorsítható a lekérdezés, ilyenkor a routing táblában tárolt IP címeket nem kísérli meg DNS használatával feloldani az eszköz
- Pl. `$ route -n`

Útválasztás

- Default gateway konfigurálása a `route` parancs használatával:
  - `$ sudo route add default gw <default gw IP címe>`
- Route létrehozása:
  - `$ sudo route add -net <IP cím> netmask <netmask>`
    `gw <gateway IP címe> <saját interfész neve>`
  - Pl.:

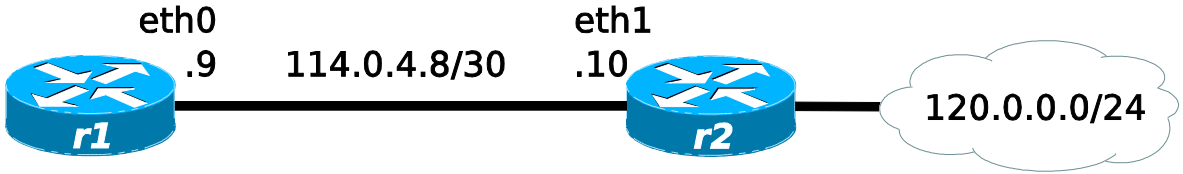



```
r1$ sudo route add -net 120.0.0.0 netmask
     255.255.255.0 gw 114.0.4.10 eth0
```

- Törlés:
  - `$ sudo route del ...`

Útválasztás

- Default gateway konfigurálása `ip route` használatával:
  - `$ sudo ip route add default via <IP cím>`
- Route létrehozása:
  - `$ sudo ip route add <IP cím>/<netmask> via <IP cím>`
  - Pl.:



  - `r1$ sudo ip route add 120.0.0.0/24 via 114.0.4.10`
- Törlés:
  - `$ sudo ip route del ...`
- Pl. lekérdezés:
  - `$ ip r`

traceroute

- A `traceroute` két végpont közötti útvonal felderítését végzi el
- Az útvonal követésénél azon IP címeket sorolja fel, melyek a csomagot fogadták
- Mivel a csomagok az oda és a vissza úton eltérő útvonalakon haladhatnak, így előfordulhat, hogy a `traceroute` által visszaadott IP címek nem feleltethetőek meg ugyanazon routereknek
- Pl. `$ traceroute google.com`

Wireshark

- A Wireshark egy csomagelemző
- Egy vagy több interfészen beérkező és távozó csomagokat figyelhetünk meg vele
- Lehetőséget biztosít arra, hogy beletekintsünk az egyes csomagok fejléceibe, ha több rétegben történt betokozás, akkor az összes réteg fejlécét is képes visszafejteni

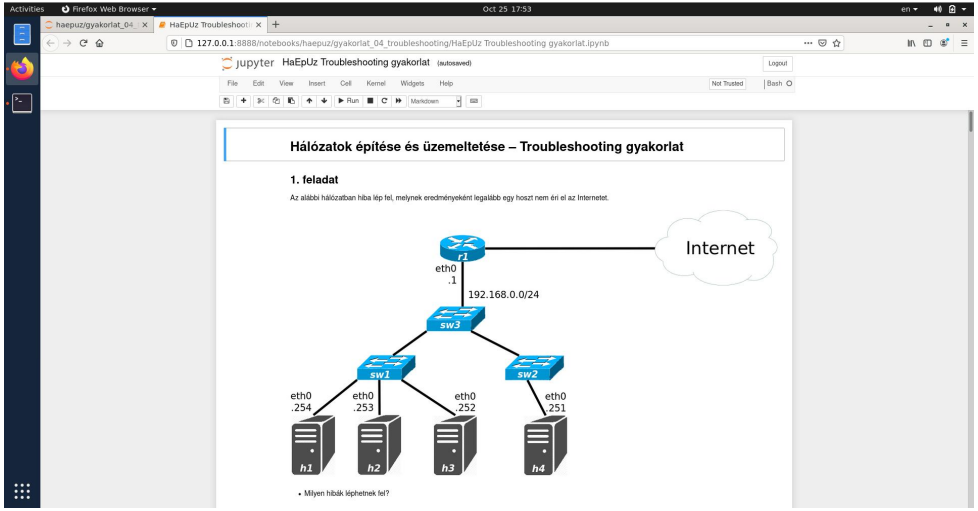
tcpdump

- A `tcpdump` segítségével egy interfészen áthaladó csomagokat tudjuk megjeleníteni
- Nem használ GUI-t
- Pl. `$ sudo tcpdump -i eth0`

