Debreceni Egyetem
Informatikai Kar

Egy telephelyi kommunikációs hálózat elemzése, tesztelése és technológiáinak ismertetése

Témavezető:
Dr. Almási Béla
Egyetemi docens

Készítette:
Ruszcsák Péter
Mérnök informatikus (BSc)

Debrecen
2009

Tartalomjegyzék

1. Bevezetés

Napjaink gazdasági világát az információáramlás nagymértékben befolyásolja. Egy versenyképes vállalatnak elengedhetetlen szükséglete egy jól működő kommunikációs hálózat. Manapság a vállalati hálózatokra egyre nagyobb terhet rónak, bővül a hálózati szolgáltatások palettája (távoli adatbázisok, IP telefónia, hálózati fájltárolás, videó konferencia, levelezés, stb.). A hatékony információ és forgalom áramláshoz jól megtervezett, biztonságosan működő hálózat szükséges. A hálózat legkisebb mértékű meghibásodása is nagy mértékű kiesést jelenthet. A megfelelő hibatűrés érdekében alkalmazhatóak a redundáns felépítésű rendszerek, melyek ugyan drágábbak, de a hálózat kiépítésével járó többlet költségek elenyészőek a hálózat meghibásodásának következtében várható kiesésekhez képest.

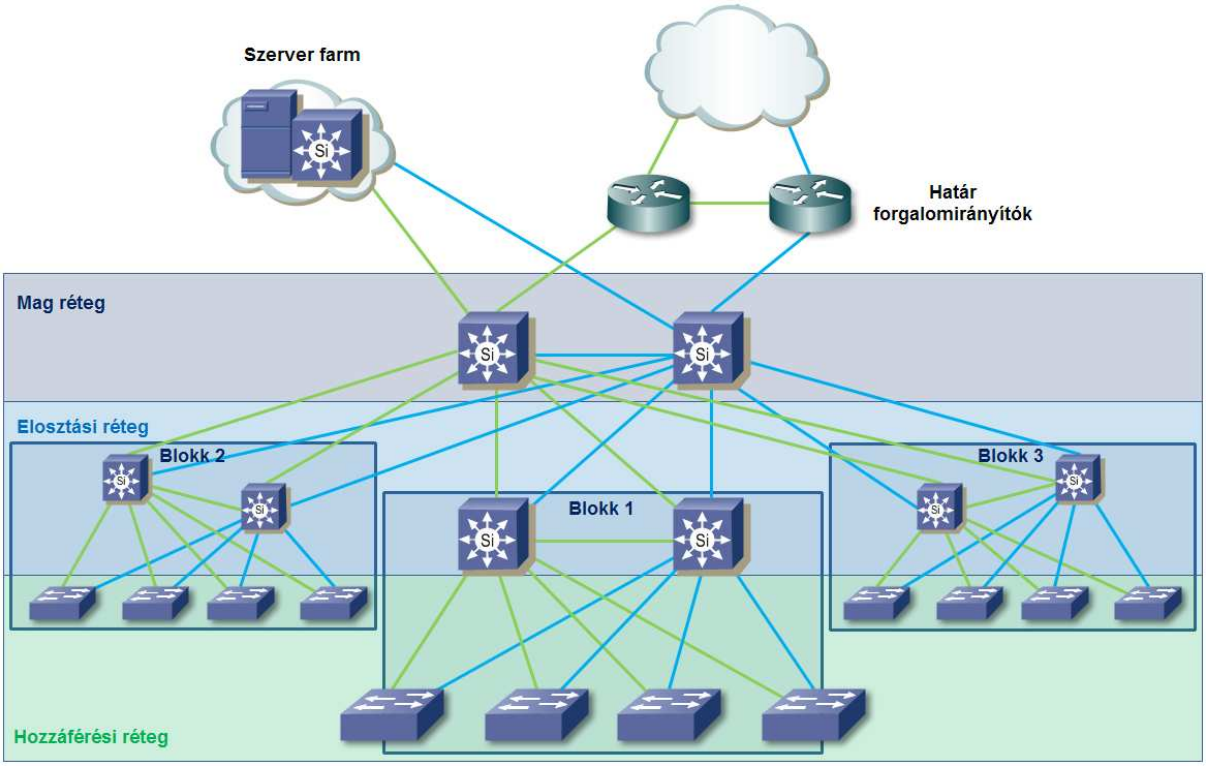
1.1 A dolgozat célja

A telephelyi hálózatokban, azon belül is a kapcsolt LAN hálózatok esetében sokféle technológia vehet részt a működésben. Jelenlegi munkahelyemen lehetőségem nyílt részletesebben megismerni az ilyen jellegű technológiákat. Dolgozatom célja, hogy bemutassam ezek működést, felépítését. A technológiák részletesebb tanulmányozása során – mely elengedhetetlen a hálózatok működésének megismerése érdekében – teszthálózatot építettem, amelyen tesztelni tudom a protokollok működését. A munkám során tapasztalt problémákat szimuláltam az általam készített teszthálózatban. Dolgozatomban ismertetem ezeket a problémákat, és hibaelhárítási folyamatát, valamint kitértek az alapvető hálózat biztonsági technológiákra, amelyek védelmet nyújtanak a belső támadások ellen.

A dolgozat olvasásához szükséges egy bizonyos szintű jártasság a hálózatok témakörében.

2. Telephelyi hálózat felépítése

Az 1. ábrán egy telephelyi hálózat felépítését láthatjuk. A telephelyi hálózat általában egy korlátozott kiterjedésű földrajzi területen helyezkedik el és egy vállalat számára nyújt kommunikációs szolgáltatásokat.



1. ábra: Egy telephely (campus) három rétegű hálózati topológiája

A felépítésben a redundancia nagy szerepet játszik. A hálózatban redundáns eszközök és redundáns kapcsolatok biztosítják a folyamatos rendelkezésre állást. A telephely különböző blokkjai egy-egy épületet reprezentálhatnak, amelyet egy gerinchálózat kapcsol össze. A szerver farm eszközei az egész telephely számára nyújtanak szolgáltatásokat, ezért azok elhelyezése általában központosított. A határ forgalomirányítók a telephelyen kívüli kommunikációt teszik lehetővé a belső hálózat részére. A topológia hierarchikus, háromrétegű felépítést követ, amelynek előnyei a forgalom hatékony áramlása, könnyű kezelhetőség és méretezhetőség.

2.1 Háromrétegű hálózati topológia

Telephelyi hálózatunkon különböző típusú forgalom haladhat attól függően, hogy egy felhasználó milyen szolgáltatást vesz igénybe. A forgalom három típusát különböztetjük meg.

- Helyi – ha a szolgáltatás és a felhasználója azonos szegmensbe helyezkedik el
- Távoli – ha a szolgáltatás és a felhasználója különböző szegmensbe helyezkedik el
- Vállalati – ha a szolgáltatás egy központi helyen helyezkedik el

A hatékony forgalom kezelés érdekében a forgalmi típusokat el kell szeparálnunk, amit a hálózat hierarchikus megtervezésével érhetünk el. Logikai és fizikai funkciók alapján három rétegre oszthatjuk hálózatunkat, melyek a hozzáférési réteg, az elosztási réteg és a mag réteg.

2.1.1 Hozzáférési réteg (Access layer)

A hozzáférési réteg az a pont ahol a végfelhasználók csatlakoznak a hálózathoz. A következők jellemzik:

- Alacsony port-költség
- Nagy mennyiségű port-szám
- Skálázható felszálló (uplink) összeköttetések a magasabb rétegek felé
- Felhasználói funkciók, mint VLAN tagság, forgalom és protokoll szűrés, valamint QoS

2.1.2 Elosztási réteg (Distribution layer)

A hozzáférési réteg kapcsolatot biztosít a hozzáférési és a mag réteg között. A következők jellemzik:

- Hozzáférési eszközök forgalmának egyesítése
- Harmadik rétegbeli csomagkezelés
- Biztonsági és házirend alapú összekapcsolási funkciók (hozzáférési listák, csomagszűrés)
- QoS funkciók
- Skálázható és rugalmas nagy sebességű kapcsolat a mag és hozzáférési réteg irányába

2.1.3 Mag réteg (Core layer)

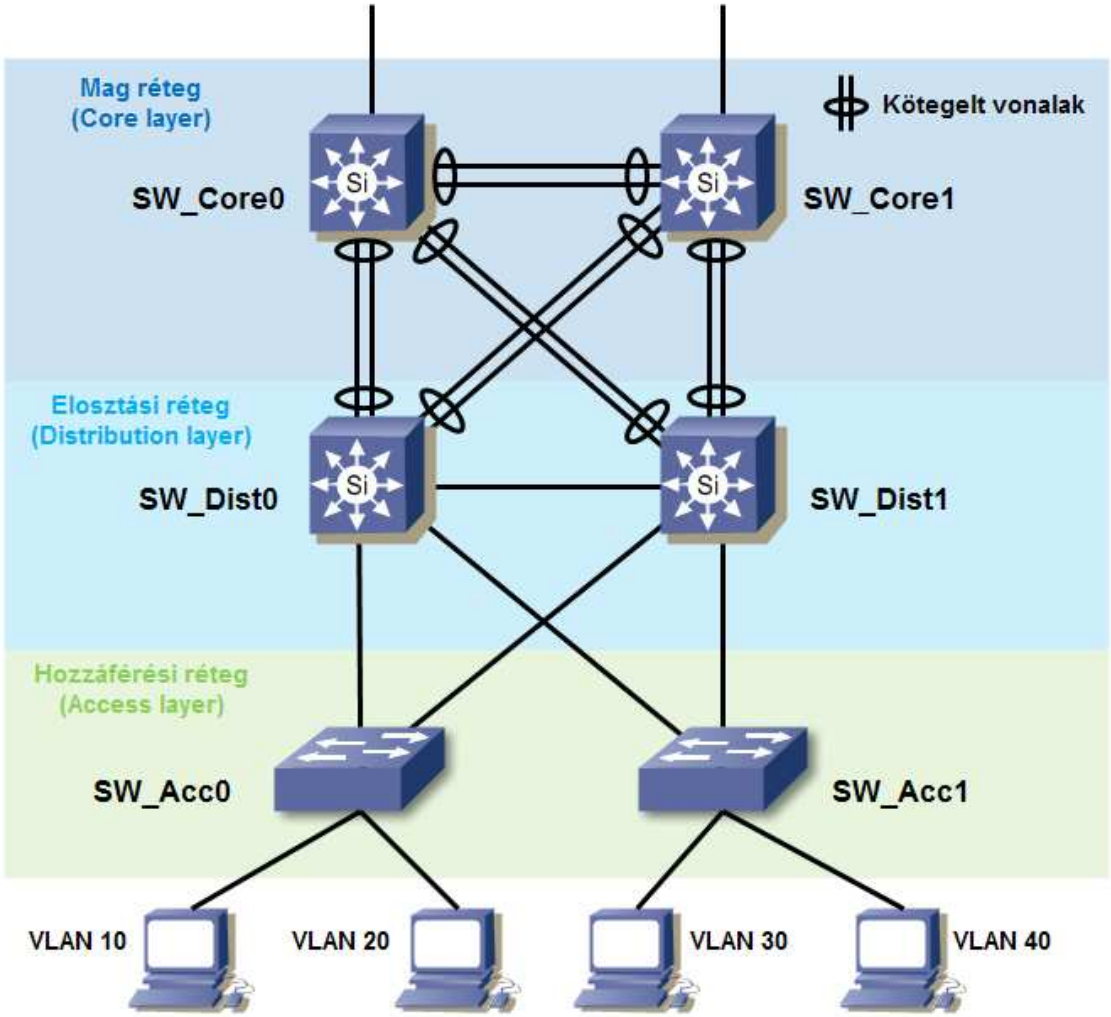
A mag réteg kapcsolatot biztosít a hozzáférési rétegek között. Mivel e rétegnek a teljes telephely nagy mennyiségű adatát kezelnie kell, a lehető leghatékonyabban kell kapcsolnia a forgalmat. A réteg eszközei biztosítanak kapcsolatot a szerver farm és a telephelyen kívüli kommunikációt biztosító határ forgalomirányítók felé. A következők jellemzik:

- Nagy teljesítményű kapcsolatok
- Nincs költséges és felesleges csomag manipuláció
- Redundancia és rugalmasság a magas rendelkezésre állás érdekében
- Magas szintű QoS funkciók

([1] 1. fejezet)

2.2 Teszthálózat bemutatása

A dolgozatban szereplő technológiákat alkalmazva felépítettem egy teszthálózatot, amelyen tanulmányozom és tesztelem a hálózat tulajdonságait. A teszthálózatot felhasználom a harmadik fejezetben, ahol az érintett technológiák működését mutatatom be az eszközök konfigurációinak felhasználásával. Továbbá felhasználom a negyedik és ötödik fejezetben, tesztelés és hibaelhárítás bemutatásának céljára.



2. ábra: A teszhálózat felépítése

2.2.1 Teszthálózat felépítése

A hálózatban a redundancia fontos tényező a folyamatos rendelkezésre állás érdekében. A hálózati elemek meghibásodhatnak, ezért szükséges tartalék eszközöket és vonalakat alkalmaznunk. A topológia háromrétegű hierarchikus felépítést követi, ahol minden rétegnek megvan a saját szerepe. A hozzáférési rétegben két második rétegbeli kapcsolót alkalmaztam. Gyakorlati környezetben az eszközök az épület egy-egy szintjének biztosíthatnak elérést a hozzá csatlakozó állomások számára. A felette lévő elosztási réteg nevéből is adódóan elosztási célokat szolgál, például lehet ez egy épületen belüli elosztás, a különböző szintek között. A rétegben a redundancia megvalósítása érdekében két kapcsolót alkalmaztam, amelyek redundánsan csatlakoznak a szomszédos rétegek eszközeivel. A harmadik mag réteg összekapcsolja a hálózat külön álló blokkjait és hozzáférést nyújthat a közös erőforrásokhoz és határeszközök felé. A hozzáférési és mag rétegben harmadik rétegbeli kapcsolókat használtam, mivel a rájuk hárult feladatokat egy hagyományos második rétegbeli kapcsoló nem képes ellátni.

3. Technológiák bemutatása

A hálózatok megértéséhez elengedhetetlen a technológiák részletesebb ismerete. Jelen fejezet célja, hogy feltárja a technológiákkal kapcsolatos kérdéseket, bemutassa a működési elvüket. Az egyes alfejezetek végén ismertetem a technológiák alkalmazásának módját és működését az általam épített teszthálózatban.

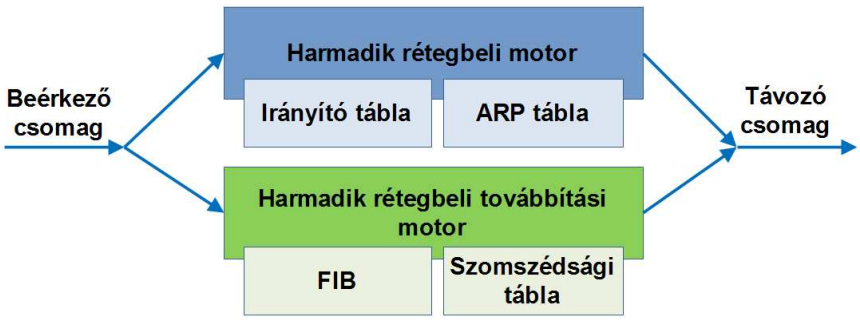
3.1 Többrétegű kapcsolás

A többrétegű kapcsolóban ötvözve van a második rétegbeli kapcsolás és a harmadik rétegbeli forgalomirányítás. Segítségével többrétegű kapcsolási eljárást használhatunk, amely hatékonyabbá teszi csomagjaink továbbítását. A többrétegű kapcsolás egy irányítási processzort és egy kapcsolási motort használ fel. Az első új csomag az irányítási processzor segítségével lesz továbbítva, majd a további csomagokat a jóval gyorsabb kapcsolási motort használja fel.

