

Szegedi Tudományegyetem

Informatikai Intézet

**Terhelés-kiegyenlítés AP-asszisztált roaming
segítségével OpenWrt-en**

Szakdolgozat

Készítette:

Südi Tamás

programtervező informatikus
szakos hallgató

Külső témavezető:

Dr. Kelemen András

egyetemi docens

Belső témavezető:

Csirik János

egyetemi tanár

Szeged

2023

Tartalomjegyzék

1. Az OpenWrt rendszer bemutatása	5
1.1. Telepítés fizikai eszközre	5
1.1.1. Telepítés ASUS TUF AX4200 eszközre	5
1.2. Telepítés virtuális környezetbe	7
1.2.1. Virtuális környezet Linux alatt	7
1.2.2. Virtuális környezet Windows alatt Hyper-V segítségével	7
1.2.3. Virtuális környezet Oracle VirtualBox segítségével	7
1.3. Hálózat beállítása OpenWrt rendszeren	9
1.4. A WAN port beállítása	9
1.5. A WLAN beállítása	9
2. Vezeték nélküli hálózati protokollok és fogalmak	10
2.1. Az IEEE 802.11 Wi-Fi szabványcsalád	10
2.2. Wi-Fi Access Point	10
2.3. Wi-Fi Station	10
2.4. SSID	11
2.5. BSS	11
2.6. Frekvenciasávok és csatornák	11
2.7. Átviteli sebesség	11
2.7.1. Moduláció és kódolás	11
2.7.2. Időosztás és más osztási technikák	12
2.7.3. Interferencia	12
2.7.4. A távolság hatása	12
2.8. Authentikáció	12
2.9. Asszociáció	12
2.10. DHCP	13
2.11. Wi-Fi irányítási protokollok	13
2.11.1. 802.11r Fast BSS Transition	13
2.11.2. 802.11k Radio Resource Measurement	14
2.11.3. 802.11v Wireless Network Management	14
2.12. Az OpenWrt hostapd és parancsai	15
2.12.1. A Wi-Fi kliensek listázása	15
2.12.2. A 802.11v transition request küldése	16
3. Terheléskiegyenlítés	17
4. Eredmények	17
4.1. Mérési környezet és metrikák	17
4.1.1. Prometheus	17
4.1.2. Grafana	17
4.1.3. A lokális tesztelés környezete	17
4.1.4. A kollégium hálózatán végzett mérések	17

Feladatkiírás

A témavezető által megfogalmazott feladatkiírás. Önálló oldalon szerepel.

Tartalmi összefoglaló

A dolgozat célja a centralizált AP-asszisztált roaming megvalósítása OpenWrt alapú rendszerekre. A dolgozat célja továbbra, hogy bemutassa az OpenWrt rendszert, a roaming technikákat és a szükséges 802.11 protokollokat.

A dolgozat első fejezete az OpenWrt rendszer bemutatásával és telepítésével foglalkozik. Összehasonlítja a virtuális fejlesztői környezet előnyeit és hátrányait a fizikai környezettel, telepítés, használat, sebesség és konfigurálhatóság szempontjából. A dolgozat második fejezete a hálózati protokollokkal, a roaming technikákkal és a szükséges 802.11 protokollok bemutatásával foglalkozik, ismerteti az AP-asszisztált roamingot. A dolgozat harmadik fejezete a terhelés-kiegyenlítési algoritmusokkal foglalkozik. A dolgozat negyedik részében az éles környezetben történő tesztelésről és a kapott eredményekről szól. A dolgozat utolsó fejezete a szakirodalmat és a kapcsolódó projekteket mutatja be.

1. Az OpenWrt rendszer bemutatása

Az OpenWrt egy Linux-alapú, nyílt forráskódú, hálózati eszközökhöz készült operációs rendszer.

Az OpenWrt a gyártói firmware helyett telepíthető a támogatott eszközökön. Ahhoz képest számos előnyt kínál a felhasználóknak, beleértve a letisztultságot, a nagyobb testreszabhatóságot, több funkciót és jobb biztonságot.

A legtöbb komponens és a build rendszer a GNU General Public License Version 2 licenz alatt érhető el, azonban néhány, elsősorban a nem OpenWrt-ben létrehozott részek más licensek alatt állnak. [5] [7] [8]

1.1. Telepítés fizikai eszközre

Egyes eszközök már rendelkeznek OpenWrt vagy OpenWrt alapú firmwarrel, azonban legtöbbször ennek telepítése a felhasználó feladata.

A támogatott eszközök listája a <https://openwrt.org/toh/start> oldalon található. Itt az eszköz támogatottságától és népszerűségétől függően megtalálhatóak annak specifikációi, a hozzá tartozó firmwarek letöltési linkje és a telepítési, visszaállítási útmutatók.

A telepítés folyamata eszközönként eltérő lehet, de általában az alábbi módokon történhet: az eszköz webes kezelőfelületen keresztül, FTP-n keresztül, SD-kártya vagy USB-meghajtó segítségével, soros port használatával. Ez azonban a garancia elvesztésével és az eszköz meghibásodásával járhat.

