Kommunikációs hálózatok

# Helyi hálózatok mérés

## Segédlet

Németh Krisztián

BME TMIT

2024. 03. 03.

v1.0

*Kezdje e mérést a „Bevezető a méréshez” c. dokumentum áttanulmányozásával!*

*Ebben a Segédlet c. dokumentumban foglaltuk össze a méréshez szükséges elméleti ismereteket és néhány praktikus tudnivalót. Nem feltétlen szükséges a mérés megkezdése előtt végigolvasni, de mérés közben érdemes időről időre megállni, és fellapozni e dokumentumban a releváns részeket.*

# Tartalomjegyzék

# 1 Elméleti ismeretek

## 1.1 Ethernet keretek

Az Ethernet egy helyi hálózati technológiai család összefoglaló neve. Azért hívjuk „helyi” hálózatnak, mert mérete jellemzően nem nagyobb, mint egy lakás, kisvállalat, vagy nagyobb vállalatnak egy részlege. Az egyetemen sokszor még az egy tanszékhez tartozó gépeket is több Ethernet hálózatra bontva kötik össze. TCP/IP hálózatokban logikailag az IP réteg alatt helyezkedik el.

Az Ethernetet az IEEE 802.3 szabványcsoport definiálja. Az adatkapcsolati rétegben csomagkapcsolt formában zajlik az átvitel. Ennek alapegysége az Ethernet keret. A keretet egy bevezető (preambulum, angolul preamble) és keret kezdeti határoló (start of frame delimiter, SFD) előzi meg, amelyek a szinkronizáció céljára szolgálnak. A keret különböző fejlécmezőkkel kezdődik, melyeket az 1. ábra mutat be (1 oktet = 1 bájt = 8 bit). A fejléc után a hasznos adattartalom (payload) következik, amely rendszerint további rétegek (L3, pl. IP) protokoll adategységeit ágyazza be. Ezt követően egy 32 bites ellenőrző összeg (cyclic redundancy check, CRC) következik, amely a keret integritásának ellenőrzését szolgálja. Legvégül az ún. keretköz (interframe/interpacket gap, IFG/IPG) következik.

| Réteg | Preambulum | Kezdeti határoló (SFD) | MAC cél cím | MAC forrás cím | 802.1q címke (opcionális) | EtherType vagy hossz | Hasznos adat | CRC | IFG/IPG |
|---|---|---|---|---|---|---|---|---|---|
| | 7 oktet | 1 oktet | 6 oktet | 6 oktet | (4 oktet) | 2 oktet | 46-1500 oktet | 4 oktet | 12 oktet |
| L2: Ethernet keret | | | ← 64–1522 oktet → | | | | | | |
| L1: Ethernet csomag és IPG | ← 72–1530 oktet → | | | | | | | | ← 12 oktet → |

*1. ábra. 802.3 Ethernet keretszerkezet (forrás: wikipedia)*

### 1.1.1 MAC címek

MAC (Medium Access Control, közeghozzáférés-vezérlés) címeknek nevezik az Ethernetben használatos azonosítóit az egyes végpontoknak (azaz tipikusan azok hálózati kártyáinak). A MAC címek 6 oktet (48 bit) hosszúságúak.

Bár ez a 6 oktet együtt képez egy oszthatatlan MAC címet, mégis logikailag több részre tagolható. Az első három oktet ún. szervezetileg egyedi azonosító (Organisationally Unique Identifier, OUI), a másik három oktet pedig hálózati csatoló (network interface controller, NIC) specifikus egyedi sorszám. A cím bizonyos bitpozíciói speciális jelentéssel bírnak:

- Az OUI első oktetjének 0. helyiértékű bitje:
    - 0: Unicast cím
    - 1: multicast cím
- Az OUI első oktetjének 1. helyiértékű bitje:
    - 0: globálisan egyedi (OUI által rögzített)
    - 1: lokálisan nyilvántartott

A csupa 1-es (ff:ff:ff:ff:ff:ff) cím ún. üzenetszórási (broadcast) cím, amelyet az üzenetszórási tartományban levő minden csomópont megkap. Az unicast címre szóló csomagok akkor veszi át a csatoló, ha az ő fizikai címére szól. Az egyéb célcímre küldött keretek elkapásához ún. promiszk (promiscuous) módban kell lennie a csatolónak. Ez utóbbi azt jelenti, hogy a csatoló azokat a kereteket is feldolgozza, amelyek nem neki szólnak.

Mivel a legtöbb hardvergyártó regisztrált OUI-val rendelkezik (akár többel is), számos adatbázis[1] létezik, amelyből megtudható egy MAC címről, hogy maga az eszköz mely gyártótól származik.

### 1.1.2 802.1q címke

A 802.1q címke, más néven VLAN címke (angolul tag) a virtuális helyi hálózatok kialakításához nélkülözhetetlen. A VLAN-okról a következő fejezetben szólunk részletesen.

### 1.1.3 EtherType

Ez a két oktetes mező a keretbe ágyazott protokoll adategység típusát határozza meg. A mérések során ún. Ethernet II más néven DIX (DEC+Intel+Xerox) keretekkel fogunk találkozni. Jellemző EtherType értékek:

- 0x0800: IPv4 adatcsomag
- 0x0806: ARP keret
- 0x86dd: IPv6 adatcsomag

A PC-n az erre szolgáló forgalomelkapó alkalmazásokkal (pl. Wireshark, tcpdump stb.) elmenthetők ezek a keretek, de csak a MAC célcímtől a hasznos adat végéig tartó mezőket fogjuk látni, a preamble-t, SFD-t és CRC-t már nem!

## 1.2 ARP

TCP/IP hálózaton IP csomagokat küldünk, melynek fejlécében többek között megadjuk a forrás- és cél-IP-címeket. Például, ha le akarok tölteni egy oldalt a helyi hálózaton lévő webszerverről, akkor a DNS megmondja a gépemnek a hozzá tartozó IP címet, és a böngésző kiküld oda egy http kérést egy IP csomagban.

Igen ám, de a számítógépemben (telefonomban) Ethernet (és/vagy WiFi) kártya van, ami Ethernet (WLAN) kereteket tud kiküldeni és fogadni. Az addig rendben is van, hogy egy ilyen keretbe beágyazzuk az IP csomagot, sőt az Ethernet (WLAN) keret forráscím mezője is adott (a hálózati kártyám MAC címe), de mi legyen a célcím? Mármint a cél Ethernet (WiFi) cím?

Nyilván a webszerver hálózati kártyájának MAC címe. Rendben, de ezt honnan tudjuk meg? Erre van az ARP protokoll, és az oprendszerben az azt megvalósító funkciók. Először is lehet, hogy már kommunikáltunk nemrég ezzel az IP címmel, és akkor még emlékezhetünk a MAC címére. Van tehát a gépünkben egy táblázat a nemrég

[1] https://www.wireshark.org/tools/oui-lookup.html

használt IP címek-MAC címek összerendeléséről. Ezt Windows alatt a parancssori `arp -a`, Linux alatt az `arp` paranccsal meg is nézhetjük.

No de mi van akkor, ha a keresett IP cím nincs benne e táblázatban? Erre a problémára készítették az Address Resolution Protocol-t (ARP), magyarul a címfeloldó protokollt. A protokoll nem új, 1982-ben specifikálták a RFC 826 számú dokumentumban, bár azóta többször kibővítették. Az ARP elvben különböző felső, illetve alsó rétegbeli protokollal is működik, mi azonban csak IP (felső) és Ethernet/WLAN (alsó) protokollokkal használjuk e mérésen, sőt igazából csak IP-vel és Ethernettel.

Az ARP működése első közelítésben elég egyszerű: a gépünk kiküld az ff:ff:ff:ff:ff:ff broadcast MAC címre egy kérést, amiben leírja, hogy keresi az adott IP címhez tartozó MAC címet. Ezt minden helyi hálózaton lévő eszköz feldolgozza. Aki magra ismer, az válaszol, immár csak a kérőnek, megadva a saját MAC címét. Ezután már mehet az Ethernet keretekbe ágyazott IP csomagokban a kommunikáció. No és persze a megszerzett MAC cím – IP cím összerendelést a gép oprendszere felveszi a korábban említett ARP táblázatba, hogy ne kelljen minden egyes csomag előtt újra körbekérdezni.

## 1.3 VLAN-ok

Egy Ethernet hálózaton belül a broadcast üzeneteket minden végpont megkapja. Ilyen broadcast üzenetek a címfeloldó protokoll (ARP) üzenetei is. Klasszikusan az egymáshoz fizikailag közel lévő számítógépeket ugyanabba az Ethernet kapcsolóba (switchbe) dugjuk. Ha egy vállalatnál az egyik osztály (pl. pénzügy) gépei egymáshoz közel vannak és egy másik osztályhoz (pl. HR) tartozó gépek egy másik emeleten találhatóak, akkor ezeknek könnyű egy-egy külön Ethernet hálózatot készíteni. Ez több szempontból is előnyös:

- menedzselhetőség: minden osztály rendszergazdája a saját hálózatáért felelős[2]
- skálázhatóság: a broadcast üzenetek osztályon belül maradnak, nem árasztják el az egész vállalatot
- biztonság: az esetleges LAN-on belüli visszaélések lehetősége is korlátozott

Ha azonban egy másik példában egy emeleten keverednek a két (vagy több) osztály számítógépei, akkor is célszerű lenne az elkülönítés. Két Ethernet kapcsoló felesleges beruházás lehet, ha egynek a befogadóképessége is elegendő, eggyel viszont az elkülönítés nem valósul meg.