3.1.1 Cisco Express Forwarding (CEF)

A többrétegű Catalyst kapcsolók egy úgynevezett Cisco Express Forwarding (CEF) technológiát használnak a hatékony csomagtovábbításra.

A CEF eljárást használó többrétegű kapcsolók két alapvető részből állnak. Egy harmadik rétegbeli motorból, amely felépíti az irányítási információkat. Továbbá egy harmadik rétegbeli továbbítási motorból, amely az információk felhasználásával hardveresen kapcsolja a csomagokat (2. ábra). (A részfejezet a [1] 12. fejezet alapján készült)



3. ábra: Többrétegű kapcsolás

3.1.1.1 Forwarding Information Base (FIB)

A 3. rétegbeli motor tartja karban az irányítási információkat, származzon az dinamikus irányító protokolltól vagy statikus útvonalból. A 3. rétegbeli továbbítási motor által használt FIB egy olyan tábla, amelyben az irányító táblától származó információk szerepelnek specifikusságuk szerint sorba rendezve. A FIB dinamikus, mivel amint az irányítótáblában változás történik, a FIB is értesül a változásról és egyezteti tábláját.

Abban az esetben, ha egy csomag nem kapcsolható a FIB tábla alapján, akkor a 3. rétegbeli motor fogja elvégezni a továbbítást.

3.1.1.2 Szomszédsági tábla

A szomszédsági tábla a FIB-nek szolgáltat második rétegbeli információkat. A tábla az ARP táblából épül fel és tartalmazza azon csomópontok fizikai címeit, amelyek egy ugrásnyira elérhetők az eszközünktől.

Abban az esetben, ha a szomszédsági tábla nem tartalmazza a szükséges információkat, akkor a 3. rétegbeli továbbítási motor nem képes továbbítani a csomagot és az a 3. rétegbeli motorhoz kerül. A 3. rétegbeli motor ARP kérést küld, hogy begyűjtse az IP-cím eléréséhez szükséges fizikai címet. Amíg a FIB bejegyzés az ARP eredményre vár, az ugyanerre a célja igyekvő újabb csomagok azonnal eldobásra kerülnek, hogy a várakozási sor ne teljen meg és váljon a motor túlterhelté. Miután a megfelelő fizikai cím bekerült az ARP táblába, a szomszédsági táblába is megtanulja.

3.2 Virtuális helyi hálózatok (VLAN)

VLAN-okat arra használjuk, hogy egy nagy telephelyi hálózatot több kisebb szórási tartományra osszunk szét. Ennek előnye az, hogy minimalizáljuk a szórási forgalom kiterjedését egy logikai szegmensre.

3.2.1 A VLAN-ok szerepe

A kapcsolt hálózatokat VLAN-okra oszthatjuk szét. Definíció szerint a VLAN egy egyetlen szórási tartomány. Ha egy VLAN-ba tartozó eszköz szórási üzenetet küld, azt minden eszköz megkapja, amelyek ugyan azon VLAN-ba tartoznak. Azonban, ha egy eszköz egy másik VLAN-ba csatlakozik, akkor nem fogja megkapni ezt a szórási üzenetet.

Egy VLAN-ba tartozó állomások egy logikai hálózati szegmensen keresztül kommunikálnak egymással. Egy fizikai szegmensbe tartozó eszközöknek egy fizikai kábel szegmensbe kell csatlakozniuk. Azonban, a VLAN tagjai bárhol elhelyezkedhetnek egy telephelyi hálózatban, mivel a kapcsolókat konfigurálhatjuk egy VLAN térképpel, ami logikai kapcsolatot biztosít a VLAN tagjainak. ([5] 8. modul)

3.2.2 VLAN tagság

Két lehetőségünk van arra, hogy egy kapcsolón VLAN-okat konfiguráljunk:

- Statikus VLAN konfiguráció
- Dinamikus VLAN konfiguráció

A statikus VLAN-ok rögzített port-VLAN összerendelésen alapulnak, egy adott kapcsoló portot rendelünk hozzá egy bizonyos VLAN-hoz. Egy végfelhasználói eszköz akkor válik a VLAN tagjává, ha a fizikailag a porthoz csatlakozik. Ebben az esetben nincs szükség protokollokra, az állomás automatikusan taggá válik. Általában az eszköznek nincs is arról tudomása, hogy a VLAN létezik, az számára transzparens.

A dinamikus VLAN-ok esetében a MAC-címek alapján biztosítják a VLAN tagságot az állomásoknak. Amikor egy eszköz a kapcsoló portjára csatlakozik, a kapcsoló egy adatbázisból keresi ki, hogy az eszközt milyen VLAN-hoz rendelje. Az adatbázisban

adminisztrátorok a felhasználó MAC-címéhez tudják rendelni az adott VLAN-t a VLAN Tagsági Vezérlő Kiszolgáló (VMPS) segítségével.

3.2.3 VLAN Trönk

Egy kapcsoló hozzáférési portja egyetlen VLAN-ba tud tagságot biztosítani az eszköz számára. Tegyük fel, hogy két kapcsolón ugyan azok a VLAN-ok szerepelnek. Ha az egyik kapcsolóra csatlakoztatott állomás kommunikálni szeretne ugyan abban a VLAN-ba szereplő másik kapcsolóra csatlakoztatott állomással, akkor a két kapcsoló közt egy vonalat kell kiépíteni az adott VLAN forgalmához. Minden egyes VLAN-nak külön vonalat kellene kiépíteni, ami nem lenne túl költséghatékony módszer.

A trönk vonal azonban több VLAN forgalmát is tudja szállítani egyetlen kapcsoló porton keresztül. A trönk vonalat általában egy másik kapcsolóval vagy egy forgalomirányítóval való összeköttetésre használjuk. Ilyen típusú vonal nem egy adott VLAN-hoz van társítva, hanem több, illetve az összes aktív VLAN forgalmat is szállítani tudja egyetlen fizikai trönk vonalon.

Ahhoz, hogy a trönk vonalon több VLAN forgalma is haladhasson, a kapcsolónak meg kell jelölnie a kereteket, hogy tudja pontosan melyek azok a VLAN-ok, amiket küld vagy fogad a vonalon. Amikor egy keretnek trönk vonalon kell áthaladnia, a fejlécébe egy VLAN azonosító kerül (802.1Q protokoll használata esetén). Amint a vonal másik végén lévő kapcsoló megkapja a keretet, megvizsgálja annak VLAN azonosítóját, majd leválasztja a keret fejlécéről azt, ezután a megfelelő portra irányítja tovább a keretet. ([5] 9. modul)

Két típusú VLAN azonosító módszert használhatunk:

- Inter-Switch Link (ISL) protokoll
- IEEE 802.1Q protokoll

3.2.3.1 Inter-Switch Link (ISL)

Az ISL protokoll egy Cisco szabadalmazott VLAN azonosító eljárás, amely a teljes keretet becsomagolja. A protokoll-t általában Ethernet médián alkalmazzák, viszont támogatja a Token Ring-et, FDDI-t és az ATM-et is.

Amikor egy keretnek trönk vonalon kell áthaladnia az ISL egy 26 byte-os fejrésszel és egy 4 byte-os farok részel toldja meg. A fej egy 15 bit-es azonosítót (VLAN ID) tartalmaz, amivel

azonosítja a VLAN-t. A farok részben pedig egy CRC érték foglal helyet, ami az újonnan becsomagolt keret adatintegritás ellenőrzésért felel.

3.2.3.2 IEEE 802.1Q

A 802.1Q (dot1q) is képes VLAN azonosításra trönk vonalon. Azonban ez a protokoll nyílt szabvány, amely lehetővé teszi azt, hogy trönk kapcsolatot alakítsunk ki különböző gyártótól származó eszközök között is.

A dot1q más eljárást használ, mint az ISL, nem csomagolja be az egész keretet, hanem a keretbe beágyaz egy azonosító (VLAN ID) címkét. Ez egy 4 byte-os mező és rögtön a forrás cím után helyezkedik el.

802.1Q trönk használ egy úgynevezett natív VLAN-t, amelybe az azonosítatlan (VLAN ID nélküli) keretek kerülnek. A trönk vonal mindkét végén egyeznie kell a konfigurált natív VLAN-nak, különben problémák merülhetnek fel a kapcsolatban (alapbeállításként az 1-es a natív VLAN).

3.2.4 Dynamic Trunking Protocol (DTP)

Ahhoz, hogy egy port trönk-é váljon konfigurálhatjuk statikusan vagy létrejöhetnek dinamikusan is, a Cisco specifikus DTP protokoll segítségével. A protokoll két kapcsoló között trönk vonalat alakít ki. Egy fizikai kapcsolóport több típusú módban lehet, attól függően, hogy akarjuk kialakítani a kapcsolatot:

- Access – A port egyetlen VLAN-hoz rendelhető hozzá (nem jön létre trönk vonal).
- Trunk – A port trönk vonalat alakít ki szomszédos porttal.
- Dynamic Desirable – Aktívan trönk vonalat próbál kialakítani szomszédjával. A trönk vonal akkor fog létrejönni, ha a szomszéd port *trunk*, *dynamic desirable* vagy *dynamic auto*-ként van beállítva.
- Dynamic Auto – Passzívan vár a szomszédra, hogy trönk vonalat alakítsanak ki. A trönk vonal akkor fog létrejönni, ha a szomszéd port *trunk* vagy *dynamic desirable*-ként van beállítva.

([2] 2. fejezet)

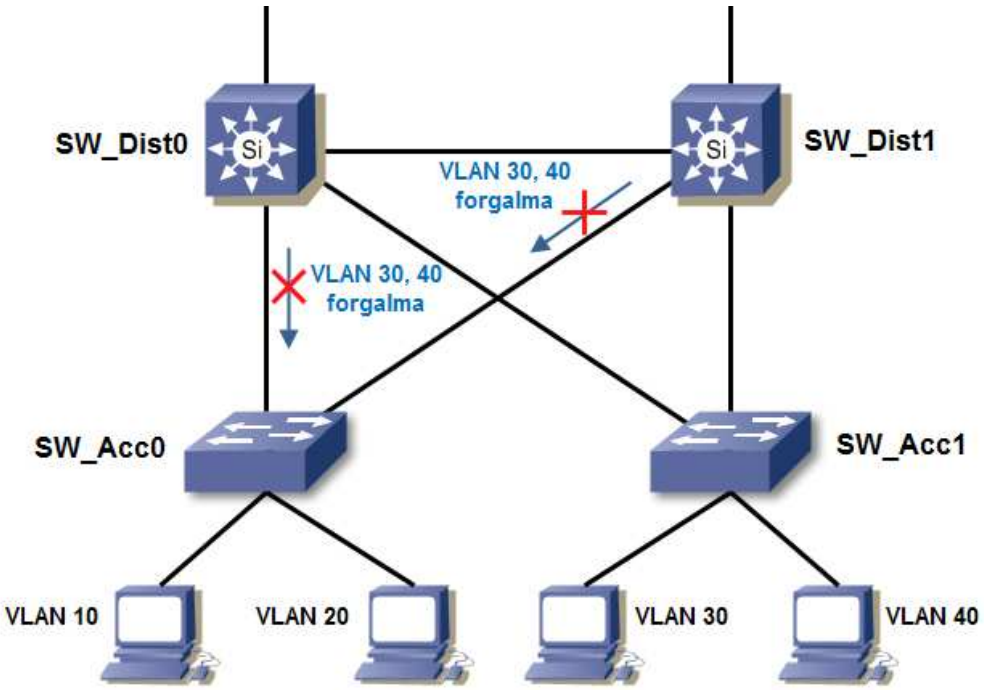
3.2.5 VLAN-ok implementálása a teszthálózatban

A teszthálózatban szereplő állomások különböző munkacsoportokat reprezentálnak. Annak érdekében, hogy a szórási üzenetek csak a saját munkacsoporton belül áramoljanak, logikailag szegmentáltam a hálózatot VLAN-ok létrehozásával.

Ahhoz, hogy az állomások részt vegyenek a VLAN-okba a hozzáférési kapcsolók megfelelő portjait statikusan hozzárendeltem az adott VLAN-okhoz.

```
!
interface FastEthernet0/5
switchport access vlan 10
switchport mode access
```

A kapcsolók között szükség volt trönk vonalak létrehozására, hogy több VLAN forgalma is haladhasson a vonalakon. Ezért a megfelelő portokat trönk módba helyeztem. Azonban SW_Acc0 és SW_Acc1 kapcsolók nem nyújtanak az összes VLAN-nak hozzáférést. Annak érdekében, hogy a vonalak ne terhelődjenek felesleges forgalommal, csak a szükséges VLAN-okat engedélyeztem a trönk vonalakon (4. ábra).



4. ábra: A felesleges VLAN forgalom letiltása

```
!
interface FastEthernet0/2
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,10,20
 switchport mode trunk
```

3.3 VLAN Trunking Protocol (VTP)

Az előző fejezetben bemutatott VLAN konfigurációk könnyen megoldhatók kisebb hálózatokban. Azonban nagyszámú kapcsolók esetén a kiépítés és a karbantartás már nem egyszerű dolog.

A Cisco kifejlesztett egy eljárást, amivel kezelni tudjuk VLAN-ainkat az egész telephelyi hálózatban. Ez a VLAN Trunking Protocol, amely lehetőséget ad arra, hogy a kapcsolók VLAN információkat cseréljenek egymással. A VTP-vel egy központi helyről tudunk VLAN-okat létrehozni, törölni és átnevezni az összes érintett kapcsolón. (A részfejezet a [1] 6. fejezet alapján készült)

3.3.1 VTP tartományok

A VTP a hasonló VLAN érdekeltségű tartományokra osztja szét a hálózatot. Egy kapcsoló csak egy VTP tartományba tartozhat és csak e tartományon belül oszthatja meg információit a többi kapcsolóval.

A kapcsolók a VTP tartományon belül számos attribútumot oszthatnak meg szomszédjaikkal. Minden attribútum információkat tartalmaz a tartományról, mint pl. revíziós szám, ismert VLAN-ok és adott VLAN paraméterek. Amikor egy VLAN-t hozzáadunk egy kapcsolóhoz, akkor az eszköz értesíti a többi kapcsolót a változásról egy VTP hirdetménnyel. A hirdetmény lehetőséget ad arra, hogy a többi eszköz az új VLAN forgalmát tudja fogadni a trönk portján keresztül.

3.3.2 VTP módok

Ahhoz, hogy egy kapcsoló a VTP tartományban részt vehessen, konfigurálnunk kell, hogy mely működési módban szerepeljen. A VTP mód meghatározza, hogy a kapcsoló milyen módon dolgozza fel a hirdetett VTP információkat. A következő módok közül választatunk:

Szerver mód: A VTP szervereknek teljes körű joga van VLAN-ok kreálására és módosítására a VTP tartományon belül. Minden VTP információt hirdet az adott tartományban, míg minden kapott információval szinkronizálja saját adatbázisát. Alapállapotban minden kapcsoló szerverként van konfigurálva. Minden egyes VTP tartományban legalább egy szervernek szerepelnie kell, amely képes arra, hogy VLAN-okat kreáljon, módosítson, illetve töröljön.

