14. Lokális hálózatok és programozás

(Kidolgozta: Jurík Flórián)

1. Írja le és magyarázza meg a LAN hálózatok infrastruktúráját, azok eszközeit, a szórási és az ütközési tartományt.

**LAN hálózatok infrastruktúrája:**

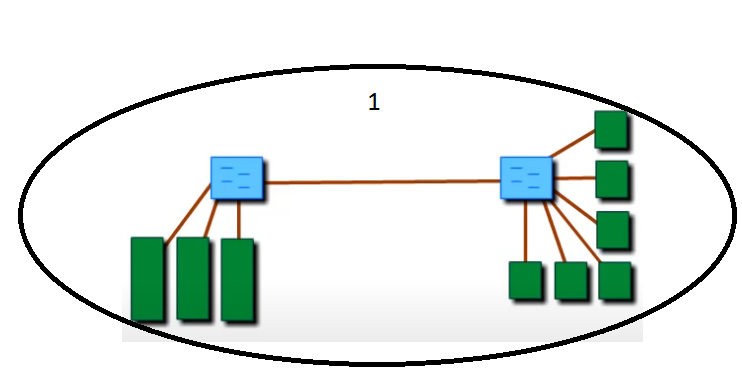
A LAN (Local Area Network – Helyi hálózat) infrastruktúra egy kis földrajzi területen belül (ház, iskola, iroda, épület) köti össze a berendezéseket.

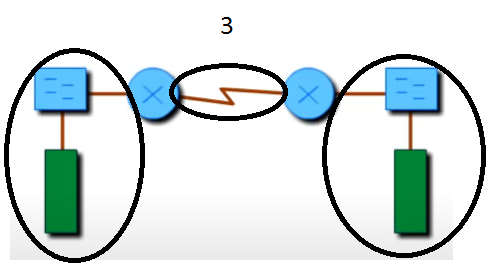
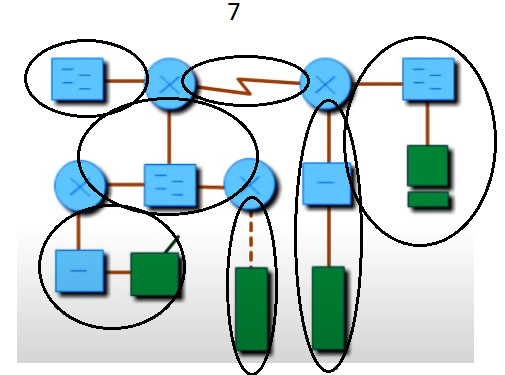
**LAN hálózatok eszközei:**

Végfelhasználói eszközök (telefon, PC, laptop…), közvetítő eszközök (router, switch, hub...), adatátviteli közegek (optikai vagy réz szál, vezeték nélküli).

