VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Joni Kaunio

# TIETOVERKON PROTOKOLLIEN SIMULOINTI

Tekniikka ja liikenne
2012

VAASAN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma

# TIIVISTELMÄ

| | |
|---|---|
| Tekijä | Joni Kaunio |
| Opinnäytetyön nimi | Tietoverkon protokollien simulointi |
| Vuosi | 2012 |
| Kieli | suomi |
| Sivumäärä | 59 |
| Ohjaaja | Antti Virtanen |

Tämän opinnäytetyön tarkoituksena oli luoda Vaasan ammattikorkeakoulun tietotekniikan-koulutusohjelmaan laboratoriotyö Vaasan Technobothnialle, jossa tutkittaisiin Wireshark-ohjelman avulla verkossa tapahtuvaa liikennettä. Jo suunnittelun alussa päädyttiin käyttämään ohjelmaa nimeltä Ostinato, jonka avulla pystyttiin simuloimaan opinnäytetyöhön tarvittavaa liikennettä halutuilla protokollilla ja standardeilla.

Opinnäytetyötä suunniteltaessa käytiin monia mahdollisia tutkimuskohteita läpi, mutta lopulta työhön valikoitiin erilaisien tietoliikenteessä tapahtuvien viestien, lähiverkkostandardien, IP-protokollien sekä VLANin tutkiminen ja simuloiminen. Lähteinä käytettiin oppikirjoja sekä internetistä löytyviä RFC-dokumentteja.

Itse laboratoriotyön toteutus osoittautui alkusuunnitelmaa helpommaksi, sillä työn toteuttamiseen ei tarvita kuin yksi kone, jossa käyttäjällä on järjestelmänvalvojan oikeudet. Myöskään uusien laitteiden/ohjelmien hankkimiseen ei tarvinnut käyttää ylimääräisiä resursseja, sillä työn suorittamiseen vaaditut ohjelmat olivat avoimen lähdekoodin ohjelmistoja. Näin ollen työssä käytettävät ohjelmat ovat ideaalisia esim. opetuskäyttöön.

Työn tärkein tavoite on opettaa ja havainnollistaa, millaisina erilaiset protokollat ja standardit näkyvät verkkoliikenteessä. Samalla harjoituksen tekijät saavat hieman kosketusta käytäntöön. Luodun simulointiympäristön yksinkertaisuus tukee oppimista, sillä usein monella eri laitteella toteutettava testiympäristö voi olla vika-altis sekä opiskelijalle liian monimutkainen.

Avainsanat Wireshark, Ostinato, simulointi

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

# ABSTRACT

| | |
|---|---|
| Author | Joni Kaunio |
| Title | Network Protocol Simulation |
| Year | 2012 |
| Language | Finnish |
| Pages | 59 |
| Name of Supervisor | Antti Virtanen |

The purpose of this thesis was to create a laboratory exercise to be used in Technobothnia Vaasa for Vaasa University of Applied Sciences. In this work the everyday online traffic would be examined with the help of a program called Wireshark. In the beginning of the designing process it became clear that the best alternative for simulating the required traffic was a program called Ostinato.

When designing the thesis many possible subjects were thought of but in the end the decision was made to include the following aspects to the thesis: different types of messages that are sent in the Communications Network, Local Area Network Standards, IP-protocols and VLAN. These choices would be examined and simulated. A couple of textbooks and some RFC-documents, which can be found from the Internet, were used as sources for theory.

The realization of the laboratory work turned out easier than originally expected because there's a need for one computer only to carry out the work with. The only thing that is needed from the machine is that the user has administrative rights. And there's also no need to use extra resources to acquire new hardware or software because all the programs that are needed for this thesis are open source. That makes them ideal for instance to be used in educational use.

The main objective of this thesis is to educate and illustrate how different protocols and standards can be seen in network traffic. In the process the students who are performing the laboratory work get a little sense of practice. The simplicity of the created simulation environment promotes the learning experience because when done with multiple units the test environment can be prone to failure and too complex to the students.

Keywords Wireshark, Ostinato, simulating

# LYHENNELUETTELO

| | |
|---|---|
| **ARP** | Address Resolution Protocol,<br>IP-osoitetta vastaavan MAC-osoitteen selvitys |
| **BSD** | Berkeley Software Distribution,<br>toinen Unix-päähaara ja siitä polveutuvat järjestelmät |
| **CFI** | Canonical Format Indicator**,**<br>sääntöjen mukaisen rakenteen osoitin |
| **DHCP** | Dynamic Host Configuration Protocol,<br>IP-osoitteiden jako uusille lähiverkon laitteille |
| **DSAP** | Destination Service Access Point,<br>vastaanottavan sovelluksen tunniste |
| **DSCP** | Differentiated Services Code Point,<br>luokittelee paketin IP-otsakkeissa |
| **ECN** | Explicit Congestion Notification,<br>laajennus, jolla vältetään verkon ruuhkautumiset |
| **FCS** | Frame Check Sequence,<br>kehystarkiste |
| **IEEE** | Institute of Electrical and Electronics Engineers,<br>teknisten ammattilaisten muodostama yhteisö |

| | |
|---|---|
| **ICMP** | Internet Control Message Protocol,<br>kontrolliprotokolla viestien nopeaan lähettämiseen laitteiden välillä |
| **IGMP** | Internet Group Membership Protocol,<br>mahdollistaa liittymisen multicast-ryhmiin IPv4-protokollassa |
| **IPv4** | Internet Protocol version 4,<br>pakettikytkentäisessä verkossa oleva Internet-kerroksen protokolla |
| **IPv6** | Internet Protocol version 6,<br>Kts. IPv4 |
| **LAPB** | Link Access Procedure Balanced,<br>nykyisen pakettiverkon yhteystason protokolla |
| **LLC** | Logical Link Control,<br>vastaa kahden osapuolen välillä olevasta siirtoyhteystason yhteydestä |
| **MAC** | Medium Access Control,<br>määrittelevät tietoverkon osapuolet |
| **MLD** | Multi Listener Discovery,<br>IPv6-protokollan versio IGMP:stä |
| **OUI** | Organizationally Unique Identifier,<br>24-tavuinen IEEEn tarjoama tunniste |

| | |
|---|---|
| **PRTY** | Priority,<br>prioriteetti, jonka avulla kehystä voidaan viedä verkossa muiden ohi |
| **RFC** | Request for Comments,<br>tiedemiehien ja tutkijoiden julkaisemia suosituksia |
| **RIF** | Routing Information Field,<br>kertoo MAC-osoitteen tavuissa bittien järjestyksen |
| **SFD** | Start of Frame Delimiter,<br>aloittaa Ethernet-kehyksen |
| **SNAP** | Subnetwork Access Protocol,<br>tekee Ethernet II-kehyksistä yhteensopivia IEEE 802.3-kehysten kanssa |
| **SSAP** | Source Service Access Point,<br>lähettävän sovelluksen tunniste |
| **TCP** | Transmission Control Protocol,<br>protokolla, johon suurin osa Internet sovelluksista luottaa (esim. WWW, sähköposti) |
| **TPID** | Tag Protocol Identifier,<br>16-bittinen kenttä, joka tunnistaa kehyksen VLAN-tunnistetuksi kehykseksi |
| **UDP** | User Datagram Protocol,<br>yhteydetön protokolla, joka välittää sanomat tarpeen vaatiessa lähettäjältä vastaanottajalle |

| | |
|---|---|
| **VLAN** | Virtual Local Area Network,<br>virtuaalinen lähiverkko, mahdollistaa liikenteen rajaamisen ainoastaan haluttujen koneiden väliseksi liikenteeksi. |

SISÄLLYS

# 1 JOHDANTO

Tämän työn tavoitteena on suunnitella ja toteuttaa laboratoriotyö Vaasan ammattikorkeakoulun tietotekniikan osastolle. Työssä on tarkoitus tutkia ja analysoida tietoverkoissa tapahtuvaa liikennettä, jonka analysointiin käytetään Wireshark-ohjelmaa.

Wireshark on ilmainen avoimesta lähdekoodista koostuva pakettianalysoija yleisimmille käyttöjärjestelmille. Sitä käytetään mm. verkon ongelmatilanteiden tutkimiseen, analysointiin sekä opetustilanteisiin. Lähetyksien simulointiin käytettävä Ostinato on ilmainen ohjelmisto, joka on julkaistu GNU General Public Licensen alla.

Lähetystavoista työhön valikoitiin mukaan broadcast (yleislähetys), multicast (ryhmälähetys) sekä unicast (täsmälähetys). Broadcastia käytetään yleisesti ennalta määräämättömälle vastaanottajamäärälle lähetettäessä, multicastia, jos tietty ryhmä on tiedossa sekä unicastia, jos tietty osoite on tiedossa. Lähiverkkostandardeista mukaan valittiin Ethernet II sekä IEEE 802.3. Ethernet II sekä IEEE 802.3 ovat kehyksiä, joita käytetään verkkoliikenteessä. IP-protokollista mukaan otettiin IPv4- sekä IPv6-protokollat sekä yhtenä tunnelointimenetelmänä 6over4-tunnelonti. IP-protokollat mahdollistavat koneiden välisen kommunikoinnin ja näistä tällä hetkellä yleisimmin käytössä oleva on IPv4-protokolla. IPv6-protokolla on tulevaisuutta ja se on tehty korvaamaan IPv4-protokollan. Viimeisenä mukaan otettiin VLAN eli virtuaalilähiverkko, joka mahdollistaa fyysisen tietoliikenneverkon jakamisen loogisiin osiin.

Yllä mainitut protokollat, standardit sekä lähetystavat sisältyvät tästä opinnäytetyöstä tehtyyn laboratoriotyöhön, jonka tarkoituksena on antaa sen tekeville mahdollisuus simuloida ja verrata keskenään verkkoliikenteessä tapahtuvaa liikennettä.

# 2 OPINNÄYTETYÖN KUVAUS

Tämän työn tarkoituksena on kehittää uusi laboratoriotyö Vaasan ammattikorkeakoulun Technobothnia-laboratorioon opetuskäyttöön. Työhön kuului myös testi-/demoympäristön kehittäminen, jolla voisi simuloida erilaisia tilanteita. Tämä osoittautui kuitenkin tarpeettomaksi, sillä verkosta löytyi ohjelmia, joiden avulla pystyttiin toteuttamaan tämän opinnäytetyön vaatimat simuloinnit. Opinnäytetyöhön otettaisiin mukaan seuraavat asiat:

Tietoliikenneverkon lähetystavat

- Broadcast
- Multicast
- Unicast

Lähiverkkostandardit

- IEEE 802.3
- Ethernet II

IP-protokollat

- IPv4
- IPv6
- 6over4-tunnelointi

VLAN-viesti

Tietoliikenneverkoissa olevat lähetystavat valittiin siksi, koska broadcastia, multicastia ja unicastia käytetään päivittäin verkossa tapahtuvassa liikenteessä. Anycast on myös yksi verkossa tapahtuvista lähetystavoista, mutta koska sille ei löytynyt hyvää simulointitapaa, se jäi pois tästä työstä. Lähiverkkostandardit valittiin tähän työhön siksi, että Ethernet-tekniikka on yleisin lähiverkkotekniikka ja sitä käytetään lähiverkkojen lisäksi eri alue- ja liityntäverkoissa. IP-protokollat valittiin siksi, että harjoitustyötä tekevän on hyvä nähdä konkreettisesti, kuinka IPv4- ja IPv6-

protokollat eroavat toisistaan todellisuudessa. Tässä työssä käytetty simulointitapa on yksinkertaisin tapa näyttää se. VLAN-viestin lähettäminen valittiin siksi, koska simulointitavalla on helppo näyttää harjoitustyötä tekevälle, kuinka tunnisteellinen ja tunnisteeton VLAN-viesti eroaa toisistaan.

# 3 LÄHETYSTAVAT JA SIMULOITAVAT PROTOKOLLAT

## 3.1 Tietoliikenneverkkojen lähetystapoja

Ensimmäisenä keskitytään erilaisiin tietoliikenneverkoissa tapahtuviin lähetystapoihin. Tämän työ tekemisessä käytettiin broadcastia, multicastia sekä unicastia.

### 3.1.1 Broadcast

Broadcast on lähetystä, jolla ei ole ennalta määrättyä vastaanottajamäärää. Sitä käytetään kun halutaan saada tietoa kaikista verkossa olevista koneista tai laitteista. IPv4-verkoissa broadcast-lähetykseen käytetään IP-osoitetta 255.255.255.255. Pelkällä nollalla IP-osoitteessa tarkoitetaan määrittämätöntä ja sillä viitataan yleensä verkkoon. Esim. 64.0.0.0 tarkoittaa verkkoa numero 64, kun taas 64.255.255.255 tarkoittaa kaikkia verkossa olevia (RFC 919).

Ethernet-verkossa broadcast-osoite on FF:FF:FF:FF:FF:FF, joka leviää kaikkialle aliverkkoon (RFC 894).