E probléma feloldására született a virtuális helyi hálózatok (Virtual LAN, VLAN) koncepciója. Ilyet használva csak egy Ethernet kapcsoló kell, és beállíthatjuk például, hogy annak melyik portja melyik VLAN-hoz tartozik. A VLAN-okat az Ethernet kapcsoló egymástól elkülöníti, köztük nem lehetséges Ethernet üzenetszórás, vagy bármi más Ethernet szintű (2. rétegbeli) kommunikáció.

Ezen túlmenően arra is lehetőség lesz, hogy összekössük mondjuk két emelet Ethernet kapcsolóit úgy, hogy mindkettőhöz vegyesen csatlakoznak a két osztály gépei, és ezt az egészet logikailag két VLAN-ra bontsuk. Ehhez az is kell, hogy a két kapcsoló közötti kommunikációban az Ethernet kereteket megjelöljük, hogy meyik VLAN-hoz tartoznak. Erre való az ún. VLAN címke (a fejlécben a 802.1q címke).

A címkék használatának módját az IEEE 802.1q szabvány részletezi, ezt néhol .1q-ként rövidítik. A címke felhelyezése (*tagging*) azt jelenti, hogy egy interfészen belépő/kilépő csomagra az adott azonosítójú címke kerül fel, vagyis az Ethernet fejlécbe beszúrásra kerül az opcionális 4 oktet. Az első kettő a protokoll azonosító (tag protocol identifier, TPID) 0x8100 értékkel, a második kettő pedig címkeinformáció (tag control information, TCI): 3 bitnyi prioritásazonosító (priority code point, PCP), 1 bites DEI (drop eligible indicator) és 12 bitnyi VLAN azonosító (VID). Egy fizikai port lehet címke nélküli (untagged) és címkézett (tagged) tagja az adott VLAN-nak.

---

[2] Kicsit sántít a példa, nyilván nincs minden osztálynak külön rendszergazdája. Azonban pl. a BME VIK-en a tanszékeknek jellemzően vannak saját rendszergazdáik, talán ez jobb példa lett volna.

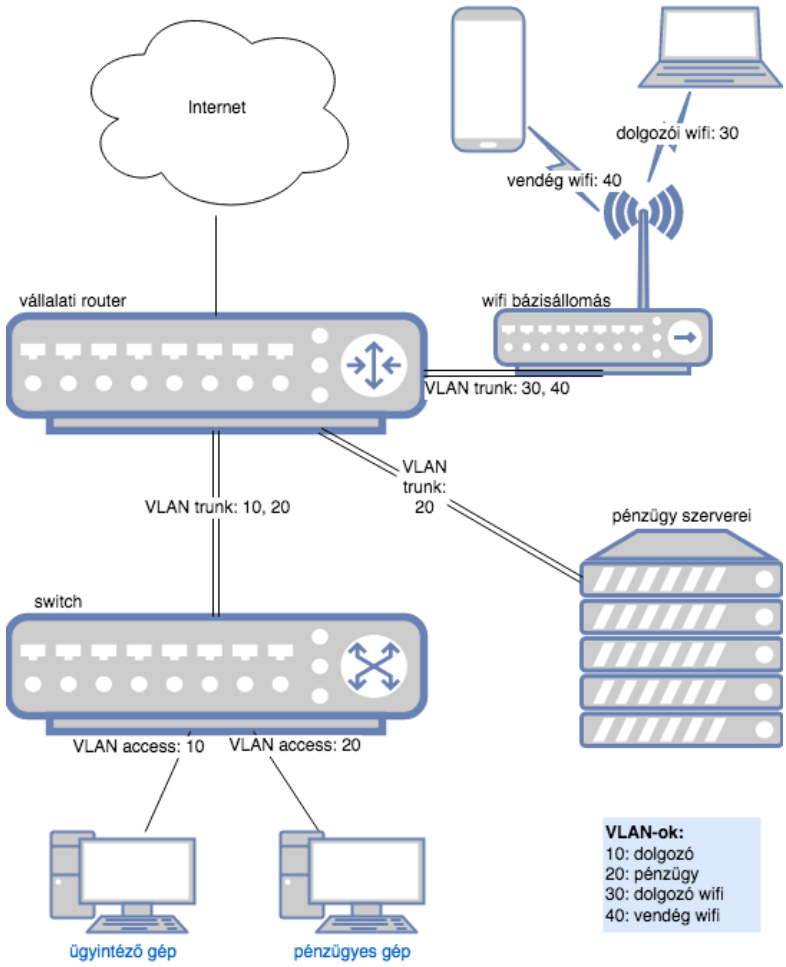
Untagged módban az interfész bár része a VLAN üzenetszórási tartományának, nem kerül címke a rajta keresztül forgalmazott keretekre. Ehhez fontos megjegyezni, hogy egy port csak egyetlen VLAN-nak lehet untagged üzemmódú tagja, a többi VLAN tagsága már címkézett módú kell, hogy legyen.

A *trunk* kapcsolatokon jellemzően ilyen címkézett keretek haladnak, kivéve azokat, amelyeken ilyen címke nincs: ezek az ún. natív VLAN-hoz tartoznak. Mielőtt a csomag eléri azt az untagged módban működő portot, a címkét el kell távolítani. Ez a művelet az *untagging*.

Tehát minden VLAN önálló üzenetszórási tartományként (broadcast domain) viselkedik. Minden ilyen VLAN-nak azonosítót (címkét, tag-et) adunk a 0...4095 tartományból, amelynek praktikusan telephelyi szinten egyediek. Az infrastruktúra L2 és L3 eszközein ezt a logikai topológiát implementáljuk:

- a (tipikusan) gerinc kapcsolatokon, amelyeken több VLAN forgalmát is át kell vezetni, mindig címkézett kereteket forgalmazunk (abban az esetben is, ha jelen állapotban csak egyetlen VLAN forgalmát kell továbbítania, mert így a későbbiekben egyszerűen rendelhetünk további VLAN-okat a gerinckapcsolathoz)
- azokon a hozzáférési portokon, amelyeken végponti eszközök kapcsolóadnak egy VLAN-hoz, hozzáférési címke nélkül továbbítjuk az Ethernet kereteket

A hozzáférési portokról érkező gerinc portra továbbítandó Ethernet kereteket taggeljük (felcímkézzük), a kilépőket untaggeljük (leszedjük a címkét). A gerinckapcsolatokon (uplink portokon) ezen felül praktikusan VLAN szűrést is végzünk: eldobjuk azokat a kereteket, amelyek nem tartoznak a gerinckapcsolathoz rendelt VLAN-ok egyikéhez sem, a többieket a szokásos módon továbbítjuk.




*2. ábra. Tipikus kisvállalati hálózat*

Ahogy a 2. ábra mutatja, az egyes VLAN-ok tetszőlegesen alakíthatók ki a 802.1q-t támogató hálózati eszközön keresztül. Így pl. a 20-as pénzügyes VLAN hozzáférhető a switch adott hozzáférési portján keresztül a pénzügyes munkaállomásról, de e switch másik, 10-es hozzáférési portján található ügyintézői gép nem kommunikálhat az adott alhálózaton.

Fontos megjegyezni, hogy a VLAN-ok önálló üzenetszórási tartományok, melyek között csak IP útválasztással lehetséges a kommunikáció, ezt nevezzük inter-VLAN routing-nak. Amennyiben egy Ethernet kapcsoló L3-as üzemmódot is támogat, úgy ez a funkció aktiválható rajta. Így bár MAC-cím szerint nem tudnak egymással kommunikálni az eltérő VLAN-ban található csomópontok, az inter-VLAN routing segítségével a hálózati rétegben, IP-címzéssel már lehetséges kettejük között az üzenetváltás az adott eszközön keresztül.

## 1.4 Tűzfal

A tűzfal tipikus feladata bizonyos rétegen áthaladó hálózati forgalom nyomon követése és egy meghatározott szabályrendszer mentén a forgalom korlátozása vagy átengedése. A hálózati eszközökben a leggyakoribb a csomagszintű (jellemzően a 4. rétegig) feldolgozást végző tűzfal. Kiegészítő funkciók is előfordulnak, ilyen pl. a címfordítás (network address translation, NAT) és portfordítás (port address translation, PAT). Léteznek alkalmazás szintű tűzfalak is, de ezek alkalmazása a kommunikációs útvonalakon nem triviális a manapság egyre gyakoribb titkosítás miatt, komoly heurisztikára van szükség a működésükhöz.

A tipikus csomagszintű tűzfalnak 3 alapvető szabályrendszere, szakszóval lánca van: forward, input, output. Az első a rajta áthaladó, a második a konkrét eszközt megcélzó, a harmadik az eszközből kiinduló forgalomra alkalmazandó. Minden lánchoz rendelhetünk egy irányelvet: megengedő vagy tiltó, attól függően, hogy mi történjen azokkal a csomagokkal, amelyekre a lánc egyetlen szabálya sem teljesül.