**Szórási tartomány (Broadcast domain):**

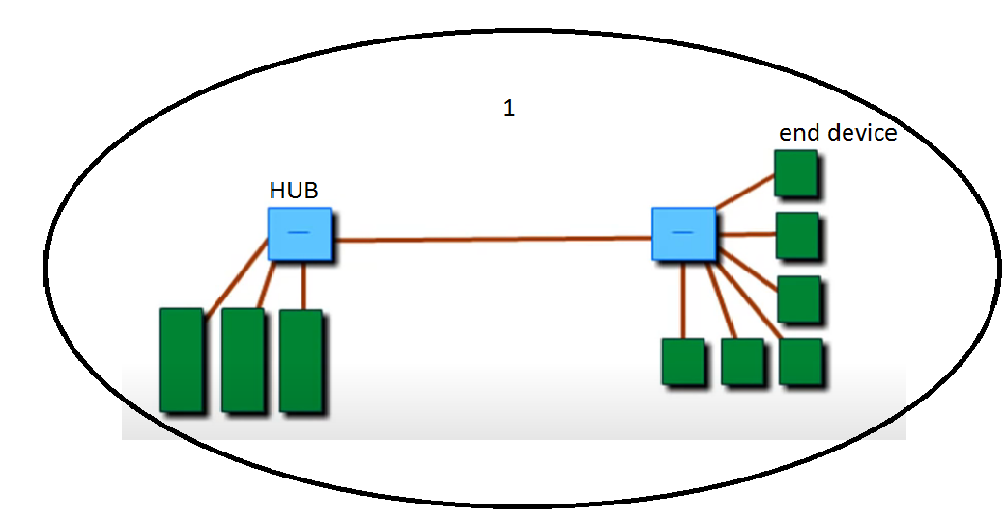
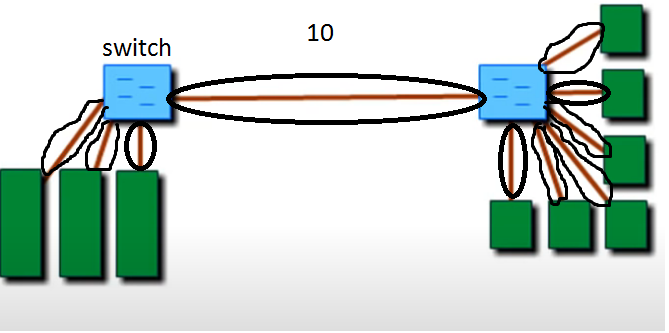
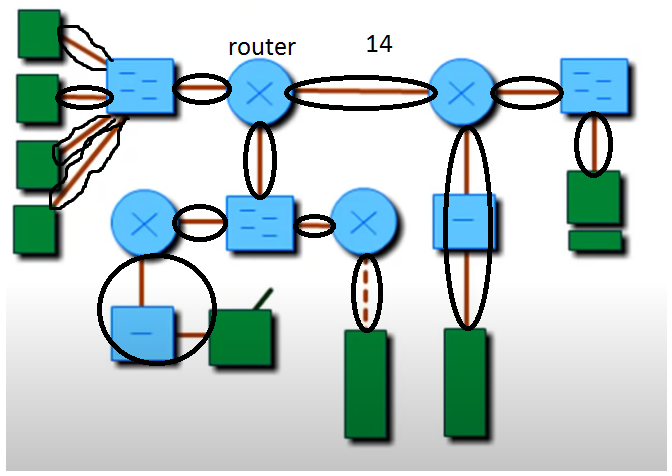
Ha egy számítógép broadcast üzenetet küld ki, azt minden közvetítő eszköz tovább küldi a többi eszköznek kivéve a routert. Tulajdonképpen ez azt jelenti, hogy mennyi különböző hálózat található. A routerektől függ az, hogy mennyi szórási tartomány (broadcast domain) található a hálózatban.

Példák különböző számú szórási tartományt tartalmazó hálózatokra:

****

**Ütközési tartomány (Collision domain):**

Eszközök csoportja, amelyek packetjei (csomagjai) össze tudnának ütközni. A packetek áramlásától függ. Ha két különböző packet összeütközik kollízió jön létre és újra kell őket küldeni. Közvetítő eszközök segítségével lehet őket csökkenteni. Minél kevesebb a hálózatban a hub, annál kevesebb az ütközési tartomány is, mivel ez csak az 1. rétegen (Layer 1) működik és egy bejövő adatot csak továbbküldi a többi porton, ami felesleges és ezzel csak tágítja az ütközési tartományt. A switch és a router viszont szétosztja őket, mert már tudják mit hová kell küldeniük.

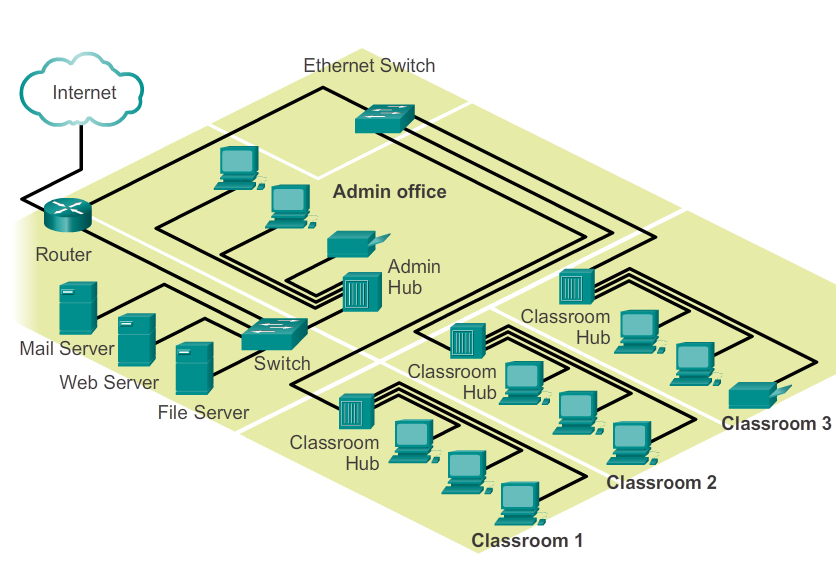
****Példák különböző számú ütközési tartományt tartalmazó hálózatokra:

1. Írja le a topológiák típusait és a „Point-to-Point a Broadcast Multiaccess Area“ fogalmát.