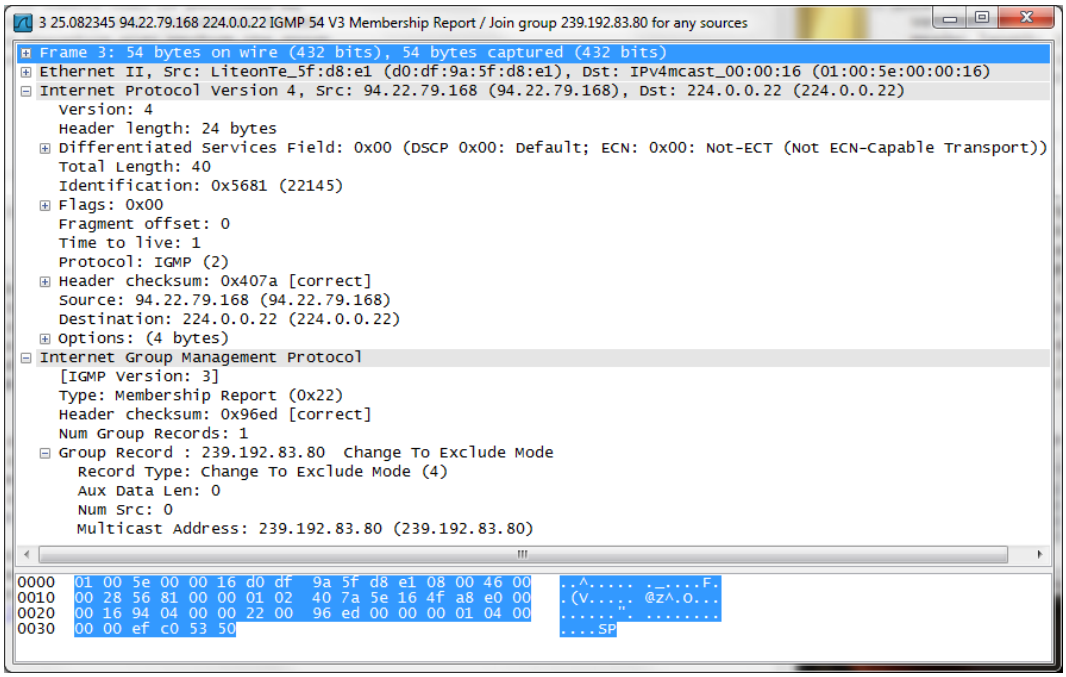
IPv4:sen seuraaja IPv6 ei sisällä perinteistä broadcast-lähetystä vaan se on korvattu multicast-viestillä, joka lähetetään osoitteeseen FF02::1. Tässä FF02 tarkoittaa sitä, että viesti on tarkoitettu kaikille samassa aliverkossa oleville ryhmän jäsenille ja ::1 antaa ryhmätunnisteeksi kaikki aktiivilaitteet ja koneet (Hakala, Vainio 2005, 220–221). FF02::2 taas lähettää tiedon kaikkiin reitittimiin (RFC 4291).

### 3.1.2 Multicast

Multicast on tietoliikenneverkoissa olevaa joukko-, moni- tai ryhmälähetystä, missä multicast-lähetys lähetetään lähettäjältä monelle vastaanottajalle. Lähettäjä lähettää yhden tietosähkeen multicast-ryhmän osoitteeseen, jossa välittäjäreitittimet hoitavat sen kopioinnin ja lähettämisen kaikille vastaanottajille, jotka ovat liittyneet vastaavaan multicast-ryhmään. Multicast-ryhmään liittyminen on mahdollista IPv4:ssä IGMP-protokollaa käyttäen. Liittyminen multicast-ryhmään tapahtuu siten, että ryhmään pyrkijä pyytää jäsenyyttä ryhmään paikallisen reitit-

timen kautta samalla, kun reititin kuuntelee näitä pyyntöjä ja ajoittain lähettää eteenpäin kyselyitä tilauksesta (RFC 1112).

Kuvassa 1 on Wiresharkilla otettu kaappaus IGMP-liikenteestä avattaessa Spotify-ta.




**Kuva 1.** Kaappaus IGMP-liikenteestä

Kuvassa 1 avatussa Internet Protocol Version 4-kohdan Protocol-kohdassa oleva numero 2 kertoo sen, että kyseessä on IGMP-protokolla. Tätä alempana olevassa Internet Group Management Protocol -kohdassa oleva Type-kohdan 0x22 kertoo sen, että kyseessä on IGMPv3 Membership Report. Group Record -kentässä annetaan multicast-ryhmän osoite, johon liitytään. Kohde-osoite 224.0.0.22 kertoo liikenteen olevan IGMPv3-protokollan mukaista.

IPv4:ssä multicast-lähetyksen kohdeosoite on luokkaa D, eli osoite voi olla väliltä 224.0.0.0 – 239.255.255.255. Ethernetin multicastiin on varattu tietty alue, joka on 01:00:5E:00:00:00/25 (RFC 1112).

IP-multicast-osoitteen muuttaminen Ethernetin multicast -osoitteeksi tapahtuu siten, että luokan D osoitteesta siirretään alimmat 23-bittiä Ethernetin multicast -

osoitteeseen. Esimerkiksi, jos luokan D osoite on 224.10.9.6, on Ethernetin multicast -osoite 01:00:5E:0A:09:06 (Harju 2005).

IPv6 käyttää multicast-lähetyksiin osoitelohkoa, jolla on etuliite ff00::/8. IPv6:n Ethernetin multicast -osoite on 33:33:xx:xx:xx:xx, josta 4 viimeistä paria muodostuu IPv6-osoitteen 4 viimeisestä parista. Esimerkiksi jos vastaanottavan kohteen osoite on ff02::1:ff69:13CA, olisi tätä vastaava Ethernet-osoite 33:33:FF:69:13:CA (RFC 4291).

### 3.1.3 Unicast

Unicast-lähetys on liikennettä, jota lähetetään verkossa yksittäiselle laitteelle. Tähän laitteeseen viitataan vastaanottajan IP-osoitteella. Mm. IP-osoitteiden jakelu toimii osaksi unicast-lähetyksenä. IP-osoitetta haluttaessa laitteelta lähetetyn DHCP Discover -viestin (broadcast-lähetys) jälkeinen palvelimelta tuleva DHCP Offer -viesti on unicast-lähetystä, sillä se on osoitettu suoraan siihen laitteen MAC-osoitteeseen, mistä DHCP Discover -viesti lähti alun perin liikkeelle. Jos palvelimen tarjoama IP-osoite sopii asiakaslaitteelle, lähettää se DHCP Request -viestin (broadcast-lähetys) palvelimelle. Palvelin, joka laitteelle IP-osoitetta tarjosi, vastaa kyselyyn myöntävästi lähettämällä unicast-lähetystä olevan DHCP Acknowledgement -viestin laitteelle. Myös ARP-keskustelussa tapahtuva ARP-reply on unicast-lähetystä. (Hakala, Vainio 2005, 211–212, 205). Unicastia käytetään myös Ylen Areena-palveluun sekä Nelosen Ruutu-palveluun ladatuissa lähetyksissä.

## 3.2 Lähiverkkostandardit

Tässä kappaleessa keskitytään IEEE 802.3- ja Ethernet II -standardeihin sekä tekijöihin, jotka erottavat ne toisistaan.

Molempien sekä IEEE 802.3- että Ethernet II -kehyksen, ensimmäinen kenttä koostuu tahdistuskuviosta (preamble), joka auttaa verkkokorttia tunnistamaan mistä kohtaa kehys alkaa. Tahdistuskuvio on pituudeltaan 7 tavun pituinen. Tämän jälkeen tulee 1 tavun mittainen kehyksen alun erotin (SFD), jonka jälkeen

tulevat vastaanottajan ja lähettäjän MAC-osoitteet. Molemmat ovat 6 tavun pituisia (Hakala, Vainio 2005, 144–145).

Seuraavaksi tulevat kohdat, joissa IEEE 802.3- ja Ethernet II -kehys eroavat toisistaan. Ethernet II -kehys sisältää 2 tavun mittaisen tyyppi-kentän. Tämä ilmoittaa, mille ylemmän tason protokollista kehyksen sisältämä hyötykuorma (payload) tulee ohjata. IEEE 802.3 -kehys sisältää samassa kohtaa 2 tavun mittaisen pituus-kentän, joka ilmoittaa hyötykuorman pituuden tavuina. Jos IEEE 802.3 -kehys ei sisällä hyötykuormaa, lisätään sen tilalle LLC-kapselin perään täytettä (padding) sen verran, että kentän vähimmäispituudeksi tulee 46 tavua. Viimeisenä kenttänä molemmissa kehyksissä tulee 4 tavun kokoinen 32-bittinen varmistussumma (FCS), joka lasketaan otsikkotiedoista ja hyötykuormasta (Hakala, Vainio 2005, 145).

Kuvassa 2 on ylläkuvattujen Ethernet II- sekä IEEE 802.3 -kehysten sisällöt.

Ethernet II-kehys

| Preamble | SFD | Kohde | Lähde | Tyyppi | Hyötykuorma | FCS |
|---|---|---|---|---|---|---|
| 7 tavua | 1 tavu | 6 tavua | 6 tavua | 2 tavua | enintään 1500 tavua | 4 tavua |

IEEE 802.3-kehys

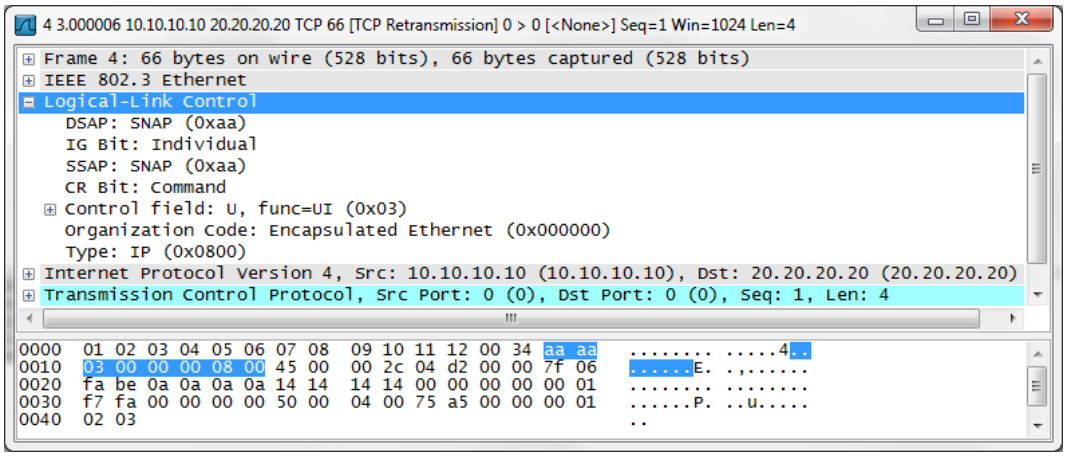
| Preamble | SFD | Kohde | Lähde | Pituus | Hyötykuorma | FCS |
|---|---|---|---|---|---|---|
| 7 tavua | 1 tavu | 6 tavua | 6 tavua | 2 tavua | enintään 1500 tavua | 4 tavua |

**Kuva 2.** Ethernet II- ja IEEE 802.3-kehykset

IEEE 802.3 -kehyksessä oleva LLC-kerros vastaa 2 osapuolen välissä olevasta siirtoyhteystasosta. Tyypillinen IEEE 802.3 -kehyksen LLC-sanoma sisältää lähetettävän (DSAP) ja vastaanotettavan (SSAP) sovelluksen tunnisteet, suora- ja ryhmäosoitusta indikoivan bitin (I/G), komentoa ja vastausta indikoivan bitin (C/R) sekä ohjaustiedon (LAPB) (Granlund 2007, 246–247).

LLC-protokollan tunnetuimpia toteutuksia on SNAP-protokolla, jonka avulla 802.2-protokollaa pyrittiin tarjoamaan siirtoyhteystasolla. (Granlund 2007, 248).

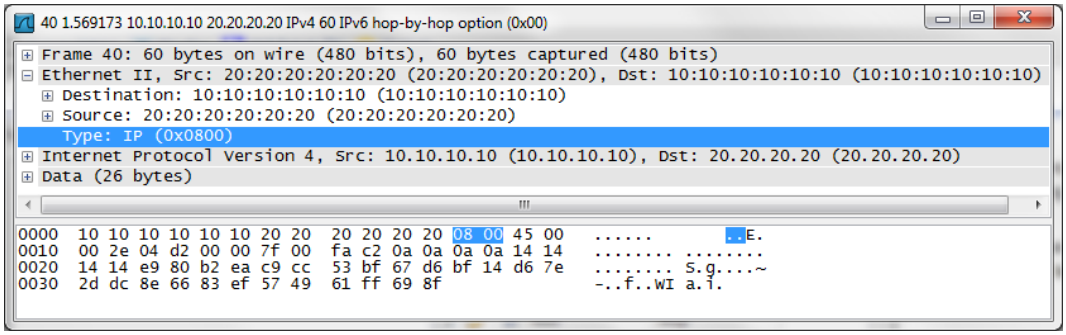
Kuvassa 3 on Ostinatolla lähetetty ja Wiresharkilla kaapattu IEEE 802.3-kehyksen sisältävä viesti, jossa on mukana LLC ja SNAP.




**Kuva 3.** IEEE 802.3-kehys

Kuvassa 3 olevassa kaapatussa kehyksessä SNAP-protokollalla on vakioarvot DSAP:ssa ja SSAP:ssa (0xaa) ja Control-kentän arvo on 0x03. Kuvassa korostettu /aa/aa/03/00/00/00/08/00 kertoo sanoman sisältävän TCP/IP -protokollan tietoa (Granlund 2007, 269).

Kuvassa 4 on Ostinatolla lähetetty ja Wiresharkilla kaapattu Ethernet II -kehys.




**Kuva 4.** Ethernet II-kehys

Kuvassa 4 olevassa näkyy korostettuna tyyppikenttä, joka erottaa Ethernet II -kehyksen IEEE 802.3-kehyksestä. Tyyppi-kentässä oleva heksadesimaaliluku 0800 kertoo sen, että ylemmän tason protokollana toimii IPv4-protokolla. Jos tyyppi-kentässä olisi heksadesimaaliluku 86dd, olisi ylemmän tason protokollana IPv6.

### 3.3 IP-protokollat

Tässä kappaleessa selvitetään IPv4- sekä IPv6-protokolla sekä niiden erot. Samalla tutustutaan työssä käytettyyn 6over4:ään.