Feladatok

VM: HaEpUz 2022 (GUI: rdp)

- Jupyter notebook indítása
  - *Terminal*: ./notebook.sh
  - *Firefox*: haepuz / gyakorlat_04_troubleshooting / HaEpUz Troubleshooting gyakorlat.ipynb
- Segédlet + Feladatok
- Egyszerre csak egy Firefox legyen elindítva
- Kulcskérdés
  - *Terminal*:
    - cd ~/.ssh
    - ssh-keygen [enter, enter, enter]
    - cat id_rsa.pub >> authorized_keys



- Dedikált menedzsment hálózat
  - 10.0.0.0/24
  - a feladatok nem erre vonatkoznak!

2. feladat (VM)

- A hálózatban jelenleg minden hoszt végez átvitelt.
  - Figyelje meg, hogy milyen irányban folyik adatátvitel és dokumentálja az eredményeit!
    - `ssh` segítségével lépjen be az egyes hosztokra (pl. *h1* esetén használja az `$ ssh -X h1` utasítást)!
    - Indítsa el a Wireshark monitorozó eszközt (`$ sudo wireshark &`), majd figyelje meg az *eth0* interfész forgalmát!
    - Jegyezze fel a következőket:
      - az adott hosztra honnan érkezik forgalom és ez milyen típusú (protokoll)?
      - az adott hosztról merre indul forgalom és ez milyen típusú (protokoll)?
      - a végpontoknak mi a MAC címük?
  - A megfigyelései alapján milyen bejegyzéseket kellene tartalmaznia a router ARP cache-ének?
  - Milyen utasítások használatával derítené fel, hogy egyezik-e az ARP cache tartalma az elvártakkal?

3. feladat (VM)

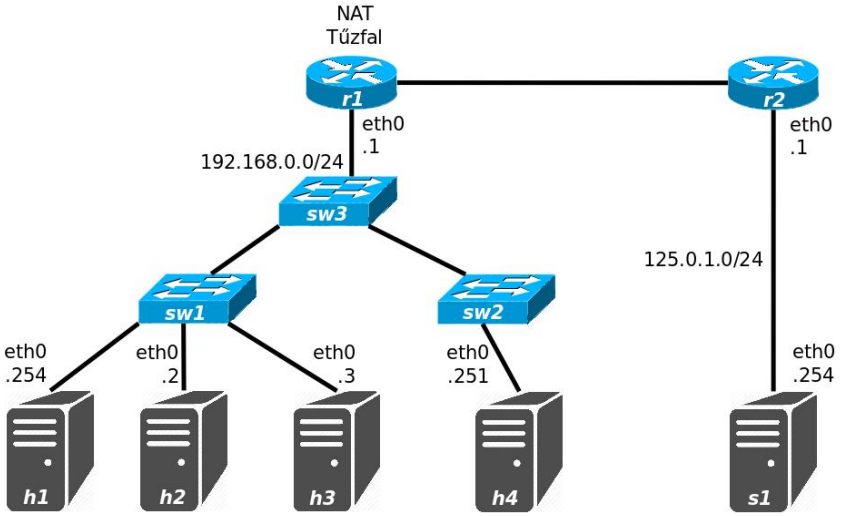
- A hálózatban a *h1* hoszt csak részlegesen tud kommunikálni a hálózattal: bizonyos hosztokat elér, míg másokat nem, valamint az *s1* szerverrel sem képes kommunikálni. A hoszt IP címe és hálózati maszkja statikusan van konfigurálva.
- Mi lehet a probléma?
  - Tesztelje a többi hoszt elérhetőségét `ping` segítségével!
  - Ellenőrizze a *h1* interfészének konfigurációját és vesse össze a hálózat ábrájával!
  - Javítsa a hibát!