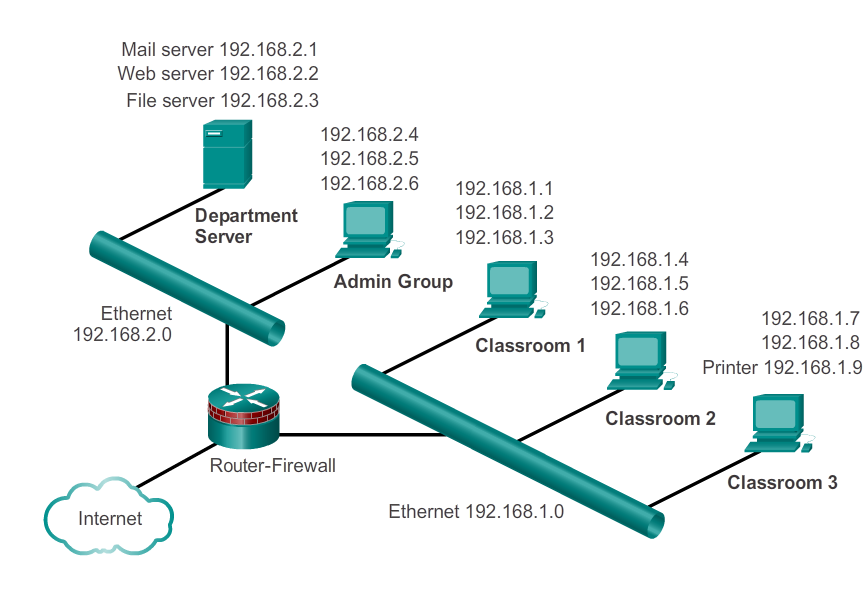
**Topológia fogalma, típusai:**

A hálózat topológiája a hálózati eszközök elrendezését és a köztük lévő összeköttetéseket jelenti.

Fizikai topológia:

A fizikai összeköttetésekre utal, meghatározza a végberendezések és a köztes eszközök (pl.: routerek, switchek és vezeték nélküli hozzáférési pontok) kapcsolódási módját. A fizikai topológia általában pont-pont vagy csillag típusú.

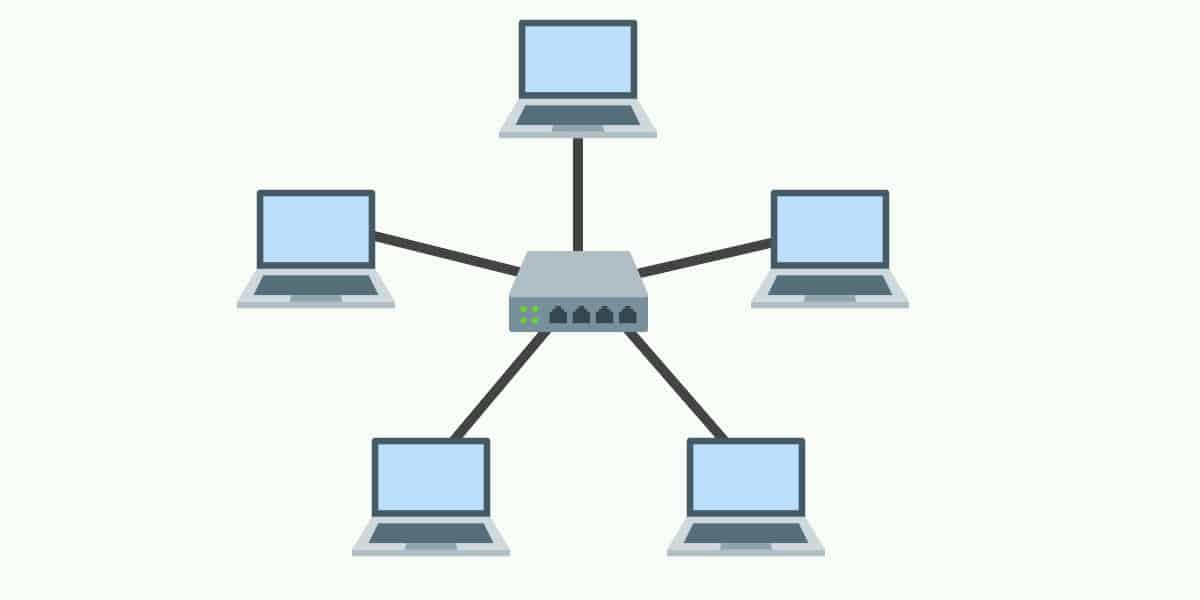
Logikai topológia:

Arra utal, hogy a hálózat miként szállítja a kereteket egyik állomástól a másikig. Ez az elrendezés az állomások közötti virtuális kapcsolatokból áll. A hozzájuk tartozó logikai jelutakat az adatkapcsolati réteg (data link layer) protokolljai határozzák meg. Az adatkapcsolati réteg a hálózat logikai topológiáját "figyeli" a közeghozzáférés vezérlése közben. A logikai topológia az, amely befolyással van a keretezés típusára és a közeghozzáférés módjára.