[10] [11]

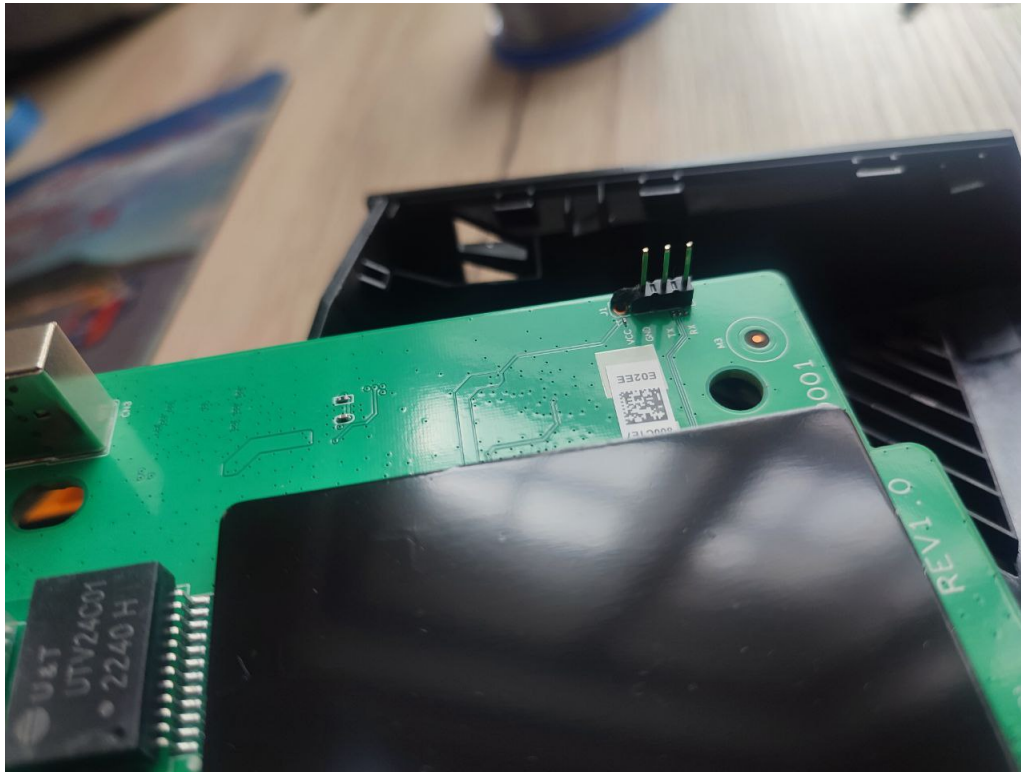
1.1.1. Telepítés ASUS TUF AX4200 eszközre

Az én általam használt ASUS TUF AX4200 eszközre az alaplapon található soros porton keresztül telepítettem az OpenWrt-t. A telepítést el lehet végezni egy Raspberry PI segítségével is, de nekem rendelkezésemre állt egy USB-soros port átalakító, így azt használtam. [6]

Az eszköz szétbontása rendkívül egyszerű, a lábakban található négy csavar eltávolítása után az előlap könnyedén lepattintható.

Az alaplapon egy négy csatlakozós sorosport található. A jelölés alapján a legfelső a táp, a második a földelés, a harmadik a TX, a negyedik pedig az RX. Mivel az eszköz rendelkezik saját táppal, ezért azt nem csatlakoztattam.

Mivel az eszközön fejlesztést és tesztelést tervezek végezni, így proaktivitás céljából a soros portra egy 3 pin-es csatlakozót forrasztottam, hogy a későbbiekben gyorsabban tudjam csatlakoztatni a számítógéphez.



```
1 Please choose the operation:
2   1: Load System code to SDRAM via TFTP.
3   2: Load System code then write to Flash via TFTP.
4   3: Boot System code via Flash (default).
5   4: Entr boot command line interface.
6   7: Load Boot Loader code then write to Flash via Serial.
7   9: Load Boot Loader code then write to Flash via TFTP.
```

Ezután elindítottam az eszközt és a 4-es gomb nyomva tartásával megszakítottam a boot folyamatot és beléptem egy CLI-be.

Eztán az eszköz LAN1 portját egy Ethernet kábellel a számítógémemhez csatlakoztattam és a számítógémem hálózatát statikusra állítottam, hogy felvegyem a 192.168.1.66 címet és a /24-es maszkot.

A routeren az alábbi parancsok megadásával elindítottam a lecsúpasított OpenWrt rendszert:

```
1 setenv ipaddr 192.168.1.1
2 setenv serverip 192.168.1.66
3 tftpboot 0x46000000 openwrt.bin
4 bootm 0x46000000
```

Ez az OpenWrt fájl nem tartalmazza a webes kezelőfelületet, ezért scp parancs segítségével húztam át a teljes firmwaret a routerre.

```
1 scp sudta@192.168.1.66:/tmp/openwrt-mediatek-filogic-asus_tuf-ax4200-
  squashfs-sysupgrade.bin /tmp
2 sysupgrade -n /tmp/openwrt-mediatek-filogic-asus_tuf-ax4200-squashfs-
  sysupgrade.bin
```

1.2. Telepítés virtuális környezetbe

A virtuális környezet egy olyan szoftveres megoldás, ami lehetővé teszi azt, hogy a felhasználó egyszerre futtasson több, akár különböző operációs rendszert is a számítógépén.