Kliens mód: A VTP klienseken nem engedélyezett a VLAN-ok létrehozása, cseréje és törlése. Ehelyett, figyelik a többi kapcsoló VTP hirdetményeit és ezek szerint módosítják saját VLAN adatbázisukat. A kapott VTP információkat a tartományban továbbküldik a szomszédjainak a trönk vonalakon, vagyis közvetítőként is működnek.

Transzparens mód: A VTP transzparens kapcsolókon lehet VLAN-t kreálni, módosítani, törölni, de azok nem szerepelnek a VTP folyamatban. A kapcsoló nem hirdeti VLAN konfigurációját és nem szinkronizálja adatbázisát a kapott hirdetményekhez. A VTP 1-es verziója esetén a fogadott hirdetményeket nem küldik tovább trönk vonalaikon, csak a 2-es verziótól továbbítják azokat a szomszédoknak és viselkednek közvetítőként.

3.3.3 VTP hirdetmények

A VTP hirdetmények multicast üzenetként kerülnek elküldésre. Amikor egy kapcsoló kap egy ilyen keretet, a multicast cím alapján azonosítani tudja, és ez által feldolgozza.

Alapértelmezetten a VTP tartományok nem biztonságosan kerülnek elküldésre. Viszont jelszavak segítségével csak azok a kapcsolók képesek feldolgozni a VTP információkat, amelyeknél a megfelelő kódszó van konfigurálva.

A hirdetmények tartalmaznak egy revíziós számot, amely az adott hirdetmény életkorára, és az elkövetett változtatásokra utal. Ez a szám 0-ról indul és minden VLAN változásnál eggyel növekszik. Egy VTP tartományban minden kapcsoló eltárol egy revíziós számot, amit a legutolsó hirdetményből kapott.

Amikor új hirdetmény kerül feldolgozásra a kapcsoló megvizsgálja a revíziós számát és összeveti sajátjával:

- Ha nagyobb, akkor felülírja VLAN adatbázisát az új VLAN információkkal és továbbküldi szomszédjainak.
- Ha egyenlő, akkor elveti.
- Ha kisebb, akkor válaszol a saját frissebb adatbázis információjával.

A revíziós szám miatt fontos, hogy körültekintően járjunk el, amikor egy új kapcsolót csatlakoztatunk VTP tartományunkba. Előfordulhat, hogy az új eszközön lévő revíziós számnak nagyobb az érteke, mint a tartományban szereplő aktuális érték. Így az új kapcsoló hirdetménye fogható fel a legfrissebbnek, amit többi kapcsoló elfogad és felülírja meglévő VLAN adatbázisát. Ez nagyon sok esetben vezet problémákhoz.

A revíziós szám az NVRAM-ban van tárolva és csak a következő esetekben nullázódik:

- Ha a kapcsolót transzparens módba állítjuk, majd vissza szervernek.
- Ha a VTP tartomány nevét egy nem létezőre állítjuk, majd vissza az eredetire.

A hirdetményeknek három formája lehet abból adódóan, hogy honnan származnak:

Összegző hirdetmény – A VTP szerver kapcsolók 300 másodpercenként küldik és minden alkalommal, amikor a VLAN adatbázisban változás történik. A *összegző hirdetmények* olyan információkat tartalmaznak, mint a VTP verzió, tartomány neve, revíziós szám, időbélyeg, és a következő *részhirdetmény* száma. VLAN konfigurációs változásoknál az *összegző hirdetményt* követi egy vagy több *részhirdetmény* a pontos VLAN adatokkal.

Részhirdetmény – A VTP szerver kapcsolók küldik VLAN konfigurációs változás után. Pontos adatokat tartalmaz a VLAN változásokról, mint kreálás, törlés vagy felfüggesztés, aktiválás, VLAN névváltozás, VLAN MTU változása

Kliens által kért hirdetmény – A VTP kliens kérhet VLAN információkat, amelyekben hiányt szenved. Például egy kliens kapcsoló újraindult és a VLAN adatbázisa törlődött vagy hallott egy összegző hirdetményről, amelynek nagyobb a revíziós száma a sajátjánál. Miután a kliens hirdetményt kért, a VTP szerver *összegző* és *részhirdetményekkel* válaszol, hogy felfrissítse a kliens adatbázisát.

3.3.4 A Virtual Trunking Protocol implementálása a teszthálózatban

A VLAN-ok konfigurálását a VTP segítségével hajtottam végre, hogy a kapcsolók VLAN adatbázisai egyszerűbben kezelhetők legyenek. A VLAN-okat elegendő volt a VTP szerverek egyikén (SW_Dist0, SW_Dist1) létrehozni és a többi azonos VTP tartományban lévő kapcsolók megtanulták azokat. A tartományban két VTP szervert konfiguráltam a redundancia megvalósítása érdekében.

3.4 Spanning Tree Protocol (STP)

3.4.1 Redundáns hálózat

Napjainkban egyre nagyobb mértékben támaszkodnak a számítógépes hálózatokra. A vállalatok elvárják az állandó rendelkezésre állást hálózatukban. Ennek eléréséhez olyan hálózati topológiát kell terveznünk, amely egy-egy eszköz meghibásodása esetén is működőképes marad.

A hibatűrés redundanciával érhető el. A nagy megbízhatóságú hálózatokba redundáns útvonalakat és készülékeket kell elhelyeznünk. Egy redundáns topológiában egyetlen elem meghibásodása nem okoz teljes körű leállást. Amint egy eszköz vagy vonal meghibásodik egy másik eszköz vagy útvonal veszi át a helyét.

A redundáns kapcsolatok miatt viszont irányítási hurkok keletkeznek a hálózatban, amelyek szórási viharokhoz és a szolgáltatások leállásához vezethetnek. A szórásos kereteket a kapcsolók az összes portjukon kiküldik, kivéve amelyen az beérkezett. A redundáns vonalakon keresztül a kapcsolók újra és újra megkapják egymástól a szórási üzeneteket. Ezt a jelenséget szórási viharnak nevezzük, és mindaddig folytatódik, amíg az egyik eszközt ki nem száll a hálózat működéséből. A kapcsolók és a végponti eszközök folyamatosan terhelté válnak a szórásos keretek feldolgozása miatt. A túlterheltség miatt az eszközök nem lesznek képesek a felhasználói adatforgalom továbbítására. Ezért a hálózat látszólag leáll vagy drasztikusan lelassul. ([5] 7. modul)

3.4.2 Feszítőfa protokoll (Spanning Tree Protocol)

Az irányítási hurkok azért jönnek létre, mert a redundáns kapcsolók nem foglalkoznak egymás jelenlétével. Ezért fejlesztették ki a feszítőfa protokollt, hogy megoldja ezt a problémát és ki tudjuk használni a redundáns kapcsolatok előnyeit. A protokoll segítségével a kapcsolók megismerik egymást és ki tudnak alakítani egy logikailag hurokmentes topológiát.

Az STP kommunikál az összes hálózatba csatlakozott kapcsolóval. Minden kapcsolón lefut a feszítőfa algoritmus a szomszédoktól kapott információk alapján. Az algoritmus kiválaszt egy referencia pontot (gyökérponti híd) a topológiában és kiszámítja az összes elérhető redundáns útvonalat ehhez a ponthoz. Miután megtalálta az összes útvonalat, az algoritmus kiválasztja azokat, amelyeken keresztül továbbítódni fognak a keretek, a többit pedig blokkolja. Azonban, ha egy továbbításra használt port meghibásodik vagy szétkapcsolódik, akkor a feszítőfa algoritmus újraszámolja a topológiát és a megfelelő portok újra aktiválódhatnak. ([1] 9. fejezet)

3.4.2.1 Bridge Protocol Data Unit (BPDU)

Az adatok melyek szükségesek a feszítőfa működéséhez BPDU-k formájában cserélődnek. Mivel a kapcsolók nem tudják, hogy milyen eszközök vannak körülöttük, ezért multicast üzenetként küldik ki a BPDU kereteket. Célcímnek a jól ismert STP multicast címet használják a 01-80-c2-00-00-00-t.

Mező	Bájtok száma
Protokoll azonosító	2
Verzió	1
Üzenet típusa (Konfigurációs vagy TCN BPDU)	1
Flagek	1
Root Bridge ID	8
Root útvonal költség	4
Küldő kapcsoló ID	8
Port ID	2
Üzenet életkora	2
Maximum Age	2
Hello Time	2
Fordward Delay	2

5. ábra: A BPDU üzenetek felépítése

A BPDU-nak két típusa létezik:

- Konfigurációs BPDU, ami a feszítőfa kiszámításához használandó.
- Topológia változást közlő (TCN) BDPU, amely a hálózati topológia változását jelzi.

A BDPU üzenetek cseréjének az a célja, hogy megválasszák a referencia pontot, amely az alapja a stabil feszítőfa topológiának. Az üzenet mezői, mint az azonosító, útvonal költség, idő értékek a kapcsolókhoz kötődnek. Ezek együttese segítségével érhetjük el azt, hogy a hálózat kapcsolói összefoghatók lesznek egy feszítőfába és ki tudják választani a közös referencia pontot. Alapbeállításként a BDPU-k 2 másodpercenként küldődnek ki a kapcsoló minden portján, hogy az adott topológia információkat kicseréljék egymással és gyorsan felfedezzék a hálózati hurkokat.

3.4.2.2 Root Bridge

A kapcsolók által közösen választott referencia pontot Root Bridge-nek (gyökérponti híd) nevezzük. Ahhoz, hogy a választási folyamat végbemenjen, szükség van minden kapcsolónak egy azonosítóra. Ez az azonosító a Bridge ID és arra használják az eszközök, hogy azonosítani tudják egymást. A Bridge ID 8 bájt hosszú és a következő mezőket tartalmazza:

- Híd prioritás (2 bájt): Értéke 0 és 65,535 között lehet és alapértelmezetten 32,768 minden Cisco Catalyst kapcsolón.
- MAC-cím (6 bájt): Ez az eszköz fizikai címe. A gyártó által lett beleégetve az eszközbe, egyedi és nem változtatható meg. Általában a cím megtalálható az eszköz hátlapján.

A kapcsolók első indulásuk után saját magukat nevezik ki Root Bridge-nek mivel még nem tudnak a többi eszköz kilétéről. A kapcsolók között egy választási folyamat megy végbe, hogy megválasszák a közös referencia pontjukat. Minden eszköz elkezdi küldeni saját BPDU-it és Root Bridge-ként saját magukat tüntetik fel. Amikor a kapcsolók megkapják többiek BPDU-ját megvizsgálják a Bridge ID-t és összevetik az általuk ismert Root Bridge ID-val. Ha az ID alacsonyabb (jobb), akkor kicseréli sajátját és az új Root Bridge ID-t fogja hirdetni. A választási eljárás végén a legjobb Bridge ID-val rendelkező kapcsoló fogja betölteni a Root Bridge szerepét. Abban az esetben, ha több kapcsolónak ugyan akkora a prioritása, akkor a kisebb MAC-címmel rendelkező eszköz fogja megnyerni a választást. ([1] 9. fejezet)

3.4.2.3 Port állapotok

A referencia pont megválasztása után, a nem root kapcsolóknak meg kell választaniuk az feszítőfában résztvevő portok szerepét. Ahhoz, hogy a port megtanulja szerepét néhány állapoton keresztül kell végighaladnia.

A következő állapotok léteznek:

Lezárás (Blocking): Minden port ilyen állapotban kezd. *Lezárt* állapotban a port nem fogad, küld kereteket és a kapcsoló nem kezdi el feltölteni CAM tábláját. Ehelyett BPDU-kat fogad a szomszédos kapcsolóktól, hogy megismerje őket. Ahhoz, hogy elkerüljük a kapcsolási hurkokat a portok ilyen állapotba kerülnek. Alapbeállításként 20 másodpercet tölt a port *lezárt* állapotban.

Figyelő (Listening): A fogadott BPDU-k alapján a feszítőfa eldöntötte, hogy a port részt vesz a kerettovábbításban. *Figyelő* állapotban a port még nem tud adatot küldeni, illetve fogadni. Azonban képes küldeni és fogadni BDPU-kat. A kapcsoló informálja szomszédjait, hogy aktívan részt vesz a feszítőfa folyamatban. Alapértelmezetten 15 másodpercet tölt a port *figyelő* állapotban.

Tanuló (Learning): A port felkészül a kerettovábbításra és a kapcsoló elkezdi feltölteni CAM tábláját, a rajta áthaladó keretek forráscímeinek megjegyzésével. A port továbbra is küld és fogad BDPU-kat. Alapértelmezetten 15 másodpercet tölt a port *tanuló* állapotban.

Továbbító (Forwarding): A port aktívan részt vesz a topológiában. Kereteket továbbít és küldi és fogadja a BDPU-kat.

Letiltott (Disable): Valójában ez nem egy feszítőfa port állapot. Ilyenkor a port adminisztratívan le van kapcsolva és nem vesz részt aktívan a feszítőfa folyamatban.

([4] 3. modul)

A 4. ábrán látható a port állapotainak folyamata.

6. ábra: Port állapotok folyamata

3.4.2.4 STP számlálók

Ahhoz, hogy a hálózat helyesen konvergáljon mielőtt kapcsolási hurok alakulna ki, az STP számlálókat használ. Ezek a számlálók is arra szolgálnak, hogy a kapcsolók megállapítsák szomszédjaik aktívak-e még és a MAC-tábla információk mikor évüljenek el.

A következő három STP számláló létezik:

Hello időintervallum (Hello time): Ennyi időközönként küld a Root Bridge konfigurációs BDPU-kat. Értéket minden kapcsolón lehetséges megváltoztatni. Azonban, az összes nem root kapcsoló átveszi az értéket a Root Bridge-n konfiguráltéval. Alapértelmezetten 2 másodperc az értéke.

Maximum Kor (Maximum Age): Ez az intervallum, ameddig a kapcsoló eltárolja a BPDU-t. Minden kapcsoló egy másolatot tart a legújabb BDPU-ról. Amint nem kap több BDPU-t az idő lejárta után eldobja, és a kapcsoló azt feltételezi, hogy topológia változás történt. Ezen felül a port ennyi időt tölt *lezárt* állapotban, mielőtt *figyelő* állapotba kerülne. Alapértelmezetten 20 másodperc az értéke.

Továbbítási késleltetés (Forward Delay): Ez az időintervallum, ameddig egy port *figyelő* és *tanuló* állapotban van. Alapértelmezetten 15 másodperc az értéke.

([1] 9. fejezet)

3.4.2.5 Port szerepek

Ahhoz, hogy a feszítőfa hurokmentes topológiát alakítson ki, meg kell határozni melyek azok a portok amelyek, továbbíthatják a forgalmat és melyeket kell tiltania. Ezért a nem root kapcsolók különböző szerepbe helyezik portjaikat a beérkező BDPU-k alapján. A kapcsoló a következő komponenseket vizsgálja meg a BPDU-ban:

- Legalacsonyabb útköltség
- Legalacsonyabb küldő Bridge ID (BID)
- Legalacsonyabb küldő port ID

A kapcsoló először az útköltséget vizsgálja meg, amelyet a vonal sebessége és a vonalak számából számol ki. A vonal költségét a szabványban lefektetett értékek határozzák meg (7. ábra). Mindig az a port kerül *továbbító* módba, amelynek a legkisebb a költsége az adott cél felé, a többi port pedig marad tiltott állapotban.