**Point-to-point fogalma:**

Egy topológia fajta. Közvetlenül köt össze két csomópontot. Rendelkezik fizikai és logikai megvalósítással, amelyek annyiban különböznek, hogy a fizikaiak fizikailag vannak összekötve a logikaiak pedig logikailag.

**Broadcast Multiaccess Area:**

A számítógépek hozzá vannak kötve egy hubhoz vagy switchhez és ezeken keresztül kommunikálnak, a csillag topológia is így működik.

1. Írja le és magyarázza meg a „shared medium“-hoz való hozzáférés módszereit - CSMA/CD és CSMA/CA.

**Shared medium fogalma:**

Néhány topológia egy megosztott közvetítő eszközön osztozik. Ezért az a neve, hogy shared medium. Ilyenkor fennáll a probléma, hogy a végeszközök egyszerre küldenek adatokat. Ennek a kontrolálására két metódus létezik.

* Contention-based access: Az összes eszköz versenyez az összeköttetés használatáért, de adat ütközés esetén van egy tervük.
* Controlled access: Mindegyik eszköznek megvan a saját ideje, hogy mikor használhatja az összeköttetést.

**CSMA, CSMA/CD és CSMA/CA fogalma:**

CSMA (Carrier Sense Multiple Access): Ez egy folyamat amely kideríti, hogy az összeköttetésen épp van-e jel.

CSMA/CD (Carrier sense multiple access with collision detection): A végfelhasználói eszköz figyeli a hálózatot adatjel után kutatva. Ha nem észlel jelet, akkor elküldi saját adatait. Ezt észleli a többi berendezés, amelyek szintén kommunikálni akarnak ezért várnak egy kicsit, ezután rendszeresen megfigyelik az összeköttetést egészen addig, amíg nem észlelik üresnek és küldhetik saját adataikat.

CSMA/CA (Carrier sense multiple access with collision avoidance): Lényegében ugyanaz mint az előző annyi különbséggel, hogy az adat küldése előtt a végfelhasználói eszköz küld egy értesítést, hogy szándékozik használni az összeköttetést. Csak akkor küldi át az adatokat miután megkapta az engedélyt a használatra. Ezt a módszert főleg vezeték nélküli technológiáknál használják.

1. Írja le a Unicast, Multicast, Broadcast (limited + directed) kommunikáció lehetőségeit.

**Unicast (egyes küldés):** Annak a folyamata, hogy például egy számítógép a packetet egy másik számítógépnek küldi.

**Multicast (többes küldés):** Annak a folyamata, hogy például egy számítógép a packetet egy kiválasztott csoportnak küldi.

**Broadcast (szórás):** Annak a folyamata, hogy például egy számítógép a packetet mindenkinek elküldi a hálózaton belül úgynevezett broadcast cím segítségével.

Fajtái:

* Limited broadcast: A packetek nem mehetnek lokális hálózaton kívülre.
* Directed broadcast: A packetek kimehetnek lokális hálózaton kívülre is.

1. Egy konkrét példán számolja ki és magyarázza meg a (privát IPv4) címtartomány meghatározásának lépéseit egy vállalati hálózat részére VLSM felhasználásával és a következő követelményekkel: LAN1 = 63 IPs; LAN2 = 62 IPs; LAN3 = 24 IPs; LAN4 = 2 IPs
   1. Számolja ki az egyes LAN-ok címtartományait és a nem felhasznált IPv4 címek számát. Határozza meg az összes alhálózatban: Network IP, prefix, SnM, Range IP és BC IP.

VLSM (Variable Length Subnet Mask – Változó hosszúságú alhálózati maszk): Az IP-címek hatékonyabb kihasználására használják a hálózatokban. A fenti példa kiszámításához szükség lesz egy /24 alhálózati maszkkal rendelkező random IP-címre, ami most legyen 192.168.1.0. Azért választottuk a 24-es alhálózati maszkot, mivel összeadva a LAN-okat összesen 151 IP-címet kell kiosztanunk, amit felkerekítünk a 2 hatványaira, így lesz ez 28. Ezt a 8-as számot pedig ki kell vonni a 32-ből (alhálózati maszk mérete bitekben), így kapjuk meg a 24-et. Mivel a 24-es alhálózati maszkot használjuk, az IP-címek a következőképpen lesznek felhasználva:

192.168.1.0 – hálózat

192.168.1.1 – 192.168.1.254 - felhasználható IP-k

192.168.1.255 – broadcast

Mivel nekünk nem kell egy LAN-ban 254 IP így ezek szét lesznek osztva a LAN-oknak megfelelően. A szétosztást tanácsos mindig a legnagyobbtól kezdeni.