A virtuális környezet rengeteg előnnyel járhat egy fejlesztő számára, mint például az extra eszköz használatának elkerülése, az egyszerűbb fájlátvitel, a kijelző és a billentyűzet használata, a gyorsabb hardver, mentések készítése és visszaállítása és ezek megosztása más fejlesztőkkel. Azonban az ilyen környezeteknek is lehetnek hátrányai, mint például, hogy a hálózati kártya nem rendelkezik a szükséges hardveres támogatással és nem olyan megbízható a teljesítménye, mint egy erre tervezett eszköznek.

1.2.1. Virtuális környezet Linux alatt

A Linux-alapú operációs rendszerek népszerűek a fejlesztők körében, mivel ingyenesek és számos olyan funkcióval rendelkeznek, amelyek lehetővé teszik a hatékony és kényelmes munkavégzést.

Ezen a platformon több virtuális környezet is elérhető, mint például a VirtualBox, a VMware Workstation vagy a KVM.

A nyílt forráskódú Kernel-based Virtual Machine (KVM) egyik előnye, hogy képes átadni PCI csatlakozású eszközöket is virtuális gépnek, így alacsony szintű hozzáférést biztosít a hardverhez. [4] [12]

Több virtuális környezet kezelő alkalmazás is támogatja mind az USB, mind a PCI csatlakozású eszközök átadását. Én ezek közül a virt-manager nevű programot választottam.

Hogy egyszerűsítsem a telepítési folyamatot [gpu-passthrough-manager](#) nevű szoftvert használtam, amely segítségével felkészítettem a rendszeremet a PCI csatornán keresztül csatlakozó WLAN-vezérlő átadására.

Ezután már csak meg kellett adnom a grub rendszerbetöltőnek, hogy a rendszert a vfio-pci.ids=8086:06f0 kernelparaméterrel indítsa el, amely a laptopom WLAN-vezérlő azonosítóját jelöli.

A számítógépem architektúrájának megfelelően a [openwrt-22.03.2-x86-64-generic-ext4-combined.img](#) képet használtam a virtuális gép létrehozásához. Ez nem tartalmaz semmilyen telepítőt, helyette grub rendszerbetöltő segítségével indítja el a rendszert.

1.2.2. Virtuális környezet Windows alatt Hyper-V segítségével

A Hyper-V a Microsoft Windows natív virtualizációs környezete.

Ezt használja például a fejlesztők által kedvelt Windows Subsystem for Linux (WSL) Version 2 is, amely lehetővé teszi a Linux kernel futtatását Windows alatt, így alacsonyabb szintű hozzáférést biztosít a hardverhez. [16] [14] Bár a Hyper-V képes bizonyos eszközök átadására a virtuális gépnek, de ezeket a funkciókat a WSL2 nem támogatja.

A Hyper-V eszközátadás protokolljának fejlesztésekor a cél eszközök nem hálózati kontrollerek, hanem perifériák és tárolóeszközök voltak. [18]

1.2.3. Virtuális környezet Oracle VirtualBox segítségével

Az Oracle VirtualBox egy multiplatform virtuális környezet, amely használatához először le kell tiltani a Hyper-V-t Windows alapú rendszereken. [3] Ezt legegyszerűbben a host

rendszeren futó parancssorból lehet elvégezni:

```
1 bcdedit /set hypervisorlaunchtype off
```

Habár az Oracle Virtualbox nem támogatja a .img kiterjesztésű képeket, de tartalmazza a VBoxManage nevű programot, amely segítségével az img kép átkonvertálható .vdi kiterjesztésű virtuális lemezzé.

```
1 & 'C:\Progra~1\Oracle\VirtualBox\VBoxManage.exe' convertfromraw --  
format VDI '.\openwrt-22.03.3-x86-64-generic-ext4-combined.img' '.\  
openwrt.vdi'
```

Az újonnan létrejött openwrt.vdi kép segítségével létrehozható a virtuális gép. Alapértelmezetten a host gépről nem érhető el a virtuális gép hálózata, azonban ez megoldható port-forwarding szabályok felvétele segítségével a Virtuális gép beállításainak Hálózat / adapter1 / speciális / port forwarding menüpontjában.

1.3. Hálózat beállítása OpenWrt rendszeren

1.4. A WAN port beállítása

A Wide Area Network (WAN) port a routert egy másik hálózathoz csatlakoztatja, amely lehet egy másik router, de akár közvetlenül az internet is.

Az OpenWrt indítása után lehetséges, hogy az internethez való csatlakozás nem sikerül. Ennek ellenőrzését a ping parancs segítségével lehet elvégezni:

```
1 root@OpenWrt:/# ping vanenet.hu
2 ping: bad address 'vanenet.hu'
3 root@OpenWrt:/# ping 1.1.1.1
4 PING 1.1.1.1 (1.1.1.1): 56 data bytes
5 ping: sendto: Network unreachable
```

Ha a ping parancs nem sikerül, akkor valószínűleg a WAN porthoz tartozó interface konfigurációja nem megfelelő. Ezt a /etc/config/network fájlban lehet szerkeszteni.