A szabályainkat azután az adott láncokhoz kapcsolódóan sokféle módon definiálhatjuk, pl. cél-/forráscím, protokoll, portszám, interfész, kapcsolat állapot, és sok egyéb, hálózati protokollokhoz kapcsolódó módokon. A szabályban meg kell adnunk egy műveletet is, amely az illeszkedéskor bekövetkezik, pl. elfogadás (accept), eldobás (drop), stb.

A tűzfal működésekor a csomagtovábbításkor a feldolgozandó csomagok az érvényben levő szabályláncok szerint lesznek megítélve: ha egy szabály illeszkedik, a kapcsolt művelet végrehajtódik. Ha nem és tovább folytatható (nem mondta a szabály, hogy stop) az illesztés folytatódik a következő szabállyal. A szabályok után pedig az alapelv lép érvénybe.

Például beállíthatjuk, hogy egy tartományba engedjen be a tűzfal minden olyat csomagot, ami egy adott másik IP címtartományból érkezik (forráscím), vagy adott portra igyekszik (pl. http: 80-as port), azonban minden más csomagot dobjon el.

## 1.5 Mikrotik RouterOS

A RouteOS a Mikrotik saját fejlesztésű, Linux alapú, de zárt forráskódú firmware-e, elsősorban a saját RouterBoard alapú platformján (switchek, routerek, bázisállomások) történő alkalmazásra. Előnye az egységesség (minden hardverre ugyanolyan firmware), ill. a hosszú támogatás, azaz hogy a régi hardverekre is felrakható a korszerű firmware. Az eszköz megvásárlásakor kapjuk a firmware-t is, de bizonyos funkciók licencdíjasok, pl. az x86 PC-re telepítés is.

A RouterOS konfigurálása legkényelmesebben a WinBox ingyenes grafikus alkalmazással történhet (a „gyakorlati útmutató” vezet be a használatába), de webböngészőből (WebFig) és gyakorlott üzemeltetők parancssorból (ssh-n vagy a GUI-kon keresztül) is végezhetik.

Vállalati jellegű, több AP-ból álló hálózat kialakításához a központi menedzsmentet segítő *CAPsMAN* (Controlled Access Point system Manager) funkció segíti: az AP-k CAP-pá (controlled access point) minősíthetők, onnantól csak AP feladatokat (hozzáférés vezérlés, felhasználói azonosítás) látnak el és ezek a beállítások (pl. RADIUS szerver, WPA/WPA2 kulcsok) központilag konfigurálhatók, de akár központilag is frissíthetők.

### 1.5.1 hAP ac$^2$

A méréshez használt router 1+4 vezetékes Gigabit Ethernet portjából az #1-es (Internet/POE In) a WAN kapcsolathoz van rendelve (de akár LAN porttá is konfigurálható). A vezetékes portokat ether1-től ether5-ig azonosítja a RouterOS.

A routerünk rendelkezik két rádiós interfésszel is: az egyik a 802.11b/g/n szabvány kiszolgálásához, *wlan1* néven. A 802.11ac szabványt pedig a *wlan2* nevű interfész működteti. Az interfészek mindegyike egyenként is letiltható.

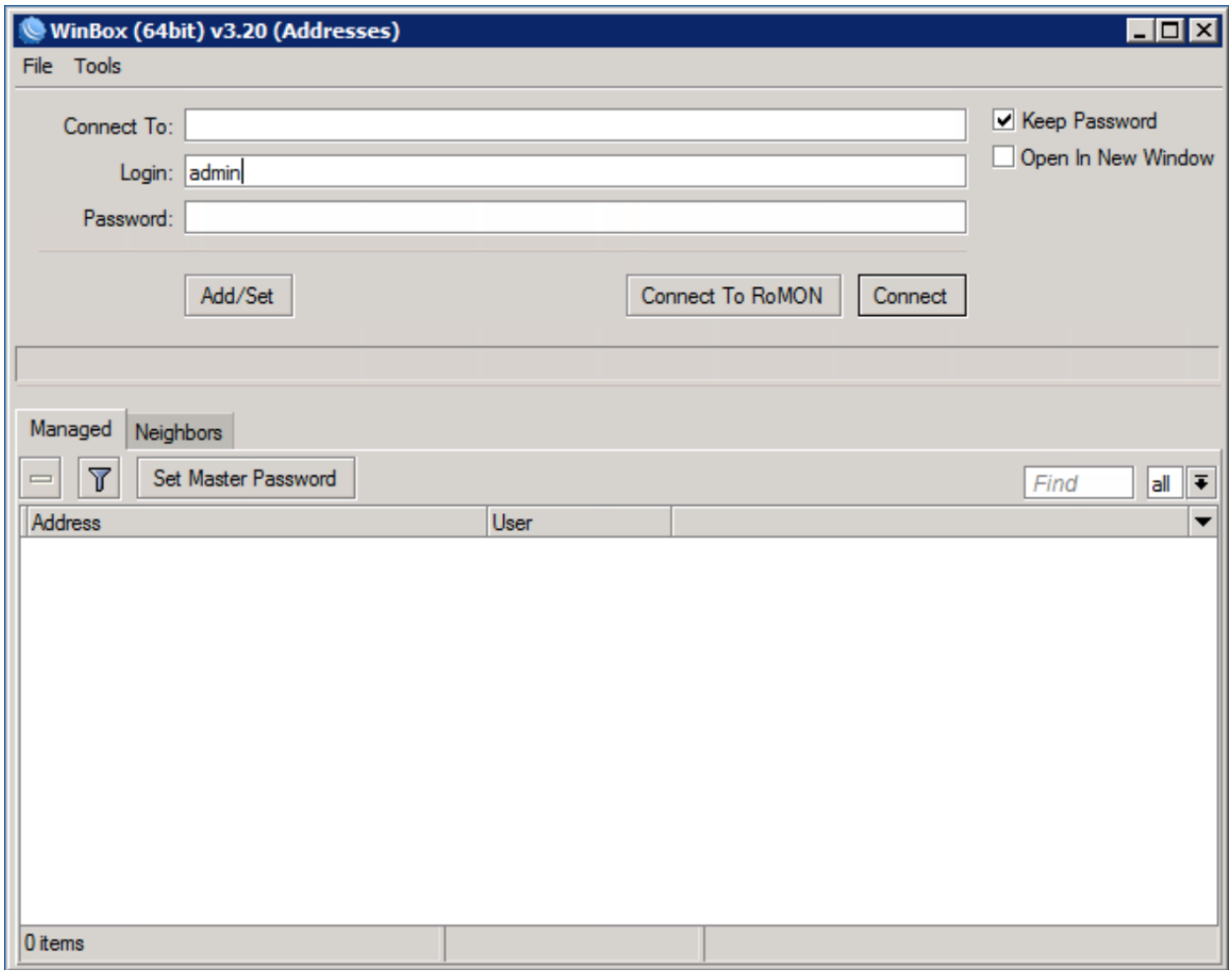
# 2 Gyakorlati tudnivalók

## 2.1 A WinBox használata

### 2.1.1 Csatlakozás, főmenü

A program indításához a következő ikont keressük: 

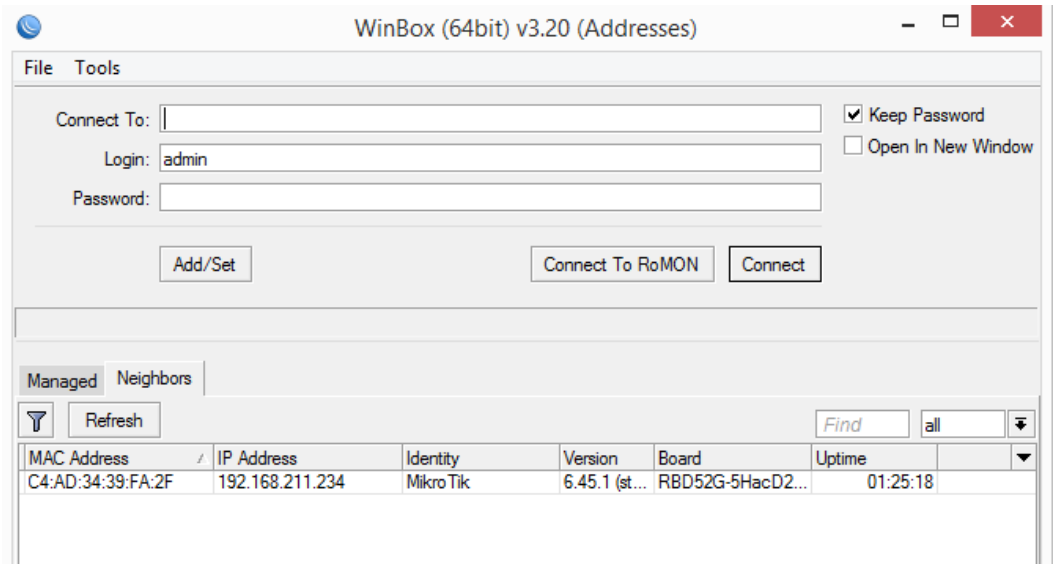

Indítás után az alábbi ablakot láthatjuk:




A *Managed* fülön a *Connect To* mezőbe a kezelni kívánt eszköz IP-címét vagy MAC-címét írjuk, valamint az adminisztrátori login nevet (alapértelmezés szerint *admin*) és a hozzá tartozó jelszót (alapértelmezetten üres). A WinBox megjegyzi az utolsó sikeres csatlakozott eszköz címét, ill. továbbiakat is felvehetünk itt.