Az IP-címek utáni /x számok a prefixet jelentik, amelyek a subnet mask rövidített változatai.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Host | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Alhálózati maszk | /24 | /25 | /26 | /27 | /28 | /29 | /30 | /31 | /32 |

A LAN1-nek 63 IP címre van szüksége. Megnézzük a táblát és látjuk, hogy van ott egy, ami képes adni 64-et. Ez nem elegendő mivel a 64 valójában 62 + 2(hálózat és broadcast) így az eggyel továbbit fogjuk választani ami 128. Szóval a LAN1 hálózata a következőképpen fog kinézni:

192.168.1.0 /25 amin belül:

192.168.1.0 – hálózat

192.168.1.1 – 192.168.1.126 – felhasználható IP-k

192.168.1.127 – broadcast

A következő számítás pedig 192.168.1.128-tól fog kezdődni. A LAN2 pontosan kielégíti a 64-et (62+2), szóval 192.168.1.128/26 lesz, amin belül:

192.168.1.128 – hálózat

192.168.1.129 – 192.168.1.190 - felhasználható IP-k

192.168.1.191 – broadcast

A LAN3-nak 24 IP-re van szüksége, ami belefér a 32-be (30 + 2) szóval 192.168.1.192/27 lesz, amin belül:

192.168.1.192 – hálózat

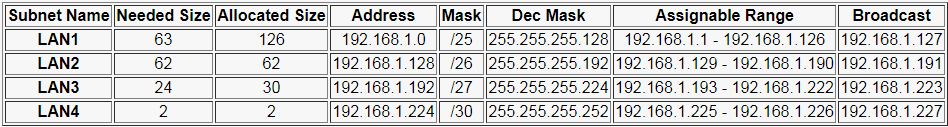
192.168.1.193 – 192.168.1.222 - felhasználható IP-k

192.168.1.223 – broadcast

A LAN4-nek csupán 2 IP-re van szüksége, amelynek megfelelő található a táblázatban, de nem felhasználható mivel még + 1-1 kell a hálózatnak és a broadcastnak is így a következőt kell választani, ami 4 lesz és így fog kinézni a következő felbontás - 192.168.1.224/30 amin belül:

192.168.1.224 – hálózat

192.168.1.225 – 162.168.1.226 – felhasználható IP-k

192.168.1.227 – broadcast

* 1. Írja le a DHCPv4 protokoll célját és tevékenységét. Vázolja fel felhasználásának lehetőségeit (ip helper address,...).

**DHCP (Dynamic Host Configuration Protocol):** Létezik DHCP kliens (UDP 68 portra kapja az üzeneteket) és DHCP szerver (UDP 67). A szerver dinamikusan adja meg a kliens IP-címét meghatározott választási lehetőségek közül. A IP megkapásának a folyamata a következőképpen történik.

1. – A számítógép üzenetet küld a szervernek, hogy szüksége van egy IP- címre

2. – A szerver felkínál neki egyet

3. – A számítógép elfogadja

4. – A szerver elismeri az elfogadást

Ez a kommunikáció broadcast üzenetek küldésével megy végbe.

A szerver csak egy bizonyos időre adja oda az IP-címet, hogy ne legyenek feleslegesen elhasználva, ezért a kliensnek a cím megtartása érdekében időnként üzenetet kell küldenie, hogy még használja azt, vagyis megújítja a szerződésüket.

**IP helper address:**

****Interfészre beírható parancs ami arra szolgál, hogy ha a szerver és a kliens nincsenek egy hálózaton, úgy is tudjanak kommunikálni. Azért nem tudnának komunikálni mert a router alapból nem engedi át a broadcast üzeneteket.

**Network time protocol:**

Számítógépes rendszerek óráinak szinkronizálására szolgáló hálózati protokoll.

**DNS (Domain name system) protocol:**

Ennek segítségével a felhasználónak nem kell úgy kikeresnie az oldalakat, hogy IP-címeket ír be, hanem megadhat domain neveket is.

**Bootstrap protocol:**

Felhasználása hasonló a DHCP-hez csak sokkal elavultabb. Nem támogatja a telefonokat. A rendszer betöltése során használják, amikor a számítógép elindul.

**TFTP (Trivial File Transfer Protocol):** Legfőképpen arra használják, hogy a fájlokat a lokális hálózaton továbbítsa (konfigurációmentések, stb.). UDP protokollt használ. Nincs semmilyen biztonsággal felszerelve.