Ha a csatlakozó hálózaton DHCP szerver üzemel, akkor az alábbi módosításokkal lehet a DHCP protokollt engedélyezni:

```
1 config interface 'lan'
2     option device 'br-lan'
3 +   option proto 'dhcp'
4 -   option proto 'static'
5 -   option ipaddr '192.168.1.1'
6 -   option netmask '255.255.255.0'
7 -   option ipassign '60'
```

A változtatások érvényesítéséhez újra kell indítani a network szolgáltatást.

Ezt a következő parancs segítségével lehet elvégezni:

```
1 service network restart
```

Ezután a ping parancs segítségével lehet ellenőrizni, hogy a WAN port beállítása sikeres volt-e:

```
1 root@OpenWrt:/# ping vanenet.hu
2 PING vanenet.hu (185.33.54.12): 56 data bytes
3 64 bytes from 185.33.54.12: seq=0 ttl=54 time=8.351 ms
4 64 bytes from 185.33.64.12: seq=1 ttl=54 time=8.104 ms
5 64 bytes from 185.33.64.12: seq=2 ttl=54 time=7.570 ms
```

1.5. A WLAN beállítása

Az eszközhöz készített image általában tartalmazza a szükséges drivereket, ekkor a WLAN beállítása egyszerűen megoldható a webes felületen keresztül.

Lehetséges azonban, hogy ezeket manuálisan kell telepíteni. Az opkg csomagkezelő segítségével lehet beszerezni a szükséges csomagokat. A telepítéshez először frissíteni kell a csomaglistát:

```
1 root@OpenWrt:/# opkg update
```

Ez a parancs letölti az internetről a RAM-ba a csomaglistát, majd a list parancs segítségével megtekinthető a letöltött csomagok listája.

Ezután a gyártóhoz tartozó kernel modult is telepíteni kell. Intel eszközök esetében ezt a `kmod-iwlwifi` csomag biztosítja. A telepítés után engedélyezni kell a modult, majd újra kell indítani a rendszert. [1]

```
1 root@OpenWrt:/# opkg install kmod-iwlwifi
2 root@OpenWrt:/# modprobe iwlwifi
3 root@OpenWrt:/# reboot
```

Ha ezután sem jelenik meg a WLAN interface, akkor valószínűleg nincsen telepítve a szükséges meghajtó. A `dmesg` parancs segítségével lehet megtekinteni a rendszerüzenekeket, és ebből kideríteni, hogy melyik drivert kell telepíteni.

```
1 [ 4.953303] iwlwifi 0000:07:00.0: Direct firmware load for iwlwifi-
   QuZ-a0-hr-b0-39.ucode failed with error -2
2 [ 4.957677] iwlwifi 0000:07:00.0: minimum version required: iwlwifi-
   QuZ-a0-hr-b0-39
3 [ 4.958681] iwlwifi 0000:07:00.0: maximum version supported: iwlwifi-
   QuZ-a0-hr-b0-66
```

A fenti üzenet azt jelenti, hogy a `iwlwifi-QuZ-a0-hr-b0-xx.ucode` fájl hiányzik, és ezt kell telepíteni. A népszerűbb driverek elérhetőek az OPKG csomagkezelőben, de gyártói oldalakról is letölthetőek. Az Ubuntu operációs rendszer készítői egy olyan tárolót tartanak fent, ami tartalmazza a legtöbb gyártóhoz tartozó illesztőprogramokat.

Ezt az alábbi linken lehet elérni: <https://git.launchpad.net/ubuntu-kernel/ubuntu/+source/linux-firmware/tree/>

Innen az OpenWrt eszköz `/lib/firmware/` könyvtárába kell másolni a megfelelő fájlt. A fenti példában ez a `iwlwifi-QuZ-a0-hr-b0-66.ucode` fájl lenne.

2. Vezeték nélküli hálózati protokollok és fogalmak

2.1. Az IEEE 802.11 Wi-Fi szabványcsalád

Az Institute of Electrical and Electronics Engineers (IEEE) nemzetközi szakmai szervezet 802 szabványcsaládja definiálja a legismertebb vezeték hálózati szabványokat. Ennek a kollekciónak a része a 802.11 Wi-Fi család is, amely a lokális vezeték nélküli mikrohullám alapú kommunikációt szabványosítja.

Az osztályban a legismertebb szabványok az a, b, g, n, ac, ax 802.11 protokollok. A szabványok közötti különbség a frekvenciasávban, a jelerősségben, a csatornaszélességben, a modulációs technikában és ezáltal a maximális adatátviteli sebességben rejlik.

2.2. Wi-Fi Access Point

Az Wi-Fi Access Point egy olyan hálózati eszköz, amely lehetővé teszi a vezeték nélküli csatlakozást a hálózathoz.

2.3. Wi-Fi Station

A "Wi-Fi állomás" (STA) egy olyan általános kifejezés, amely egy olyan eszközt jelent, amely csatlakozik egy Wi-Fi hálózathoz. Ez lehet egy laptop, egy okostelefon, egy IoT

eszköz vagy egy vezeték nélküli repeater is.