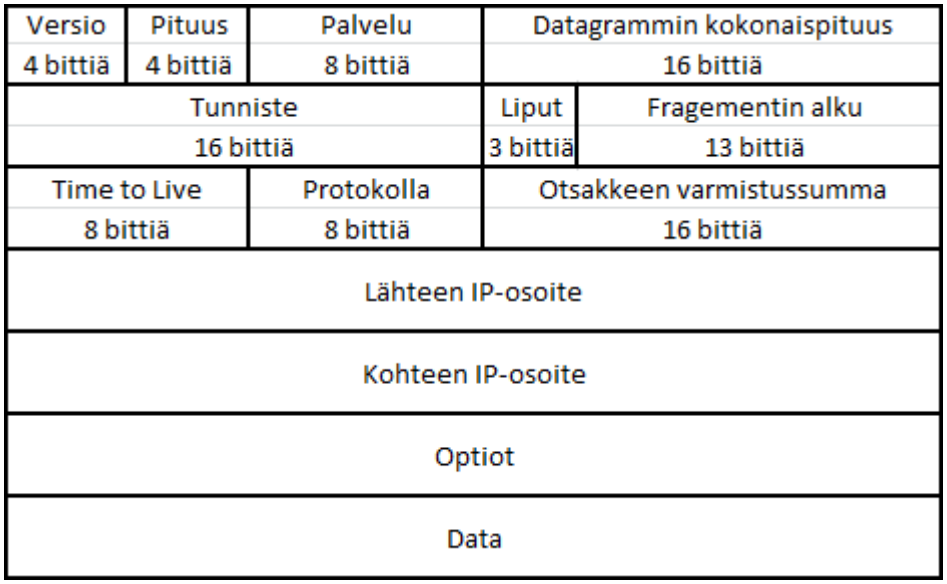
IPv4 on protokolla, johon nykyinen Internet-liikenne pääasiassa perustuu. Sen osoitteisto on 32-bittinen, joten mahdollisia osoitteita on hieman yli 4 000 000 000. Näistä kuitenkin jää huomattava osa käyttämättä, osoitteiden alkuperäisestä luokkajaosta johtuen. IPv4:ssä käytetään 3 eri lähetys- ja osoitemuotoa: broadcastia, multicastia ja unicastia. (Hakala, Vainio 2005, 215–216).

IPv4-protokollan osoiteluokat on jaettu 5 eri luokkaan: A, B, C, D ja E. Taulukossa 1 on jaoteltuna nämä luokat, niiden osoitealueet sekä näiden mahdollistama koneiden määrä tai tarkoitus (Hakala, Vainio 2005, 192).

**Taulukko 1.** IPv4-osoiteluokat

| Luokka | Osoitealue | Koneiden määrä/tarkoitus |
|---|---|---|
| A | 0.0.0.0–127.255.255.255 | 16,7 miljoonaa |
| B | 128.0.0.0–191.255.255.255 | 65,5 tuhatta |
| C | 192.0.0.0–223.255.255.255 | 256 |
| D | 224.0.0.0–239.255.255.255 | Multicast |
| E | 240.0.0.0–255.255.255.255 | Kokeilut |

Kuvassa 5 on IPv4-protokollan otsakkeen rakenne.




**Kuva 5.** IPv4-otsakkeen rakenne

Versio-kenttä kertoo, mikä versio protokollasta on kyseessä. Pituus-kenttä kertoo otsakkeen pituuden 32 bitin sanoina. Palvelu-kentässä olevat bitit 0-2 kertovat otsakkeen kiireellisyyden, bitti nro 3 kertoo mahdollisen viiveen, bitti nro 4 kertoo välitystason ja bitti nro 5 kertoo luotettavuustason (Hakala, Vainio 2005, 310–311).

Aiemmin biteille 6-7 ei ole käyttötarkoitusta. Nykyään näiden bittien avulla saadaan verkon ruuhkautumisesta ilmoitus ilman pakettien kadottamista. Tämä on kuitenkin valinnainen ominaisuus jota käytetään vain silloin, kun molemmat päätepisteet tukevat sitä (RFC 3168).

Datagrammin kokonaispituus ilmoitetaan tavuina 16 bittinä. Tunniste-kenttä yksilöi IP-otsakkeen. Liput kertovat, saako otsaketta fragementoida. Lipuissa oleva bitti 0 on varattu, bitti 1 kertoo, saako pilkkoa (0=saa, 1=ei saa) ja bitti 2 kertoo fragmenttien määrän (0=viimeinen, 1=lisää tulossa). Fragmentin alkukenttä kertoo 13 bitillä, mistä kohtaa alkuperäisen IP-otsakkeen data-kentän tavusta alkaa nykyinen fragmentti. Time to Live-kenttä ilmaisee sekuntteina ajan, minkä datagrammi on olemassa. Protokolla-kenttä kertoo IP-otsakkeen siirtämän ylemmän tason protokollan tyypin. Tärkeimpiä koodeja ovat 1 (ICMP), 2 (IGMP), 4(IP), 6

(TCP) sekä 17 (UDP). Otsakkeen varmistussummakenttä sisältää varmistussumman (Hakala, Vainio 2005, 310–311).

IPv6 on uuden sukupolven protokolla ja sillä onkin 2 merkittävää eroa verrattuna IPv4:een. Ensimmäiseksi, osoitteisto on 128-bittinen sekä toiseksi, siinä yhdistyvät looginen ja fyysinen osoite. Osoitteet ovat IPv6:ssa luokattomia, mutta niissä on hierarkia, joka kertoo minkä operaattorin asiakkaalle kukin verkko kuuluu tai missä verkko sijaitsee maapallolla. IPv4-protokollasta poiketen, IPv6:ssa ei ole ollenkaan broadcastia vaan se on korvattu multicast-lähetyksellä, joka lähetetään kaikille ryhmässä oleville aliverkon koneille. IPv6 mahdollistaa myös anycast-lähetykset, joissa datagrammi lähetetään tietylle tai tietyille koneryhmän laitteille. IPv6-protokollassa osoitteiden esitysmuoto on kaksoispisteillä erotettu 16-bittinen heksadesimaalimuoto IPv4-protokollassa olleiden pisteillä erotettujen desimaalilukujen sijaan (Hakala, Vainio 2005, 216–217).

Kuvassa 6 on IPv6-protokollan otsakkeen rakenne.

<table>
<tr><td>Versio<br>4 bittiä</td><td>Liikenneluokka<br>8 bittiä</td><td colspan="2">Vuonohjaus<br>20 bittiä</td></tr>
<tr><td colspan="2">Hyötykuorman pituus<br>16 bittiä</td><td>Seuraava otsake<br>8 bittiä</td><td>Hyppyjen max. määrä<br>8 bittiä</td></tr>
<tr><td colspan="4">Lähteen IP-osoite<br>128 bittiä</td></tr>
<tr><td colspan="4">Kohteen IP-osoite<br>128 bittiä</td></tr>
<tr><td colspan="4">Data</td></tr>
</table>

**Kuva 6.** IPv6-otsakkeen rakenne

Kuten IPv4-otsakkeessa, myös IPv6-otsakkeessa oleva versio-kenttä kertoo käytetyn protokollan version. Seuraavaksi tulevan liikenneluokka-kentän tarkoituksena on turvata sovelluksille niiden tarvitsemat kaistanleveydet. Liikenneluokka-kenttä kertoo myös, että mihin liikennekategoriaan otsakkeeseen sisältyvä data kuuluu. Kentässä olevat arvot 0-7 kertovat datan kuuluvan TCP-liikenteeseen ja arvot 8-15 kertovat datan kuuluvan UDP-liikenteeseen. Vuonohjaus-kenttä antaa ohjeita otsakkeen välittämiseksi eteenpäin mahdollisimman nopeasti ja virheittä. Hyöty-

kuorman pituus-kenttä kertoo siirrettävän datan määrän tavuina. Seuraava otsake-kenttä kertoo mahdollisista tulevista lisäotsakkeista. Kuten IPv4-otsakkeessa, myös IPv6-otsakkeessa saattaa syntyä fragmentoitumista. Jos sitä kuitenkin syntyy, tarvittavat fragmentointitiedot löytyvät Extension Headerista (Seuraava otsake). Hyppyjen Max. määrä-kenttä vastaa IPv4-otsakkeen Time to Live-kenttää, ja se kertoo kuinka monen reitittimen kautta otsake saadaan välittää (Hakala, Vainio 2005, 312–314).

Työssä käytetty 6over4 on osa IPv6-protokollaa, joka on tarkoitettu siirtämään IPv6-paketteja IPv4-verkon päällä. IPv6-paketit siirretään IPv4-paketeissa IPv4-protokollan tyypillä 41 (IPv6), joka on sama mitä käytetään IPv6-pakettien tunnelointiin IPv4-kehysten sisällä. IPv4-otsake sisältää IPv4 muodossa olevat kohde- ja lähdeosoitteet. IPv4-paketin runko sisältää IPv6-otsakkeen, jota seuraa välittömästi hyötykuorma (RFC 2529).

### 3.4 VLAN

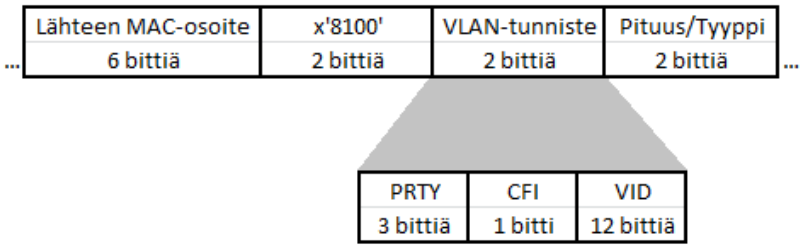
VLANin, eli virtuaalisen lähiverkon, perusideana on luoda kytkinten avulla useista eri verkoissa olevista koneista broadcast domaineja, ja näin ollen samalla rajata tapahtuvaa levitysliikennettä. Virtuaaliset lähiverkot mahdollistavat myös unicast- ja multicast-viestien rajoittamisen vain tiettyjen koneiden välillä tapahtuvaksi liikenteeksi (Hakala, Vainio 2005, 87–93).

Virtuaaliset lähiverkot on mahdollista toteuttaa 4 tavalla: porttien perusteella (port based VLAN), MAC-osoitteen perusteella (MAC-based VLAN), verkkokerroksen perusteella (OSI layer 3 based VLAN) tai policy-määritysten perusteella (policy based VLAN). Porttien perusteella tehtävässä virtuaalisessa lähiverkossa yhden tai useamman kytkimen portit ryhmitellään omiksi VLANeikseen. MAC-osoitteen perusteella tehtävässä virtuaalisessa lähiverkossa kytkimen asetusohjelmalla muodostetaan työasemien ja palvelinten verkkokorttien MAC-osoitteista ryhmiä, jotka muodostavat VLANin. Verkkokerrokseen perustuva virtuaalinen lähiverkko voidaan muodostaa kytkimillä, joissa on valmiiksi reititysominaisuus mukana. Tällä tavalla muodostettaessa VLANit muodostetaan aliverkkojen tai IPX-verkkonumeroiden perusteella. Policy-perusteinen VLAN toteutetaan policy-

määrityksiin perustuvilla kytkimillä. Niissä olevilla hallintaohjelmilla voidaan käyttäjät ryhmitellä eri VLANeihin eri perusteilla. Määrityksinä voidaan käyttää verkko-osoitetta, protokollan tyyppiä tai muita mahdollisia protokollien otsakkeista saatavia tietoja (Hakala, Vainio 2005, 96–99).

Kytkin määrittää kehyksen välittämisen eteenpäin tuloportista vastaanottajan lähtöporttiin 3 sääntötyypin perusteella. Tulosäännöt määrittävät saapuvan kehyksen virtuaaliverkon joko lähettäjän tai kytkimen mukaan, välityssäännöt määrittävät kehyksen lähettämisen eteenpäin sekä lähtösäännöt minkä VLAN-tunnisteen (Tag) kehys saa lähtiessään ulos kytkimestä (Hakala, Vainio 2005, 100–101).

Kuvassa 7 näkyy VLAN-tunnisteen sisältävä kehys.




**Kuva 7.** VLAN-tunniste

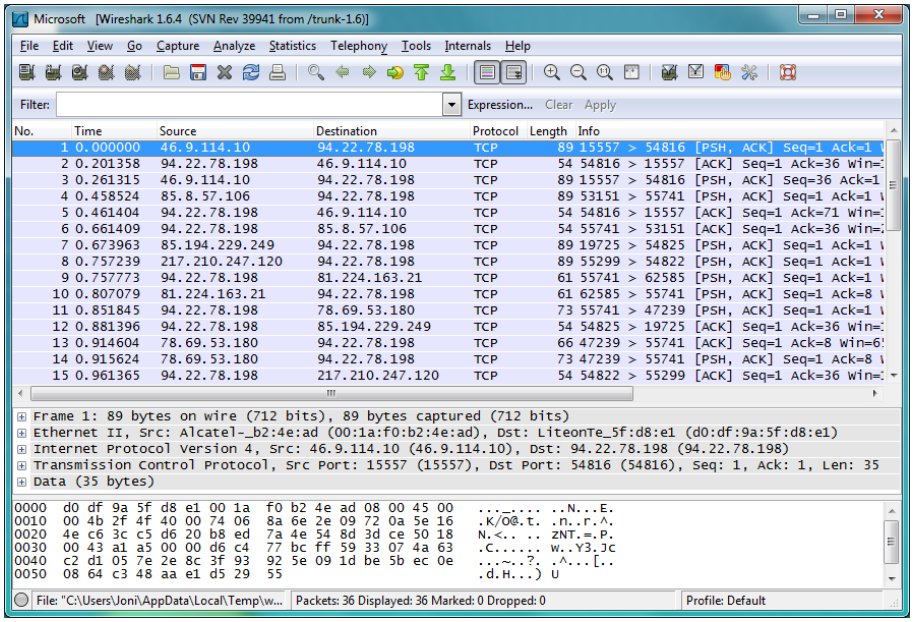
Lähteen MAC-osoite-kentän jälkeisen x8100-kentän avulla tunnistetaan, onko kehyksessä VLAN-tunniste. VLAN-tunniste sisältää 3 eri kenttää: ensimmäisenä PRTY-kentän, joka kertoo käyttäjän prioriteetin, jonka avulla voidaan kehystä reitittää verkossa muiden ohi. Toisena CFI-kentän, joka kertoo onko kehyksessä RIF-kenttä sekä viimeisenä tunnisteen, joka yksilöi sen tietyn lähiverkon, jolle kehys kuuluu (Granlund 2007, 267).