A fenti módon (Managed fül) tehát mi adjuk meg a menedzselni kívánt eszköz címét. Lehetőség van arra is, hogy az automatikusan felismert környező eszközök közül válasszunk, erre van a Neighbors fül.

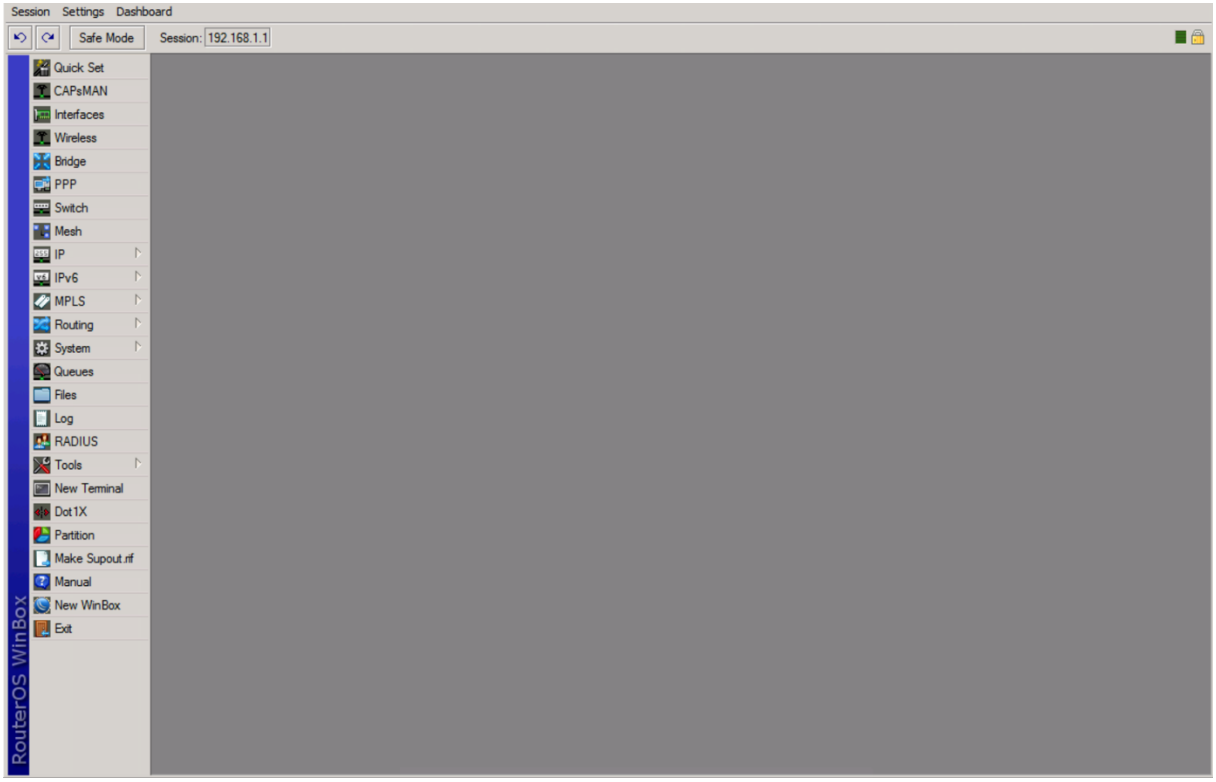
A *Neighbors* fülön tehát a felderített RouterOS-t futtató eszközök listáját találhatjuk:




A beazonosítást segítheti az *identity* mező (ha már adtunk egyedi nevet) ill. a MAC- és IP-cím. Figyeljünk arra, hogy mindig ahhoz az eszközhöz csatlakozzunk, amelyet valóban konfigurálni szeretnénk!

A csatlakozáshoz a *Connect* gombra kattintsunk! (A RoMON – router management overlay network – egy központosított menedzsment réteg, a mérések során nem ezzel dolgozunk, így a *Connect to RoMON* gomb helyett a *Connect*-et használjuk!)

Csatlakozás után ehhez hasonló kép fogad minket:



Az elvégzett konfigurációs beállítások ugyan azonnali érvényűek, de hasznos lehet az ún. Safe Mode aktiválása. Ha bekapcsoljuk, a WinBox-ból való kilépéskor megerősítést fog kérni a véglegesítésre. Ha ez nem történik meg, akkor visszavonja a session-ben elvégzett összes beállítást. Ezáltal megelőzhetjük, hogy kizárjuk magunkat, ill. hogy emiatt reset-elni kelljen az eszközt.

A menüsor alatti sáv dashboard kiegészíthető pár érdekes mérőeszközzel, amelyek a Dashboard menün keresztül érhetők el: dátum, idő, CPU terhelés, memóriafoglaltság és uptime (bekapcsolás óta eltelt idő).

A főablak bal oldalsávja a főmenü, legfontosabb pontjai:

- Quick Set: gyorsbeállító "varázsló", a leggyakoribb funkciók gyors beállításához (pl. tipikus otthon wifi router beállításhoz)
- CAPsMAN: központosított Wi-Fi hálózat kiépítéséhez szükséges funkciók
- Interfaces: interfész-ek, VLAN-ok
- Wireless: rádiók, vezetéknélküli profilok
- Bridge: hálózati híd kialakítása
- PPP: pont-pont kapcsolatok kialakítása (pl. betárcsázós, PPPoE)
- Switch: beépített hálózati kapcsoló konfigurálása
- Mesh: vezetéknélküli eszközök hálózata (decentralizált)
- IP, IPv6: IP beállítások
- Routing: útvonalválasztás
- System: az eszköz adminisztrációja (pl. pontos idő, licenszek, beállítások mentése, stb.)
- Dot1X: végponti autentikációs beállítások
- Log: rendszerüzenetek naplója
- Tools: hasznos segédeszközök (ping, forgalomgenerátor, stb.)
- New Terminal: parancssori ablak (pl. scripteléshez)

Azon főmenüpontok, amelyek mellett jobbra nyíl található, második szinttel is rendelkeznek, a levélelemek egy-egy ablakot nyitnak meg. A WinBox többablakos felület, ami rendkívül kényelmessé teszi az amúgy borzasztó sok funkció áttekintését. Az ablakok rendelkezhetnek fülekkel, ill. egy eszközsorral, ami tipikusan így néz ki:

- +/- gomb: hozzáadás/törlés (kontextustól függően), némely funkciónál a + lenyíló menüt is takar (ha lefele mutató nyíl található a jel mellett)
- pipa/kereszt: kijelölt elemek engedélyezése/letiltása. Figyelem, a használata azonnali érvényű!
- sárga cetli (komment): megjegyzés fűzése az elemhez. A listában általában az elem felett fog megjelenni, hacsak az "Inline Comments"-et nem engedélyezzük a Settings menüben.
- szűrő: ha a kontextusban értelmezhető, a listában bizonyos elemekre lehet vele keresni

### 2.1.2 Interfaces menü

A legfontosabb fülek:

- *Interface* fül: az egyes interfészek listája, beleértve a rádiósat, virtuálisat, VLAN-t, bridge-et stb. Itt le is tudjuk őket tiltani (ld. pipa/kereszt toolbar gomb), illetve a legfontosabb statisztikák is megjeleníthetők
- *Interface List* fül: az interfészeket lehet csoportosítani, hogy akár együtt is hivatkozhatók legyenek (pl. tűzfalban stb.), alapból a LAN és WAN csoportok vannak itt
- *Ethernet* fül: a vezetékes ethernet csatolók
- *VLAN* fül: definiált VLAN-ok, új hozzáadása esetén meg kell adnunk legalább a VLAN azonosítót és a szülő interfész nevét

### 2.1.3 Wireless menü

E mérésben nem használunk WiFi-t, így erre most nem lesz szükség. Egy korábbi méréshez kellett, ezért ha már megvolt, benne hagytuk e doksiban a Wireless menü legfontosabb füleinek alábbi leírását, de most nyugodtan átugorhatják.

- *WiFi Interfaces* fül: fizikai, virtuális (további Wi-Fi hálózatok létrehozásához), WDS (lefedettség kiterjesztéséhez), bizonyos mezők megjelenítéséhez az *Advanced Mode* bekacsolása szükséges a jobboldali gombsoron
    - *SSID* mező: itt tudjuk megadni, mi legyen a létrejövő hálózat neve
    - *Security Profile*: az azonos fülön létrehozható profil hozzárendelése (pl. az autentikáció paramétereinek megadásához)
    - *WPS Mode*: Wi-Fi Protected Setup (gombnyomásra érvényesíthető automatikus biztonsági beállítások), amelyet nem fogunk használni, hiszen mi gondoskodunk a paraméterezésről
    - *Frequency Mode*: *regulatory-domain*: a helyi szabályzásnak megfelelő frekvencia használat; *manual-txpower*: mint az előbbi, csak teljesítménykorlát nélkül; *superchannel*: teszteléshez, minden, a rádió által támogatott frekvencia engedélyezésével
    - *Country*: az ország kiválasztása, amiből az eszköz tudja, milyen frekvenciákat és maximális teljesítményt használhat
    - *Antenna Gain*: nyereség dBi-ben, a maximális kimenő teljesítmény kiszámításához (egész érték; minimum 3, router antennái kb. 2.5 dBi nyereségűek)
    - *VLAN Mode*: az alapértelmezés a *no tag*, vagyis nem tageljük a WiFi forgalmat ezen a hálózaton, ha viszont tagelni szeretnénk, akkor állítsuk *use tag*-re és alább adjuk meg a VLAN azonosítót
    - *VLAN ID*: VLAN azonosító (alapértelmezés szerint 1)
    - Hide SSID: pipáljuk be, ha nem kívánjuk hirdetni a hálózatot

Egyéb részletes beállítások lehetségesek a további füleken.