2.4. SSID

Az SSID (Service Set Identifier) egy karakterlánc típusú azonosító, amelyet a hálózat sugároz a környezetébe. Ez által a felhasználók könnyen azonosíthatják és megkülönböztethetik a hálózatokat egymástól.

2.5. BSS

A Basic Service Set (BSS) arra utal, hogy az adott objektum egy primitív építőeleme a szolgáltatásnak. Wi-Fi hálózatok esetében ez egyetlen hozzáférési pont, amelyet a BSSID-ja azonosít. Ez általában az adott eszköz MAC címe.

Fontos megjegyezni, hogy egy BSS több SSID-t is sugározhat és egy fizikai eszközbe több BSS is beépíthető. A 2.4 és 5 GHz-et is támogató dual-band AP-k általában két BSS-t is tartalmaznak.

2.6. Frekvenciasávok és csatornák

A legegyszerűbben Wi-Fi szabványok a 2,4 GHz és az 5 GHz elnevezésű frekvenciasávokat használják.

A 2,4 GHz-es sáv a régebbi és elterjedtebb frekvenciasáv, amely a 2400 és 2483 MHz közötti tartományba esik. Mivel ez a sáv szabadon használható, azt más eszközök, például mikrohullámú sütők, Bluetooth eszközök vagy rádiók is használhatják.

Az 5 GHz-es sáv csak az újabb Wi-Fi szabványok támogatják, mint például a 802.11ac és a 802.11ax. Ez a 5160 és 5895 MHz közötti szélesebb frekvenciasávot használja. Ez nagyobb sáv szélességet és kevesebb interferenciát kínál, ezáltal jobb teljesítményt eredményez. Azonban terjedése gyengébb, ezért a társánál rövidebb hatótávolságot biztosít.

A Wi-Fi kontextusában a frekvenciasávok mellett beszélhetünk csatornákról is, amelyek az adott frekvenciasávokhoz rendelt azonosítók.

Az eszközök általában több szomszédos csatornát, azaz szélesebb frekvenciasávokat, például 20, 40, 80 vagy akár 160 MHz-et is felhasználhatnak, ha képesek erre a helyi szabályozások is engedélyezik ezt. [17]

Azonban annak ellenére, hogy egy eszköznek van tanúsítványa egy adott frekvenciasávhoz, nem feltétlenül tudja használni az összes csatornát abban.

2.7. Átviteli sebesség

Az átviteli sebesség azt jelenti, hogy adott időablakban milyen mennyiségű adatot képes átadni az eszköz a hálózaton.

Az elérhető sebességet általában az alábbi tulajdonságokkal írjuk le:

2.7.1. Moduláció és kódolás

A moduláció és kódolás (MCS) tábla a használható jelek tulajdonságait, például a frekvenciájukat, a fázisukat és az amplitúdójukat határozza meg.

Az eszközök számos MCS érték közül választhatnak, az eszközök támogatása, a jel erőssége és a kommunikáció minősége függvényében. Ezeket a <https://mcsindex.com/> oldalon lehet megtekinteni. Az alacsonyabb MCS értékek általában alacsonyabb sebességet, míg a magasabb MCS értékek magasabb sebességet biztosítanak.

2.7.2. Időosztás és más osztási technikák

Az időosztás egy olyan technika, amely lehetővé teszi, hogy több eszköz egyidejűleg kommunikáljon az adott hálózaton. Ilyenkor egy adott időintervallum felosztásra kerül több szeletre, amelyekben a kliensek egymást váltva kommunikálhatnak.

A vezeték nélküli hálózatok más osztási technikákat is alkalmazhatnak.

A Code Division technika segítségével az eszközök különböző kódolások használatával egymástól függetlenül és egyidejűleg használják ugyanazt a frekvenciasávot. A kódok merőlegessége lehetővé teszi az adatok elkülönítését a fogadó oldalon.

A MIMO (Multiple-Input Multiple-Output) rendszerek segítségével az eszközök különböző irányokba mutató antennákat használnak, amelyek lehetővé teszik az adatok irányított átvitelét és fogadását. Ezáltal az eszközök egymástól függetlenül és egyidejűleg kommunikálhatnak ugyanazon a frekvenciasávon, minimalizálva az interferenciát és növelve a kapacitást.

2.7.3. Interferencia

Az interferencia a vezeték nélküli hálózatokban előforduló jelenség, amely akkor következik be, amikor egy adott jel egy másik jelre hatást gyakorol. Ekkor a zavaró jelet zajnak nevezzük.

Ennek eredményeképpen a jel torzul és az átvitel minősége romlik, azonban ez nem feltétlenül eredményez hibás adatokat, mivel a kódolási technikák ezt megakadályozzák ellenőrzés és javítás segítségével.

2.7.4. A távolság hatása

A távolság növekedésével a jel erőssége csökken, több zavaró jel kerül annak útjába és elnyelődhet vagy rosszabb esetben visszaverődhet, ezáltal zajt okozva.

2.8. Authentikáció

Az autentikáció a hálózatban történő azonosítási folyamatot jelenti, amely során a felhasználó azonosítja magát a hálózat felé. Ez általában jelszóval vagy valamely egyéb közös kulcsú titkosítás (Shared Key) használatával történik. Az autentikáció bekapcsolása erősen ajánlott, de nem kötelező.