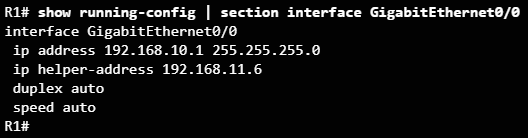
* 1. Egy PT példán „show“ és ICMP parancsok segítségével határozza meg a DHCPv4 protokoll konfigurációját és működését az útválasztón.

**Router1#show ip dhcp binding** – Megnézni, hogy melyik IP-címeknek vannak most lefoglalva.

**Router1#show ip dhcp conflict** – Ha 2 gép valahogy ugyanazt az IP-t kapta akkor ez a parancs kimutassa érzékelt konfliktusokat.

**Router1#show ip dhcp database** – Kimutatja a jelenlegi aktivitást a DHCP adatbázisban

**Router1# show ip dhcp server statistics** – A DHCP szerver statisztikáit mutatja ki

**Router1# show ip dhcp pool** – Megmutatja a DHCP pool-t és információkat róla. A pool azon IP-címek csoportja, amelyek még kioszthatók.

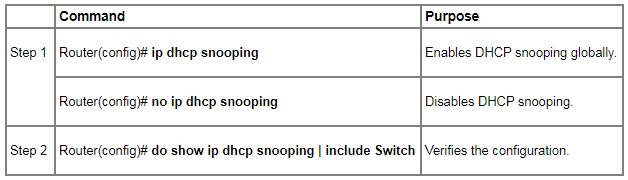
És még a többi általános show parancsban is meg lehet őket találni.

* 1. Írja le a DHCPv4 elleni támadásokat (starvation és spoofing – MiTM) és a programban demonstrálja az ellenük való védekezés lehetőségeit az útválasztón.

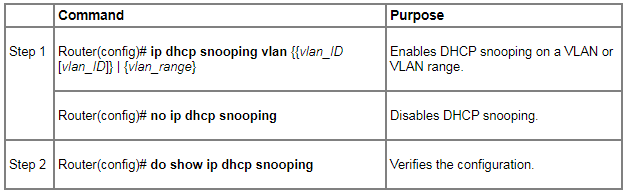
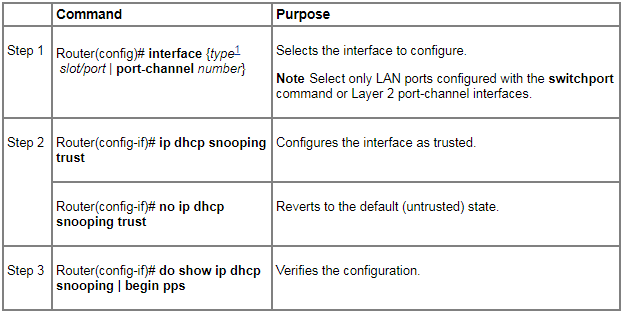
**DHCP starvation:** Valaki elküld egy csomó DHCP DISCOVER üzenetet, amelyekre ha a szerver válaszol, akkor gyorsan elfogyasztja a felhasználható IP-ket. Miután elfogyasztotta az IP-ket létrehozhat egy saját DHCP szervert és hamis gatewayt ír be aminek következtében ő rajta keresztül fognak menni az adatok - MiTM.

**DHCP spoofing:** Valaki létrehoz a meglévőn kívül egy DHCP szervert a hálózatban és a broadcast üzeneteket kihasználva megvan rá az esély, hogy a hamis DHCP szerver hamarébb válaszol. Rossz gatewayt megadva ugyan minden információ el lesz küldve az internetre, de azon a félen keresztül fog menni aki felálította a hamis szervert. Ezt hívják MiTM támadásnak.

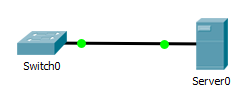
**MiTM (MAN in THE MIDDLE):** Két fél közötti kommunikációt megzavar egy harmadik, aki közöttük lesz. Mindkét fél számára a másik félnek adja ki magát ezzel hasznos információkat gyűjtve.

**DHCP snooping** – A routerek és switchek portjait be lehet konfigurálni megbízhatóvá és nem megbízhatóvá. A DHCP kliens nem megbízható porttól nem fog DHCP offert elfogadni.

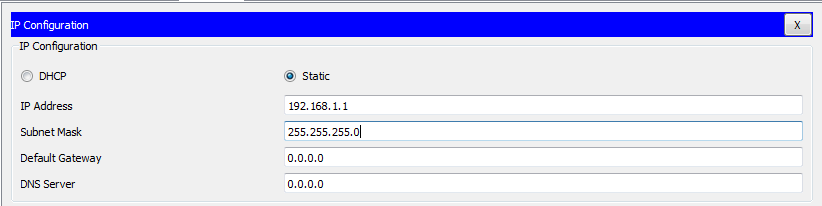
A VLAN-okon alapból nincs ez megengedve, így lehet engedélyezni:

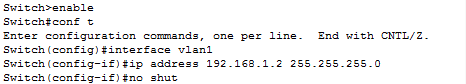
A portokat így lehet megbízhatóvá beállítani:

**DAI (Dynamic ARP Inspection):** A DHCP snooping által létrehozott adatbázis alapján összeköti az IP-ket a MAC-címekkel, így csökkentve az ARP poisoning esélyét, amely MiTM-hez vezethet.