- *Registration* fül: itt láthatók az asszociált (felcsatlakozott) felhasználók, megjelenítve a fizikai címüket és egyéb átviteli jellemzőiket, a kapcsolatot itt meg is lehet szakítani
- *Connect List*: ha az eszközzel más vezetéknélküli hálózathoz szeretnénk csatlakozni, itt lehet megadni a kapcsolatot
- *Security Profiles*: egy-egy Wi-Fi hálózat kapcsolódási jellemzőit állíthatjuk be itt: biztonsági protokollok, kulcsok, autentikációs metódus, stb. Először itt kell definiálni őket, majd fel lehet használni a *WiFi Interfaces*-ben (AP üzemmódhoz) vagy a *Connect List*-nél a kliens módhoz.

A kimenő teljesítményt közvetlenül nem tudjuk szabályozni pusztán a *WiFi Interfaces* → *Tx Power* fülön, mert a firmware igyekszik az adóteljesítményt, ill. a maximális kimenő teljesítményt (EIRP) a helyi szabályozáshoz igazítani. Hogy hatásosan le tudjuk csökkenteni a tényleges kisugárzott teljesítményt, a *WiFi Interfaces* → *Wireless* → *Antenna Gain* mezőt emeljük fel 30 dBi-re.

### 2.1.4 Bridge menü

- *Bridge* fül: hídkapcsolatok listája
- *Ports*: az interfész-híd összerendelések
- *VLANs*: itt az Interfaces menüben létrehozott VLAN-okat rendelhetjük bridge-ekhez, valamint taggelési (címkézési), untaggelési (címke levételi) szabályokat adhatunk meg. A már létrehozottaknál láthatjuk az aktuális taggelés, untaggelési mechanizmust is

…

### 2.1.5 IP menü

- *ARP*: ARP táblázat: megnézhetjük, milyen MAC-IP összerendeléseket ismer az eszköz. Indokolt esetben mi is vehetünk fel új bejegyzéseket
- *Addresses*: IP-címek listája: az interfészeknek – beleértve a VLAN-okat is – IP-címeket oszthatunk, amelyeket átjáróként (vagy menedzsment címként) használhatunk
- *DHCP Client*: ha valamely interfészen IP-címet akarunk igényelni, itt adhatjuk meg. Az alapkonfigurációnak rendszerint része, hogy a WAN kapcsolatul szolgáló interfészre DHCP-n kér címet.
- *DHCP Server*: ha valamely VLAN-on (alapkonfiguráció szerint a *bridge*-en) IP-címet szeretnénk osztani, itt tudunk DHCP szervert definiálni:
    - *DHCP* fül: szerver/interfész lista
    - *Networks* fül: a DHCP szervernek ismerie kell az adott hálózat címét, maszkját és átjáróját (tipikusan az *Addresses*-ben megadott cím), itt tuduk meadni
    - *Leases*: itt listázhatjuk, milyen MAC-címre milyen IP-ket osztogattunk, ill. azok meddig érvényesek. Vehetünk fel statikus bejegyzéseket és érvényteleníthetünk is.
    - *Options/Option Sets*: ha speciális DHCP paramétereket, pl. boot szerver cím, stb. szeretnénk hirdetni, akkor itt vehetjük fel. A DNS szerver nevét, átjárót nem kell külön felvenni itt.
- *DNS*: az eszköz DNS gyorsítótárának beállításai
- *Firewall*: tűzfal beállítása
    - *Filter Rules*: szűrőszabályok kezelése; a # a sorrendet mutatja, a ::: a megjegyzéseket. A feltételmezőkön túl az illeszkedő csomagok mennyiségét (*Bytes* oszlop) és számát (*Packets* oszlop) is láthatjuk, ez segíthet a hibakeresésben
    - *NAT*: cím/port fordítási szabályok
    - *Connections*: kapcsolat nyilvántartó táblázat

…

- *Pool*: IP-címtartományok megadása, ezekből tud a DHCP szerver címeket kiosztani. A pool-nak nevet adhatunk, megmondhatjuk, mettől meddig osztható ki (pl. 192.168.88.2-192.168.88.10), illetve azt, hogy melyik legyen a következő pool, ha ez már kifogyott
- *Routes*: útvonaltáblázat, átjárók. Statikusakat is vehetünk fel, ha szükséges. Interfész, VLAN definiálásakor ide automatikusan bekerül az útvonal.

…

### 2.1.6 System menü

- *Clock*: rendszeridő és dátum, akár NTP szinkronnal is
- *Identity*: az eszköz hosztneve, itt tudjuk módosítani is
- *Packages*: kiegészítő csomagok letöltése
- *Password*: adminisztrátori jelszó módosítása
- *Reset Configuration*: a gyári beállítások visszatöltése
- *Reboot*: újraindítás. Elég ritkán kell.

…

## 2.2 A Wireshark és használata

A Wireshark egy nyílt forráskódú protokoll vizsgálatot segítő szoftver. Hálózati forgalmat lehet vele elkapni és lementeni (pcap/pcapng formátumba), amit később is megnyithatunk, elemezhetünk, visszajátszhatunk (pl. tcpreplay-jel). A szoftver a csomagok bináris tartalmának megjelenítésén túl képes a számára ismert protokollok dekódolására, fejlécek megjelenítésére. Ezen túl bizonyos protokollok (pl. TCP) esetén képes a csomagfolyamok lekövetésére, azaz az összetartozó csomagok együttes megjelenítésére is.

Csomagelkapásnak, angolul capture-nek nevezzük, amikor a wireshark-kal adott hálózati csatolón érkező csomagokat elkapjuk és elmentjük (haladók akár át is irányíthatják, ún. nevesített csővezetékek/named pipes segítségével). Az elkapás praktikusan válogatás nélküli (promiscuous) módban zajlik, ami nemcsak az elkapást végző csomópontnak szóló csomagok lementését teszi lehetővé, hanem mindent, amit az interfész "lát". Ez például vezeték nélküli hálózati kapcsolat esetén vagy porttükrözés esetén lehet hasznos.

Az elkapást alapesetben végezhetjük GUI-val, de akár konzolról, a *dumpcap* nevű parancssoros alkalmazás segítségével is.

### 2.2.1 Capture filter

Csomagelkapás során hasznos lehet, főleg aggregált kapcsolat esetén, ha nem mentünk le minden csomagot, csak azokat, melyekre a vizsgálat során biztosan kíváncsiak vagyunk. Ilyenkor ún. capture filterrel, vagyis elkapás-szintű szűrési kifejezéssel tudjuk megadni, mit kapjon/mentsen el a Wireshark.

A legsűrűbben használt kifejezések:

- szűrés IP-címre: "host x.x.x.x" vagy "host domainnév"
- szűrés alhálózatra: "net hálózat/maszkbitek" vagy "net hálózat mask maszk"
- szűkítés forrásra: "src" előtag
- szűkítés célra: "dst" előtag
- szűrés protokollra: "proto x", ahol x "udp", "tcp", "icmp", "arp" stb., az első háromnál elhagyható a "proto"
- szűrés portra: "port x", ahol x lehet IANA szolgáltatás név is "pl. port domain" vagy "port 53"
- szűrés porttartományra, pl.: "tcp[0:2]>1500 and tcp[0:2]<1550", vagyis az első két bájtja a TCP fejlécnek (forrás)
- újabb libpcapek (0.9.1 felett): "tcp portrange x-y"
- speciális:
    - "broadcast", "multicast"

A kifejezés elemeit *and* vagy *or* logikai operátorokkal kapcsolhatjuk össze, zárójelezhetjük, illetve a *not* operátor is használható.

### 2.2.2 Display filter

Szűrni a már elkapott csomagsorozatban is lehet. Ez a szűrés csak a megjelenítést befolyásolja, az elmentett fájl tartalmát nem. Ha illeszkedik rá a kifejezés, akkor megjelenik, egyébként nem. A kifejezés szintaxisa némileg eltér a capture filtertől:

- logikai kifejezések: protokoll mezők illesztése operátorokkal, pl. ip.src, tcp.window_size
- operátorok: ==, eq, !=, contains, matches, !, pl. ip.src==192.168.0.0/16, tcp.port eq 25
- kifejezések sorozata logikai kapcsolókkal elválasztva: &&, ||, and, or, pl. http.request.uri matches "gl=se$"

- bővebben ld. Wireshark display filter reference: https://wiki.wireshark.org/DisplayFilters

### 2.2.3 Protokollelemzés

A csomaglistában a kiválasztott csomag részletes protokoll információi megtekinthetőek, rétegről rétegre lenyitva az egyes rétegeinek fejléceit.