2.9. Asszociáció

Az asszociáció a Wi-Fi hálózatokban a csatlakozási folyamatot jelenti, amely során az eszközök kapcsolatot létesítenek és azonosítják magukat a hálózaton belül. Amikor egy eszköz csatlakozni kíván egy Wi-Fi hálózathoz, asszociációs kérelemmel küld ki. A hálózat válaszolva elfogadja vagy elutasítja az asszociációs kérést. Az asszociáció során

az eszköz és a hálózat között kialakul egy kapcsolat, amely lehetővé teszi az adatátvitelt és a kommunikációt a hálózaton belül. Az asszociáció folyamata során a hálózat azonosítja és hitelesíti az eszközt, valamint hozzárendelhet egy azonosítót. Ez lehetővé teszi az eszköz számára, hogy részt vegyen a hálózati kommunikációban és hozzáférjen a hálózat erőforrásaihoz.

2.10. DHCP

A DHCP (Dynamic Host Configuration Protocol) egy hálózati protokoll, amely nemcsak a vezeték nélküli hálózatokban, hanem általában az IP-alapú hálózatokban is használatos. Ez tájékoztatja a klienst a hálózat felépítéséről és a saját azonosítójáról.

Ez a folyamat egy három lépéses úgy nevezett "kézfogással" zajlik le.

Az első lépésben a kliens egy DHCP Discovery üzenetet szór szét a hálózaton, amelyben kéri a DHCP szervert, hogy válaszoljon neki.

Erre a DHCP szerver egy DHCP Offer üzenettel válaszol, amelyben megadja a kliens számára a hálózat konfigurációját.

A kliens a DHCP Request üzenetében visszaigazolja a DHCP szervernek, hogy elfogadja a konfigurációt, amely erre egy ACK üzenettel válaszol.

2.11. Wi-Fi irányítási protokollok

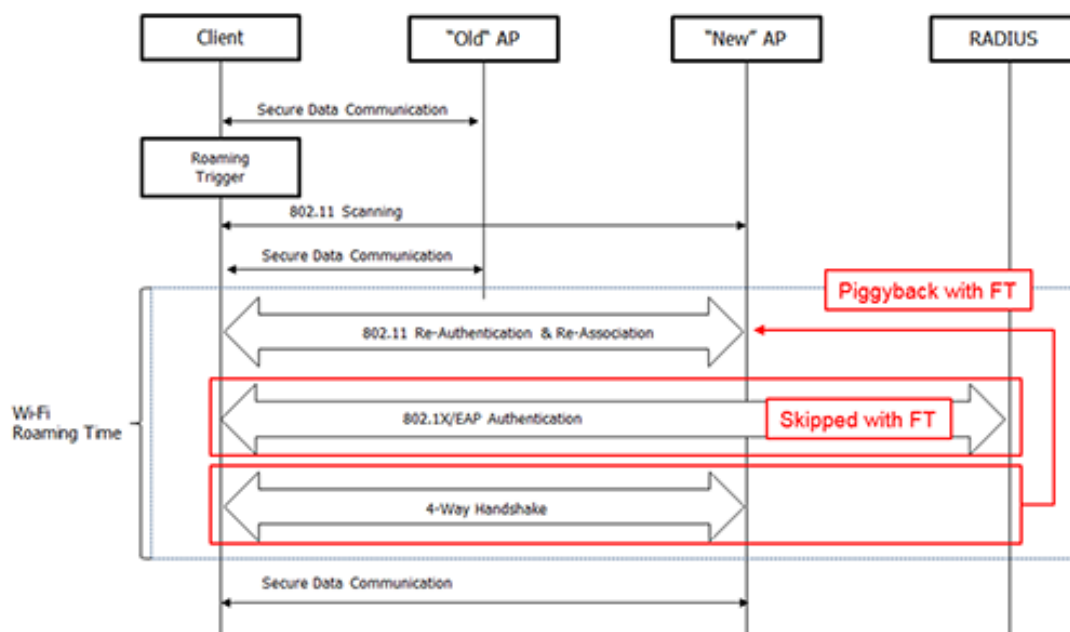
A Wi-Fi hálózatkezelési protokollok a hálózatok működését segítik. A hálózatok működéséhez szükséges az eszközök szabályozása is. Ezt a feladatot látják el a Wi-Fi irányítási protokollok. Nem minden hálózati eszköznek kell mindegyiket támogatnia, de ezek segíthetnek a biztonság és a felhasználói élmény javításában.

A továbbiakban ismertetem a roamingot segítő protokollokat a Samsung Knox "Továbbfejlesztett barangolási funkciók" című cikkének [13] és az Apple "Barangolás Wi-Fi-hálózatokon" című cikkének információ alapján. [15]

2.11.1. 802.11r Fast BSS Transition

A 802.11r Fast BSS Transition (FT) egy Wi-Fi irányítási protokoll, amely lehetővé teszi a zavartalan és gyors átmenetet az állomások számára. Ez gyorsított átmenet lehetővé teszi, hogy az STA megszakítás nélkül folytassa a kommunikációt egy másik hozzáférési ponttal, anélkül, hogy újra kellene hitelesítenie magát.