Vonal sávszélesség (Mbps)	4	10	16	45	100	155	622	1000	10000
STP költség	250	100	62	39	19	14	6	4	2

7. ábra: STP útvonal költségek

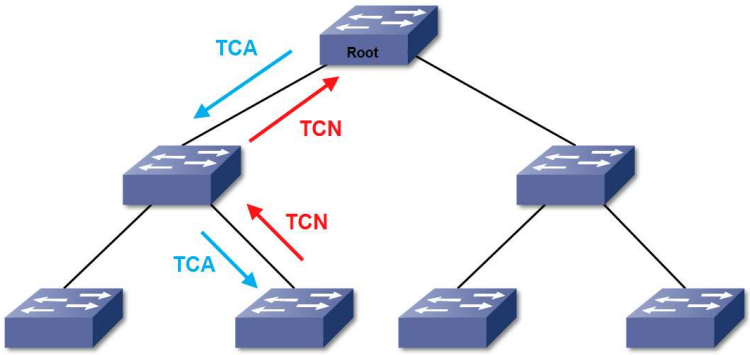
Abban az esetben, ha útiköltség és a küldő BID-je egyenlő, akkor a kapcsoló a kisebb értékű port ID-t veszi figyelembe a továbbításhoz, a többit tiltja.

Minden nem root kapcsolónak meg kell választani egy portot, amely a Root Bridge felé vezet. Ezt a portot Root Portnak nevezzük és az útvonal költségek alapján kerül választásra. Továbbá a feszítőfa minden szegmensbe választ egy úgynevezett kijelölt portot, amely továbbítja a forgalmat. A szegmens másik portja pedig lehet nem kijelölt port, amely továbbra is blokkolja a forgalmat vagy lehet root port, ha a Root Bridge-el szomszédos az eszköz. A Root Bridge összes portja kijelölt portként szerepel.

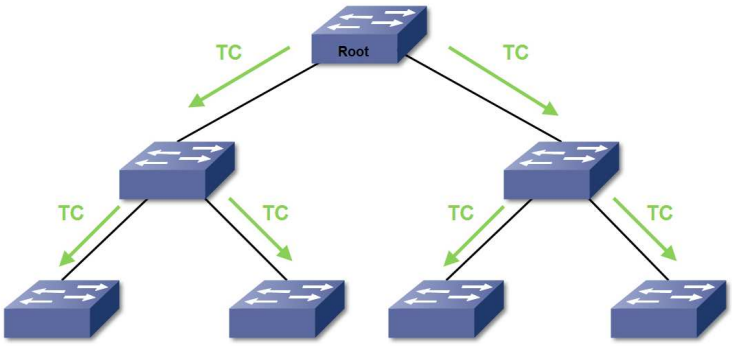
3.4.2.6 Topológia változás

Topológia változás akkor következik be, ha egy port *továbbító* állapotba kerül vagy *továbbító*, illetve *tanuló* állapotból *lezárt* állapotba kerül. Ilyenkor a kapcsoló, amely észlelte a változást egy TCN BPDU-t küld ki a root portján, hogy értesítse a Root Bridge-t. Ezek a BPDU üzenetek nem tartalmaznak adatokat, csak informálnak a változásról.

A változás bekövetkezése után a kapcsoló a *Hello Time* intervallum idejében folyamatosan küldi TCN BPDU-ját amíg nem kap visszaigazolást a szomszédjától. Amikor a szomszéd megkapja az üzenetet a változásról továbbküldi a Root Bridge irányába és elküldi visszaigazolását (TCA). Ez a folyamat addig folytatódik, amíg a Root Bridge is megkapja a TCN BDPU-t (8. ábra). Ezután Konfigurációs BPDU-ban beállítja a topológia változás flag-et, és elküldi az összes kapcsolónak a hálózatban. Ez az üzenet (TC) jelzi a topológia változást és minden kapcsoló lecsökkenti a MAC-táblájuk elévülési idejét. Így az eszköznek újra kell tanulnia az útvonalakat (9. ábra).



8. ábra: Topológia változás - első fázis



9. ábra: Topológia változás - második fázis

3.4.3 Az STP továbbfejlesztései

3.4.3.1 PortFast

Az alapértelmezett STP időzítés mellett egy portnak kezdetben 30 másodpercbe telik, hogy aktívan részt vegyen az adattovábbításban. A PortFast fejlesztéssel konfigurált hozzáférési port viszont képes egyből *lezárt* állapotból *továbbító* állapotba lépni, kihagyva a *figyelő* és *tanuló* állapotokat. Így egy hozzá csatlakoztatott végpont azonnal képes adatott küldeni és fogadni. Olyan portoknál használjuk, amelyek munkaállomáshoz vagy szerverhez csatlakoznak.

3.4.3.2 PVST

A PVST segítségével egy kapcsolón különböző STP példányokat hozhatunk létre minden egyes VLAN-nak. A kapcsolók azonosítójához hozzáadódik egy VLAN ID, amely lehetővé teszi, hogy minden VLAN-nak saját Root Bridg-e, root portjai és egyéb STP eszközei legyenek. A fejlesztéssel optimalizálhatjuk az adatforgalmat és terhelés eloszlást valósíthatunk meg a kapcsolók STP működése között.

3.4.4 Az STP védelme

Az STP-t használó hálózatok BPDU-kat használnak a kapcsolók közötti kommunikációra. Hálózatunkba támadó indokkal becsatlakoztathatnak egy idegen kapcsolót, amely BPDU-jaival megváltoztathatja és károsíthatja a feszítőfa topológiát. Az ilyen támadások ellen különböző védelmi mechanizmusokat fejlesztettek ki.

3.4.4.1 Root Guard

A Root Bridge-nek kulcsfontosságú szerepe van, mivel viszonyítási pontként szolgál a hálózat többi kapcsolója részére. Védelem nélkül azonban a támadó kapcsoló könnyen átveheti ezt a szerepet azáltal, hogy „nagyobb” BPDU-kat küld.

A Root Guard segítségével megakadályozhatjuk, hogy a rossz kapcsoló váljon a feszítőfa gyökérponti hídjává. Ha egy Root Guard-al konfigurált port kap egy BDPU-t, amelyben a hálózat legjobb BID-je van hirdetve, akkor a port „*root-inconsistent*” állapotba helyezi magát.

Ebben az állapotban a port csak küldhet és továbbíthat BDPU-t, fogadni nem képes azokat. Így megakadályozza az üzenetet küldő kapcsolót, hogy Root Bridge-é váljon. Amint a „jobb” BDPU-k küldése befejeződött, a port visszaáll a normál STP állapotába. ([4] 8. modul)

3.4.4.2 BPDU Guard

A PortFast lehetőséggel portunkat azonnal képessé tehetjük, hogy adatot továbbítson. Azonban, ha egy ilyen portra tévedésből vagy támadó okból egy kapcsolót csatlakoztatunk, kapcsolási hurkok alakulhatnak ki. A BDPU Guard ezt a hátrányt küszöböli ki oly módon, hogyha a port egy BPDU-t kap, akkor *error-disabled* állapotba kerül, lekapcsolódik.

A portok eredeti állapotába manuálisan kapcsolhatjuk vissza vagy használhatunk egy visszaállító időzítőt, amely automatikusan megteszi egy adott idő letelte után.

3.4.4.3 Unidirectional Link Detection (UDLD)

A kapcsolók kétirányú vonalakkal kapcsolódnak egymáshoz, ahol a forgalom mind a két irányba áramolhat (full duplex). Ha egy vonal fizikailag meghibásodik a kapcsolók észlelik a problémát és lekapcsoltnak nyilvánítják. Azonban előfordulhat, hogy a vonal oly módon hibásodik meg, hogy csak továbbítani vagy fogadni képes. Ilyenkor a kapcsolók úgy észlelik, mintha kétirányú maradna a vonal, miközben csak az egyik irányba halad a forgalom. Az ilyen vonalat unidirectional (egyirányú) link-nek nevezzük.

Az unidirectional link egy lehetséges veszélye az STP topológiának, mivel a vonal végén lévő port nem kap BDPU-kat. Tegyük fel, hogy egy portnak normál esetben *lezárt* állapotba kellene maradnia. Mivel a port nem kap BDPU-t, így azt feltételezi, hogy biztonságosan változtathatja STP állapotát és végül forgalmat fog továbbítani. Eredményként kapcsolási hurkot fog okozni a hálózatban.

Az UDLD az ilyen unidirectional link-ek felismerésére fejlesztették ki. A védelemmel konfigurált port időszakosan hello üzeneteket küld, amelyre válaszokat vár a vonal másik végén lévő eszköztől. Ha válaszok nem érkeznek meg a vonal unidirectional link-nek lesz nyilvánítva és lekapcsolódik.

3.4.4.4 Loop Guard

A Loop Guard fejlesztés megakadályozza, hogy egy port tévedésből lépjen *továbbító* állapotba. Ilyen eset akkor fordulhat elő, ha egy *lezárt* port valamilyen hiba folytán (pl. unidirectional link) nem kap tovább BDPU-kat. A port a *Max Kor* intervallum lejárta után *továbbító* állapotba lép és hálózati hurkot hozz létre.

A Loop Guard az ilyen problémák megoldására szolgál. Ha egy *lezárt* port meghatározott időn belül nem kap BDPU-t egy úgynevezett „loop inconsistent” lezárt állapotba kerül, mintsem *továbbítóba*. Amint a port ismét fogadja a BDPU-kat automatikusan visszakerül eredeti helyzetébe. A védelmet javasolt minden olyan porton alkalmazni, amelyeknek esélye van, hogy root vagy kiválasztott porttá váljanak. Leghatékonyabban az UDLD funkció használatával együtt működik. ([2] 8. fejezet)

3.4.5 A feszítőfa protokoll implementálása a teszthálózatban

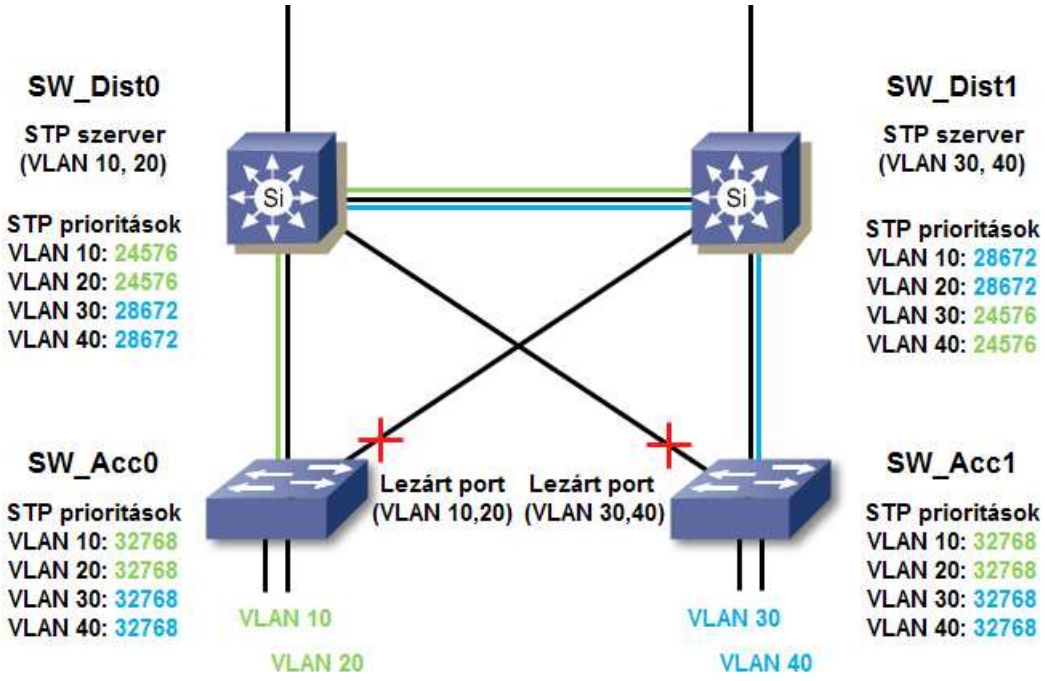
A hálózatban az STP protokoll felelős a hurkok kialakulásának megelőzésért. Az STP alapbeállítások mellett is automatikusan képes végbevinni a választási folyamatokat. Azonban az eredménye lehetséges, hogy más lesz az elvártnál. Előfordulhat, hogy a topológia leglassabb kapcsolója válik Root Bridge-é, ami nem egy ideális választás, mivel a legtöbb forgalom ezen a kapcsolón fog keresztül haladni.

Ebből adódóan ajánlott manuálisan megválasztani Root Bridge-t, néhány szempontot figyelembe véve. A nagy forgalmi terhelés miatt érdemes olyan kapcsolót választani, amely a legjobb teljesítménnyel rendelkezik. Továbbá a Root Bridge a többi kapcsoló referencia pontja, így minden eszköz egy útvonalat fog választani felé. Ezért javasolt, hogy egy központi részen helyezkedjen el és nagy sávszélességű kapcsolatokkal rendelkezzen.

A hálózat kialakítását figyelembe véve, az elosztás rétegbeli kapcsolókat választottam Root Bridge-ként. A redundáns kiépítés mellett, az STP prioritások manipulálásával lehetőségem van megosztani a terhelést a két kapcsoló között.

```
!
spanning-tree vlan 10 priority 24576
spanning-tree vlan 20 priority 24576
spanning-tree vlan 30 priority 28672
spanning-tree vlan 40 priority 28672
```

A prioritások manipulálása után az ábrán látható módon alakul ki az STP topológia. Az STP a prioritások alapján a VLAN 10 és 20 számára a SW_Dist0 kapcsolót fogja választani Root Bridge-ként. A VLAN 30 és 40 számára viszont az SW_Dist1 kapcsoló fogja betölteni a szerepet. A kapcsolók FastEthernet vonalakon keresztül csatlakoznak egymáshoz, amely költség értéke 19. Az STP a költségek alapján kiválasztja a Root Bridghe-hez vezető legkisebb költségű útvonalat, amelyet a forgalom továbbításra fog használni. Az alternatív vonalat pedig lezárja (10. ábra).



10. ábra: STP topológia

A hatékonyság érdekében STP fejlesztéseket is használtam a kapcsolókon. A hozzáférési portok, melyekre az állomások csatlakoznak, nem vesznek részt az STP folyamatokban, így portoknak szükségtelen *figyelő* és *tanuló* állapotokon keresztül haladniuk. Ezért a PortFast funkciót alkalmaztam a SW_Acc0 és SW_Acc1 kapcsolók hozzáférési portjain. Segítségével a csatlakozó állomásoknak nem kell kivárniuk feleslegesen az STP időzítőket.

```
!
interface FastEthernet0/5
spanning-tree portfast
```

A hozzáférési portok - mivel állomásokhoz csatlakoznak - normál esetben nem kapnak BPDU üzeneteket. Azonban egy támadó saját kapcsolóját használva csatlakozhat a hálózathoz

és a hamis BPDU-k segítségével megzavarhatja az STP folyamatokat (switch spoofing). Az ilyen támadások a hozzáférési rétegben a legvalószínűbbek, mivel ez a réteg érhető el legkönnyebben a felhasználók számára. Így minden hozzáférési portra alkalmaztam a BPDU Guard védelmet.

```
!
spanning-tree portfast bpduguard default
```

Továbbá az SW_Dist0 és SW_Dist1 kapcsolók hozzáférési réteg felé vezető portjaira konfiguráltam a Root Guard védelmet.

```
!
interface FastEthernet0/1
spanning-tree guard root
```

A médiumok meghibásodása az STP helytelen működését eredményezheti, és hurkokhoz vezethet. Abban az esetben, ha az SW_Acc0 és SW_Acc1 hozzáférési kapcsolók *lezárt* portjai hiba folytán nem kapnak meghatározott időn belül BPDU-kat, a portok továbbító állapotba lépnek és hálózati hurok jön létre. A probléma megoldására az UDLD és Loop Guard funkciókat alkalmaztam a kapcsolókon.

```
!
spanning-tree loopguard default
!
interface FastEthernet0/1
 udld enable
```

3.5 EtherChannel

Napjaink kommunikációs hálózataira egyre nagyon terhelés nehezedik, ezért hálózati eszközeink között gyorsabb és gyorsabb kapcsolatokra van szükségünk. A hálózati kapcsolók ethernet kapcsolatot használnak az egymás közötti adatforgalom lebonyolítására. Az ethernet folyamatosan fejlődik, napjainkban elérhető a 10-Gigabit Ethernet, amely 10 Gigabit/szekundumos sebességre képes. Azonban, hogy használni is tudjuk a technológiákat korszerűsíteni kell eszközeinket, ami költséges.