Előfordulhat, hogy a szoftver nem ismeri fel automatikusan, milyen protokollok vannak beágyazva, például olyan RTP, ami ugyan UDP-be van ágyazva, de nincs szokásos portszáma, egyedi fejléce. Ilyenkor a csomagot kijelölve explicit mondható meg, milyen protokollal állunk szemben: Decode as...

A tipikus webböngészés több párhuzamos TCP kapcsolat nyitását jelenti időről időre. A Wireshark az egyes TCP folyamokat ki tudja gyűjteni, mindegyikhez rendel egy folyam azonosítót (tcp.stream).

- Statistics → Conversations: protokoll+forrás/cél IP cím/port alapján felderíti a folyamokat és megjeleníti. Itt látható, hogy mettől meddig zajlottak a folyamok, mennyi adat mozgott rajtuk
- Statistics → I/O Graph: átviteli grafikont tudunk rajzoltatni egy vagy több kiválasztott csomagsorozatra, amit display filterrel tudunk meghatározni
- Telephony → RTP Streams: hasonló, mint a Conversations, de az RTP folyamnak bizonyos tulajdonságait is méri, pl. csomagvesztés, jitter

### 2.2.4 Bevezető a Wireshark használatába

*Ez a fejezet jóval bővebb, mint amire a méréshez valójában szükség van!*

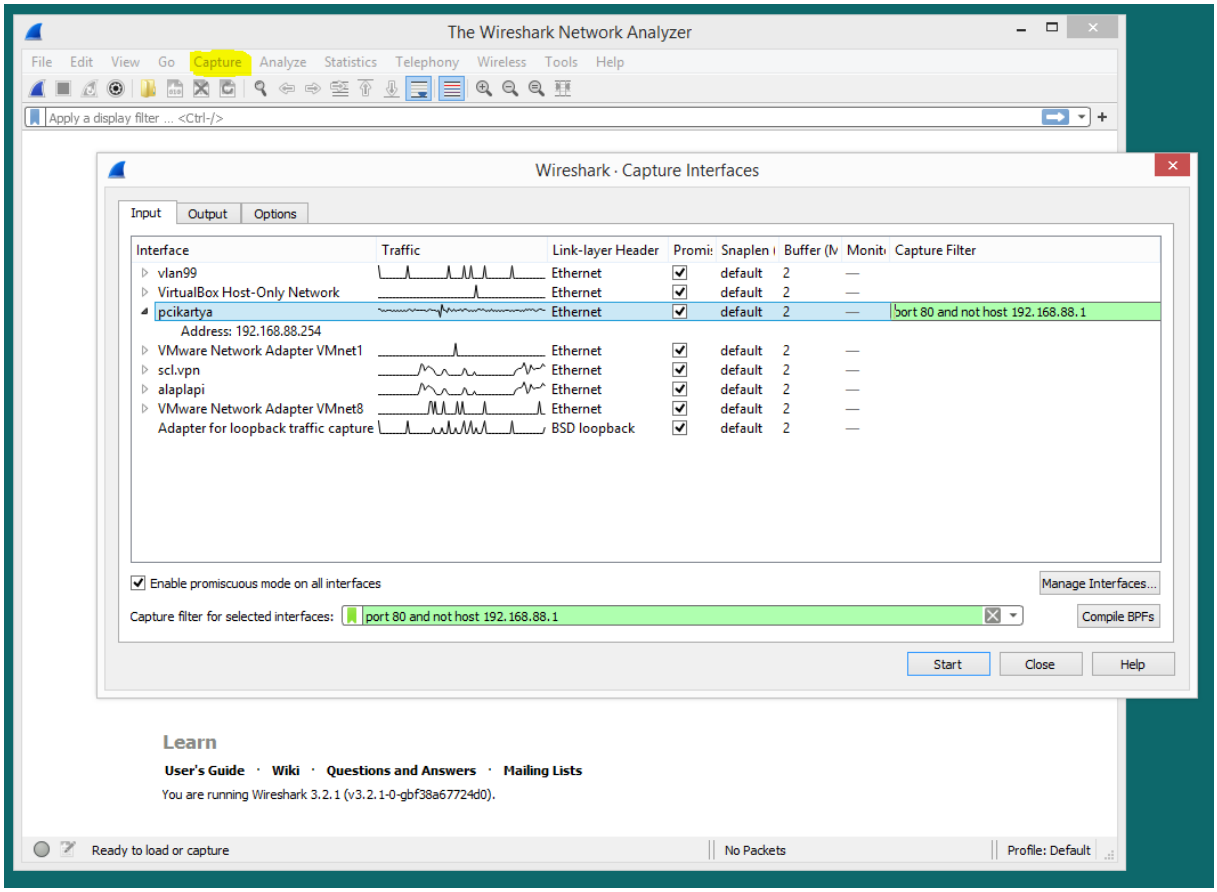
Az indításhoz a Wireshark-ot keressük meg, az ikonja ehhez hasonlatos: 


A nyitóképernyőn azon interfészek listája található, amelyekről adott jogosultsági szinttel tudunk csomagot elkapni, valamint a legutóbb megnyitott pcap fájlok (elkapott csomagokat tartalmazó fájlok) listája.

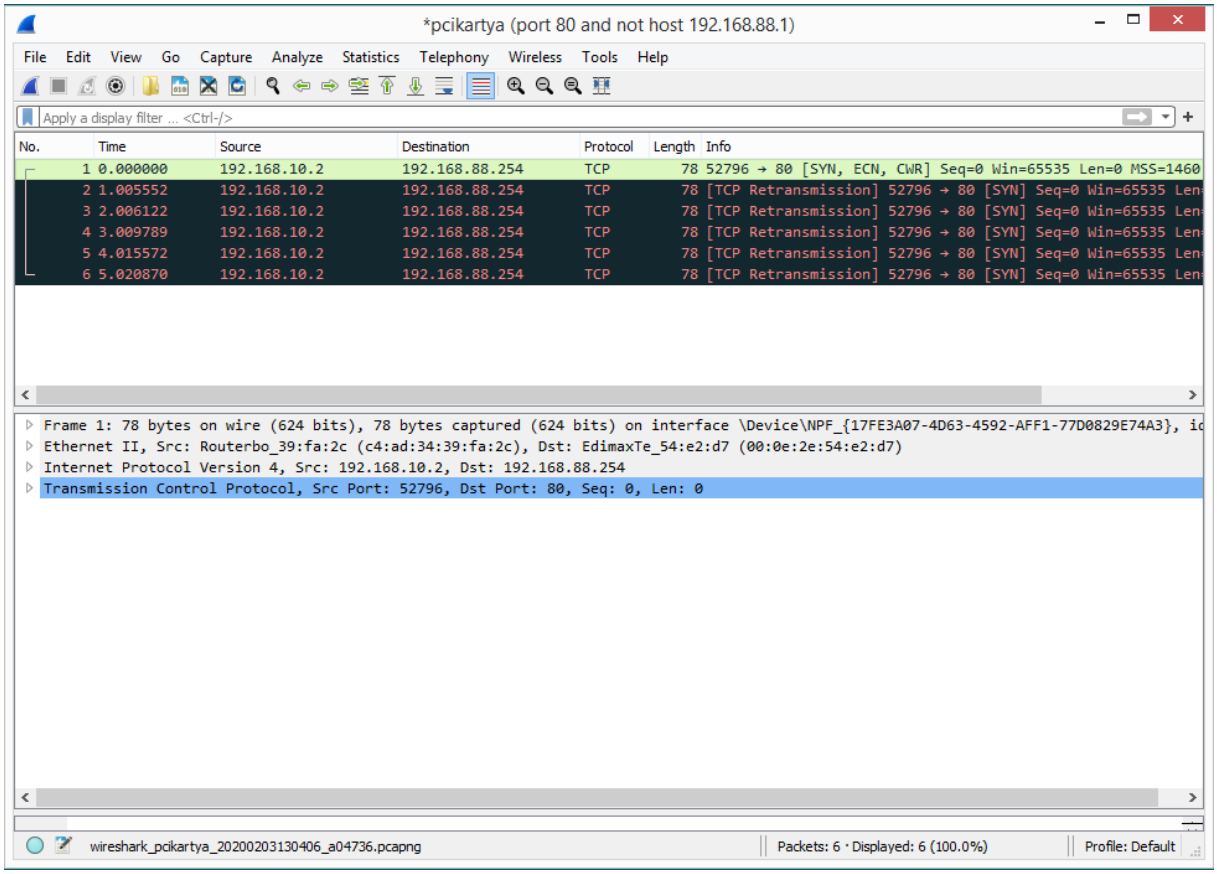
Csomagelkapás indítása adott interfészen, egyszerű capture filterrel, azaz elkapási szűrővel3:

Capture →Options, megkeressük a szóban forgó interfészt (segíthet az IP-címe és neve is) és a Capture Filter oszlopban opcionálisan megadhatunk egy szabályt, majd Start.