4. feladat

- Egy irodában az a szokás, hogy a dolgozók az asztalok között vándorolnak, és ilyenkor viszik magukkal a laptopjukat is. Mivel a wifi elérhetősége nem mindenütt megfelelő, ezért a dolgozók vezetékes kapcsolaton keresztül csatlakoznak a hálózatra. Egy dolgozó az egyik nap azonban azt veszi észre, hogy míg korábban tetszőleges asztalhoz leülve sikeresen csatlakozott a hálózathoz, most ugyan az első helyen, ahol csatlakozni próbált, ott sikerült neki, de amikor átment egy másik helyre – ahol korábban már ült valaki –, ott már nem. Furcsa módon amikor az előzőleg ott ülő kolléga visszatér, neki sem sikerül csatlakozni a hálózathoz.
- A két munkatárs értesíti is önt a szokatlan hibajelenségről. Önnek eszébe jut, hogy most lépett érvénybe a cég új hálózati policyje, mely azt célozza, hogy illetéktelenek ne csatlakozhassanak a hálózatra. Milyen problémára kezd gyanakodni? Hogyan oldaná meg? Megfelelőnek tartja-e az ilyen típusú védekezést az illetéktelenek hálózatra kapcsolódása ellen?

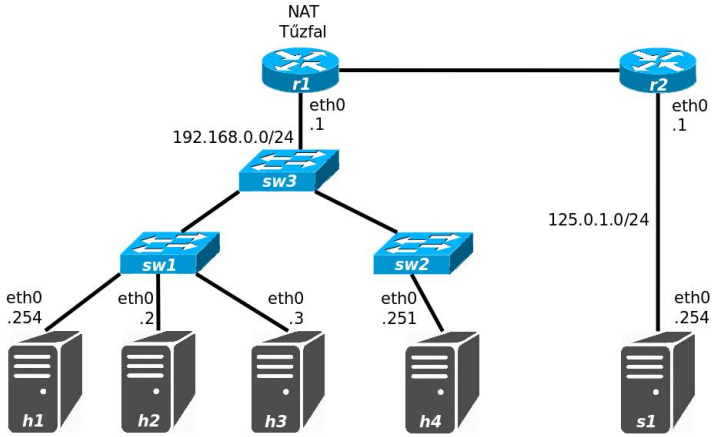
5. feladat

- Az alábbi hálózat esetén, a *h1* hoszt egyik – hálózati megoldásokban nem igazán jártas – felhasználója szeretne fájlokat megosztani az *s1* előtt ülő ismerősével.



5. feladat

- Egy egyszerű fájl szerver indítása után lekérdezi a *h1* IP címét és átküldi azt ismerősének. Az *s1* előtt ülő felhasználó ezt az IP címet használva szeretne csatlakozni *h1*-hez, azonban nem tud.
  - A *h1* hoszt felhasználója vajon milyen IP címet kérdezhetett le és milyen eszközzel?
  - Mi lehet az oka, hogy *s1* nem tud kapcsolódni *h1*-hez?



6. feladat

- Egy irodai hálózaton – ahol egyre több eszköz kapcsolódik a hálózatra – DHCP-n keresztül kapja minden hoszt a hálózati konfigurációját. Néha előfordul, hogy egy-egy hoszt nem képes csatlakozni a hálózathoz. Ilyen esetben az interfész konfigurációt lekérdezve ön azt tapasztalja, hogy az eszköz nem kapott IP címet.
  - Miért lehet ez?
  - Nagyvonalakban hogyan ellenőrizné, hogy helyes-e a megállapítása?
  - Mi lehet a megoldás a problémára?

7. feladat (VM)

- A hálózatban a *h1* hosztról nem érhető el az *s1* szerver.
- Mi lehet a probléma?
  - A *h1* hosztra történő bejelentkezés után `nslookup` használatával kérdezze le a 192.168.0.252-es IP címen elérhető DNS szervertől az *s1.tslab* IP címét!
  - Indítson `ping`et a kapott IP címre! Hogyan értékeli a kapott eredményt?
- Hogyan lehet javítani a hibát?