A Cisco kifejlesztett egy EtherChannel nevű eljárást, amellyel kötegelhetjük párhuzamos vonalainkat. Így növelhetjük kapcsolataink sávszélességét anélkül, hogy új hardverekre kellene költeni. A fejlesztéssel kettő, négy vagy akár nyolc Fast Ethernet, Gigabit Ethernet vagy 10-Gigabit Ethernet kötegelése lehetséges egy logikai EtherChannel vonallá. Ezáltal nyolc Gigabit Ethernet vonal kötegelése 16Gbps sávszélességű full-duplex (irányonként 8-8) vonalat biztosít.

Alapesetben a kapcsolók közötti többszörös vagy párhuzamos vonalak kapcsolási hurkot okozhatnak. Azonban az EtherChannel elkerüli ezt a problémát, mivel a kötegelt vonalak egy logikai vonalként úgy viselkedik, mint egy hagyományos hozzáférési vagy trönk vonal.

Az EtherChannel redundanciát is biztosít a kötegelt fizikai vonalak között. Abban az esetben, ha az egyik vonal hibázik, akkor a rajta keresztül haladó forgalom automatikusan a fennmaradt vonalakra irányítódik. A hibajavítás néhány századmásodperc időt igényel, így a végfelhasználó számára észrevehetetlen. Amint egy vonal helyreáll a forgalom automatikusan újra szétosztódik az aktív vonalakon.

Ahhoz, hogy fizikai vonalak egy logikai vonalat alkossanak néhány kritériumnak szükséges megegyezniük, amelyek a következők:

- Ethernet média típus – Egy Fast Ethernet és egy Gigabit Ethernet vonal nem kötegelhető és alkothat logikai vonalat.
- VLAN hozzárendeltség – Hozzáférési portoknál, minden portnak azonos VLAN-ba kell tartozniuk. Trönk portok esetén a natív VLAN-nak és engedélyezett VLAN-oknak szükséges megegyezniük.
- Duplexitás és sebesség – Az összes porton azonos duplexitást és sebességet kell konfigurálnunk.

Az EtherChannel-en keresztül a forgalom determinisztikus módon van felosztva a kötegelt vonalakon, de a terheltség nem feltétlenül azonos. Egy hash algoritmus eredménye fogja meghatározni, hogy egy adott keret melyik vonalon továbbítódik. Az algoritmus felhasználhatja a forrás IP, cél IP-címet vagy ezek kombinációit, forrás és cél MAC-címeket vagy TCP/UPD portszámokat. A címek alapján kiszámol egy bináris mintát, amely meghatározza, hogy mely keret mely fizikai vonalon fog keresztül haladni a kötegben. ([1] 7. fejezet)

3.5.1 EtherChannel kialakító protokollok

Az EtherChannel kialakításához dinamikus protokollokat használhatunk. A Catalyst kapcsolókon két típusa érhető el, melyek a Cisco szabadalmaztatott Port Aggregation Protocol (PAgP) és a nyílt szabványú Link Aggregation Control Protocol (LACP).

3.5.1.1 Port Aggregation Protocol (PAgP)

A PAgP automatikus EtherChannel konfigurációt és kialakítást biztosít. A PAgP csomagok segítségével a szomszédok felismerik és megvizsgálják port csoportjaik tulajdonságait. Ha egyezést találnak, együtt kialakítják a kétirányú pont-pont EtherChannel vonalat.

A PAgP képes dinamikusan változtatni a logikai vonal paramétereit, ha a köteg portjai közül valamelyik módosul. Például, ha a köteg egy fizikai vonalán VLAN, sebesség vagy duplexitást változtatunk, a protokoll újrakonfigurálja az adott paramétereket az összes kötegelt fizikai porton.

A logikai vonalat különböző módokba konfigurálhatjuk, amely meghatározza a vonal kialakulásának feltételeit. Ezek *desirable* és *auto* módok. *desirable* módban a kapcsoló aktívan felkéri a távoli végpontot, hogy kialakítsák a vonalat. *Auto* módban az EtherChannel csak abban az esetben fog kialakulni, ha a távoli kapcsoló kezdeményezi a felkérést, hasonlóan a DTP protokollhoz

3.5.1.2 Link Aggregation Control Protocol (LACP).

Az LACP a nyílt szabványú alternatívája a PAgP-nek. A protokoll hasonlóan csomagok segítségével hozza létre a kapcsolatot. Azonban az LACP különböző szerepeket rendel az EtherChannel végpontjaihoz.

LACP esetén 16 fizikai vonalat is kötegelhetünk egy logikai vonalba, viszont csak 8 működik ténylegesen. A portokhoz a port számuk alapján egy prioritást rendel, amely meghatározza mely portok fognak aktívan részt venni adott időben. A nagyobb prioritású portok aktívak, míg kisebbek tartalék állapotba kerülnek előforduló hiba esetére. Továbbá vonal a két végén lévő kapcsolónak MAC-címük alapján egy rendszer prioritás rendel a protokoll. A nagyobb prioritású eszköz fog döntést hozni arról, hogy melyek az aktív vonalak az EtherChannel-en.

Mint a PAgP-nál is láthattuk az LACP logikai vonalainak is különböző módokat állíthatunk be. Az LACP *active* és *passive* módjainak szerepe rendre megegyezik a PAgP *desirable* és *auto* szerepével.

3.5.2 Etherchannel implementálása a teszthálózatba

Az elosztási réteg eszközei a hozzáférési rétegből érkező nagy mennyiségű forgalmat továbbítják a mag réteg kapcsolói felé, ezért nagy sávszélességű vonalakra van szükség. Az Etherchannel protokollt alkalmaztam, hogy magasabb sávszélességet érjek el a hozzáférési és mag réteg között. A kapcsolókon létrehoztam egy logikai vonalat, majd az adott fizikai vonalakat hozzátársítottam. A kötegelés további előnye, hogy párhuzamos redundáns vonalakat alakítunk ki a kapcsolók között. Ezáltal vonalszakadás esetén a köteg működő vonalai képesek továbbra is a forgalom továbbítására.

```
!
interface Port-channel2
 no switchport
 ip address 172.16.0.1 255.255.255.252
```

```
!
interface FastEthernet0/11
 no switchport
 no ip address
 channel-group 2 mode desirable
!
interface FastEthernet0/12
 no switchport
 no ip address
 channel-group 2 mode desirable
```

3.6 Inter-VLAN Routing

Egy második rétegbeli kapcsolón lehetőségünk van több VLAN konfigurálására, azonban közöttük a forgalom lebonyolítására az eszköz nem képes. Az azonos VLAN-ba tartozó állomások képesek lesznek egymással kommunikálni, viszont különbözőkbe tartozók nem.

Ahhoz, hogy a különböző VLAN-ok tagjai képesek legyenek egymással kommunikálni, egy harmadik rétegbeli eszközre van szükségünk. Az eszköz lehet egy forgalomirányító, illetve lehet harmadik rétegbeli kapcsoló.

3.6.1 VLAN-ok közötti forgalomirányítás 3. rétegbeli kapcsoló segítségével

Hagyományosan a kapcsolók a második rétegbeli információk (MAC-cím), míg a forgalomirányítók a harmadik rétegbeli információk (IP-cím, maszk) alapján hozzák meg továbbítási döntéseiket. Azonban a harmadik rétegbeli kapcsolókban ötvözve vannak a forgalomirányítók és a kapcsolók funkciói. Így képesek irányítani a forgalmat, amikor a cél és forrás címek különböző VLAN-ból származnak.

Ahhoz, hogy a VLAN-ok közötti irányítást megvalósítsuk egy harmadik rétegbeli kapcsolón, szükséges egy harmadik rétegbeli virtuális interfészt (SVI) létrehoznunk a VLAN-ok számára. Egy VLAN SVI feldolgozza az összes porton beérkező forgalmat, amely az adott VLAN-hoz tartozik. ([1] 12. fejezet)

3.7 Redundancia protokollok

Amikor egy állomás a saját alhálózatán belül szándékozik adatot küldeni, egy ARP kéréssel kideríti az cél fizikai címét és közvetlenül elküldi csomagját. Azonban, ha a cél különböző alhálózatban található, akkor a küldő félnek szüksége lesz egy közbülső eszközre (pl. forgalomirányító), amely továbbítja a csomagot az adott alhálózat felé. Az ilyen eszközöket alapértelmezett átjárónak nevezzük.

A redundancia érdekében több átjárónk is lehet, azonban egy állomásnak csak egy átjáró IP-címét adhatjuk meg. Ha az állomás által ismert átjáró meghibásodik, akkor a csomagok nem képesek elhagyni az alhálózatot, mivel az állomás nem tud a tartalék eszközök címeiről. A redundancia protokollok e problémára nyújtanak megoldást.

3.7.1 Hot Standby Router Protocol (HSRP)

A HSRP egy olyan redundancia protokoll, amely több forgalomirányítót vagy többrétegű kapcsolót egy csoportba foglal és közös IP-címmel látja el az eszközöket. A csoportban megválasztásra kerül egy elsődleges (aktív) forgalomirányító, amely továbbítja a forgalmat. Továbbá a protokoll kijelöl egy tartalék forgalomirányítót, amely az elsődleges eszköz meghibásodása esetén lép életbe. A forgalomirányítók Hello üzenetekkel folyamatosan vizsgálják egymás működését, és szükség esetén változtatnak szerepükön.

A HSRP prioritás alapján határozza meg a csoport eszközeinek szerepét. A prioritás minden tagnak konfigurálható 0 és 255 között, alapértelmezetten 100-as az értéke. A csoport legnagyobb értékkel rendelkező forgalomirányítója válik aktivá. Abban az esetben, ha több eszközön is azonos a prioritás, akkor a magasabb HSRP interfész IP-címe fogja befolyásolni a választást. ([1] 13. fejezet)

3.7.1.1 Virtuális címek

A HSRP csoport tagjaihoz rendelnünk kell egy közös IP-címet, amelyet virtuális IP-címnek is nevezünk. Ezt adhatjuk meg az állomásoknak, mint alapértelmezett átjáró. Ha az állomás ARP kérést küld az adott címmel egy speciális 0000.0c07.acxx formájú MAC-címet kap válaszul, ahol az xx a csoport számának hexadecimális értéke. Az erre a címre küldött adatok a csoport aktív forgalomirányítójához kerülnek.

3.7.1.2 HSRP állapotok

A HSRP által konfigurált interfészeknek különböző állapotokon kell keresztül haladniuk, hogy elérjék az aktív állapotot. A folyamatban az eszközök információkat gyűjtenek be egymástól, hogy kialakítsák egymáshoz való viszonyukat. Az állapotok a következők:

Initial – A kezdési állapot. A HSRP nem fut az eszközön.

Learn – Az eszköz információkat gyűjt, hogy megállapíthassa a csoport közös IP-címét és aktív forgalomirányítóját.

Listen – Az eszköz megismerte a közös IP-címet. Meghatározza, hogy van-e aktív és tartalék eszköz forgalomirányító a csoportban. Ha van mindkettő, akkor ebben az állapotba marad és figyeli a Hello üzeneteket.

Speak – Az eszköz Hello üzenteket küld, és aktívan részt vesz az aktív illetve tartalék forgalomirányító választásában.

Standby – Az eszköz lesz a következő aktív forgalomirányító. Figyeli az aktív forgalomirányító Hello üzeneteit és meghibásodása esetén átveszi szerepét.

Active – Az eszköz továbbítja a forgalmat, amelyet a közös IP-nek címeztek.

([2] 5. fejezet)

3.7.1.3 HSRP optimalizálása

Preempt funckió

Az aktív forgalomirányító meghibásodása esetén a tartalék veszi át a helyét. Azonban, amikor helyreáll, az eszköz nem válik azonnal újra aktivá. Az újonnan választott aktív forgalomirányítónak először hibáznia kell, hogy az eredeti újra visszanyerje szerepét.

Így, ha HSRP csoportunkba egy új eszközt csatlakoztatunk, hiába a legnagyobb prioritással rendelkezik, nem fogja átvenni az aktív forgalomirányító szerepét. Mivel a jelenlegi aktív forgalomirányító már hamarabb átvette a szerepet.

Azonban eszközeinket konfigurálhatjuk a preempt funkcióval, amely lehetővé teszi, hogy a legnagyobb prioritással rendelkező eszköz azonnal átvegye az aktív szerepet

Interfész figyelés (HSRP Track)

Az átjáró forgalomirányítónkon egy vagy esetleg több interfész is kapcsolódhat különböző alhálózatok felé. Azonban, e vonalak meghibásodása esetén az eszköz aktív marad. Így az állomások által hozzá küldött csomagok nem kerülnek továbbításra.

A HSRP tartalmaz egy olyan funkciót, amellyel vonalhiba felismerést végezhetünk. A funkció figyeli az adott interfészeket és meghibásodásuk esetén csökkenti a forgalomirányító prioritását. A csökkentés mértékét minden vonalnak külön megadhatjuk.

A *track* funkcióval elérhetjük, hogy aktív forgalomirányítónk egy adott interfésze meghibásodása esetén a tartalék eszköz vegye át a forgalmat. Az adott interfészre egy olyan mértékű csökkentési értéket állítunk be, hogy az értékkel csökkentett prioritás kisebb legyen, mint a tartalék eszközé. Így amikor a vonal meghibásodik a prioritás alapján kénytelen lesz eldobni aktív szerepét. Amikor a vonal ismét hibamentessé válik, a forgalomirányító visszakapja eredeti prioritását és visszaveheti eredeti szerepét. (*preempt* funkcióval együtt használandó)

Terhelés eloszlás

Redundáns átjárók esetén egy csoportba egy aktív forgalomirányító kerül megválasztásra. Így a többi eszköz és vonalai kihasználatlanok maradnak. Azonban több csoport létrehozásával eloszthatjuk a forgalmi terhelést a forgalomirányítók között.