[3] Itt azt mondjuk meg, hogy ne az összes csomagot kapja el, csak bizonyosakat. Később a „display filter”, azaz megjelenítési szűrő segítségével a már elkapott csomagok közül kiválaszthatjuk a megjeleníteni kívántakat.



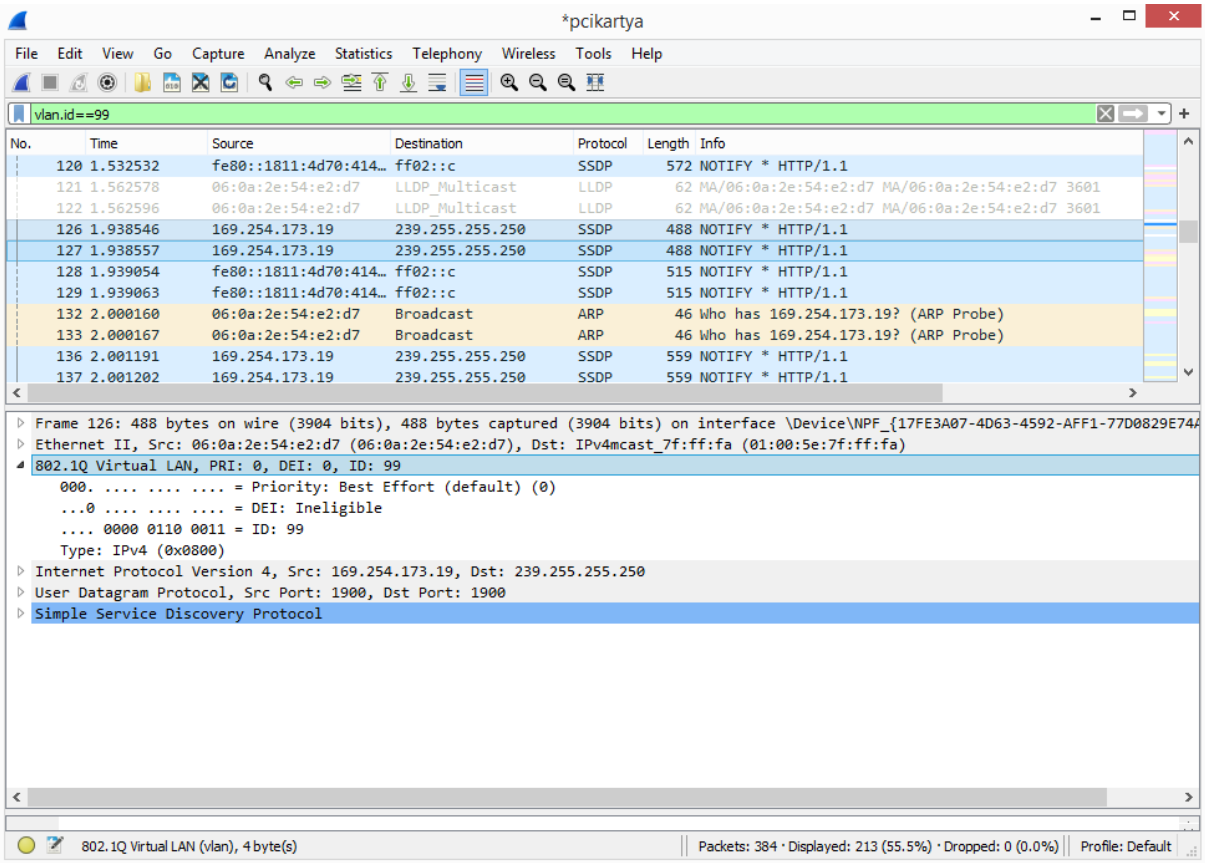

Az elkapást indítva az alábbi kiosztású képernyőhöz jutunk (a fenti csomagelkapási szabály eredményeképp). Azt látjuk, hogy a 192.168.10.2-es címről TCP kapcsolat felépítése indul (csomag #1, TCP-SYN). Normális esetben erre egy TCP-ACK csomagot kellene válaszul lássunk. Ilyet ebben a példában nem látunk, mindössze bizonyos időközönként újra érkező SYN csomagokat. Vagyis ez a TCP kapcsolat nem tudott felépülni.




A menü és eszköztár alatt egy, a böngészők címsorához hasonló szövegbeviteli sávot találunk. Ez a megjelenítési szűrő kifejezés megadására szolgál. Ha üres, akkor minden elkapott csomagot látni fogunk.

Ez alatt a csomagsorozatot fogjuk látni. Alapértelmezés szerint az elkapás szerint rendezve. A csomaglista folyamatosan bővül, egészen amig le nem állítjuk az elkapást a stop gombbal. A listában kiválasztott csomag részletesen megtekinthető a csomaglista alatti, lenyíló menüket tartalmazó blokkban. Itt minden felismert protokoll fejléce meg fog jelenni egy lenyíló menü formájában, ahol a protokollhoz kapcsolódó fejlécmezők tekinthetők meg. A beágyazás természetesen befele haladva tekinthető meg.

Az alábbi ábrán a megjelenítési szűrő használatát láthatjuk: a címsorba írt kifejezéssel csak az adott tulajdonsággal rendelkező (most a 99-es VLAN azonosítójú) csomagok jelennek meg.




### 2.2.5 Távoli csomagelkapás

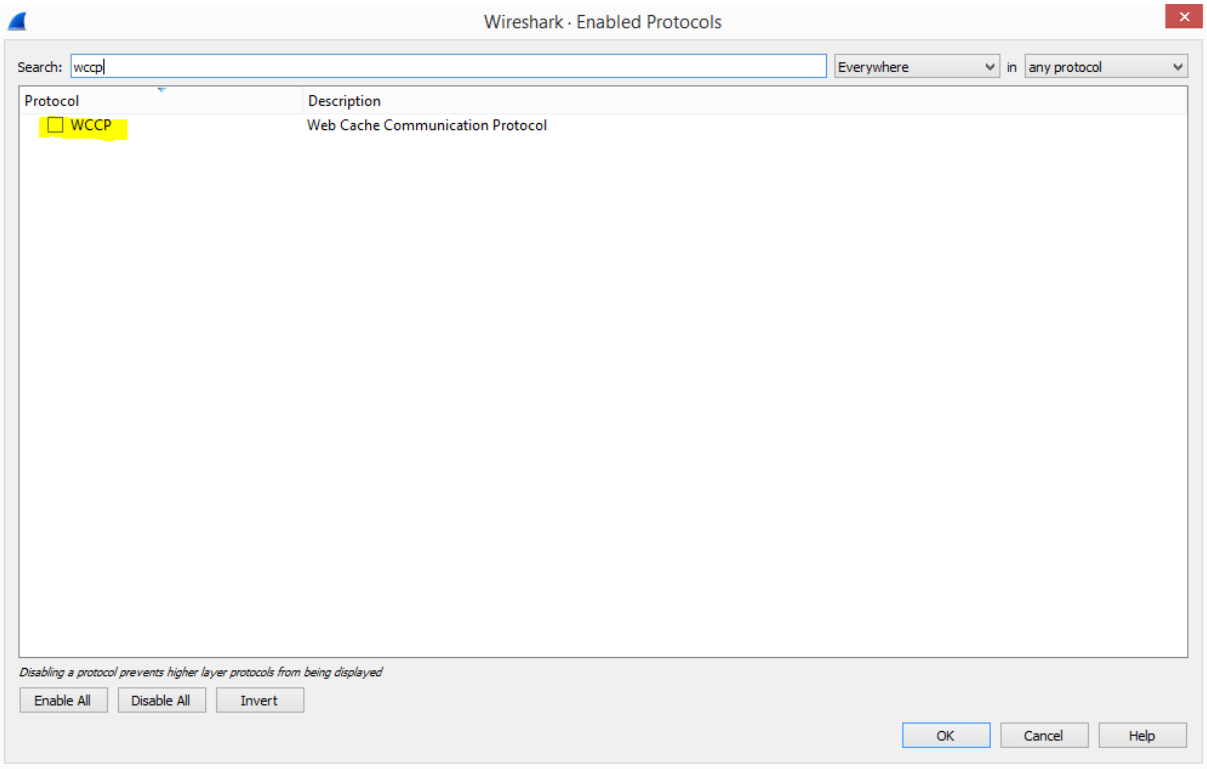
Hasznos funkció lehet, ha a routeren megjelenő forgalmat valamilyen módon ki tudjuk csatolni és egy távoli csomópont felé irányítani vagy akár lementeni. Ezt a funkciót a RouterOS *Tools → Packet Sniffer* pontjában konfigurálhatjuk.

A sniffert a *Start* / *Stop* gombbal kapcsolgathatjuk. A beállítások megváltoztatása csak kikapcsolt állapotban lehetséges. Ha a routerre szeretnénk menteni a forgalmat, érdemes egy megfelelő USB drive-ot dugni az USB portra, majd a File Name-nél megadni az elérési utat és a fájlnevet. (Az USB drive-ról csatolt fájlrendszer elérési útját a Files menüpontban kereshetjük meg.)