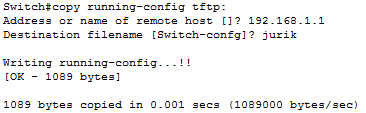
1. Hozzon létre biztonsági másolatot a konfigurációs állományból az ön vezetéknevével az egyik kapcsolóból a TFTP szerverre és egy szöveges állományt is Notepad-ban. A válasznál demonstrálja és ellenőrizze le a létrehozott biztonsági másolatok helyes működését.
2. Szükség lesz egy switchre és egy szerverre:
3. Megadjuk a helyes IP-ket, hogy tudjanak kommunikálni egymással:

Szerver: Klikk a szerverre >> Desktop >> IP Configuration



1. Switch: Mivel alapból nem lehet neki IP-t adni, fog kelleni egy vlan és annak lesz IP-je

(Pingeléssel lehet ellenőrizni, hogy tudnak-e kommunikálni.)

1. A küldés a következő parancsokkal történik (szoktak ilyenkor még egy copy running-config startup-config ot csinálni, de nem kötelező ):

(Azt, hogy megérkezett-e így lehet leellenőrizni: Szerverre kattints >> Services >> TFTP >> és ha ott van a fájlok között a te fájlod is akkor átment.)

1. A konfigurációt pedig a következőképpen lehet kinyerni a szöveges állományba:

Kattints a switchre >> config >> Running config export

Próbaként indítsuk újra switchet, és ha nem volt elmentve a running config startup config-ként akkor a switchen most nem lesz semmilyen elvégzett beállítás így beletölthetjük azt, ami a TFTP szerveren van.

Ahhoz, hogy a szerver és a switch tudjanak kommunikálni, először egy IP-címet kell adni a switchnek mivel újra lett indítva. Mindegy, hogy mi lesz az IP-cím (csak egy hálózatban legyen a szerverrel) ugyanis miután letöltötte a konfigurációt a szerverről arra fog vissza változni ami ott volt.

6. A konfigurációt a következő parancsokkal kell letölteni:

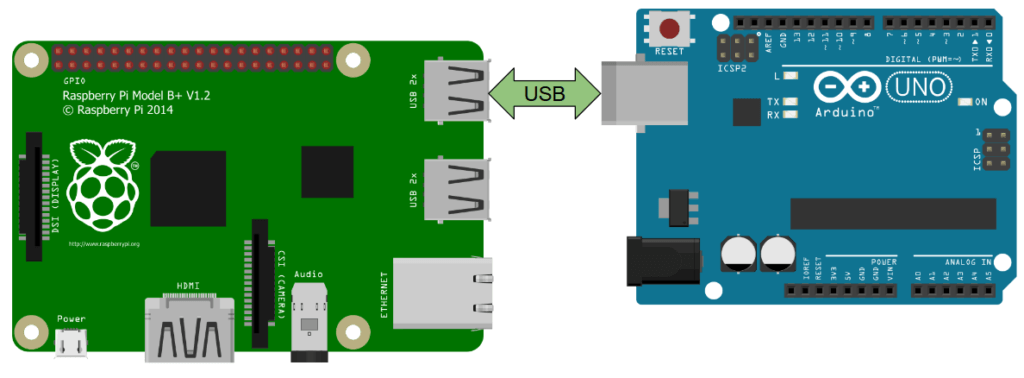
 Amikor a destination filename-t kéri, elég csak Enter-t nyomni.