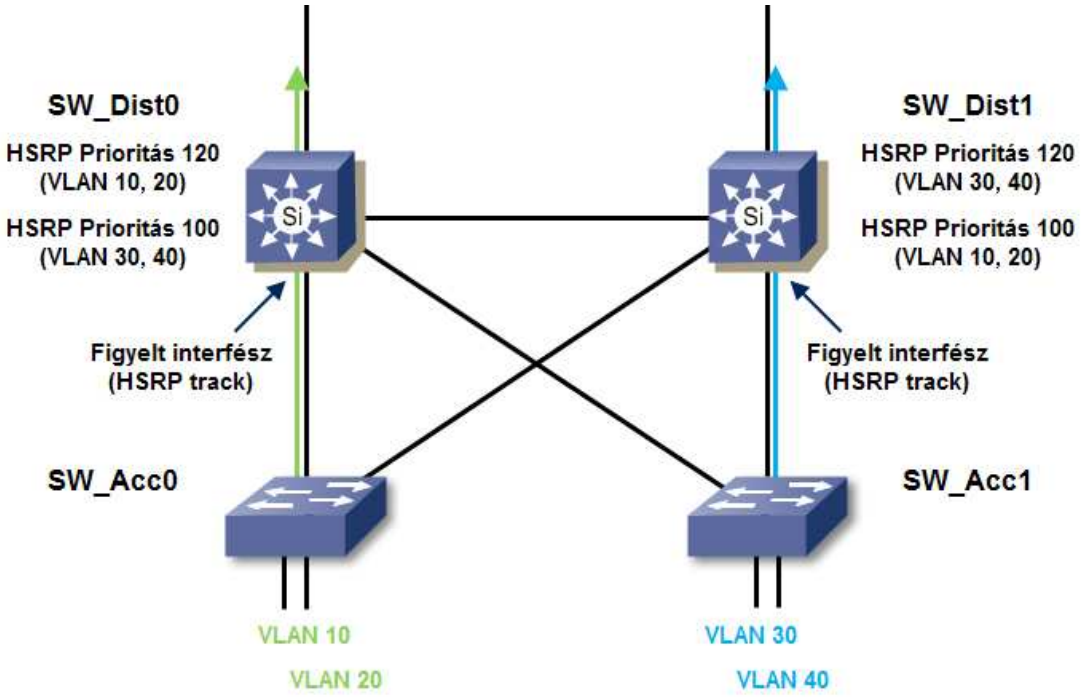
3.7.2 VLAN-ok közti forgalomirányítás és redundancia implementálása a teszthálózatban

A VLAN forgalmával nem ajánlatos a mag réteg eszközeit terhelni, ezért a VLAN közti forgalomirányítást az elosztási réteg harmadik rétegbeli kapcsolóin, virtuális interfészek segítségével hajtom végre. A hálózat redundáns kiépítését kihasználva a forgalomirányítást a két kapcsoló között megosztom. A redundáns terhelés eloszlás megvalósításához a HSRP protokollt alkalmazom.

```
!
interface Vlan10
 ip address 172.16.10.1 255.255.255.0
 standby 1 ip 172.16.10.10
 standby 1 priority 120
 standby 1 preempt
 standby 1 track FastEthernet0/0 30
!
interface Vlan20
 ip address 172.16.20.1 255.255.255.0
 standby 1 ip 172.16.20.10
 standby 1 priority 120
 standby 1 preempt
 standby 1 track FastEthernet0/0 30
!
interface Vlan30
 ip address 172.16.30.1 255.255.255.0
 standby 1 ip 172.16.30.10
 standby 1 preempt
!
interface Vlan40
 ip address 172.16.40.1 255.255.255.0
 standby 1 ip 172.16.40.10
 standby 1 preempt
```

Az aktív eszközök választásánál figyelembe vettem a kialakult STP topológiát. A VLAN 10 és 20 számára a SW_Dist0 kapcsoló az STP szerver, így a SW_Acc0 és SW_Dist1 közötti vonalat lezárta a feszítőfa protokoll. Ezért a prioritásokat oly módon konfiguráltam, hogy a 10-es és 20-as VLAN-ok állomásai számára a SW_Dist0 váljon aktív kapcsolóvá. A VLAN 30 és 40 STP szervere viszont a SW_Dist1 kapcsoló, ezért aktívként is ezt az eszközt választottam (11. ábra).

A vonal szakadásokat figyelembe véve SW_Dist0 SW_Acc0 felé vezető és a SW_Dist1 SW_Acc1 felé vezető interfészeire HSRP *track* funkcióját alkalmaztam. Ezáltal, ha a két kapcsoló figyelt interfészei lekapcsolódnak, az eszközök prioritása változni fog és a kapcsolók szerepei átértékelődnek (11. ábra).



11. ábra: HSRP

3.8 Támadások elleni védelem

3.8.1 MAC-cím elárasztás

A kapcsolók minden beérkező keret forrás MAC-címét eltárolják CAM táblájukban. Azonban, ha egy támadó folyamatosan más és más fizikai címről küldi az adatokat, a kapcsoló egy idő után nem tudja eltárolni az új címeket, mivel CAM táblájuk megtelik. Ennek következtében a kapcsoló nem tudja megállapítani mely portján továbbítsa a kereteket, így az összes portján kiküldi azokat. Eközben a támadó az elárasztott adatokat begyűjtheti. Az ilyen támadásokat MAC-cím elárasztásos támadásnak nevezzük.

3.8.2 Port Security

A Port Security funkcióval korlátozhatjuk egy port által megtanulható fizikai címek számát és megadhatjuk, pontosan mely címek legyenek engedélyezettek.

A kapcsoló a forgalom alapján dinamikusan tanulja meg a MAC-címeket. Azonban statikusan is konfigurálhatunk adott címeket. Továbbá használhatjuk a Port Security *sticky* funkcióját, amely segítségével a kapcsoló dinamikusan tanulja meg címet és eltárolja azt az aktív konfigurációba.

A védelem megszegése esetén megadhatjuk, milyen módon járjon el a kapcsoló. A következő eljárások közül választhatunk:

- Shutdown – A port azonnal *errdisable* állapotba kerül, melynek hatása ugyan olyan, mintha le lenne kapcsolva.
- Restrict – A port felkapcsolva marad, viszont a szabályt megszegő felől érkező kereteket eldobja. Az eseményről feljegyzés készül és a kapcsoló SNMP üzentet küld.
- Protect - Az előbbivel megegyező a működése, de az eseményről nem készül feljegyzés.

([4] 8. modul)

3.8.3 Spoofing támadások

Egy ártó szándékú felhasználó spoofing eszközök segítségével küldhet hamis adatokat a hálózatban szereplő állomásoknak. A támadás célja, hogy az állomások az eredeti céljuk helyett a támadónak küldjék adataikat. A támadó begyűjti az információkat, és az adatot továbbküldi az eredeti célja felé. Így a felhasználó észre sem veszi a beavatkozást. Ezek a technikák legtöbbször egy man-in-the-middle támadás részeit képezik.

3.8.3.1 DHCP Snooping

A DHCP szerver a hálózatban való szerepléshez szolgáltat szükséges információkat az állomások számára. Ilyen információ például az IP-cím és az alapértelmezett átjáró címe.

A támadó üzemeltethet egy hamis DHCP szervert, amellyel hamis információkat küld az állomásoknak. Az állomás DHCP kérésére a csaló eszköz fog válaszolni és saját IP-címét tünteti fel az alapértelmezett átjáró címe helyett. Így az átjárónak címzett csomagok a támadó kezébe kerülnek.

A DHCP snooping funkcióval azonban meggátolhatjuk a hamis üzenetek küldését. A védelem engedélyezésével a kapcsoló portjait megbízható és nem megbízható módokba konfigurálhatjuk. A megbízható port megbízik a hozzá kapcsolódó eszközökbe és engedélyezi minden típusú DHCP üzenetek áramlását. A nem megbízható port azonban csak DHCP kéréseket engedélyez. A DHCP szerverünket megbízható, míg a felhasználói állomásokat nem megbízható portokra ajánlott csatlakoztatni.

3.8.4 VLAN támadások

3.8.4.1 Kapcsoló Spoofing

Alapértelmezett beállítások mellett a kapcsolók a Dynamic Trunking Protocol használatával dinamikusan alakítják ki a trönk vonalat. A DTP egyszerűbbé teszi a kapcsolók konfigurálását, azonban lehetőséget ad a támadóknak, hogy hozzáférjenek a trönk vonalon haladó VLAN forgalomhoz. A rosszindulatú felhasználó számítógépével a kapcsoló portjához kapcsolódik és DTP csomagokat küldve kialakítja a trönk vonalat. Mivel alapértelmezetten a trönk minden VLAN forgalmat engedélyez, a támadó az egész hálózat forgalmát begyűjtheti.

A támadás megelőzhető, hogyha a kapcsoló porjait hozzáférési módba helyezzük vagy kikapcsoljuk az DTP funkciót.

3.8.4.2 VLAN ugrás

A támadó a 802.1Q protokoll hibáit kihasználva képes adatokat küldeni különböző VLAN-okba, harmadik rétegbeli eszköz használata nélkül. A támadáshoz szükséges, hogy az adott kapcsoló rendelkezzen dot1q trönk vonallal és a támadó által használt port VLAN-ja megegyezzen a trönk natív VLAN-jával.

A támadó egy plusz VLAN ID-vel jelöli meg a keretét. Ezáltal a keret duplán lesz jelölve, a szabályos VLAN ID-vel és a támadó által ellátottal. Amikor a kapcsoló megkapja a keretet, leválasztja a szabályos VLAN ID-t és mivel a trönk natív VLAN-jával megegyezik, a kapcsoló továbbítja a trönk vonalra. A másik végen lévő kapcsoló megvizsgálja a keretet és észreveszi a (hamis) VLAN jelölést. A kapcsoló a szokásos módon leválasztja a VLAN ID-t és megfelelő portjára küldi a keretet.

A támadást kétféle módon előzhetjük meg:

- A trönk natív VLAN-nak egy használaton kívüli VLAN-t adunk meg és letiltjuk a forgalmát a trönk vonalon.
- Beállítjuk a kapcsolón, hogy a natív VLAN-ba tartozó kereteket jelölje meg a trönk vonalon.

3.8.5 Kapcsoló védelme

Kapcsolónkhoz illetéktelen felhasználók is hozzáférhetnek, akik kárt tehetnek hálózatunk működésében vagy információkat szerezhetnek eszközeinkről. Ezért ajánlatos a következő védelmi eljárásokat alkalmazni:

- Biztonságos jelszó konfigurálása – Használjunk titkosított jelszót a privilegizált szintek eléréséhez! A titkosítás szótár alapú támadással feltörhető, ezért jelszavunkban használjunk speciális karaktereket.
- A konzol hozzáférés védelme – A konzol hozzáférésen keresztül teljes mértékben konfigurálható az eszköz, ezért a fizikai védelmen kívül fontos jelszóval védeni azt.

- A Vty vonalak védelme – A Vty vonal távoli hozzáférést (Telnet, SSH) biztosít kapcsolónkhoz. A vonalhoz való hozzáférést jelszóval és hozzáférési listákkal korlátozhatjuk.
- SSH használata – Az SSH protokoll a Telnet-tel szemben biztonságos távoli hozzáférést biztosít eszközünkhöz. A telnet nyílt szövegként, míg az SSH titkosítva továbbítja a forgalmat.
- Bejelentkező feliratok – A login és motd banner feliratokkal a kapcsolónkhoz hozzáférő felhasználókat figyelmeztethetjük a használati szabályokról.
- Szükségtelen szolgáltatások letiltása – Előfordulhat, hogy eszközünkön néhány szolgáltatás alapértelmezetten engedélyezve van, amelyek támadási pontot adhatnak. Vizsgáljuk meg kapcsolónk konfigurációját és tiltsuk le a szükségtelen szolgáltatásokat!
- CDP használata - A CDP funkcióval információkat kaphatunk a szomszédos Cisco eszközökről 2. rétegbeli szinten. Azonban ezeket a titkosítatlan információkat a támadó is megszerezheti, ezért csak olyan helyeken használjuk, ahol feltétlenül szükségünk van rá.
- Használton kívüli portok – A kapcsoló portjai alapértelmezetten felkapcsolt állapotban vannak és a VLAN 1 tagjai. Ezáltal támadó könnyedén hozzáférhet az eszközünkhöz. Ezért a használaton kívüli portokat kapcsoljuk le, és helyezzük egy használaton kívüli VLAN-ba. ([1] 15. fejezet)

3.8.6 A védelmi technikák implementálása a teszthálózatban

MAC-cím elárasztással a támadó felhasználói adatokhoz juthat hozzá. A támadások ellen a Port Security fejlesztést alkalmazva, korlátoztam a hozzáférési portok által engedélyezett fizikai címek számát. Továbbá a *sticky* funkciót használva a kapcsoló csak adott állomásoknak engedélyezik a hozzáférést, és dinamikusan tanulják meg, a maximalizált számú MAC-címeket.

```
!
interface FastEthernet0/5
 switchport mode access
 switchport port-security maximum 2
 switchport port-security mac-address sticky
```

A támadó hamis adatok küldésével megtévesztheti az eszközöket és károsíthatja a hálózatot vagy adatokhoz juthat hozzá. A DHCP csalások ellen a DHCP Snooping fejlesztést alkalmaztam a kapcsolókon. A védelem bekapcsolásával minden port megbízhatatlan állapotba kerül. Ezért a DHCP szerver irányába mutató portokat megbízhatóvá konfiguráltam.

```
!
ip dhcp snooping
!
interface FastEthernet0/10
 ip dhcp snooping trust
```

A VLAN ugrásos támadások ellen konfiguráltam, hogy a trönk vonalakon a kapcsoló jelölje meg a natív VLAN kereteit. Ezáltal duplán jelölt keretek esetén, a kapcsoló nem fogja leválasztani az első natív VLAN jelölést és a keret az eredeti VLAN-ján belül marad.

```
!
vlan dot1q tag native
```

Ahhoz, hogy illetéktelen felhasználók ne férjenek hozzá a kapcsolókhoz, titkosított jelszóval védem a privilegizált hozzáférést és a konzol illetve vty vonalakat. Továbbá a szükségtelen szolgáltatásokat kikapcsoltam.

```
!
enable secret 5 $1$U0yJ$s9vmsUUvqrZiiYdh9ZjWn1
!
no ip http server
```

```
!
line con 0
 password 7 030752180500
 login
line vty 0 4
 password 7 045802150C2E
 login
 transport input ssh
```

A nem használt portok támadási pontokat adhatnak a támadók számára. Ezért használaton kívüli portokat egy nem használt VLAN-ba helyeztem és lekapcsoltam, hogy elszeparáljam a működő hálózattól. Így, ha a támadó hozzá is tud csatlakozni a kapcsoló egy nem használt portjához, a működő hálózatot nem fogja elérni.

```
!
interface FastEthernet0/5
 switchport access vlan 999
 switchport mode access
 shutdown
```

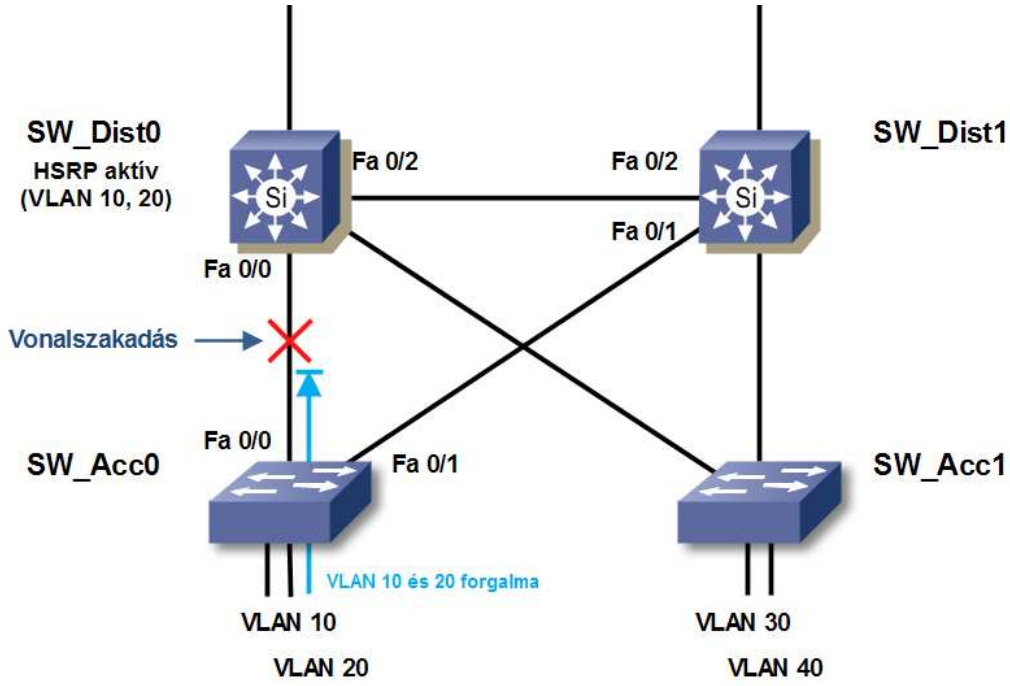
A Cisco CDP fejlesztése a támadó számára információkat szolgáltathat eszközeinkről. Azonban, néhány esetben szükség lehet a funkcióra. Például, ha Cisco IP telefont csatlakoztatunk kapcsolónkra. Ezért a hozzáférési portok kivételével, ahol felhasználók használhatnak IP telefont, kikapcsoltam a szolgáltatást.

```
!
interface FastEthernet0/1
 no cdp enable
```

4. Esettanulmány

A fejezetben az általam épített teszthálózaton tesztelem a hálózat redundáns felépítését. A topológiában egy hibát szimulálva megvizsgálom, hogy a hálózat hogyan reagál a változásra, és milyen eljárásokat hajt végre a probléma áthidalásához.

Az SW_Dist0 és SW_Acc0 között vonalszakadást szimulálok azáltal, hogy a kapcsolók közötti vonalat lekapcsolom az interfészen kiadott *shutdown* paranccsal (12. ábra). A következőkben azt vizsgálom, hogy a kapcsolókon konfigurált STP és HSRP protokollok az elvártak szerint reagálnak-e a szimulált hibára.



12. ábra: A vonalszakadás helye

A SW_Acc0 kapcsoló eredetileg FastEthernet 0/0 portját használta az adatok továbbítására. Azonban a vonalszakadás után a redundáns útvonalat kényszerül használni a FastEthernet 0/1 portján keresztül.

Az STP protokoll reagál a változásokra és VLAN 10 és 20 számára SW_Acc0 kapcsoló FastEthernet 0/1 portja az eredetileg *lezárt* állapotából *továbbító* állapotba kerül (13. ábra). A kapcsoló közben TCN üzeneteket küld a változásról a Root Bridge-nek, hogy arról minden kapcsoló értesüljön.

SW_Acc0 kapcsoló STP debug kimenetén látható a port állapotának változása és az üzenet küldése a topológia változásról:

**Oct 8 21:23:35.407: STP: VLAN10 Fa0/0 -> blocking*
**Oct 8 21:23:35.407: STP: VLAN10 new root port Fa0/1, cost 38*
**Oct 8 21:23:35.443: STP: VLAN10 Fa0/1 -> listening*
**Oct 8 21:23:37.407: STP: VLAN10 sent Topology Change Notice on Fa0/1*
**Oct 8 21:23:50.455: STP: VLAN10 Fa0/1 -> learning*
**Oct 8 21:24:05.475: STP: VLAN10 sent Topology Change Notice on Fa0/1*
**Oct 8 21:24:05.479: STP: VLAN10 Fa0/1 -> forwarding*

```
SW_Acc0#show spanning-tree vlan 10 brief

VLAN10
  Spanning tree enabled protocol ieee
  Root ID     Priority    24576
              Address     cc03.0844.0001
              Cost        38
              Port        2 (FastEthernet0/1)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32768
              Address     cc01.0844.0001
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                   Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet0/1      128.2    128    19 FWD    19 28672 cc02.0844.0001 128.2
FastEthernet0/5      128.6    128    19 FWD    38 32768 cc01.0844.0001 128.6
```

13. ábra: STP port állapot változás

A HSRP protokoll is észlelni fogja a változást, mivel a SW_Dist0 kapcsolón konfigurált *track* funkcióval (lásd 3.7.1.3) figyeli a FastEthernet 0/0 interfészt. A SW_Dist0 VLAN 10 és 20 számára eredetileg 120-as prioritás van konfigurálva, amely nagyobb a SW_Dist1 prioritás értékénél. Ezáltal a két VLAN számára a SW_Dist0 az aktív kapcsoló.

Az interfész lekapcsolódása után a *track* funkció a megadott értékkel csökkenti az eszköz HSRP prioritását. A konfigurált csökkentési érték 30, tehát a SW_Dist0 új prioritása 90 lesz. Az eszköz a változás következtében *speak* állapotba kerül, hogy információt cseréljen csoportjában az új helyzetről. Az SW_Dist0 csökkentett prioritás értéke alacsonyabb a SW_Dist1 100-as értékénél. Ezáltal SW_Dist1 átveszi VLAN 10 és 20 számára az aktív szerepet, SW_Dist0 pedig tartalékként fog részt venni (14. ábra).

```
SW_Dist0#show standby
Vlan10 - Group 1
  State is Standby
    22 state changes, last state change 00:11:43
  Virtual IP address is 172.16.10.10
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.392 secs
  Preemption enabled
  Active router is 172.16.10.2, priority 100 (expires in 7.468 sec)
  Standby router is local
  Priority 90 (configured 120)
    Track interface FastEthernet0/0 state Down decrement 30
  IP redundancy name is "hsrp-Vl10-1" (default)
```

14. ábra: SW_Dist0 prioritás értékének csökkenése

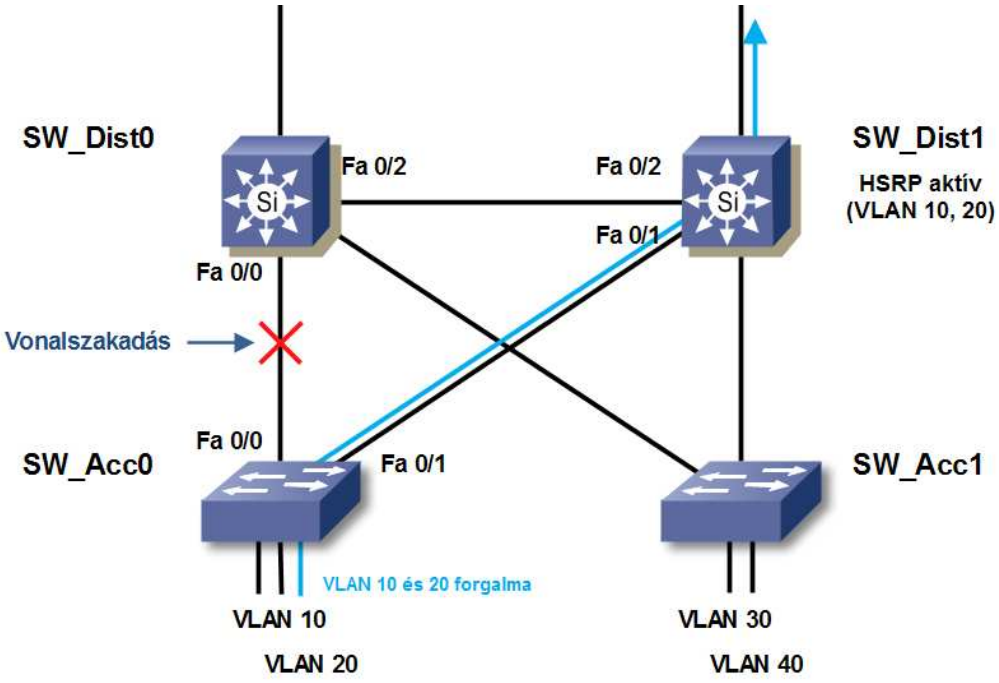
A SW_Dist1 kapcsoló debug kimenetén látható, hogy az eszköz átveszi az aktív szerepet:

**Oct 8 21:21:24.614: %HSRP-5-STATECHANGE: Vlan10 Grp 1 state Standby ->Active*

A SW_Dist0 kapcsoló debug kimenetén látható, hogy az eszköz átveszi a tartalék szerepet:

**Oct 8 21:26:00.255: %HSRP-5-STATECHANGE: Vlan10 Grp 1 state Active -> Speak*
**Oct 8 21:26:10.255: %HSRP-5-STATECHANGE: Vlan10 Grp 1 state Speak -> Standby*

A redundáns felépítés és a megfelelően konfigurált protokollok segítségével a SW_Acc0 kapcsolóhoz csatlakoztatott állomások továbbra is képesek kommunikálni a hálózattal (15. ábra).



15. ábra: A forgalom új iránya

A vonal helyreállása után a SW_Dist0 Fastethernet 0/0 interfésze újra felkapcsolt állapotba kerül. Az eszköz visszakapja eredeti prioritását és újra aktív kapcsolóként fog szerepelni (16. ábra).

```
SW_Dist0#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl10        1   120  P Active   local           172.16.10.2     172.16.10.10
Vl20        1   120  P Active   local           172.16.20.2     172.16.20.10
Vl30        1   100  P Standby  172.16.30.2     local           172.16.30.10
Vl40        1   100  P Standby  172.16.40.2     local           172.16.40.10
```

16. ábra: HSRP állapotok a vonal helyreállása után

Az STP újra *lezárt* állapotba helyezi a FastEthernet 0/1 interfészét, mivel a vonal helyreállása után a FastEthernet 0/0 felé alacsonyabb költséggel éri el a SW_Dist0 Root kapcsolót (17. ábra).

```
SW_Acc0#show spanning-tree vlan 10 brief

VLAN10
  Spanning tree enabled protocol ieee
  Root ID     Priority    24576
              Address     cc03.0adc.0001
              Cost        19
              Port        1 (FastEthernet0/0)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32768
              Address     cc01.0adc.0001
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                   Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet0/0      128.1    128    19 FWD     0 24576 cc03.0adc.0001 128.1
FastEthernet0/1      128.2    128    19 BLK    19 28672 cc02.0adc.0001 128.2
FastEthernet0/5      128.6    128    19 FWD    19 32768 cc01.0adc.0001 128.6
```

17. ábra: STP port-állapotok a vonal helyreállása

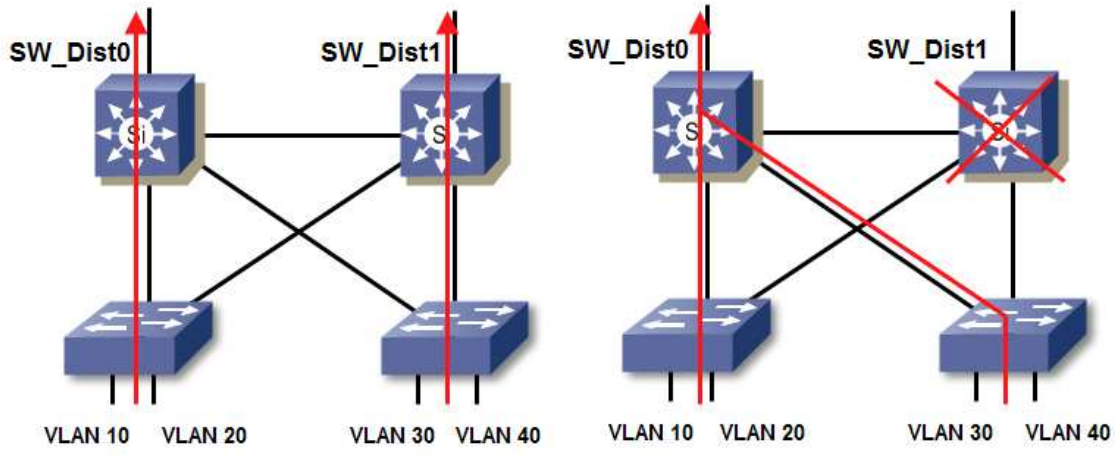
A fentebbi eredményekből látható, hogy a konfigurált protokollok megfelelően reagálnak a topológiában történő vonalszakadásokra. Továbbá a hiba elhárítása után a forgalom iránya visszaáll az eredeti helyzetébe.

5. Hibaelhárítás

A fejezetben a munkám során előfordult problémákat szimuláltam teszthálózatomban és ismertetem a probléma feltárásához és megoldásához szükséges folyamatokat.

5.1 1. Probléma

Az 18. ábrán látható SW_Dist1 kapcsoló újraindul. A kapcsoló több szolgáltatást is nyújt a VLAN 30 és 40 felhasználói számára. Mivel a hálózat redundáns kiépítésű és a megfelelő protokollokkal konfigurálva vannak, a SW_Dist0 eszköz átveszi a helyét.



18. ábra: A forgalom iránya a SW_Dist1 működése (bal) és meghibásodása (jobb) esetén

Miután a SW_Dist1 újraindult, ismét képes ellátni feladatait. Azonban a kapcsoló nem továbbít forgalmat a mag réteg felé és a SW_Dist0 vonalai túlterhelté válnak.

A kapcsolókon a HSRP státuszokat megvizsgálva láthatjuk, hogy a SW_Dist0 az összes VLAN számára aktívként, míg a SW_Dist1 tartalékként szerepel (19. ábra). A jelenlegi felállás mellett az SW_Dist0 kapcsolón az összes VLAN forgalom keresztülhalad, ezért túlterhelté válhat.

```
SW_Dist0#sh standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl10        1   120  P Active   local           172.16.10.2     172.16.10.10
Vl20        1   120  P Active   local           172.16.20.2     172.16.20.10
Vl30        1   100  P Active   local           172.16.30.2     172.16.30.10
Vl40        1   100  P Active   local           172.16.40.2     172.16.40.10
SW_Dist1#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl10        1   100    Standby  172.16.10.1     local           172.16.10.10
Vl20        1   100    Standby  172.16.20.1     local           172.16.20.10
Vl30        1   120    Standby  172.16.30.1     local           172.16.30.10
Vl40        1   120    Standby  172.16.40.1     local           172.16.40.10
```

19. ábra: Hibás működés

A HSRP prioritások alapján SW_Dist1 kapcsolónak kellene a VLAN 30 és 40 számára aktívként szerepelnie. Ezáltal terheléseloszlás valósulna meg a két eszköz között.

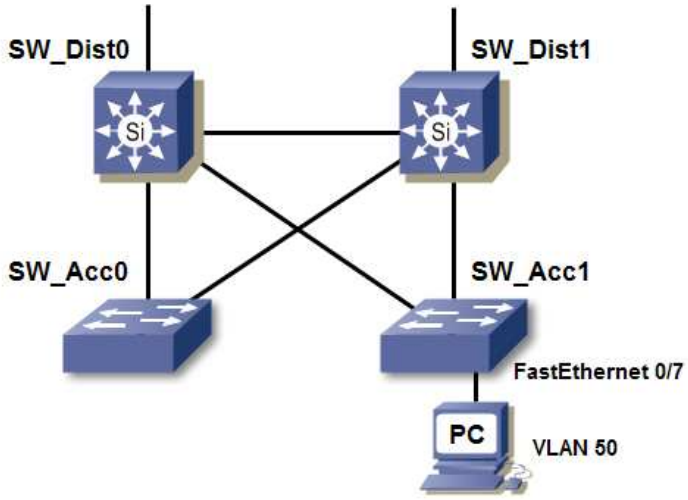
A kimeneteket megvizsgálva észrevehetjük, hogy a SW_Dist1 kapcsolón a *preempt* funkció (lásd 3.7.1.3) nincs bekapcsolva. A funkció hiányába, amikor a kapcsoló újra működőképessé vált, nem vette vissza eredeti szerepét. A problémát az adott VLAN interfészeken a *preempt* funkció bekapcsolásával orvosolhatjuk, amelyet a *standby 1 preempt* paranccsal hajthatunk végre. Eredménykép a terheléseloszlás újra létrejön és a hálózat hatékonyan működik tovább (20. ábra).

```
SW_Dist0#sh standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl10        1   120  P Active   local           172.16.10.2     172.16.10.10
Vl20        1   120  P Active   local           172.16.20.2     172.16.20.10
Vl30        1   100  P Standby  172.16.30.2     local           172.16.30.10
Vl40        1   100  P Standby  172.16.40.2     local           172.16.40.10
SW_Dist1#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl10        1   100  P Standby  172.16.10.1     local           172.16.10.10
Vl20        1   100  P Standby  172.16.20.1     local           172.16.20.10
Vl30        1   120  P Active   local           172.16.30.1     172.16.30.10
Vl40        1   120  P Active   local           172.16.40.1     172.16.40.10
```

20. ábra: Helyes terheléselosztott működés

5.2 2. Probléma

A vállalat új munkacsoportjának egy új 50-es VLAN-t hoztak létre, amelynek első tagja a SW_Acc1 kapcsoló FastEthernet 0/7 portjára csatlakoztatott állomás (21. ábra). Azonban az állomás nem képes kommunikálni a hálózat többi tagjaival.



21. ábra: Az új 50-es VLAN tagja

Az új 50-es VLAN –t a VTP szerver által konfigurálták a tartomány összes kapcsolóján. A kapcsolókon ellenőrizhetjük, hogy az adatbázisuk tartalmazza-e az adott VLAN-t, és a VLAN-hoz mely portok vannak hozzárendelve. A *show vlan* parancs kiadásával megbizonyosodhatunk arról, hogy az 50-es VLAN létezik a kapcsolón és a megfelelő FastEthernet 0/7 port van hozzárendelve.

```
SW_Acc1#sh vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/2, Fa0/3, Fa0/4, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15
10   VLAN0010                         active
20   VLAN0020                         active
30   VLAN0030                         active    Fa0/5
40   VLAN0040                         active    Fa0/6
50   VLAN0050                         active    Fa0/7
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

22. ábra: A show vlan parancs kimenete

A további tesztekből kiderült, hogy az állomás nem éri el az alapértelmezett átjáróját (SW_Dist0 és SW_Dist1), annak ellenére, hogy hozzá vezető hozzáférési és trönk vonalak aktívak (23. ábra).

```
SW_Acc1#show interfaces description
Interface                      Status         Protocol Description
Fa0/0                          up             up       ****Trunk To SW_Dist1****
Fa0/1                          up             up       ****Trunk To SW_Dist0****
Fa0/7                          up             up       ****Access port VLAN 50****
```

23. ábra: A szükséges vonalak up - up állapotúak

A trönk vonalak részletes tulajdonságait megvizsgálhatjuk a *show interface trönk* paranccsal. A kimeneten észrevehetjük, hogy az újonnan konfigurált VLAN 50 nincs engedélyezve a vonalakon (24. ábra). Ezáltal a trönk port nem fogja továbbítani az adott VLAN forgalmát.

```
SW_Acc1#show interfaces trunk

Port        Mode         Encapsulation  Status        Native vlan
Fa0/0       on           802.1q         trunking      1
Fa0/1       on           802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/0       1,30,40,1002-1005
Fa0/1       1,30,40,1002-1005
```

24. ábra: Trönk vonalak tulajdonságai

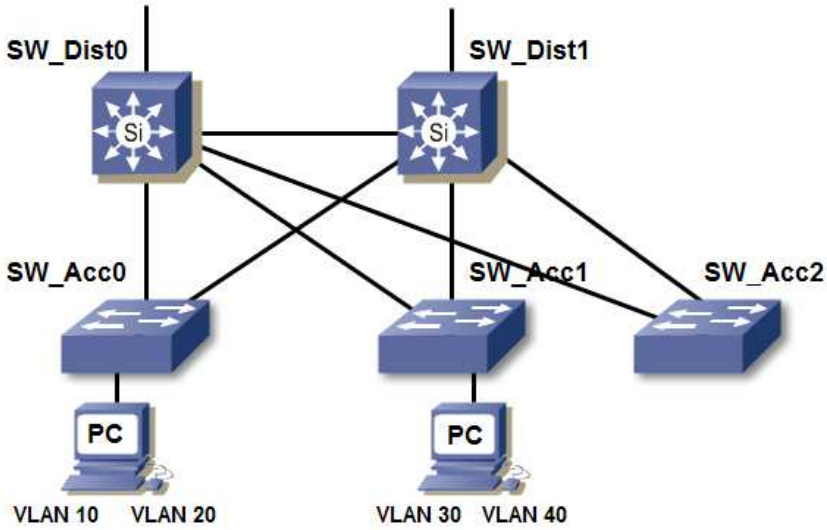
A problémát a VLAN engedélyezésével javíthatjuk, amelyet a *switchport trunk allowed vlan add 50* paranccsal hajthatunk végre a megfelelő trönk vonalakon (25. ábra). Az új beállítások után az állomás képes lesz kommunikálni a hálózat tagjaival.

```
SW_Acc1#sh interfaces trunk

Port        Mode         Encapsulation  Status        Native vlan
Fa0/0       on           802.1q         trunking      1
Fa0/1       on           802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/0       1,30,40,50,1002-1005
Fa0/1       1,30,40,50,1002-1005
```

25. ábra Új engedélyezett VLAN

5.3 3. Probléma

Bővítés céljából a hálózathoz új hozzáférési kapcsolót (SW_Acc2) csatlakoztattak (26. ábra). Azonban, üzembe helyezése után a tartomány összes állomása elvesztette a kapcsolatot a hálózat többi elemével.



26. ábra: Új hozzáférési kapcsoló a hálózatban

A kapcsolókat vizsgálva észrevehetjük, hogy az állomások által használt VLAN-ok törlődtek a kapcsolókról. Helyettük új ismeretlen VLAN-ok jelentek meg (27. ábra).

```
SW_Acc1#sh vlan

VLAN Name                                Status    Ports
---- ----------------------------------- --------- -------------------------------
1    default                             active    Fa0/2, Fa0/3, Fa0/4, Fa0/7
                                                   Fa0/8, Fa0/9, Fa0/10, Fa0/11
                                                   Fa0/12, Fa0/13, Fa0/14, Fa0/15
110  VLAN0110                            active
120  VLAN0120                            active
130  VLAN0130                            active
140  VLAN0140                            active
150  VLAN0150                            active
160  VLAN0160                            active
1002 fddi-default                        active
1003 token-ring-default                  active
1004 fddinet-default                     active
1005 trnet-default                       active
```

27. ábra: Ismeretlen VLAN-ok

A probléma oka, hogy az újonnan csatlakoztatott SW_Acc2 kapcsoló a VTP protokollt (lásd 3.3) használva saját régebbi VLAN adatbázisát küldte szét a hálózatban. Az eszköz már korábban szerepelt egy VTP tartományban és konfigurációs revíziós száma magasabb, mint a jelenlegi kapcsolókon (28. ábra). A kapcsolók a revíziós szám alapján azt feltételezik, hogy az új eszköz frissebb VTP hirdetménnyel rendelkezik, ezért adatbázisukat felülírják.

```
SW_Acc2#show vtp status
VTP Version                     : 2
Configuration Revision          : 11
Maximum VLANs supported locally : 256
Number of existing VLANs        : 11
VTP Operating Mode              : Client
VTP Domain Name                 : Building1
```

28. ábra: Magas konfigurációs revízió szám

Az eredeti VLAN adatbázist csak akkor nyerhetjük vissza, ha rendelkezünk az adatbázis biztonsági másolatával, különben újra kell konfigurálnunk.

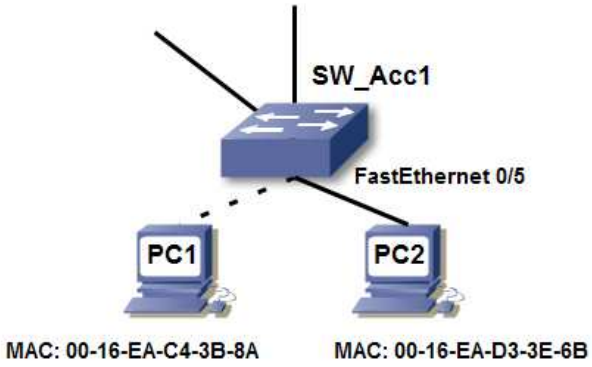
A probléma megelőzhető, ha az új kapcsoló revíziós számát lenullázzuk. Ezáltal az újonnan csatlakoztatott kapcsoló nem fogja a VTP tartomány kapcsolói VLAN adatbázisát felülírni. Az eljárást legegyszerűbben az eszköz VTP Transzparens (Transparent) módba (lásd 3.3.2) való helyezésével érhetjük el (29. ábra).

```
SW_Acc2#show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 256
Number of existing VLANs        : 11
VTP Operating Mode              : Transparent
VTP Domain Name                 : Building1
```

29. ábra: A konfigurációs revíziós szám nullázása

5.4 4. Probléma

A SW_Acc1 kapcsolóhoz csatlakozó PC1 állomást meghibásodása folytán lecserélték. Az új PC2 állomás azonban nem képes kommunikálni a hálózattal (30. ábra).



30. ábra: A PC1 állomás cseréje a PC2-re

A hálózat biztonsága érdekében a hozzáférési portok Port Security (lásd 3.8.2) védelem által vannak konfigurálva. A kapcsoló PC1 fizikai címét megtanulta és eltérő cím esetén tiltja a porton való kommunikációt. Mivel az PC2 fizikai címe különböző, a kapcsoló letiltja az adott interfészt és ezáltal az állomás nem képes a hálózati kommunikációra (31. ábra).

```
SW_Acc1#show port-security interface fastEthernet 0/5
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
```

31. ábra: A kapcsoló biztonsági okokból lekapcsolta portját

A port helyreállításához el kell távolítanunk a konfigurációból a megtanult MAC-címet, mivel a kapcsoló csak egy fizikai címet engedélyez portján. A MAC-cím eltávolítását a *no switchport port-security mac-address 0016.eac4.3b8a* paranccsal hajthatjuk végre. Mivel az interfészt a védelem megszegése miatt került lekapcsolásra, ezért manuálisan kell aktiválnunk az interfészt. Az eljárást az interfész le majd felkapcsolásával vihetjük véghez, a *shutdown, no shutdown* parancsok használatával. A műveletek végrehajtása után a port ismét felkapcsolt állapotba kerül és képes lesz az új állomástól érkező forgalom továbbítására (32. ábra).

```
SW_Acc1#show port-security interface fastEthernet 0/5
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
```

32. ábra: Helyreállított interfész

6. Összegzés

Dolgozatomban egy telephelyi kommunikációs hálózat felépítését és az abban használt technológiákat, valamint azok biztonságához szükséges alapvető védelmi technológiákat ismertettem.

Egy háromrétegű redundáns telephelyi hálózat felépítését vázoltam, elemeztem a három rétegnek milyen szerepe és tulajdonságai vannak. A technológiák részteltes megismerése érdekében teszthálózatot építettem a bemutatott technológiák alkalmazásával. A teszt hálózatott a dolgozatom során felhasználtam a technológiák működésének bemutatására és tesztelésére. Bemutattam a topológiában használt többrétegű kapcsoló használatának előnyeit, kifejtettem a Virtuális LAN-ok hasznát és szerepet a hálózatban. Továbbá bemutattam, hogy milyen protokollok szükségesek a Virtuális LAN-ok hatékony használatához és egyszerűbb kezeléséhez a hálózatban. Szemléltettem a hálózat redundáns felépítése miatt szükséges technológiákat, mint az Spanning Tree Protocol és az Hot Standby Router Protocol. Ismertettem a technológiák zavartalan működéséhez szükséges védelmi eljárásokat, majd a belső támadások elleni alapvető védelmi módszereket.

Az ismertetett technológiákat implementáltam a teszthálózatba. Mivel a hálózat redundáns felépítésű, teszteltem, hogy az általam konfigurált protokollok helyesen reagálnak e az esetleges hibákra. Ezért vonalszakadást szimuláltam a teszthálózatban és megfigyeltem, hogyan reagál a változásra a hálózat. Az eredményekből megállapítottam, hogy helyesen működnek a konfigurált technológiák.

Végül munkahelyi tapasztalataim során előfordult problémákat szimuláltam a teszthálózatban, majd ismertettem a probléma elhárításának folyamatát.

A dolgozatomból kiderül, hogy az ismertetett technológiák használata és együttműködése szükséges ahhoz, hogy a hálózat kifogástalanul működjön. Természetesen egy telephelyi hálózatban további protokollok is felelősek a működésért. Nem tértem ki a gerinchálózatban szükséges forgalomirányító protokollokra, a szerverfarm részletesebb felépítésére, a határponti eszközök működésére, QOS-re stb. Ezek kifejtése nem képezi a dolgozat célját, valamint terjedelmi korlátok miatt sem kerülnek említésre.

Dolgozatom elkészítése során részletesebben megismerkedtem egy általános telephelyi hálózat felépítésével és működésével, és sikeresen letettem a Cisco 642-812 - Building Converged Cisco Multilayer Switched Networks vizsgát.

7. Irodalomjegyzék

7.1 Nyomtatott források

[1]David Hucaby, 2006 CCNP BCMSN Official Exam Certification Guide, 4th Edition. Cisco Press, Indianapolis

[2]Brent Stewart and Denise Donohue [2007] CCNP BCMSN Quick Reference Sheets Cisco Press, Indianapolis

[3]Chris Bryant [2007], The Ultimate BCMSN Study Guide. The Bryant Advantage

7.2 Internetes források

[4] Cisco Netacad tananyag: CCNP: Building Multilayer Switched Networks 5.0 http://www.cisco.com/web/learning/netacad/index.html

[5] Cisco Netacad tananyag: CCNA 3: Switching Basics and Intermediate Routing - 3.1 http://www.cisco.com/web/learning/netacad/index.html

[6] Campus Network for High Availability Design Guide http://www.cisco.com/en/US/docs/solutions/Enterprise/Campus/HA_campus_DG/hacampusdg.html

[7] Using HSRP for Fault-Tolerant IP Routing http://www.cisco.com/en/US/docs/internetworking/case/studies/cs009.html

[8] RFC2281 - Cisco Hot Standby Router Protocol (HSRP) http://www.ietf.org/rfc/rfc2281.txt

[9] Spanning Tree Protocol Root Guard Enhancement http://www.cisco.com/en/US/tech/tk389/tk621/technologies_tech_note09186a00800ae96b.shtml

8. Köszönetnyilvánítás

Köszönetet kell mondanom Dr. Almási Béla egyetemi docensnek szakdolgozatom témavezetéséért, munkám során nyújtott szakmai segítségéért, tanácsaiért és türelméért, melyek nagymértékben hozzá járultak dolgozatom elkészítéséhez. Nem szabad megfeledkeznem mindazokról, akik jó tanácsokkal, építő jellegű kritikákkal illették munkámat.