7. feladat (VM)

- A hálózatban a *h1* hosztról nem érhető el az *s1* szerver.
- Mi lehet a probléma?
  - A *h1* hosztra történő bejelentkezés után `nslookup` használatával kérdezze le a 192.168.0.252-es IP címen elérhető DNS szervertől az *s1.tslab* IP címét!
  - Indítson `ping`et a kapott IP címre! Hogyan értékeli a kapott eredményt?
- Hogyan lehet javítani a hibát?
  - Statikus bejegyzéssel? Elég hatékony-e a kapott routing tábla, össze lehet-e vonni benne bejegyzéseket?
  - Ha kézzel nem lehet módosítani *r1* routing tábláját, de tudjuk, hogy *r2* OSPF-et futtat?

8. feladat (VM)

- A hálózat felhasználói arra panaszkodnak, hogy amikor az *s1.tslab* oldalt szeretnék betölteni, akkor azon a *Welcome to s2!* felirat olvasható. Az *s2.tslab* oldal betöltésekor is hasonló problémába ütköznek. Ön már tudja, hogy az *s1* és az *s2* a megfelelő oldalakat ajánlja ki, a *h1* felől mégsem ez látszik.
  - *h1*-re történő bejelentkezés után egy böngésző (Firefox) segítségével ellenőrizze le, hogy valóban fennáll-e a hiba! (A böngésző kissé lassan tölti be az oldalt.)
  - Mi történik, ha nem az *s1.tslab* ill. az *s2.tslab* neveket írja a böngésző címsorába, hanem azok IP címét? (Az IP címeket a hálózat ábráján találja.)
  - Mi lehet a probléma oka? (`nslookup` használatával ellenőrizze a címfeloldást!)
  - Milyen IP címen keresné a problémát okozó eszközt?

9. feladat

- Az előző feladat tanulságait felhasználva válaszolja meg a következő kérdéseket!
- Ha egy hálózatban a hosztok képesek elérni egymást nevek használatával, de egyetlen külső szervert sem képesek név alapján elérni, csak közvetlenül IP címet használva, akkor milyen problémára gyanakodna?
- Milyen megoldás jöhet szóba?

10. feladat (VM)

- Időnként a *h1* vagy a *h2* hoszt felhasználóitól hibabejelentést kap, mely szerint az *s1* szerver időnként elérhetetlen. Amikor viszont ön próbál kapcsolódni a hosztokról az *s1*-hez, minden esetben sikeres a kapcsolatfelépítés. Más hosztok felhasználóitól sem érkezik ilyen panasz, és ön a saját hosztjáról is bármikor képes elérni az *s1* szervert. Ma éppen sem a *h1*, sem a *h2* hosztot nem használja senki. Úgy dönt, kivizsgálja az esetet.
  - A *h1* hosztra történő bejelentkezés után `ping`elje meg az *s1* szervert!
  - Anélkül, hogy leállítaná az előző pinget, a *h2* hosztról szintén pingelje meg az *s1* szervert!
  - Mit tapasztal? Mit gondol, miért történhet ilyen?
  - Hasonlítsa össze a *h1* és *h2* interfész konfigurációját az `ifconfig` parancs használatával!
  - Válasszon egy szabad IP címet a hálózaton és ennek megfelelően javítsa a hibát a *h2* hoszton! Ne felejtse el beállítani a default gateway-t! Ellenőrizze, hogy a probléma még mindig fennáll-e!

11. feladat (VM)

- A hálózatban az *s1* és az *s2* szerverek 80-as portján egy-egy HTTP szervernek kellene elérhetőnek lennie. A hálózat felhasználói viszont arra panaszkodnak, hogy az *s2* szerveren lévő weboldat nem érik el. Feltételezhetjük, hogy a hálózattal minden rendben van. Mi lehet a probléma?
  - A *h1* hoszton indítson el egy böngészőt és hasonlítsa össze, hogy milyen eredményeket kap az *s1.tslab* ill. az *s2.tslab* oldalakat betöltve!
  - `netcat` használatával ellenőrizze, hogy a 80-as port nyitott-e az *s2* szerveren! (A lekérdezéskor a szerver IP címét használja!)
  - `netcat` vagy `telnet` használatával csatlakozzon az *s1* és *s2* szerverek 80-as portjára és ellenőrizze, hogy ott valóban egy HTTP szerver található-e! Értékelje az eredményeket!