A Samsung Knox cikk ábráján látható, hogy a 802.11r FT protokoll segítségével több lépés is kihagyható a roaming folyamatából, így gyorsabbá téve azt.



2.11.2. 802.11k Radio Resource Measurement

A 802.11k, más néven Radio Resource Measurement (RRM) szabvány létrehozta a csatornák optimalizált listáját, ezzel segíti a közeli, roamingcélként szolgáló hozzáférési pontok gyors megkeresését. Amikor az aktuális hozzáférési pont jele gyengül, az eszköz további hozzáférési pontokat keres a listából, ezáltal elkerülve a teljes tartomány átvizsgálását.

A szabvány része a Beacon Report is, amely lehetővé teszi az STA számára, hogy lekérje önmaga számára a barangoláshoz ajánlott hozzáférési pontok listáját. Azonban ha ezek az információk helytelenek, akkor az állomás összezavarodhat. A legjobb eredmény elérése érdekében a beacon jelentésben az AP megkaphatja az állomástól a többi AP távolságát.

TODO: network packet capture image .

2.11.3. 802.11v Wireless Network Management

A 802.11v Wireless Network Management (WNM) szabvány célja, hogy lehetővé tegye a hálózati infrastruktúra és az STA-k közötti hatékony információcserét a teljes hálózat teljesítményének optimalizálása érdekében.

A szabvány legfontosabb funkciója a BSS Transition Management (BTM).

Általában az állomások nem tudják, hogy mi történik a hálózatban, és nem tudják, hogy melyik hozzáférési pont a legjobb választás. A BTM egy AP kérheti az állomástól, hogy váltson másik csatornára vagy hozzáférési pontra.

Azonban az állomás nem köteles elfogadni a kérést, és a BTM nem garantálja, hogy az állomás elfogadja a kérést. Az AP az Abridged Bit beállításával jelezheti, hogy erősen ajánlja a váltást.

A Disassociation Imminent Bit beállításával az AP jelezheti, hogy az állomásnak a disassociation timer-ben meghatározott másodpercen belül el kell hagynia a hálózatot.

TODO: BTM packet capture image

2.12. Az OpenWrt hostapd és parancsai

A hostapd egy hozzáférési pontokhoz és hitelesítési kiszolgálókhoz készült felhasználói tartományból elérhető szolgáltatás. [2]

OpenWrt rendszeren az ubus segítségével lehet legkönnyebben kommunikálni a hostapd-val.

Az ubus a projekt neve, amely egy daemonból, egy könyvtárból és egy parancssorból áll. Ezen kívül a uhttpd-mod-ubus modul segítségével a webes felületen keresztül is elérhetővé tehető. [9]

```

1  Usage: ubus [<options>] <command> [arguments...]
2  Options:
3  -s <socket>:   Set the unix domain socket to connect to
4  -t <timeout>:   Set the timeout (in seconds) for a command to
                    complete
5  -S:           Use simplified output (for scripts)
6  -v:           More verbose output
7  -m <type>:     (for monitor): include a specific message type
8                  (can be used more than once)
9  -M <r|t>       (for monitor): only capture received or transmitted
                    traffic
10
11  Commands:
12  - list [<path>]      List objects
13  - call <path> <method> [<message>] Call an object method
14  - subscribe <path> [<path>...] Subscribe to object(s) notifications
15  - listen [<path>...] Listen for events
16  - send <type> [<message>] Send an event
17  - wait_for <object> [<object>...] Wait for multiple objects to
                    appear on ubus
18  - monitor          Monitor ubus traffic

```

2.12.1. A Wi-Fi kliensek listázása

A BSS-hez csatlakozott kliensek listázásához a következő parancsot kell kiadni:

```

1  root@OpenWrt:/ ubus call hostapd.$BSS get_clients
2  {
3    "freq": 5580,
4    "clients": {
5      "d8:f8:83:9d:52:dd": {
6        "auth": true,
7        "assoc": true,
8        "authorized": true,
9        "preauth": false,
10       "wds": false,
11       "wmm": true,
12       "ht": true,
13       "vht": true,
14       "he": true,
15       "wps": false,
16       "mfp": false,
17       "mbo": false,
18       "rrm": [113, 0, 0, 0, 0],
19       "extended_capabilities": [4, 0, 64, 0, 1, 0, 64, 64, 0, 0, 32],