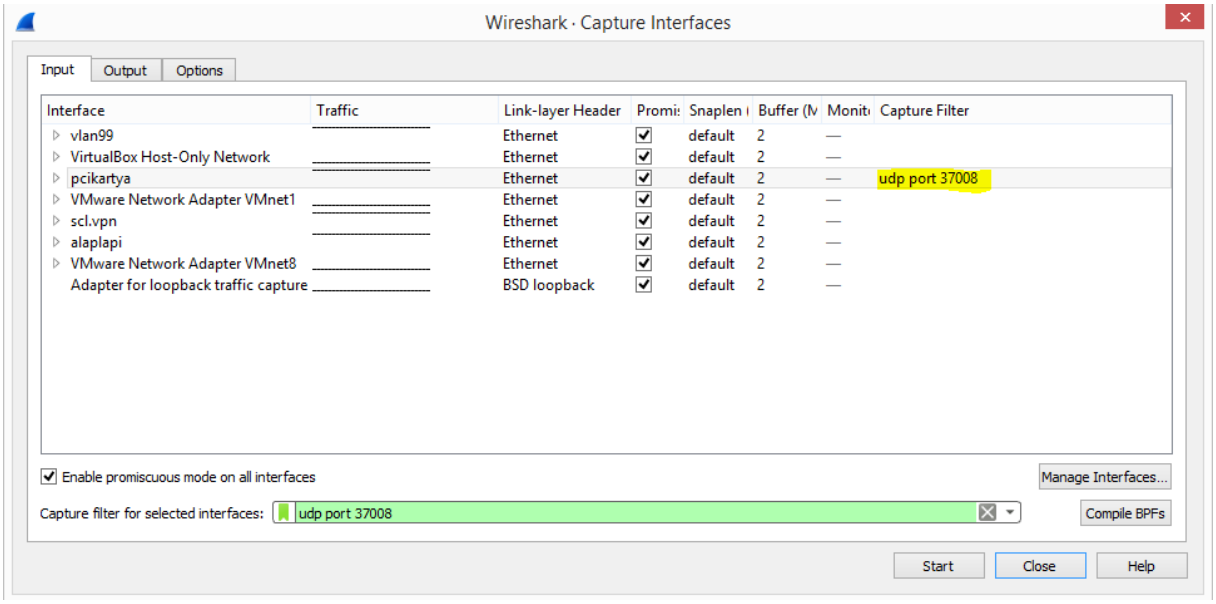
A távoli elkapáshoz szükséges beállítások a *Streaming* fülön találhatók. Az elkapott forgalom továbbítását a *Streaming Enabled* bepipálásával engedélyezhetjük. A *Server*-nél meg kell adnunk a csomópont IP-címét. Amennyiben szűrni szeretnénk a forgalmat (pl. bizonyos típusú csomagok elkapása, stb.), akkor a *Filter Stream*-et kell bepipálnunk és a *Filter* fülön különféle szűrési lehetőségeket tudunk beállítani. Érdekes lehet adott interfészen áthaladó forgalom elkapása, vagy adott IP protokoll vagy portszám szerinti elkapás.

A beállítások elvégzése (*Apply*) után a *Start*-tal indítható a streamelés. A forgalmat a 37008-as portra tartó UDP csomagokba fogja beágyazni a sniffer.

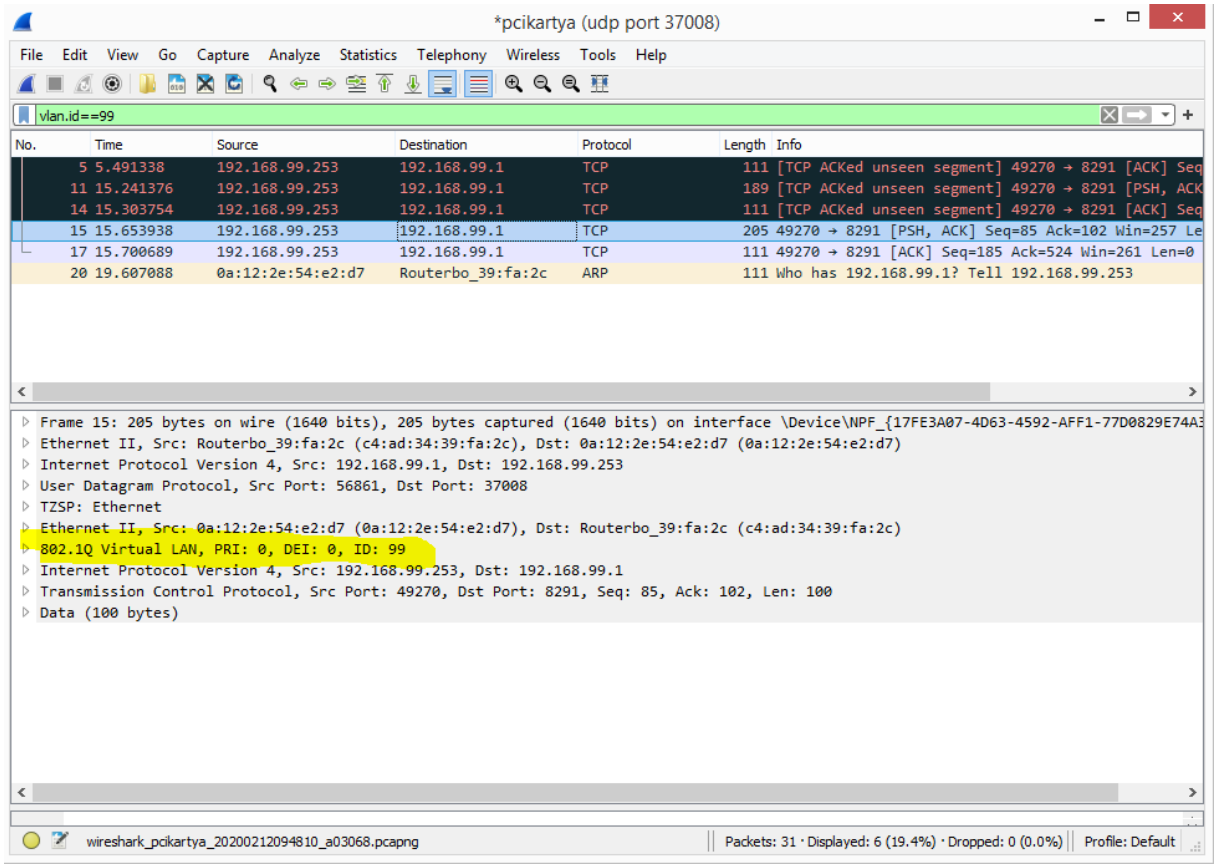
Hogy a Wireshark-kal el tudjuk kapni és meg tudjuk tekinteni ezeket a csomagokat, először is ki kell kapcsolnunk a WCCP protokoll felismerését, amelyet a Wireshark *Analyze → Enabled Protocols* menüjében kell megkeresnünk és törölnünk a pipát:




Ezután az elkapást úgy indítsuk, hogy csak a fenti portszámra érkező csomagokat kapjuk el:




Az így elkapott UDP csomagokban Ethernet keretek lesznek beágyazva, további protokollokat tartalmazva, ahogy azokat a routeren futó sniffer elkapta. Az alábbi képen egy VLAN tag-et is tartalmazó csomagot is láthatunk:


*pcikartya (udp port 37008)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
vlan.id==99
No. Time Source Destination Protocol Length Info
5 5.491338 192.168.99.253 192.168.99.1 TCP 111 [TCP ACKed unseen segment] 49270 → 8291 [ACK] Seq
11 15.241376 192.168.99.253 192.168.99.1 TCP 189 [TCP ACKed unseen segment] 49270 → 8291 [PSH, ACK
14 15.303754 192.168.99.253 192.168.99.1 TCP 111 [TCP ACKed unseen segment] 49270 → 8291 [ACK] Seq
15 15.653938 192.168.99.253 192.168.99.1 TCP 205 49270 → 8291 [PSH, ACK] Seq=85 Ack=102 Win=257 Le
17 15.700689 192.168.99.253 192.168.99.1 TCP 111 49270 → 8291 [ACK] Seq=185 Ack=524 Win=261 Len=0
20 19.607088 0a:12:2e:54:e2:d7 Routerbo_39:fa:2c ARP 111 Who has 192.168.99.1? Tell 192.168.99.253
Frame 15: 205 bytes on wire (1640 bits), 205 bytes captured (1640 bits) on interface \Device\NPF_{17FE3A07-4D63-4592-AFF1-77D0829E74A3
Ethernet II, Src: Routerbo_39:fa:2c (c4:ad:34:39:fa:2c), Dst: 0a:12:2e:54:e2:d7 (0a:12:2e:54:e2:d7)
Internet Protocol Version 4, Src: 192.168.99.1, Dst: 192.168.99.253
User Datagram Protocol, Src Port: 56861, Dst Port: 37008
TZSP: Ethernet
Ethernet II, Src: 0a:12:2e:54:e2:d7 (0a:12:2e:54:e2:d7), Dst: Routerbo_39:fa:2c (c4:ad:34:39:fa:2c)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 99
Internet Protocol Version 4, Src: 192.168.99.253, Dst: 192.168.99.1
Transmission Control Protocol, Src Port: 49270, Dst Port: 8291, Seq: 85, Ack: 102, Len: 100
Data (100 bytes)
wireshark_pcikartya_20200212094810_a03068.pcapng
Packets: 31 · Displayed: 6 (19.4%) · Dropped: 0 (0.0%)
Profile: Default