# 4 TYÖKALUJEN ESITTELY

## 4.1 Wireshark

Wireshark on lähinnä verkon analysointiin tarkoitettu ilmainen ohjelma, jonka ensimmäinen virallinen versio julkaistiin 2006. Wiresharkin edeltäjä oli vuonna 1998 päivänvalon nähnyt Ethereal. Wireshark on saatavilla usealle eri käyttöjärjestelmälle, mm. Linux, Windows, Mac OSX. Tässä työssä käytettiin Wiresharkin Windows-versiota, josta uusin versio työtä tehdessä oli 1.6.4. Työssä käytettiin Wiresharkia vaaditun liikenteen kaappaamiseen ja analysoimiseen.

Kuvassa 8 näkyy Wiresharkilla kaapattua Spotify-ohjelmassa tapahtuvaa liikennettä.



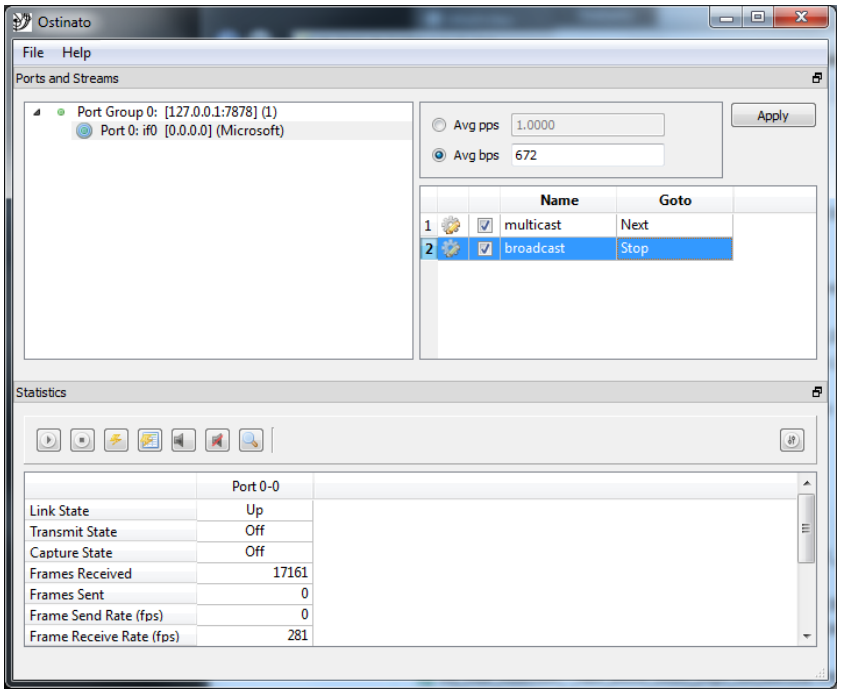

**Kuva 8.** Wiresharkilla kaapattua liikennettä

## 4.2 Ostinato

Ostinato on avoimeen lähdekoodiin perustuva Wiresharkin rinnalla toimiva verkon liikennegeneraattori, jonka ensimmäinen vakaa versio julkaistiin vuonna 2010. Työ tehtiin käyttämällä uusinta vakaata versiota Ostinatosta, joka tätä kirjoittaessa on 0.5. Työ toteutettiin Windows-ympäristössä, mutta Ostinatosta on toimivat versiot myös Linuxille, BSD:lle sekä Mac OS X:lle. Tuettuihin protokol-

liin kuuluvat mm. Ethernet II, IEEE 802.3, VLAN, ARP, IPv4, IPv6 sekä IP-tunnelointi. Ostinatoa käytettiin lähestulkoon kaiken työssä vaaditun liikenteen luomiseen.

Kuvassa 9 tilannekuva lähetyksen luomisesta Ostinatossa.




**Kuva 9.** Lähetyksen luominen Ostinatossa

### 4.3 Multicast-liikenteen luominen

Työn suorittamiseen vaadittava multicast-liikenne luotiin TfGen-ohjelman avulla. Työtä tehdessä uusin ohjelmasta julkaistu versio oli vuonna 2001 julkaistu 1.00. Kuvassa 10 näkyy TfGen-ohjelman pääikkuna.


**Kuva 10**. TfGen-ohjelma

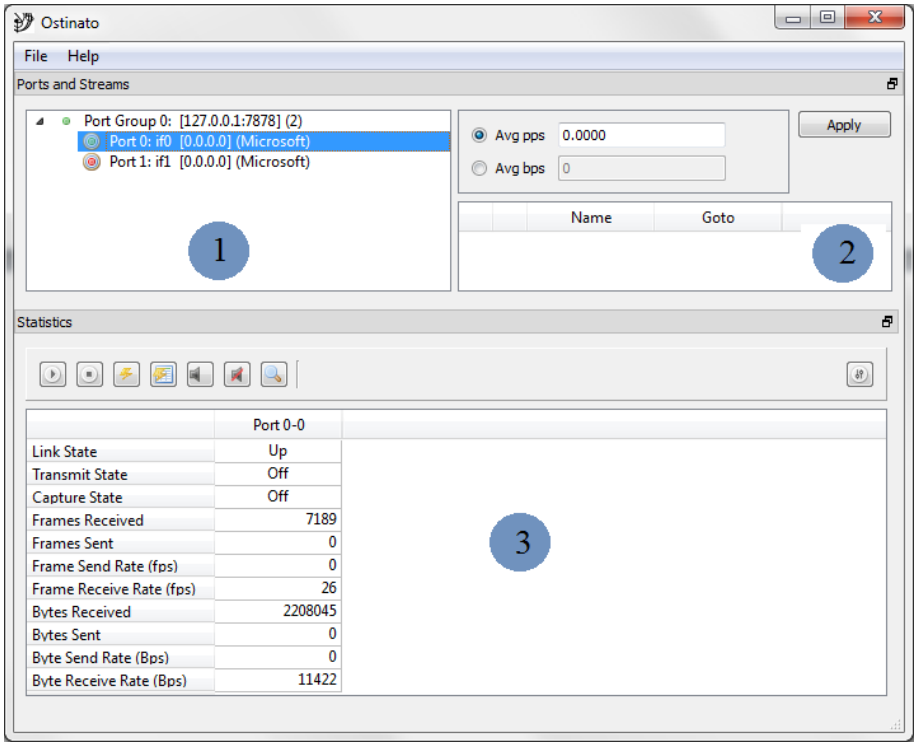
Multicast-liikenteen lähettäminen ei onnistu ilman vastaanottamista. Tähän käytettiin saman tekijän julkaisemaa Mcast-ohjelmaa. Mcast tekee PC:stä noden, eli solmun, joka pystyy vastaanottamaan multicast-liikennettä verkossa. Uusin versio 0.1, jota myös käytettiin työn suorittamiseen, on julkaistu vuonna 1997. Kuvassa 11 näkyy Mcast-ohjelman pääikkuna.


**Kuva 11**. Mcast-ohjelma

# 5 OSTINATON KÄYTTÖ

## 5.1 Yleistä

Tässä luvussa tullaan esittelemään tarkemmin Ostinatossa tehtävän streamin erilaisia mahdollisia asetuksia, itse streamin lähetystä ja kuinka niiden kaappaaminen onnistuu Wiresharkin avulla. Stream on Ostinaton käyttämä nimitys lähetettävälle kehykselle. Kuvassa 12 näkyy Ostinaton pääikkuna, johon on merkitty numeroin eri osiot. Osiossa 1 on portti-lista, jossa olevia portteja voidaan ohjata ja käyttää streamien lähetykseen. Osiossa 2 on stream-lista, johon on listattuna streamit, joita on luotu lähetystä varten. Osiossa 3 on tilasto-lista, josta näkyy portissa tapahtuva liikenne. Osion 3 vasemmassa ylälaidassa olevia ohjaimia käyttäen, pystytään esim. aloittamaan lähetys tai kaappaus.




**Kuva 12.** Ostinaton pääikkuna

Osiossa 1 olevassa portti-listassa olevat portit on jaoteltu Port Group-ryhmiin, joiden sisällä on portit, joita voi käyttää streamien lähetyksiin.

Taulukossa 2 näkyy Port Group-ryhmän erilaiset tilat sekä erilaiset porttien tilat.

**Taulukko 2.** Erilaiset tilat

| Kuvake | Tila |
|---|---|
| | Port Group ei ole yhdistetty tai port on alhaalla |
| | Port Group yrittää yhdistää |
| | Port Group on yhdistetty tai port on ylhäällä |
| | Virhe yhdistettäessä |

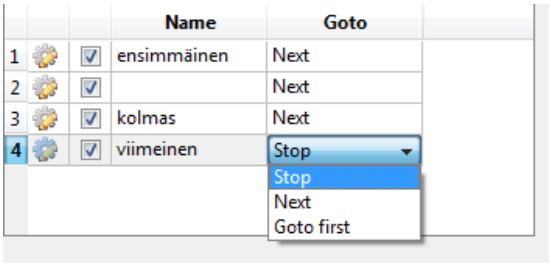
Taulukossa 3 näkyy erilaisia toimintoja, jotka tulevat esille kun osion 1 valitsee oikealle hiiren painikkeella.

**Taulukko 3**. Toiminnot

| Kuvake | Toiminta | Selitys |
|---|---|---|
| | New Port Group | Lisää uuden porttiryhmän listaan osioon 1. Tämän jälkeen yhdistää siihen, luodessa tarvitaan IP-osoite ja mahdollinen portti. |
| | Delete Port Group | Poistaa porttiryhmän listasta. |
| | Connect Port Group | Yhdistää porttiryhmän, jos yhteys on katkennut tai ei ole automaattisesti mennyt päälle. |
| | Disconnect Port Group | Katkaisee yhteyden porttiryhmään. |
| | Exclusive Port Control | Tätä työtä tehdessä vaihtoehto on vielä kokeiluasteella. Sen tarkoitus on kuitenkin toimia siten, että se varaa tietyn portin täysin Ostinaton käyttöön. |
| | Port Configuration | Antaa mahdollisuuden valita portille lähetysominaisuus kahdesta mahdollisesta; Interleaved sekä Sequential. Ensimmäisessä vaihtoehdossa streamit lähetetään limittäin, toisessa peräkkäin. |

Stream luodaan valitsemalla osiosta 1 haluttu portti, jonka jälkeen osioon 2 voidaan luoda stream kahdella tavalla; joko valitsemalla File->New Stream tai

näpäyttämällä hiiren oikealla painikkeella toista osiota ja valitsemalla New Stream pudotusvalikosta. Kun stream on luotu, voi sen nimetä haluamakseen kaksoisnäpäyttämällä name-solua. Name-solun vieressä olevaa laatikkoa käyttämällä voi streamin joko laittaa päälle tai pois. Goto-solusta voidaan määrittää, mitä tapahtuu streamin suorituksen jälkeen. Kuvassa 13 näkyy osio 2.



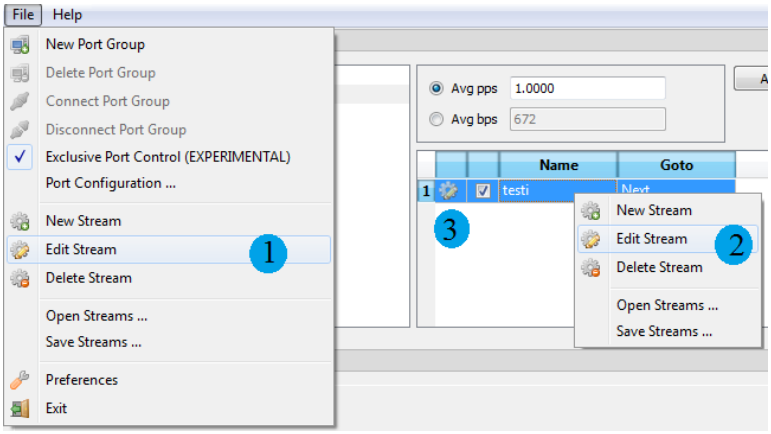

**Kuva 13**. Osio 2

Streamin asetuksia pääsee muuttamaan joko kaksoisnäpäyttämällä streamin järjestysnumeron vieressä olevaa rattaan kuvaa tai valitsemalla hiiren oikealla painikkella stream ja valitsemalla Edit Stream pudotusvalikosta.

## 5.2 Streamin asetukset

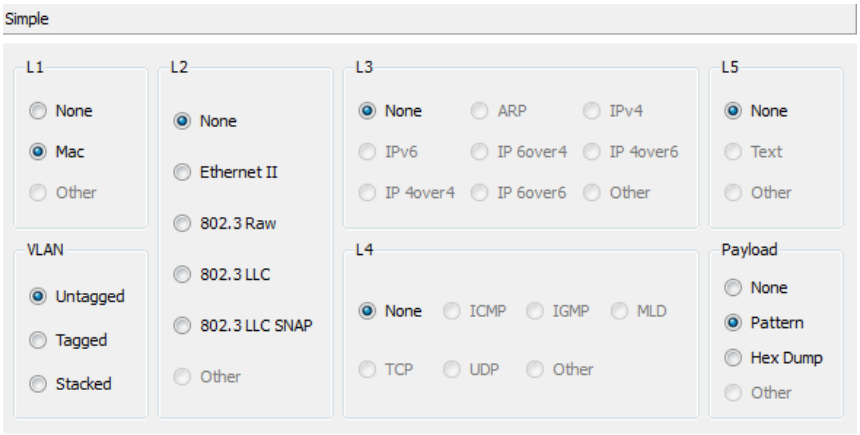
Streamin asetuksiin pääsee käsiksi 3 tavalla; joko valitsemalla vasemmalta ylhäältä File->Edit Stream (kun stream on valittuna), valitsemalla stream oikealla hiiren painikkeella ja valitsemalla Edit Stream tai kaksoisnäpäyttämällä streamissa olevaa ratasta.

Kuvassa 14 näkyy eri vaihtoehdot.



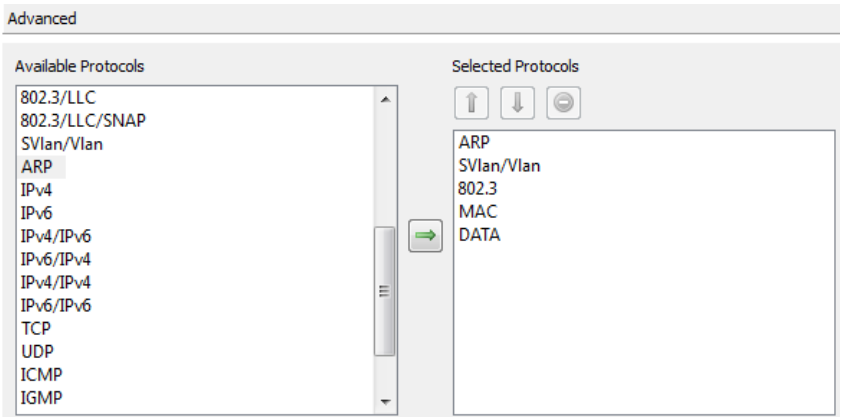

**Kuva 14**. Vaihtoehdot asetuksiin pääsemiseen

Asetusikkunan pitäisi avautua, josta pääsee asettamaan haluamansa ominaisuudet omalle streamilleen. Streamin asetukset on mahdollista valita joko yksinkertaisella (Simple) tai vaativammalla (Advanced) tavalla. Kuvassa 15 on yksinkertainen tapa tehdä stream.




**Kuva 15**. Yksinkertainen tapa

Kuvassa 16 on toinen, eli vaativampi tapa tehdä stream.




**Kuva 16**. Vaativampi tapa

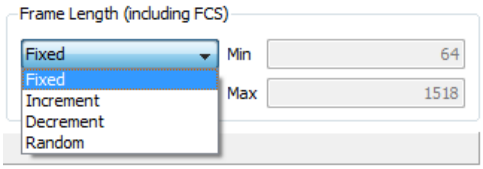
Vaativammassa tavassa on mahdollista kasata protokollat miten haluaa. Esim. vaativammassa tavassa on mahdollista pistää TCP ennen ARP:ia, mikä ei yksinkertaisessa tavassa onnistu. Tällä tavalla voidaan mm. simuloida virheellistä liikennettä. Vasemmalta puolelta ruudukkoa löytyy mahdolliset valittavat protokollat ja oikealta puolelta löytyy jo valitut protokollat. Taulukossa 4 näkyvät vaativassa tavassa olevien nappien toiminnot.

**Taulukko 4**. Toiminnot

| Nappi | Toiminta |
|---|---|
|  | Siirtää valitun protokollan vasemmasta paneelista oikealle |
|  | Siirtää valitun protokollan yhden askeleen ylöspäin |
|  | Siirtää valitun protokollan yhden askeleen alaspäin |
|  | Poistaa valitun protokollan haluttujen protokollien joukosta |

Frame Length -kohdassa asetetaan kehykselle haluttu koko. Vaihtoehtoina on joko Fixed (kehyksellä on tietty koko), Increment (kehyksen koko nousee min ja max välillä), Decrement (kehyksen koko laskee max ja min välillä) tai Random (kehyksen koko on min ja max väliltä).

Kuvassa 17 näkyy valittavat vaihtoehdot.




**Kuva 17**. Kehyksen pituuden asetus

Ikkunan alapuolella olevassa osiossa (kts. kuva 15), joka sisältää yksinkertaisen tavan, määritellään streamille halutut ominaisuudet. L1-kohdassa on aina valmiiksi valittuna MAC-vaihtoehto. Jos vaihtoehdon yrittää vaihtaa noneksi, streamin ominaisuuksien määrittäminen ei onnistu.

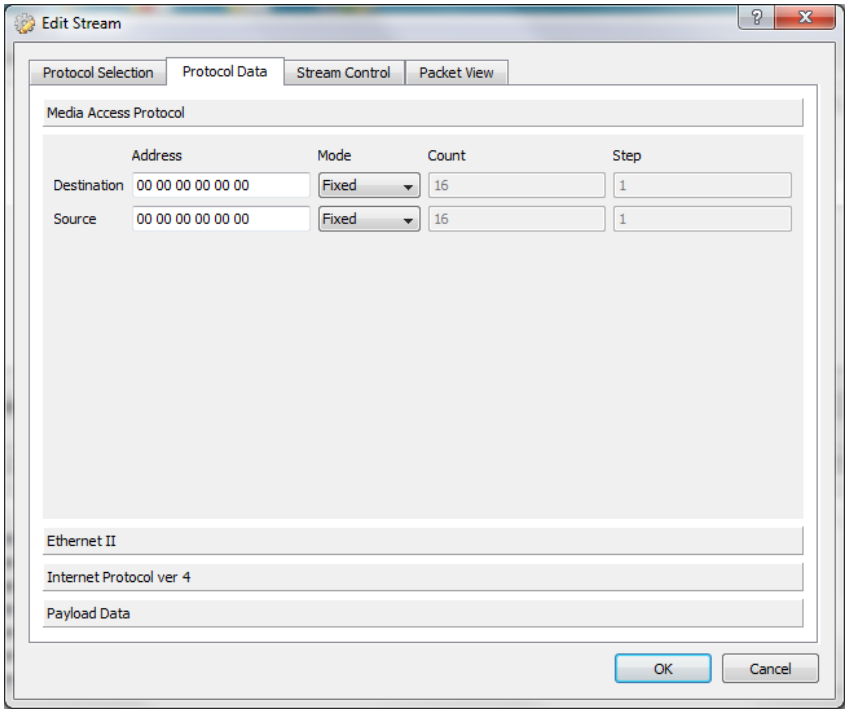
L2-kohdassa valitaan streamiin haluttu peruskerros, jonka sisällöksi voi valita joko Ethernet II:n tai jonkun IEEE 802.3:n vaihtoehdoista Raw, LLC tai LLC SNAP.

Eri vaihtoehdon valitsemalla aukeaa L3-kohtaan vaihtoehtoja, joista voi määritellä streamiin halutun verkkokerroksen, esim. kaikki vaihtoehdot ovat valittavissa Ethernet II:lla ja IEEE 802.3 LLC SNAPilla, mutta IEEE 802.3 Rawilla ei voi valita mitään L3-kohtaan. L3-kohdassa voi streamin valita sisältävän joko ARPin tai IPv4:n, IPv6:n tai erilaisia tunnelointitapoja, kuten IP6over4:n tai IP4over4:n. L4-kohdassa määritellään streamille haluttu kuljetuskerros, joka voi olla esim. ICMP, TCP tai UDP.

L5-kohta aukeaa, jos L4-kohdasta valitsee joko TCP:n tai UDP:n kuljetuskerrokseksi. Tämä kohtaa mahdollistaa sen, että TCP:n tai UDP:n sisään voidaan asettaa tekstiä. Payload-kohdassa määritellään streamille haluttu hyötykuorma. Tähän vaihtoehtoja ovat joko kuvio (pattern) tai hex dump. VLAN-kohdassa määritellään, että halutaanko streamin olevan joko tagattu, untagattu tai kasattu (stacked).

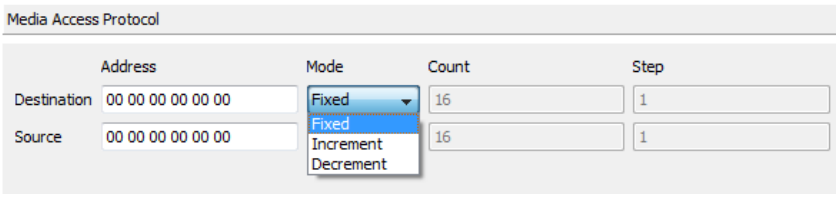
Kun streamille on valittu halutut ominaisuudet, niin siirrytään seuraavaksi Protocol Data -välilehdelle, jossa määritellään kaikki streamiin asetettavat tiedot.

Kuvassa 18 näkyy tämä välilehti.




**Kuva 18.** Protocol Data -välilehti

Protocol Selection -välilehden valinnoista riippuu tämän välilehden muuttuminen. Esim. jos olisit valinnut L2-kohtaan Ethernet II, L3-kohtaan IPv4 ja Payload-kohtaan jotain, ne näkyisivät tässä (kts. kuva 14). Ainoa vaihtoehdoista, mikä tulee jokaiseen streamiin on Media Access Protocol -kohta johon asetetaan kohteen ja lähettäjän MAC-osoitteet. Kuvassa 19 näkyy tämä välikohta.




**Kuva 19.** Media Access Protocol

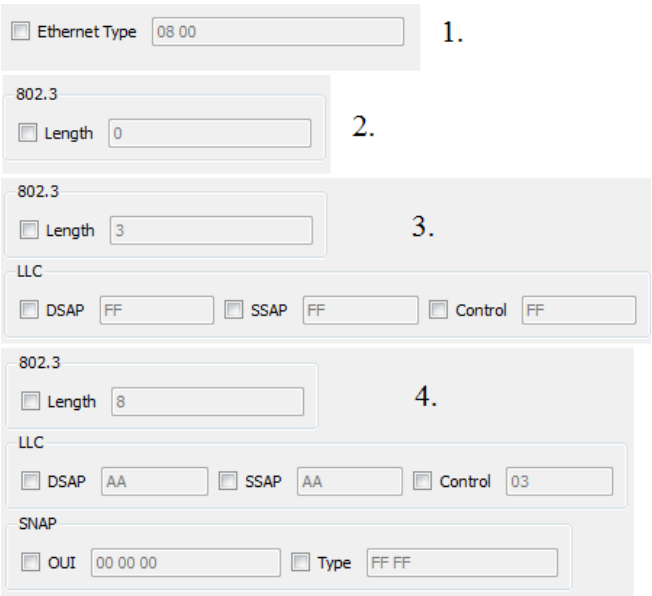
MAC-osoite voidaan joko asettaa kiinteäksi (fixed), nousevaksi (increment) tai laskevaksi (decrement). Jos valitsee jomman kumman jälkimmäisistä vaihtoehdoista, MAC-osoitteen arvo joko nousee tai laskee askelmaan (steppiin) asetetun arvon verran.

Kun MAC-osoitteet on määritelty, tulee seuraavaksi VLAN-protokollan määritys. Protcol Selection -välilehdessä on 3 mahdollista vaihtoehtoa tälle; tunnisteeton (untagged), tunnisteellinen (tagged) sekä kasattu (stacked). Kuvassa 20 näkyy tunnisteellinen vaihtoehto.


**Kuva 20.** Tunnisteellinen VLAN

Jos aiemmassa välilehdessä valitsee vaihtoehdoksi tunnisteettoman, se aiheuttaa sen, että stream ei sisällä minkäänlaista VLANia. Jos taas valitsee kasatun, streamissa on 2 VLANia päällekkäin.

Tämän jälkeen tulee L2-kohdan, eli peruskerroksen, asetus. Kuvassa 21 näkyy mahdolliset valintavaihtoehdot.

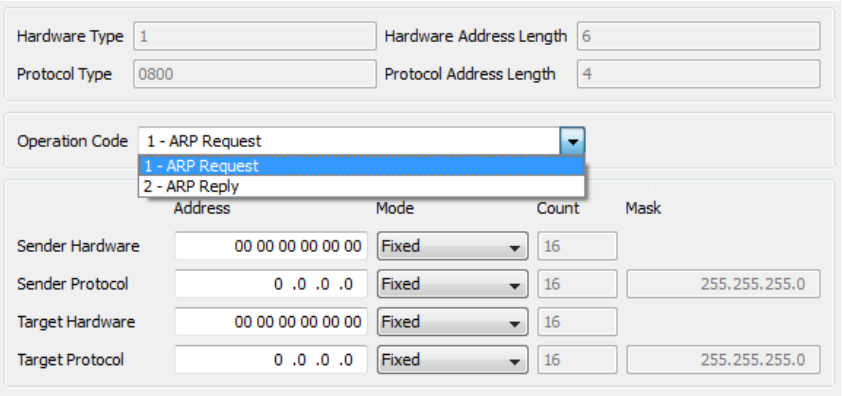



**Kuva 21.** L2-kohta

Kohdassa 1 olevaan Ethernet II:een on mahdollista asettaa Ethernetin tyyppi, joka on oletuksena 0000, mutta vaihtuu automaattisesti sitä mukaa minkä protokollan valitsee L3-kohtaan. Esim. tässä tapauksessa on valmiiksi tullut 0800, joten

valittuna on ollut IPv4-protokolla. Kohdassa 2 olevaan IEEE 802.3 Rawiin voi syöttää pituuden, kohdassa 3 olevaan IEEE 802.3 LLC:een voi syöttää edellä mainitun pituuden, DSAPin, SSAPin ja controlin arvon ja kohdassa 4 olevaan IEEE 802.3 LLC SNAP:iin voi syöttää edellä mainittujen LLC:een kohtien lisäksi OUIn ja tyypin.

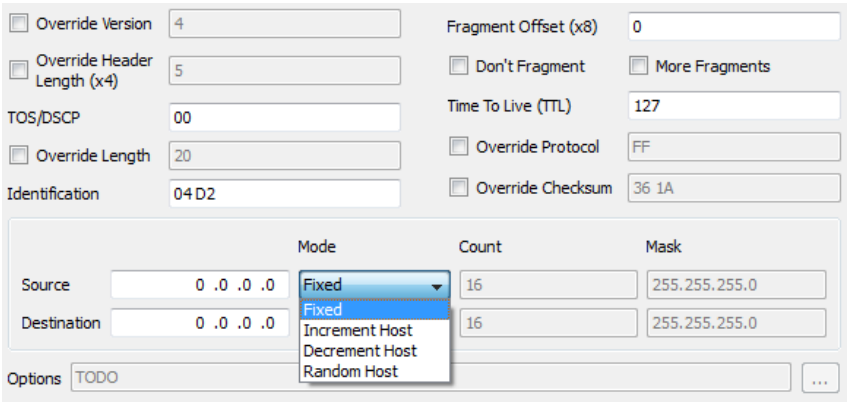
Tämän jälkeen asetetaan L3-kohdassa valitun IP-protokollan tai vaihtoehtoisesti tunnelointimenetelmän arvot. Kuvassa 22 näkyy kohta, missä voidaan asettaa ARP:n arvot.




**Kuva 22.** ARP-protokolla

Kuten kuvasta 22 näkyy, Ostinatossa voi luoda kahdenlaista ARP-protokollaa: pyyntöä ja vastausta. Kuten myös Mac-osoitteiden kohdalla kuvassa 19, on tässäkin mahdollista syöttää joko kiinteät, nousevat, laskevat tai satunnaiset osoitteet lähetyksille.

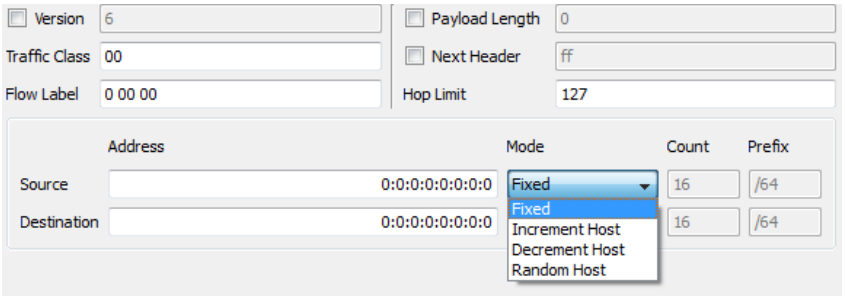
Kuvan 23 mukainen näkymä tulee, jos valitsee Protocol Selection -välilehdessä protokollaksi IPv4:n.




**Kuva 23.** IPv4-protokolla

IPv4-protokollassa on mahdollista asettaa mm. fragmentoituuko kehys, elinajan (Time To Live/TTL), lähde- sekä kohdeosoitteen.

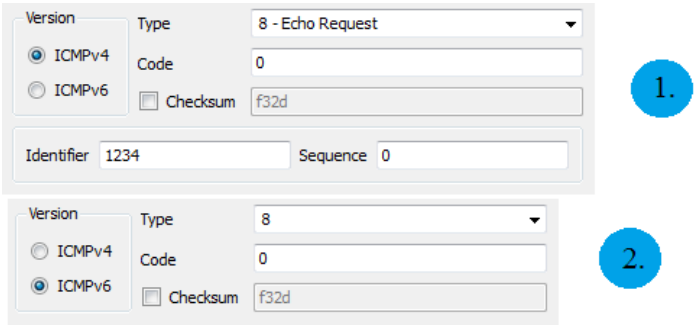
Kuvan 24 mukainen näkymä tulee, jos valitsee IPv4:n sijasta IPv6:n.




**Kuva 24.** IPv6-protokolla

IPv4-protokollan tapaan IPv6:ssa on mahdollista asettaa lähde- ja kohdeosoite. Erona kuitenkin IPv4:ään osoiteformaatti, joka tässä on 128-bittinen. Muita IPv6:ssa mahdollisesti asetettavia vaihtoehtoja ovat mm. hyötykuorman pituus sekä liikenneluokka. Muut protokollavaihtoehdot (6over4, 4over4 jne.) ovat vain yhdistelmiä yllä olevista IPv4- ja IPv6-protokollien kohdista.

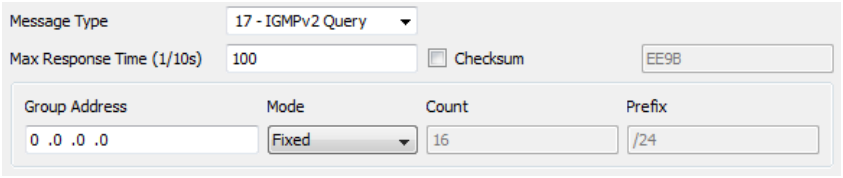
Seuraavaksi määritellään L4-kohta, eli kuljetuskerroksen asetukset. Ensimmäisenä kuvassa 25 näkyvä ICMP.




**Kuva 25.** ICMP-protokolla

Kuten kuvasta 25 näkyy, ICMP-protokollaa on mahdollista lähettää kahtena eri versiona: joko ICMPv4:nä (merkitty numerolla 1) tai ICMPv6:na (merkitty numerolla 2). Suurin ero näissä on se, että ICMPv4:ää käytetään IPv4-protokollan kera ja ICMPv6:ta IPv6-protokollan kera.

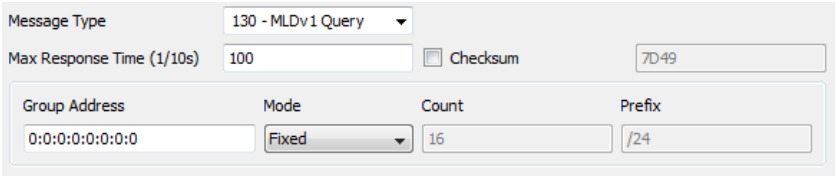
Seuraavana on vuorossa kuvassa 26 oleva IGMP-protokolla.




**Kuva 26.** IGMP-protokolla

IGMP-protokollan asetuksissa pystyy asettamaan esim. viestilleen haluamansa tyylin (Message Type), joita ovat mm. IGMPv1 Query ja IGMPv1 Report, tarkistussumman tai ryhmän osoitteen.

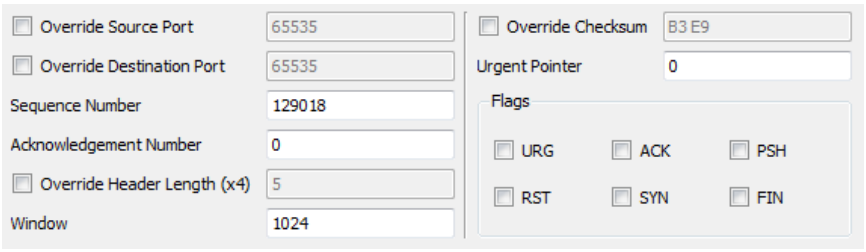
Tämän jälkeen on vuorossa kuvassa 27 oleva MLD-protokolla.




**Kuva 27.** MLD-protokolla

MLD-protokollassa on mahdollista, kuten myös ylempänä olevassa IGMP:ssä, asettaa viestin tyyppi, tarkistussumma sekä ryhmäosoite. Suurin ero tällä ja ylempänä olevalla vaihtoehdolla on se, että IGMP on lähinnä tarkoitettu toimimaan IPv4:n kanssa ja tämä, MLD, on tarkoitettu toimimaan lähinnä IPv6:n kanssa.

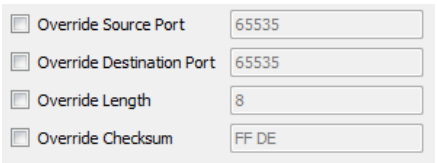
Seuraavaksi on vuorossa kuvassa 28 oleva TCP-protokolla.




**Kuva 28.** TCP-protokolla

TCP-protokollaan on mahdollista asettaa mm. oma lähde- tai kohdeportti, tarkistussumma tai valita joku annetuista lipuista.

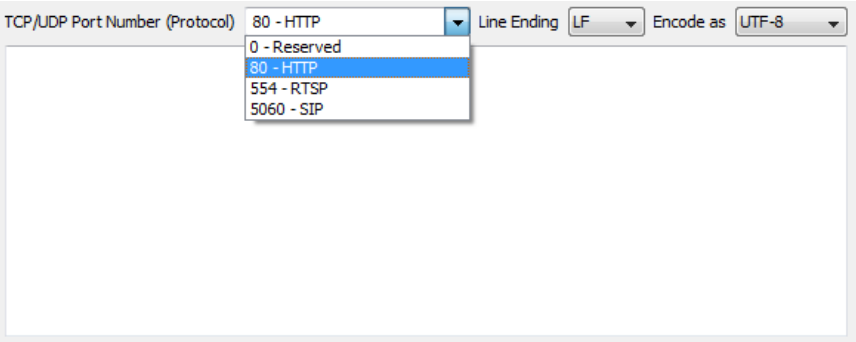
Tämän jälkeen on vuorossa kuvassa 29 oleva UDP-protokolla.




**Kuva 29.** UDP-protokolla

UDP-protokollassa on mahdollista muuttaa, samoin kuin yllä olevassa TCP-protokollassa, mm. lähde- tai kohdeportteja jne. Seuraavaksi tuleva L5-kohta avautuu, jos valitsee näistä jälkimmäisistä protokollista, TCP tai UDP,

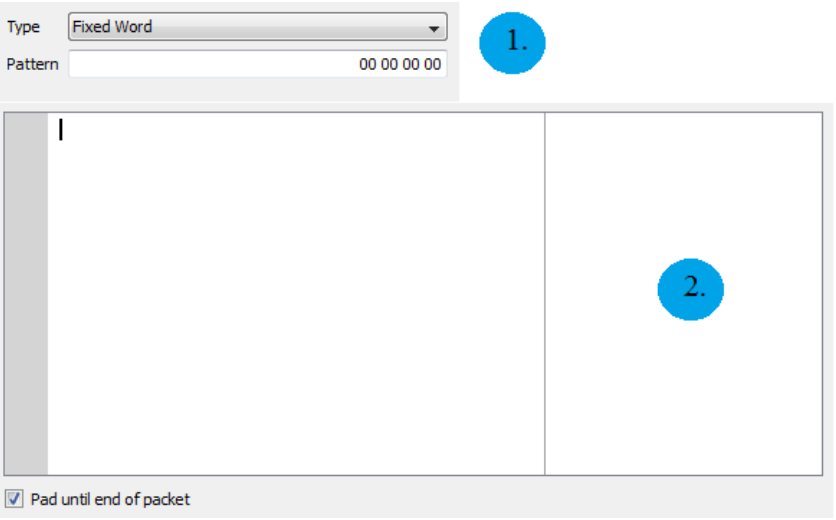
jommankumman L4-kohdassa. L5-kohta mahdollistaa tekstin lisäämisen sekä TCP- että UDP-protokollaan. Kuvassa 30 näkyy L5-kohdan valitsemisen lisäämä kohta Protocol Data -välilehteen.




**Kuva 30.** TCP:n ja UDP:n valitsemisen mahdollistava L5-kohta

Tähän tekstikenttään on mahdollista syöttää joko HTTP:tä, RTSP:tä tai SIP:tä.

Viimeisenä Protocol Data -välilehteen tulee streamin mahdollisen hyötykuorman (Payloadin) asettaminen. Protocol Selection -välilehdessä on mahdollista valita hyötykuormaksi jo aiemmin mainitut: kuvio (pattern) tai hex dump. Kuvassa 31 näkyvät molemmat vaihtoehdot.

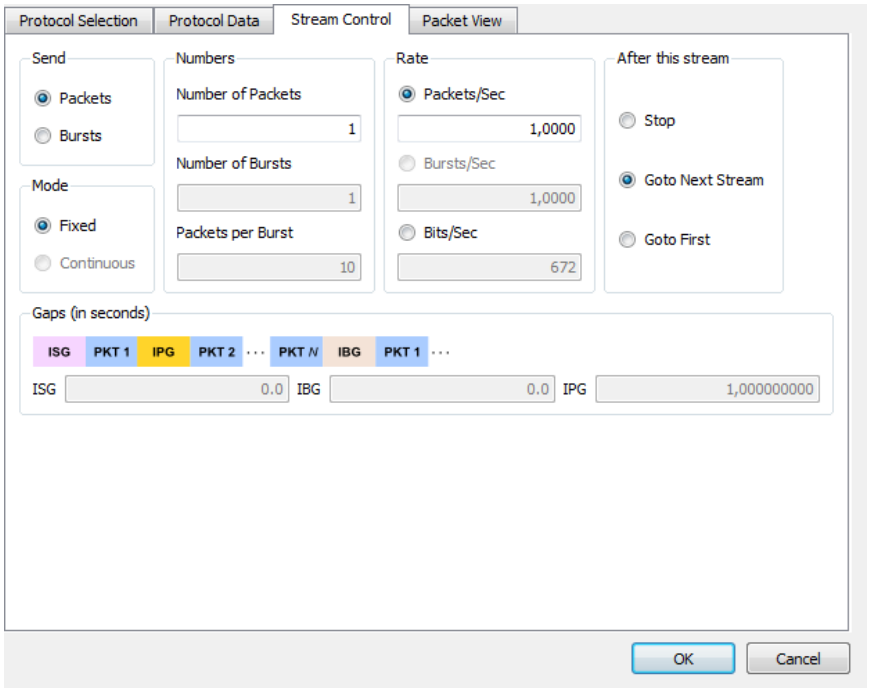



**Kuva 31.** Hyötykuorma

Numerolla 1 merkitty kohta näkyy, jos hyötykuormaksi valitsi kuvion. Kuvioksi on mahdollista valita joko kiinteä sana, laskeva tai nouseva bitti tai satunnainen

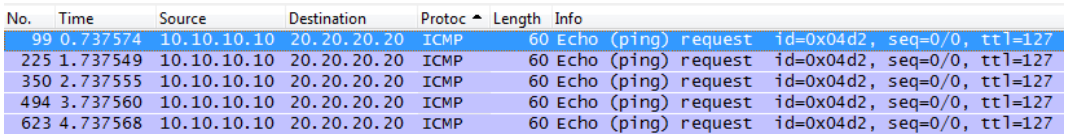
kuvio. Kiinteän sanaan voi kirjoittaa minkä tahansa sanan, rajoituksena kuitenkin se, että sanassa käytettävät kirjaimet ja numerot ovat heksa-lukuja. Numerolla 2 merkitty kohta taasen näkyy, jos hyötykuormaksi valitsi hex dumpin. Kuten kuvi-ossa oleva kiinteään sanaan, myös hex dumpiin saa syöttää mitä vain. Ainoana rajoituksena pysyy se, kuten myös edellisessä vaihtoehdossa, että tekstikenttään syötettävä numero/kirjainyhdistelmät ovat heksa-lukuja.

Seuraavaksi toiseksi viimeiseen, kuvassa 32 näkyvään Stream Control -välilehteen.




**Kuva 32.** Stream Control -välilehti

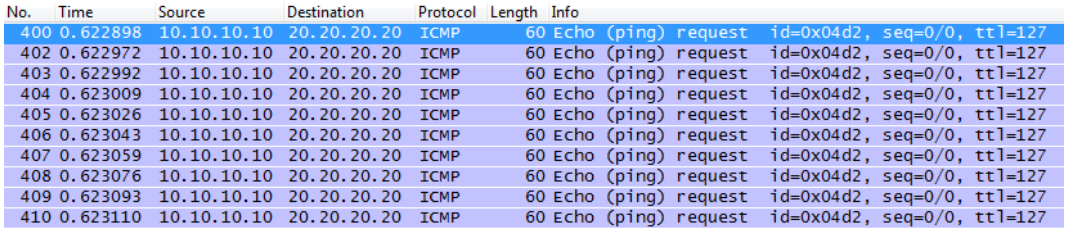
Stream Control -välilehdessä streamille asetetaan lähetysominaisuudet, eli esi-merkiksi lähteekö stream paketteina (packets) tai ryöppyinä (bursts). Paketteina lähetettäessä streamit lähtevät liikkeelle yksi kerrallaan. Tässä lähetystavassa voi-daan määritellä kuinka monta pakettia lähtee yhdessä streamissa, sekä kuinka monta pakettia lähtee yhdessä sekunnissa. Kuvassa 33 näkyy miltä stream näyttää kaapattuna, jos tavaksi valitsee paketin, pakettien määräksi 5 sekä pakettien lähe-tysnopeudeksi 1/s.



| No. | Time | Source | Destination | Protoc | Length | Info |
|---|---|---|---|---|---|---|
| 99 | 0.737574 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 225 | 1.737549 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 350 | 2.737555 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 494 | 3.737560 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 623 | 4.737568 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |

**Kuva 33.** Pakettilähetys

Kuten kuvasta 33 näkyy, paketteja on lähtenyt yhteensä 5 kappaletta sekunnin välein. Toisena lähetystapana on ryöppy, jolloin paketteja lähtee suuri määrä samaan aikaan. Tässä lähetystavassa voidaan määritellä ryöppyjen määrä, kuinka monta pakettia menee/ryöppy sekä kuinka monta ryöppyä lähtee/sekunti. Kuvassa 34 näkyy miltä stream näyttää kaapattuna, kun tavaksi on valittu ryöppy, niiden määräksi 1, pakettien määräksi/ryöppy 10.



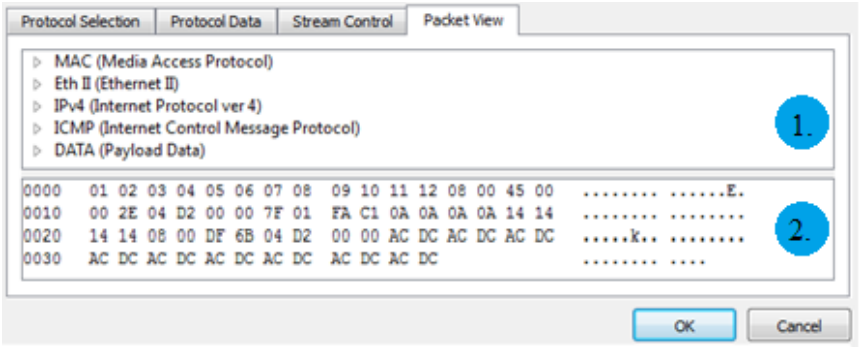
| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 400 | 0.622898 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 402 | 0.622972 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 403 | 0.622992 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 404 | 0.623009 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 405 | 0.623026 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 406 | 0.623043 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 407 | 0.623059 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 408 | 0.623076 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 409 | 0.623093 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |
| 410 | 0.623110 | 10.10.10.10 | 20.20.20.20 | ICMP | 60 | Echo (ping) request id=0x04d2, seq=0/0, ttl=127 |

**Kuva 34.** Ryöppylähetys

Kuten kuvasta 34 näkyy, paketteja lähti yhdessä sekunnissa 10 kappaletta. Kun haluttu lähetystapa on valittuna, niin tältä välilehdeltä voi asettaa vielä sen, mitä tapahtuu streamin jälkeen (kts. kuva 32). Vaihtoehtoina ovat joko pysähtyminen, jatkaminen seuraavaan streamiin tai ensimmäiseen tehtyyn streamiin palaaminen.

Kun kaikki asetukset ovat kunnossa, voi tehtyä streamia tarkastella vielä viimeisestä Packet View -välilehdestä.

Kuvassa 35 on esimerkki, miltä Packet View -välilehti voi näyttää.

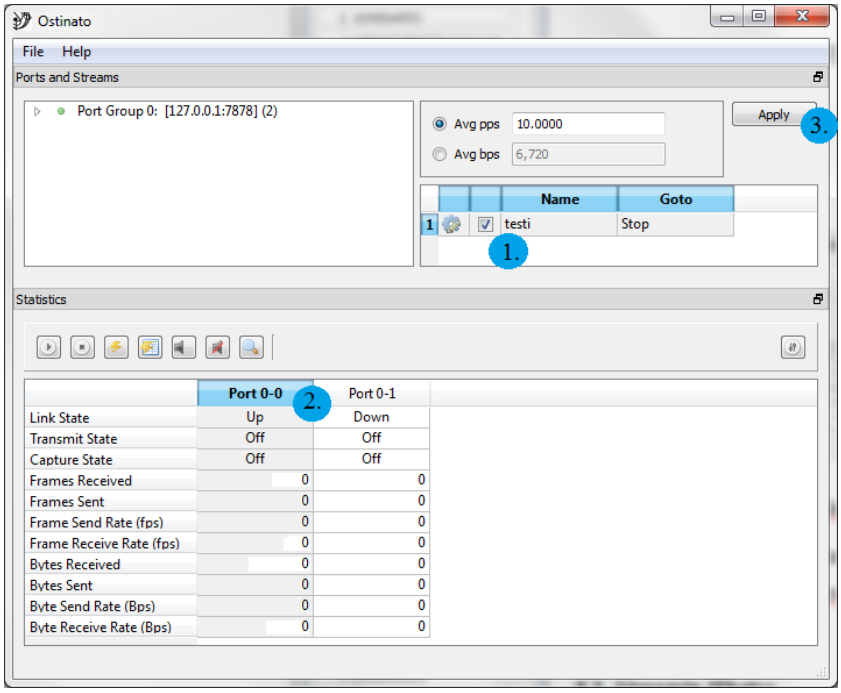



**Kuva 35.** Packet View -välilehti

Numerolla 1 merkityssä laatikossa näkyy kaikki aiemmissa välilehdissä valitut protokollat puunäkymässä (tree view) sekä numerolla 2 merkityssä laatikossa näkyy kaikki hex dumpina.

### 5.3 Streamin lähetys

Streamin valmistuttua siirrytään itse streamin lähettämiseen. Streamin lähettäminen tapahtuu pääikkunasta. Kuvassa 36 näkyy tilanne ennen streamin lähetystä.




**Kuva 36.** Lähetystilanne

Ennen lähetystä kannattaa varmistaa, että numerolla 1 merkitty streamin nimen vasemmalla puolella oleva neliö on valittuna. Kun stream on valmis lähetettäväksi, valitaan pääikkunan alhaalla olevasta Statistics-kohdasta portti, johon stream luotiin. Tässä tapauksessa oikea portti on merkitty numerolla 2. On tärkeää valita juuri sinisellä korostettu kohta, sillä jos valitsee jonkun muun kohdan, streamin lähettäminen ei onnistu. Kun oikea kohta on Statistics-kohdasta valittu, näpäytetään pääikkunan oikeassa yläkulmassa, streamin luontikohdassa, olevaa numerolla 3 merkittyä Apply-nappia. Nyt stream on vahvistettu aiemmin valittuun porttiin. Streamin lähetys tapahtuu valitsemalla Statistics-kohdasta löytyvästä nappivalikosta ensimmäisenä olevan Start Tx -napin ja streamin lopetus tapahtuu Start Tx -napin viereisestä Stop Tx -napista. Jos lähetys onnistuu, niin aiemmin valitussa portissa olevaan Transmit State -kohtaan pitäisi tulla lukemaan Off-tilan tilalle On sekä Frames Sent -kohtaan pitäisi tulla lähetettyjen pakettien määrä.

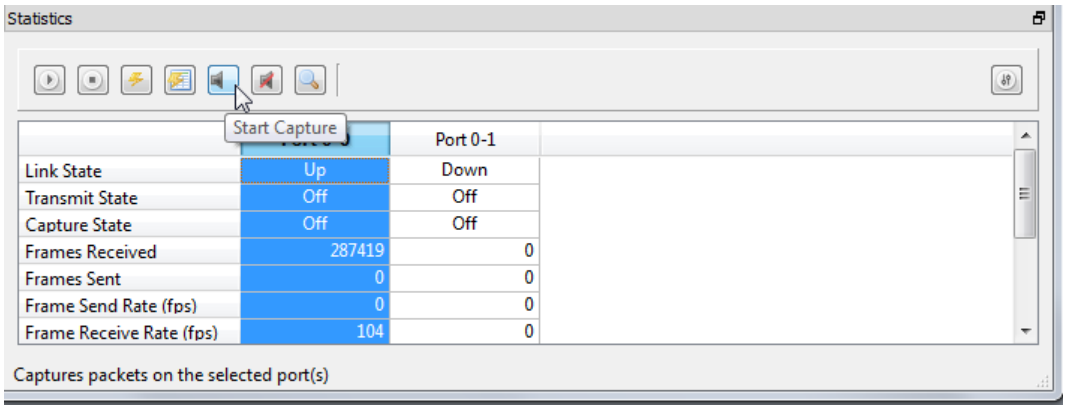
Taulukossa 5 on Statistics-kohdassa olevien nappien selitykset.

**Taulukko 5.** Statistics-kohdan nappien selitys.

| Nappi | Toiminta | Selitys |
|---|---|---|
| | Start Tx | Aloittaa lähetyksen valitussa portissa. |
| | Stop Tx | Lopettaa lähetyksen valitussa portissa. |
| | Clear Selected Port Stats | Nollaa valitusta portista aiemmin lähetetyn |
| | Clear All Ports Stats | Nollaa kaikkien porttien tiedot. |
| | Start Capture | Aloittaa valitun portin kaappaamisen. |
| | Stop Capture | Lopettaa valitun portin kaappaamisen. |
| | View Capture Buffer | Näyttää valitusta portista kaapatut paketit Wiresharkissa. |
| | | Voi valita portit, jotka haluaa näkyvän Statistics-ikkunassa. |

## 5.4 Streamin kaappaaminen

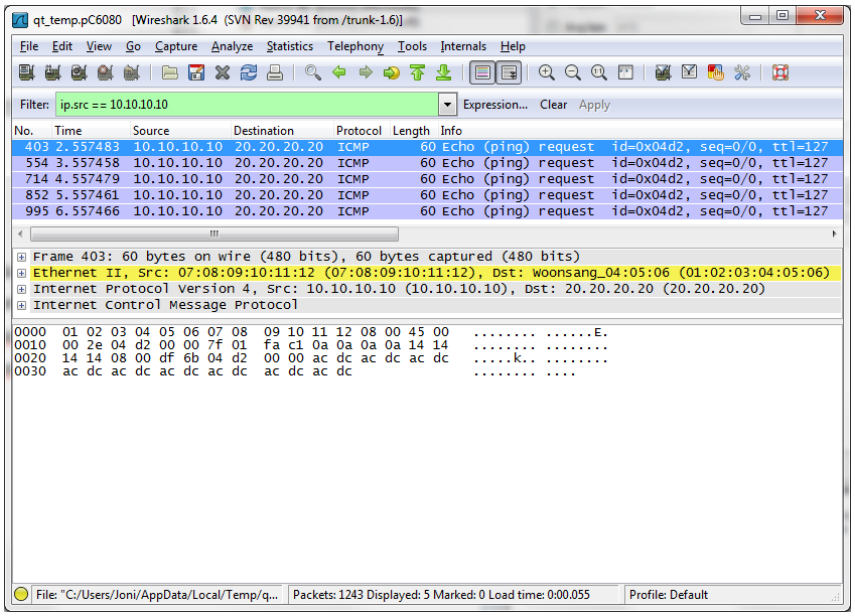
Streamin kaappaaminen onnistuu kahdella eri tavalla. Ensimmäinen tapa on käynnistää Wireshark ennen streamin lähetystä ja toinen tapa on käyttää hyväksi Ostinatoon suoraan rakennettua Wireshark-tukea. Tämä tapahtuu siten, että ennen streamin lähettämistä valitaan Statistics-kohdan napeista Start Capture. Kuvassa 37 näkyy tämä tilanne.




**Kuva 37.** Kaappaamisen aloittaminen

Jos kaappaaminen menee päälle, valitun portin Capture State -kohdassa pitäisi lukea On. Seuraavaksi lähetetään stream kappaleessa 5.3 mainitulla tavalla. Kun stream on lähetetty, näpäytetään Start Capture -napin vieressä olevaa Stop Capture -nappia. Tämän jälkeen Capture State -kohdassa pitäisi lukea Off. Kun lähetys ja

kaappaus on suoritettu, kaapatun streamin saa näkyviin näpäyttämällä View Capture Buffer -nappia. Portti, josta stream lähetetiin, pitää olla edelleen valittuna. Jos kaikki menee oikein, kuvan 38 mukainen Wireshark-ikkuna pitäisi avautua.