```

```

20     "aid": 3,
21     "signature": "wifi4|probe:0,1,45,191,255,221(0050f2,8),htcap:19
    ef,htagg:17,htmcs:0000ffff,vhtcap:039071f6,vhtrxmc:0000fffa,
    vhttxmc:2000fffa|assoc:0,1,33,36,48,45,127,191,255,221(001735,8)
    ,70,59,221(0050f2,2),htcap:19ef,htagg:17,htmcs:0000ffff,vhtcap
    :039051f6,vhtrxmc:0000fffa,vhttxmc:2000fffa,txpow:1600,extcap
    :0400400001004040000020",
22     "bytes": { "rx": 6035050799, "tx": 1296173698 },
23     "airtime": { "rx": 17207474893, "tx": 15158693919 },
24     "packets": { "rx": 4254865, "tx": 1606104 },
25     "rate": { "rx": 240190000, "tx": 240190000 },
26     "signal": -54,
27     "capabilities": {
28         "vht": {
29             "su_beamformee": true,
30             "mu_beamformee": true,
31             "mcs_map": {
32                 "rx": {
33                     "1ss": 9, "2ss": 9, "3ss": -1, "4ss": -1, "5ss": -1, "6
    ss": -1, "7ss": -1, "8ss": -1
34                 },
35                 "tx": {
36                     "1ss": 9, "2ss": 9, "3ss": -1, "4ss": -1, "5ss": -1, "6
    ss": -1, "7ss": -1, "8ss": -1
37                 }
38             }
39         }
40     }
41 }
42 }
43 }

```

2.12.2. A 802.11v transition request küldése

```

1 root@OpenWrt:/ ubus call hostapd.Mora_Network bss_transition_request
    '{"addr": "50:3D:C6:5B:81:E5", "reassoc_delay": 10, "
    disassociation_timer": 5, "validity_period": 30, "abridged": 1}' &&
    sleep 5 && ubus call hostapd.Mora_Network del_client '{"addr" :
    "50:3D:C6:5B:81:E5", "deauth": true, "ban_time": 10000 }'

1 root@OpenWrt:/ ubus call hostapd.Mora_Network bss_mgmt_enable '{ "
    neighbor_report": true, "beacon_report": true, "link_measurements":
    true, "bss_transition": true }'

1 root@OpenWrt:/ ubus call hostapd.Mora_Network link_measurement_req '{
    "addr": "FE:ED:AD:BE:EE" }'

```


3. Terheléskiegyenlítés

4. Eredmények

4.1. Mérési környezet és metrikák

4.1.1. Prometheus

4.1.2. Grafana

4.1.3. A lokális tesztelés környezete

4.1.4. A kollégium hálózatán végzett mérések

Függelék

A program forráskódja

A függelékbe kerülhetnek a hosszú táblázatok, vagy mondjuk egy programlista:

Nyilatkozat

Alulírott Südi Tamás, programtervező informatikus szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet XY Tanszékén készítettem, XY diploma megszerzése érdekében. Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem Diploma-munka Repozitóriumában tárolja.

Szeged, 2023. május 19.

.....
aláírás

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani **X. Y-nak** ezért és ezért ...

Hivatkozások

- [1] *Configuring WiFi6 as AP in linux with hostapd*. <https://community.intel.com/t5/Wireless/Configuring-WiFi6-as-AP-in-linux-with-hostapd/m-p/1223560>.
- [2] *hostapd*. <https://w1.fi/hostapd/>.
- [3] *How to get VirtualBox 6.0 and WSL working at the same time*. <https://stackoverflow.com/questions/58031941/how-to-get-virtualbox-6-0-and-wsl-working-at-the-same-time>.
- [4] *KVM*. https://web.archive.org/web/20230324055810/https://www.linux-kvm.org/page/Main_Page.
- [5] *OpenWrt About*. <https://web.archive.org/web/20230315035827/https://openwrt.org/about>.
- [6] *OpenWrt Asus TUF-AX4200 support*. <https://web.archive.org/web/20230519011324/https://forum.openwrt.org/t/asus-tuf-ax4200-support/155738>.

- [7] *OpenWrt FAQ*. <https://web.archive.org/web/20221123091403/https://openwrt.org/faq/general>.
- [8] *OpenWrt Home*. <https://web.archive.org/web/20230315035827/https://openwrt.org/start>.
- [9] *OpenWrt hostapd*. <https://openwrt.org/docs/techref/ubus>.
- [10] *OpenWrt installation*. <https://web.archive.org/web/20220609112758/https://openwrt.org/docs/guide-user/installation/start>.
- [11] *OpenWrt Stock Firmware*. <https://web.archive.org/web/20230316170518/https://openwrt.org/docs/guide-user/installation/openwrt-as-stock-firmware>.
- [12] *PCI passthrough*. https://web.archive.org/web/20230327035420/https://wiki.archlinux.org/title/PCI_passthrough_via_OVMF.
- [13] *Samsung Know Enhanced roaming algorithm*. <https://web.archive.org/web/20230519013635/https://docs.samsungknox.com/admin/knox-platform-for-enterprise/kbas/kba-115013403768.htm>.
- [14] *What's new in Hyper-V on Windows*. <https://web.archive.org/web/20230425004856/https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/system-requirements-for-hyper-v-on-windows>.
- [15] *Wi-Fi network roaming with 802.11k, 802.11r, and 802.11v on iOS, iPadOS, and macOS*. <https://web.archive.org/web/20230503143419/https://support.apple.com/en-us/HT202628>.
- [16] *Windows Subsystem for Linux*. <https://web.archive.org/web/20230323193209/https://learn.microsoft.com/en-us/windows/wsl/about>.
- [17] *WLAN csatornák*. https://en.wikipedia.org/wiki/List_of_WLAN_channels.
- [18] *WSL2 USBIP*. <https://web.archive.org/web/20230310122219/https://learn.microsoft.com/en-us/windows/wsl/connect-usb>.