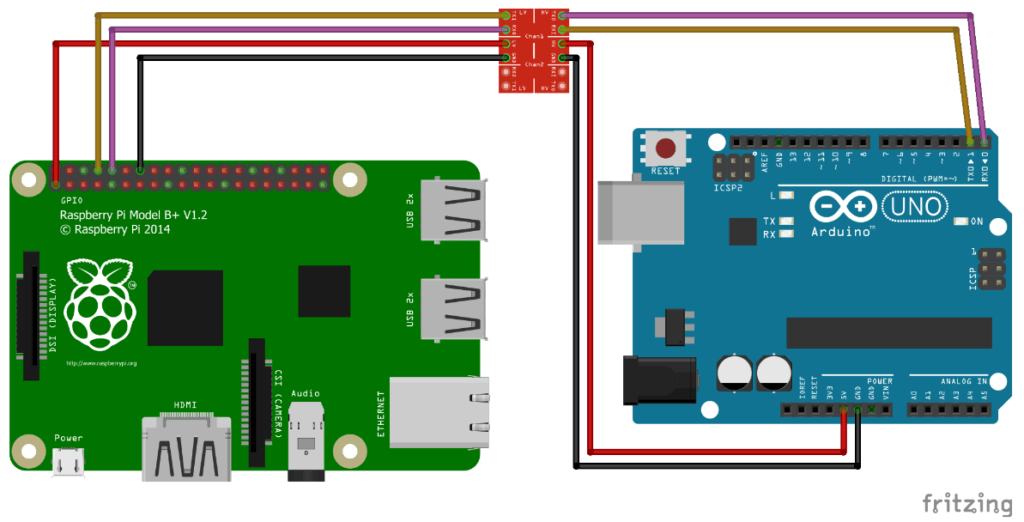
Ellenőrizzük a konfigurációt a show run paranccsal és ha minden olyan mint amikor először elmentettük akkor jól megcsináltuk a letöltést.

1. Írja le az IoT keretén belül a Raspberry PI és az Arduino együttműködését és a kommunikációjuk lehetőségeit.

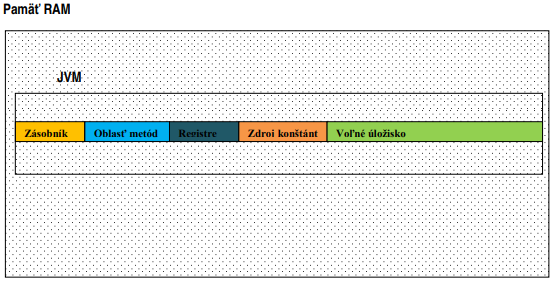
A két eszköz összedolgozásával és a megfelelő szenzorok beszerzésével elég összetett projekteket lehet csinálni, amelyek valós idejű adatokat küldhetnek az internetre is.

Az Arduinot és Raspberry Pi-t sorosan kétféleképpen lehet összekötni, hogy kommunikáljanak egymással:

* USB-vel: Ilyenkor először az Arduinot a számítógéphez kell kötni, hogy feltöltsük rá a futtatni kívánt programot, ezután pedig a Raspberry Pihez.
* Kivezetésekkel (pinekkel): Lehetséges ez a megoldásmód is, de ez már összetettebb, mivel a Raspberry Pi és Arduino fajtájától függően szükség lesz egy feszültség szintváltóra is.



1. Magyarázza el és mutassa be a memória dinamikus allokációját Java-ban. *(Megjegyzés: Szlovák feliratokkal kiegészítve a programozás órán tanultakkal való összevethetőség, jobb megértés céljából.)*



Magyarázat:

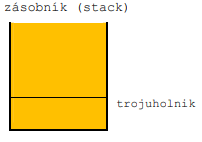
Zásobník – angol: stack, magyar: verem

Voľné úložisko – angol: heap, magyar: (heap-et használják)

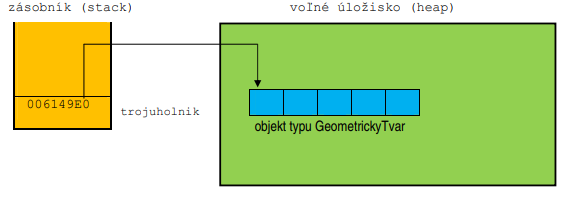
Ha készítünk egy Java projektet a JVM( Java Virtual Machine) a RAM egy részét a fenti ábrán látható részekre ossza fel. Minket most csak a stack és a heap fog érdekelni. Az összes primitív változó (int, float, char...) a stackben van létrehozva. Az objektumok (a classok példányai) pedig a heap-ben. Ebben az a jó, hogy csak addig vannak fenntartva, amíg szükség van rájuk. Ezzel sok helyet képes megspórolni a program.

1. Létrehozunk egy példányt (inštancia)

Ezt a program még csak a stackbe menti el.



1. Most pedig létrehozunk egy objektumot a new szó segítségével.

A new arra szolgál, hogy keres egy szabad helyet a heap-ben, amit ha talál akkor annak a címét visszaküldi és hozzáköti a trojuholník változóhoz.

Ha már az objektum nincs használva, akkor el kell tüntetni. Ezt a Java automatikusan csinálja a garbage collector segítségével, amely folyamatosan pásztázza a heap-et és ha talál egy objektumot, amellyel már nincs összeköttetés akkor azt kitörli.

Az egész dolog hosszan van szétírva, de egy sorban elvégezhető az egész:

