

**VIZSGAREMEK**

Készítették:

Ötvös Bence – Fazekas Dominik – Somogyi Bánk

Konzulensek:

Csontos Dénes, Kormos-Tóth László Erik, Fülöp Tibor

Miskolc 2026.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

**Informatikai rendszer- és alkalmazás-üzemeltető technikus**

# Dokumentáció

Ötvös Bence – Fazekas Dominik – Somogyi Bánk

2025-2026

**Vizsgaremek Hálózati Dokumentáció**

# Tartalomjegyzék

[1. Tartalomjegyzék 3](#_Toc219375529)

[2. Bevezetés 4](#_Toc219375530)

[3. Topológia 4](#_Toc219375531)

[4. Megvalósításhoz használt eszközök 5](#_Toc219375532)

[5. IP Táblázat 5](#_Toc219375533)

[6. Telephelyek 6](#_Toc219375534)

[**a) Ruby Telephely (komplex)** 6](#_Toc219375535)

[1) SSH 7](#_Toc219375536)

[2) Vlanok 8](#_Toc219375537)

[3) Etherchannel 11](#_Toc219375538)

[4) HSRP 13](#_Toc219375539)

[5) Dinamikus Forgalomirányítás 15](#_Toc219375540)

[6) PPP 16](#_Toc219375541)

[7) Tunnel 18](#_Toc219375542)

[8) NAT 20](#_Toc219375543)

[**b) Weiss Telephely** 21](#_Toc219375544)

[9) DHCP 22](#_Toc219375545)

[**c) Blake Telephely** 24](#_Toc219375546)

[10) Windows Server 25](#_Toc219375547)

[11) Linux Server 26](#_Toc219375548)

[7. Dokumentáció linkek: 26](#_Toc219375549)

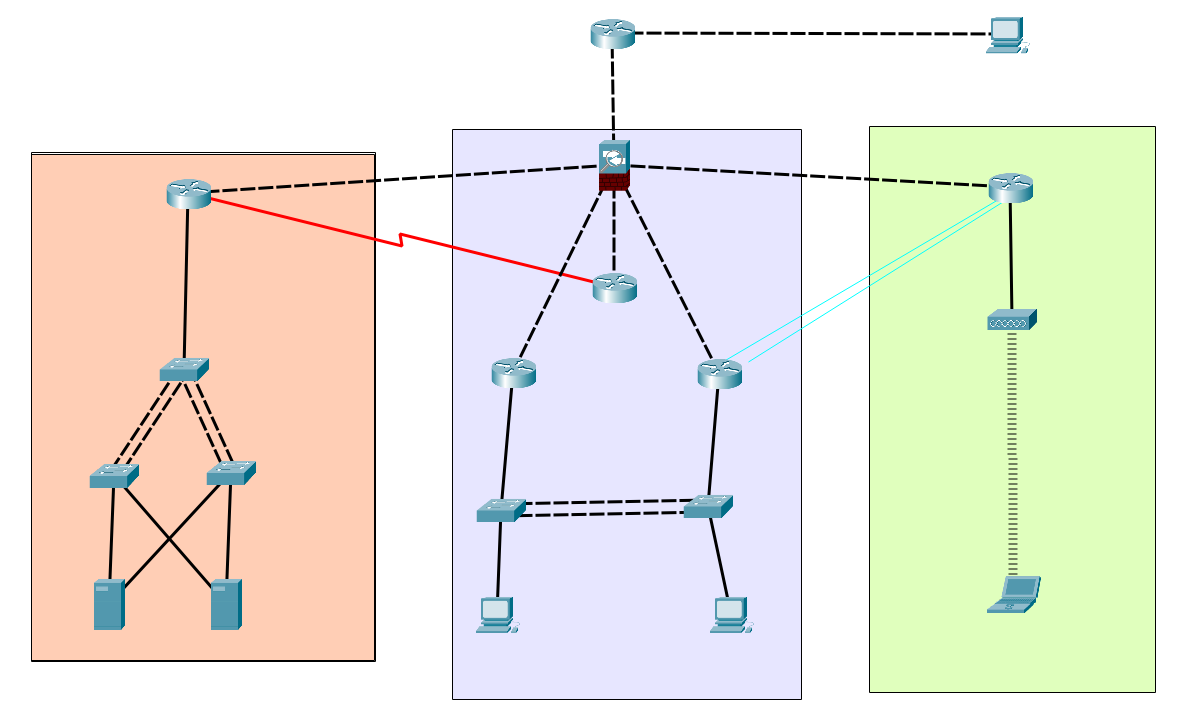
# Bevezetés

A projekt célja egy **biztonságos, skálázható és nagy rendelkezésre állású hálózati infrastruktúra** kialakítása, amely képes kiszolgálni különböző telephelyek (Blake, Ruby és Weiss) eltérő igényeit, miközben egységes címzési, routing- és biztonsági technológiákat alkalmaz.

**Ez a dokumentáció bemutatja:**

* a hálózat logikai és fizikai felépítését
* az alkalmazott technológiákat
* az IP-címzési tervet
* a telephelyek közötti kapcsolatok megvalósítását

# Topológia



**Weiss**

**Blake**

**Ruby**

ASA

PPP

Etherchannel

Tunnel

DHCP

Vlan 20

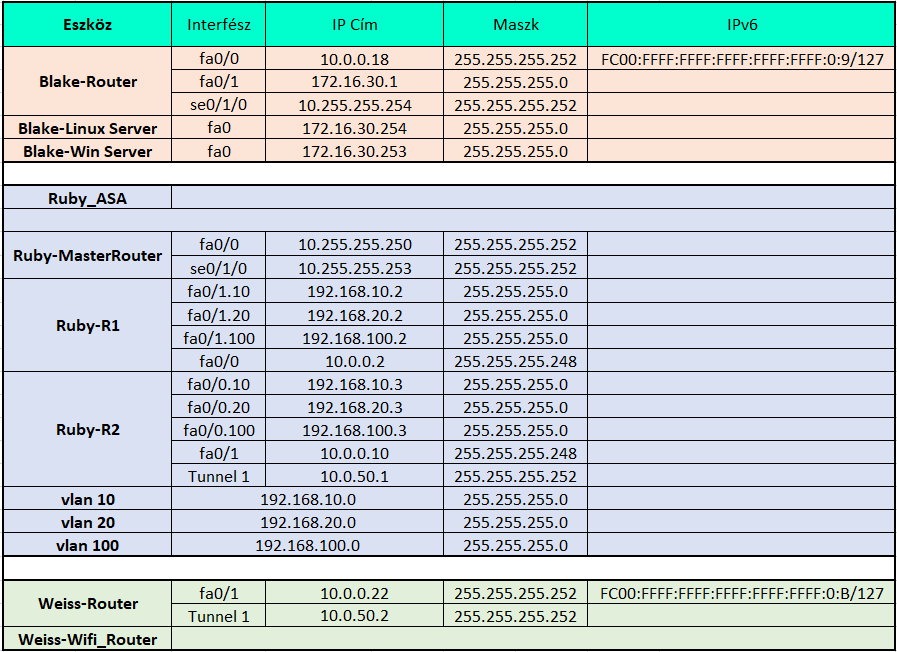
Vlan 10

Linux

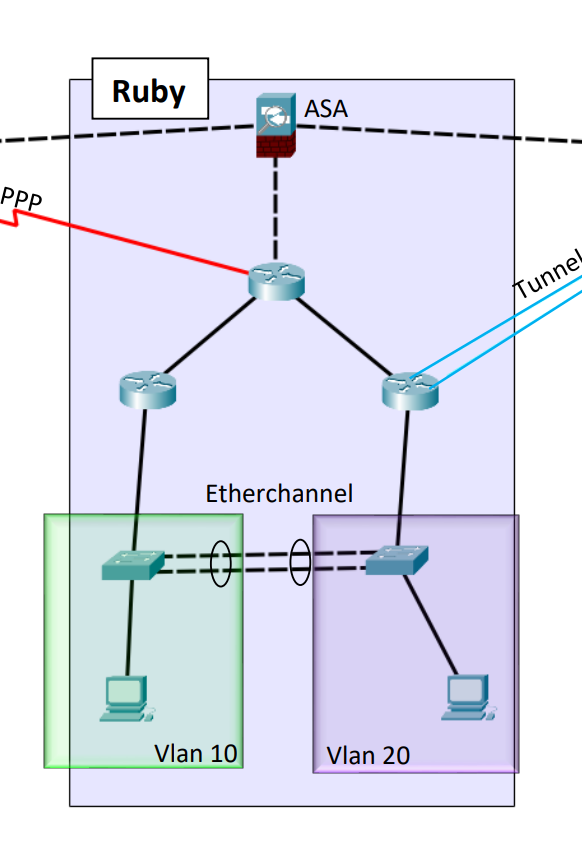
Windows

# Megvalósításhoz használt eszközök

# IP Táblázat



# Telephelyek



## Ruby Telephely (komplex)

Használt technológiák:

SSH

Vlan

Etherchannel

HSRP

OSPF

PPP

Tunnel

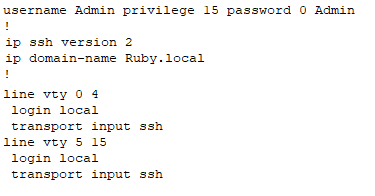
NAT

### SSH

Az SSH biztonságos távoli elérést biztosít számítógépek között, lehetővé teszi hogy távolról parancsokat futtassunk vagy fájlokat küldjünk a másik gépnek.

’Admin’ felhasználót és 15-ös titkosítási szinttű ’Admin’ jelszót állítottunk be. Az ip domain-name ’Ruby.local’ lett, a ’0 4’ és a ’5 15’ vonalon alkalmaztuk az ssh-t.

Parancsok SSH konfigurálásához:



Előnyei:

* Biztonságos
* Hitelesítés több módon
* Platformfüggetlen
* Sokoldalú
* Széles körben elterjedt

### Vlanok

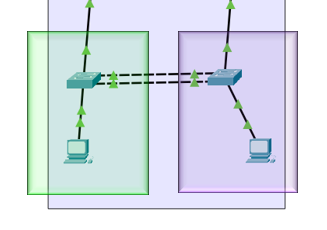
Vlanok segítségével egy fizikai hálózatot logikailag több, egymástól elkülönített hálózatra oszthatunk.

Előnyei:

* Biztonság
* Broadcast forgalom csökkentése
* Rugalmas hálózattervezés
* Költséghatékony
* Jobb menedzselhetőség

Létrehoztuk a Ruby telephelyen lévő switcheken a Vlan10-et és Vlan20-at a hálózatok elkülönítése érdekében.

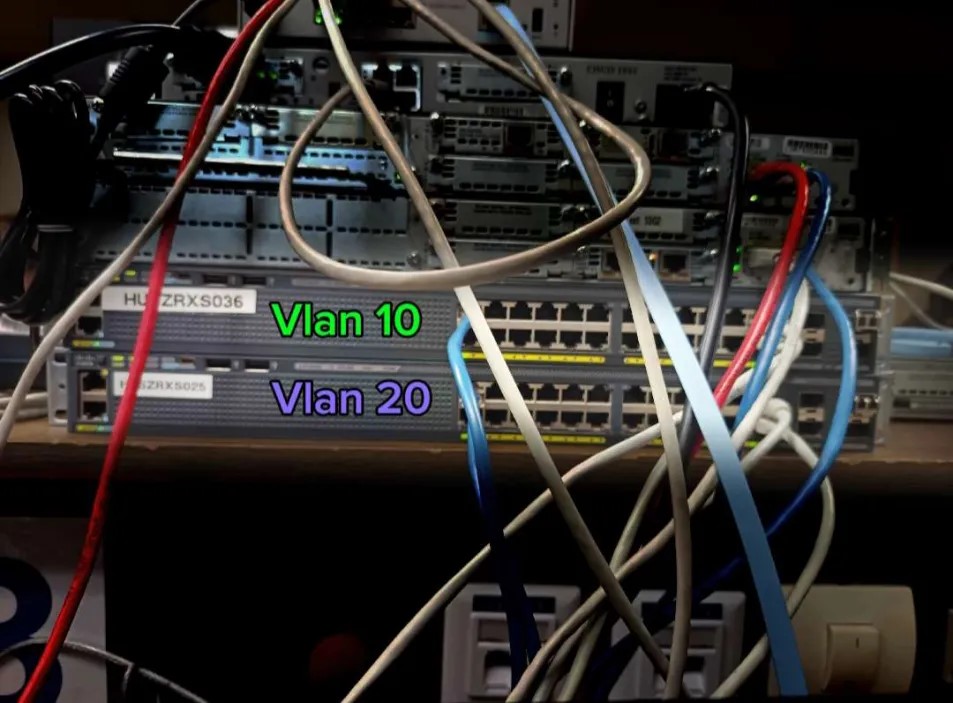
Vlanok a Ruby telephelyen





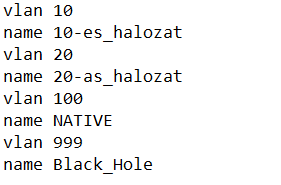
Vlan 20

Vlan 10

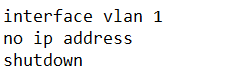


Ezzel növeljük a hálózat biztonságát és javítjuk a teljesítményét mivel elkülönítjük a forgalmat.

Vlanok létrehozása és elnevezése

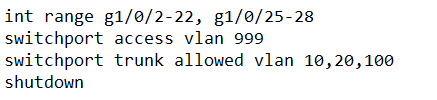


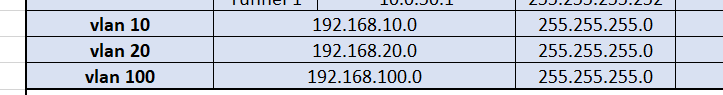
Vlan 1-et megszüntettük, mivel nem volt rá szükség.



Native Vlan-nak a Vlan 100-at állítottuk be.

Létrehoztuk a vlan 999-et, a ”Black Hole” vlant, amibe bekerült összes nem használt lekapcsolt port, tehát az összes port a g1/0/1, g1/0/23 és a g1/0/24-en kívül.



Vlan 10 kapta a 192.168.10.0-ás hálózatot, vlan 20 a 192.168.20.0-ásat, vlan 100 pedig a 192.168.100.0-ásat.

Ruby-SW1

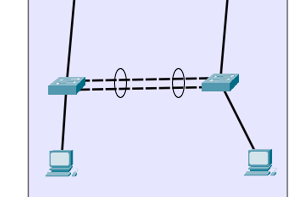




Ruby-SW2

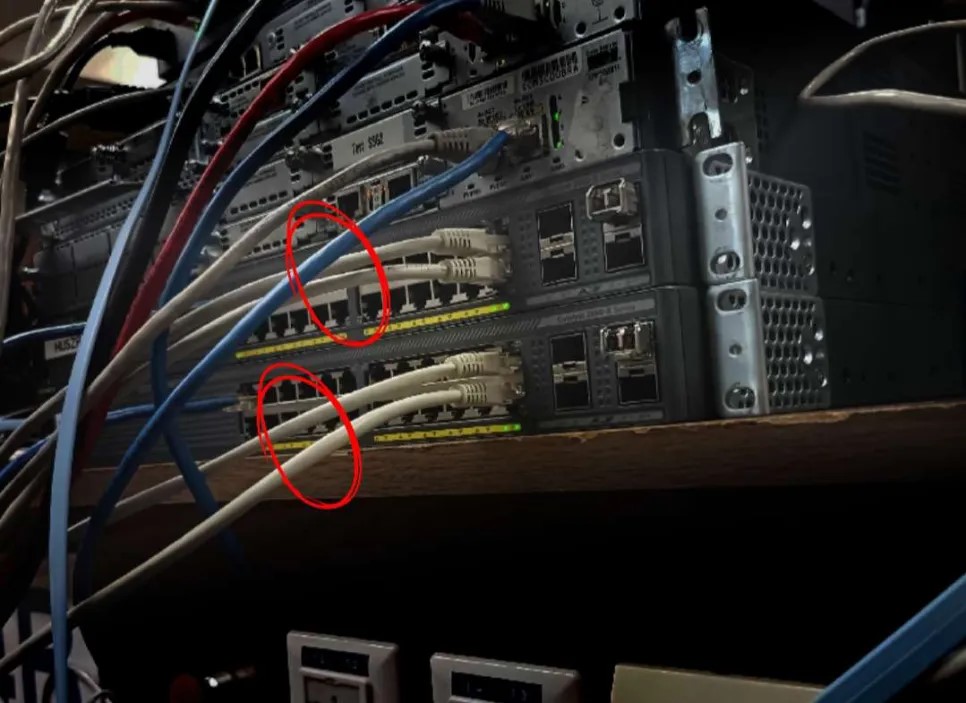
### Etherchannel

Az Etherchannel lehetővé teszi hogy egy switch két portját összefogjuk, így kettő helyett egyetlen összeköttetés jön létre.



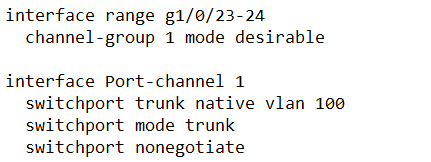
Két port összefogva Ruby-telephelyen

A két Ruby-Switch (Ruby-SW1 és Ruby-SW2) g1/0/23 és g1/0/24-es portjain konfiguráltuk.



Az etherchannel Cisco-s ’PAgP’ fajtáját konfiguráltuk (mode desirable).

Konfiguráció mindkét switchen



Előnyei:

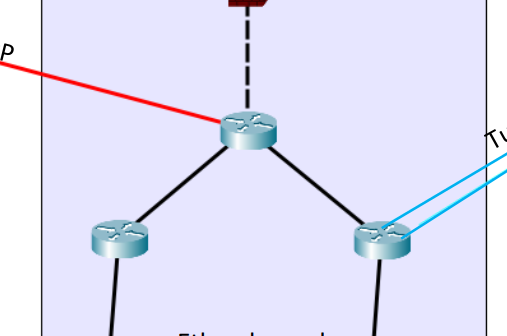
* Nagyobb sávszélesség
* Redundancia
* STP kompatibilis
* Skálázható
* Hatékonyabb erőforrás-kihasználás

### HSRP

HSRP-vel létrehozhatunk egy virtuális átjárót több routeren keresztül, ahova az elsődleges átjáró meghibásodása esetén átirányítható a forgalom.

Célja: Redundanciát biztosít a hálózatban, meghibásodás esetén nem szakad meg a kapcsolat.

Mi a Ruby telephely routerein alkalmaztuk:



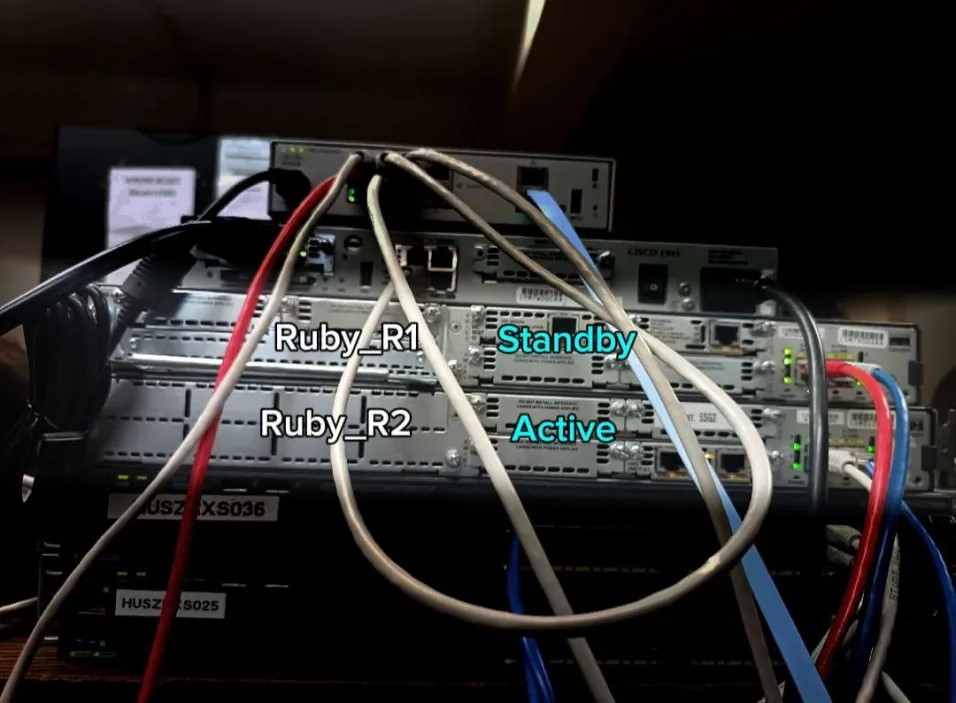
Aktív Router

Készenléti Router

R2

R1

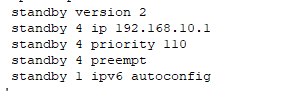
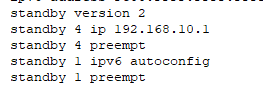
Ha az ’Aktív’ router meghibásodik vagy megszakad a kapcsolat, a készenléti router átveszi a helyét amíg a probléma meg nem oldódik, így nincs leállás.



Parancsok a HSRP konfigurálásához:

Ruby R1

Ruby R2

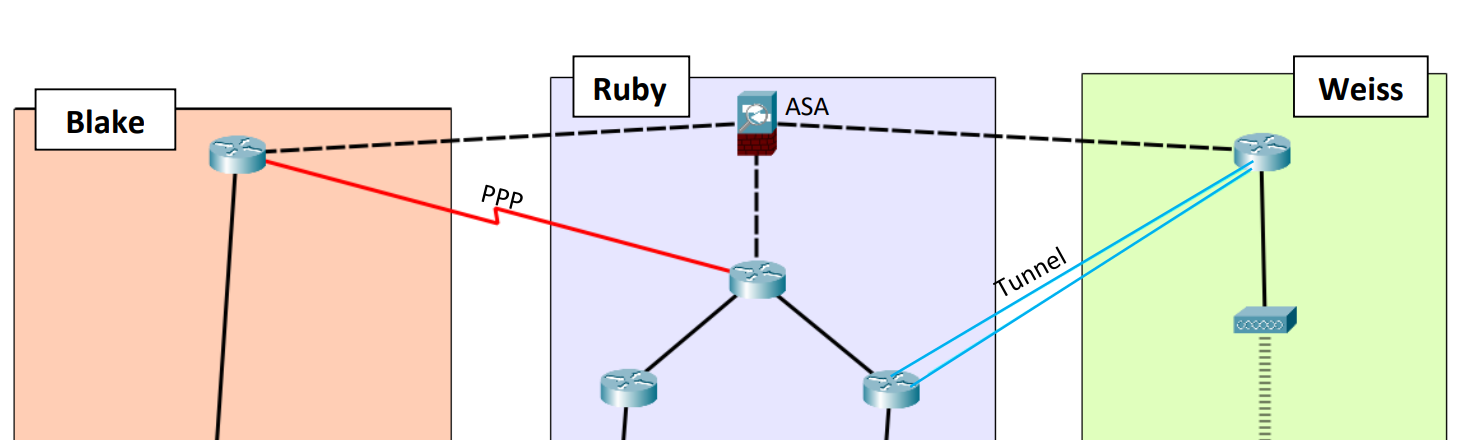


Előnyei:

* Magas rendelkezésre állás
* Automatikus átváltás
* Egyszerű kliensoldalon
* Stabil és kiforrott
* Preemption támogatás

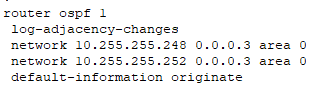
### Dinamikus Forgalomirányítás

A dinamikus forgalomirányítás automatikusan kiválasztja a legjobb útvonalat a forgalom elosztása érdekében.

Mi az OSPF útválasztási protokoll-t használtuk.

OSPF-el konfigurált eszközök

Konfigurálás során megadjuk azokat a hálózatokat, amelyeket az adott forgalomirányító ismer.



Master Router

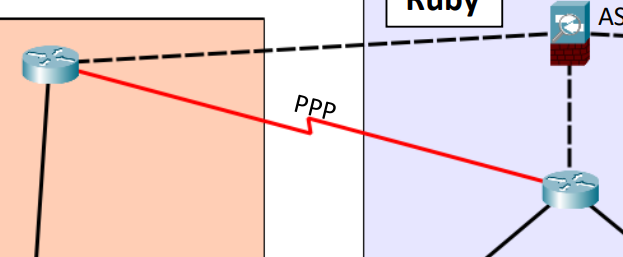
Előnyei:

* Automatikus útvonalfrissítés
* Skálázhatóság
* Kevesebb manuális konfiguráció
* Optimális útvonalválasztás
* Redundancia kihasználása

### PPP

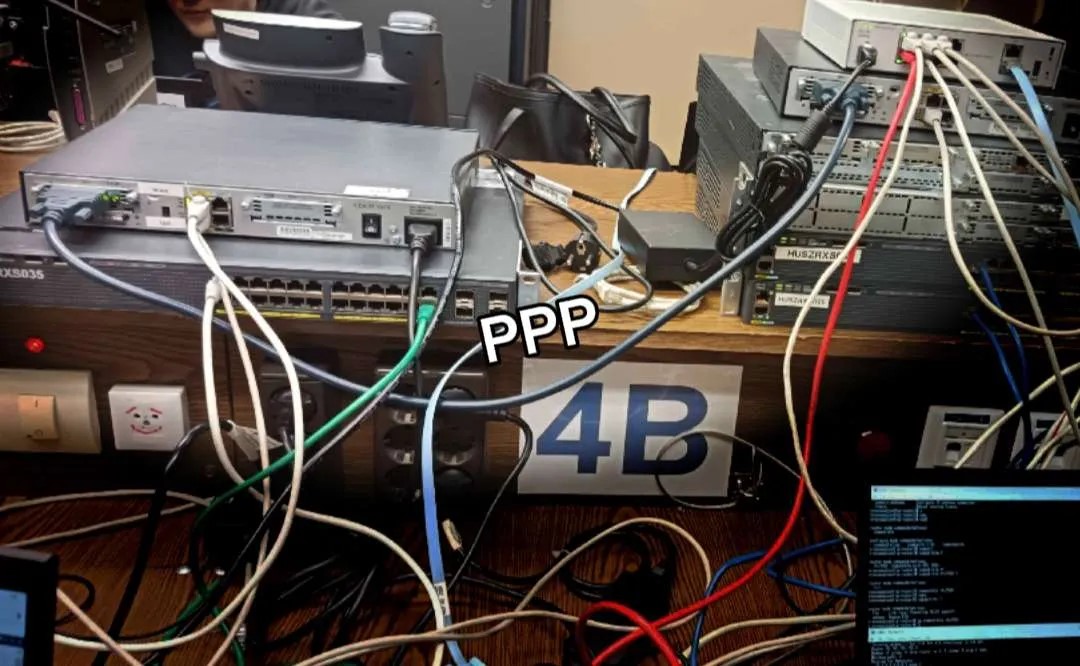
A PPP (Point-to-Point Protocoll) két hálózati eszköz között biztonságos, szabályozott és megbízható adatkapcsolat létrehozására szolgál.

MasterRouter és Weiss\_R1 közötti serial kapcsolaton alkalmaztuk:



**Blake\_Router**

**MasterRouter**



Mindkét routeren létrehozunk egy felhasználót a másik router nevével.

Master Router



Hitelesítésnek ’chap’-ot állítottunk, újjabb és jobb mint a ’pap’.



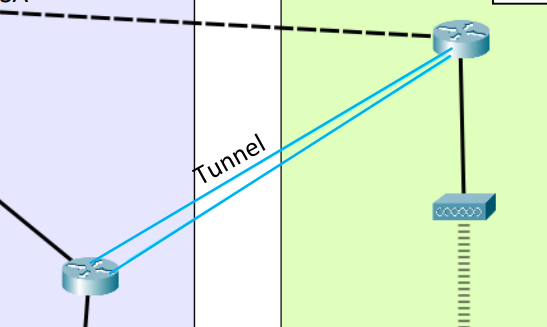
Előnyei:

* Egyszerű és stabil
* Hitelesítést támogat
* Több hálózati protokoll
* Széles körben támogatott
* PPP over Ethernet (PPPoE)

### Tunnel

A tunnel segítségével egy hálózati protokoll csomagjait egy másik, idegen protokollon keresztül továbbítják, mintha egy „alagúton” utaznának, így biztonságosan vagy nem kompatibilis hálózatok között képesek kommunikálni.

Ruby\_R2 és Weiss\_R1 között alkalmaztuk:

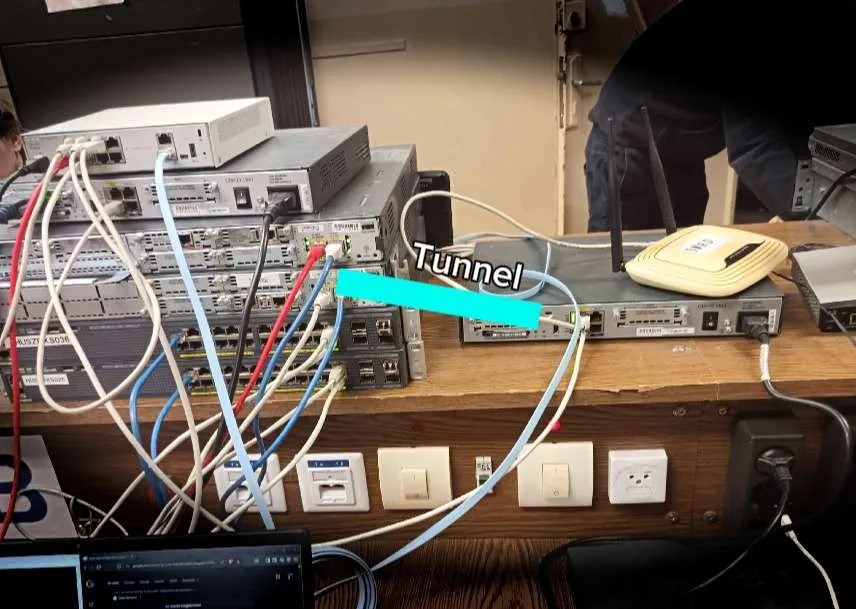


**WeissR1**

**RubyR2**

Tunnel ’igazi’ útvonala (pirossal):





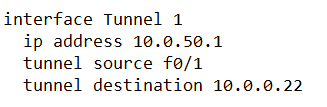
Ruby-R2 tunnel IP-nek 10.0.50.1-et állítottunk, Weiss-R-en pedig 10.0.50.2-t.



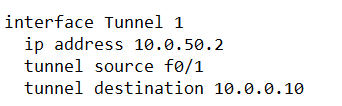
Ruby-R2

Weiss-R





Ruby R2



Weiss-router

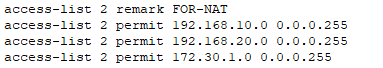
### NAT

A NAT egy routeren vagy tűzfalon működik amely a belső (privát) IP-címeketpublikus IP-címekre fordítja, amikor az eszközök az internet felé kommunikálnak.

Előnyei:

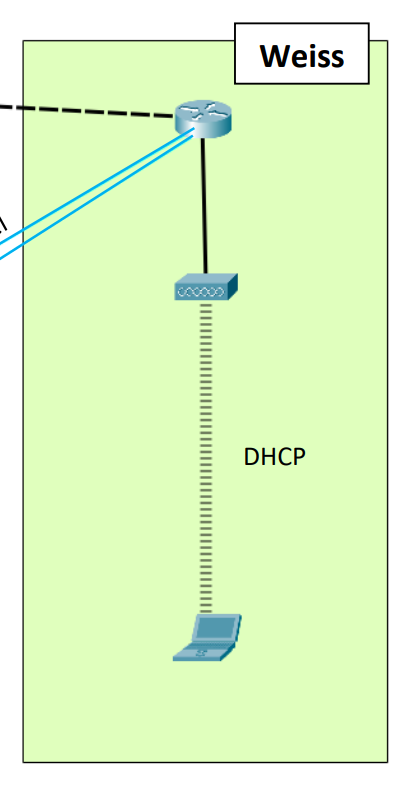
* IPv4 címmegtakarítás
* Alap biztonság
* Egyszerű belső címzés
* Függetlenség az internetszolgáltatótól
* Széles körben támogatott

Parancsok a NAT konfigurálásához:



ACL Nat-hoz





## Weiss Telephely

### DHCP

DHCP segítségével automatikusan osszon ki hálózati beállításokat az eszközöknek.

A DHCP segítségével a kliens automatikusan megkapja:

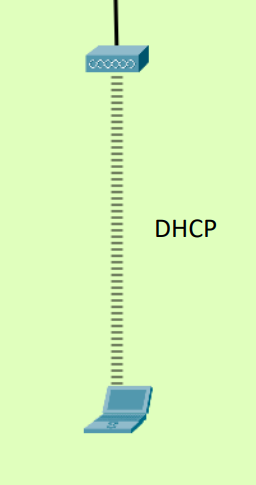
* IP-címét
* Alhálózati maszkot
* Alapértelmezett átjárót (gateway)
* DNS szerver címét

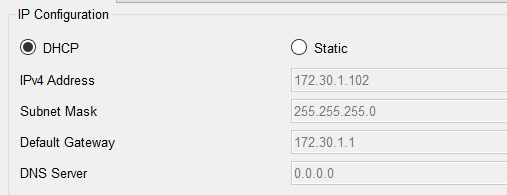
Így nem kell kézzel konfigurálni minden egyes eszközt.

Előnyei:

* Automatizálás
* Kevesebb konfigurációs hiba
* Központi menedzsment
* Rugalmas
* Támogat statikus kiosztást is

A Weiss telephelyen a laptopok az Access Point-tól kapnak címet DHCP-n keresztül

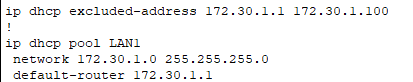


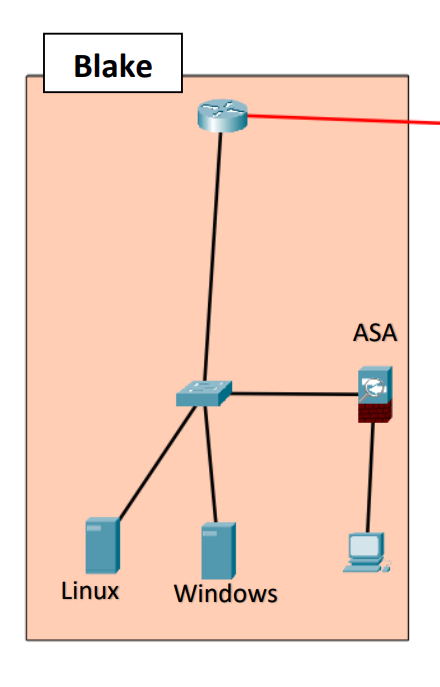


DHCP az egyik laptopon

Parancsok a DHCP konfigurálásához:

Weiss-router





## Blake Telephely

### Windows Server

### Linux Server

# Dokumentáció linkek:

Prezentáció: [Cím nélküli prezentáció (5).pptx](Cím%20nélküli%20prezentáció%20(5).pptx)

Videó:

Packet Topológia: [TOPO.png](../Pictures/TOPO.png)

Élő berendezés:

Tesztelési Dokumentáció: <Projekt_Word_Tesztelés.docx>