**Kuva 38.** Kaappaus Wiresharkissa

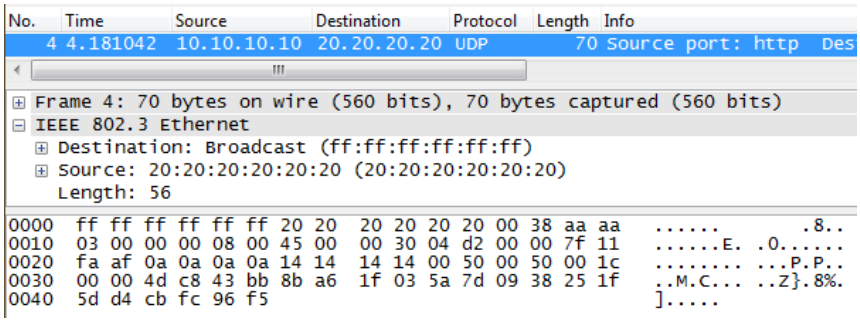
# 6 LABORATORIOTYÖ

## 6.1 Yleistä

Työssä simuloidaan erilaisia tietoliikenteen protokollia käyttämällä Ostinatoa, jolla luodaan suurin osa tarvittavasta liikenteestä. Kaikki analysointia varten tarvittava liikenne kaapataan Wiresharkilla. Työn toteutus onnistuu koneella kuin koneella, ainoana vaatimuksena on se, että koneelle on järjestelmänvalvojan oikeudet. Työssä tarvittavien ohjelmien: Ostinaton, TfGenin sekä Mcastin käyttäminen ei vaadi asentamista, vaan ne ovat purettavissa zip-paketeissa. Ainoastaan Wireshark vaatii erillisen asennuksen.

## 6.2 Tietoliikenneverkkojen lähetystapoja

Työssä broadcast-lähetyksen luomiseen käytettiin Ostinatoa. Yksi mahdollisista Ostinatossa suoritettavista tavoista on lähettää jo aiemmin kappaleessa 3.1.1 mainittuun Ethernet-verkon broadcast-osoitteeseen FF:FF:FF:FF:FF:FF streami. Pelkästään broadcast-liikenteen kaappaaminen onnistuu Wiresharkissa joko lisäämällä kaappausta aloittaessa Capture Filter -kohtaan ether broadcast tai kaappauksen ollessa käynnissä Display Filter -kohtaan eth.dst==ff:ff:ff:ff:ff:ff. Kuvassa 39 on Ostinatolla luotu ja Wiresharkilla kaapattu Ethernet-verkkoon lähetetty broadcast-lähetys.



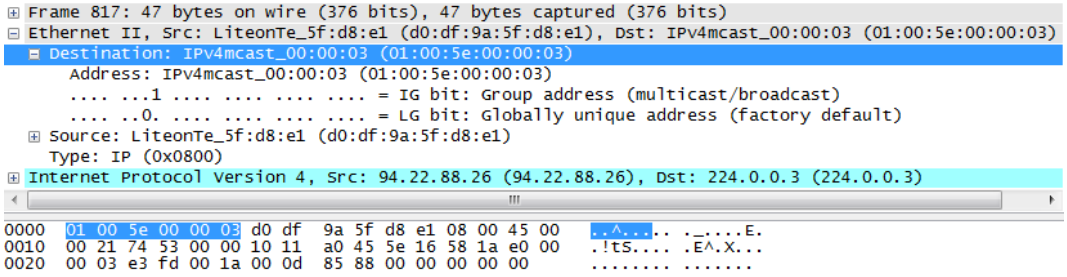

**Kuva 39.** Broadcast-lähetys

Kuvassa 39 näkyvä Destination-kohta kertoo, että kyseessä on broadcast-lähetys.

Työssä tarvittava multicast-liikenne luotiin kahdella eri tavalla. Multicast-liikennettä pystyy kaappaamaan Wiresharkilla kahdella eri tavalla: lisäämällä

ennen kaappauksen aloitusta Capture Filter -kohtaan ether multicast tai lisäämällä kaappauksen ollessa käynnissä Display Filter -kohtaan (eth.dst[0] & 1) && eth.dst!=ff:ff:ff:ff:ff:ff. Multicastin kaappaaminen onnistuu pelkästään ensimmäiselläkin parametrilla, mutta myös broadcast-liikenne kannattaa rajata pois siksi, että Ethernet broadcast on myös määriteltävissä Ethernet multicastiksi (broadcastissa ja multicastissa ensimmäisen tavun viimeinen bitti on 1).

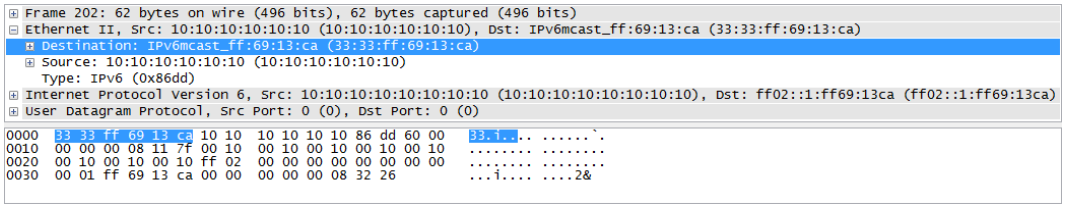
Ensimmäiseksi liikenne luotiin käyttämällä kappaleessa 4.3 mainittuja TfGen- ja Mcast-ohjelmia. Kuvassa 40 näkyy näiden avulla syntyneestä liikentestä Wiresharkilla kaapattu kehys.




**Kuva 40.** Ulkopuolisilla ohjelmilla luotu multicast

Kehyksessä oleva kohteen Ethernet-osoite sekä IPv4-osoite kertovat, että kyseessä on IPv4- protokollan mukaista multicast-lähetystä. Kohteen Ethernet-osoite on luotu käyttämällä IPv4-osoitetta. Kuten jo luvussa 3.1.2. aiemmin mainittiin, otetaan IP-osoitteen 3 viimeistä kohtaa, muutetaan ne desimaaliluvuista heksadesimaaliluvuiksi. Näin ollen alla olevasta IPv4-osoitteesta 224**.0.0.3** tulee 01:00:5e**:00:00:03**.

Seuraavaksi luotiin samantyyppinen multicast-lähetys käyttämällä Ostinatoa. Kuvassa 41 näkyy Ostinatolla luotu ja Wiresharkilla kaapattu Multicast-lähetys.

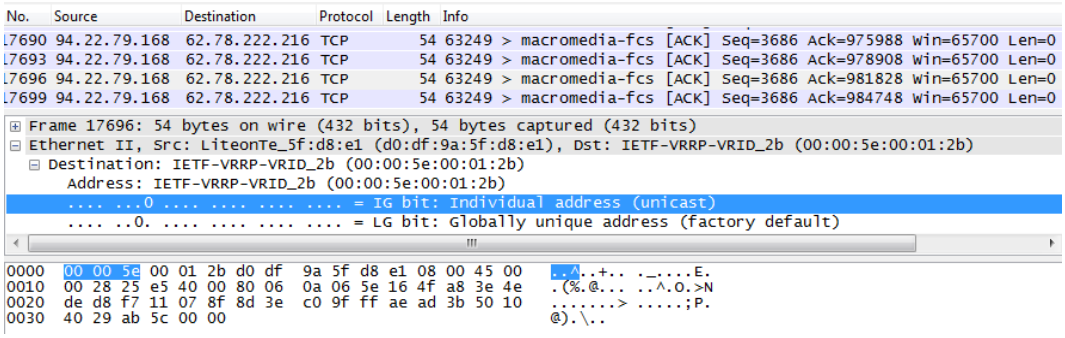



**Kuva 41.** Ostinatolla luotu multicast-lähetys

Yllä olevan kehyksen Ethernet-osoite sekä IP-osoite kertovat, että kyseessä on ollut IPv6-protokollan mukaista multicast-lähetystä. IPv6-osoitteessa sen kertoo alku ff02. Lähetykseen käytetty IPv6-osoite on saatu ottamalla viimeiset 24 bittiä unicast-osoitteesta ja lisäämällä ne etuliitteessä FF02::1:FFXX:XXXX/104 olevien X:ien paikalle. Esim. jos IPv6-osoite olisi ollut fe80::2fe:ff:fe**69:13ca**, niin uudeksi osoitteeksi saataisiin ff02::1:ff**69:13ca**. IPv6:n Ethernetin multicast-osoite on saatu siirtämällä IP-osoitteen ff02::1:**ff69:13ca** kaksi viimeistä paria Ethernet-osoitteen tyhjille paikoille. Näin Ethernet-osoitteeksi on saatu 33:33:**ff:69:13:ca**.

Työssä tarvittu Unicast-liikenne saatiin kaappaamalla verkosta tulevaa liikennettä. Tähän käy mm. ARP tai Nelosen tarjoama Ruutu.fi-palvelu. Pelkästään unicast-liikennettä saadaan kaapattua syöttämällä Wiresharkissa ennen kaappauksen aloittamista Capture Filter-kohtaan not broadcast and not multicast. Pelkkää ARP-liikennettä saa kaapattua lisäämällä kaappauksen ollessa käynnissä Display Filter-kohtaan arp.

Kuvassa 42 näkyy Wiresharkilla kaapattua liikennettä ruutu.fi-sivustolta.




**Kuva 42.** Verkosta kaapattua Unicast-liikennettä

Unicast-liikenteen tunnistaa 2 seikasta. Kohteena on tietty IP-osoite sekä Ethernetissä on ensimmäinen tavun viimeinen bitti on 0.

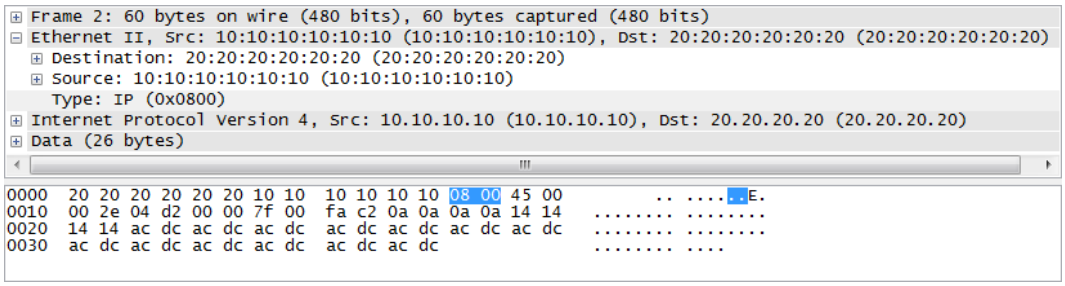
### 6.3 Lähiverkkostandardien simulointi

Ethernet II- ja IEEE 802.3-kehyksien lähettäminen toteutettiin Ostinatolla ja kaapattiin Wiresharkilla. Varsinaista suodinta ei näiden standardien kaappaamiseen löydy, vaan ne pitää kaapata käytetyn IP-osoitteen perusteella. Tätä helpottamaan molemmat lähetykset on tarkoitettu tehtäviksi muuten täysin samoilla asetuksilla, ainoastaan tutkittava standardi on muutettu. Se, mitä protokollaa käyttäen nämä lähettää, ei varsinaisesti ole mitään väliä, sillä molemmat standardit ovat samanlaisia sekä IPv4- että IPv6-protokollassa. Kuvassa 43 on myöhempää käyttöä varten tässä kappaleessa tutkitut standardit.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| DID | SID | Tunnus | Tieto-osa | | | | | Ethernet II-kehys |
| DID | SID | Pituus | DSAP | SSAP | CTL | Tieto-osa | | IEEE 802.3 LLC |
| DID | SID | Pituus | DSAP | SSAP | CTL | SNAP | Tieto-osa | IEEE 802.3 SNAP |

**Kuva 43.** Kehykset (Granlund 2007, 268)

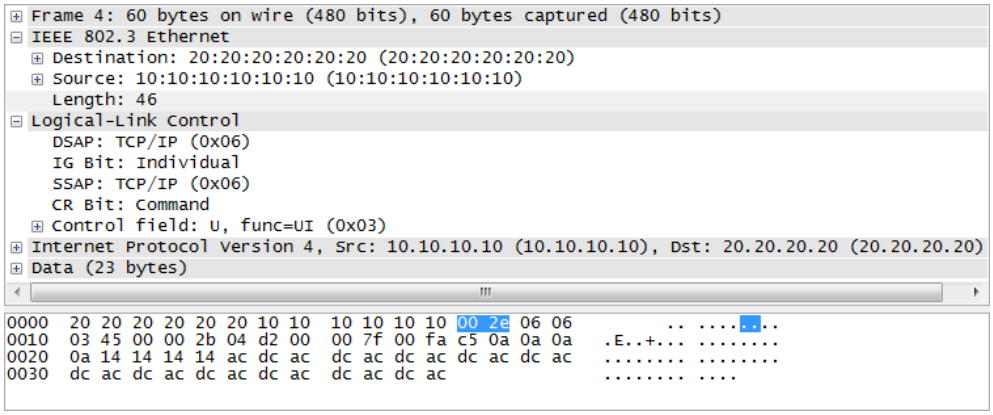
Tätä kohtaa varten tehdyt lähetykset on lähetetty IPv4-protokollaa käyttäen. Kuvassa 44 on Ethernet II-standardilla lähetetty kehys.




**Kuva 44.** Ethernet II-kehys

Kun kuvassa 44 olevaa lähetettyä kehystä vertaa kuvassa 43 olevaan Ethernet II-kehykseen huomataan, että molemmista löytyy samat komponentit. Destination=DID, Source=SID, tunnus=Type sekä Tieto-osa on lopussa oleva Data-kenttä. Tyyppi-kentässä oleva arvo 0x0800 kertoo, että kyseessä oleva viesti sisältää IPv4-protokollan. Jos viesti sisältäisi IPv6-protokollan, olisi tyyppi-kentässä arvona 0x86DD.

Seuraavaksi lähetettiin IPv4-protokollan sisältävä IEEE 802.3-standardin mukainen viesti, joka näkyy kuvassa 45.



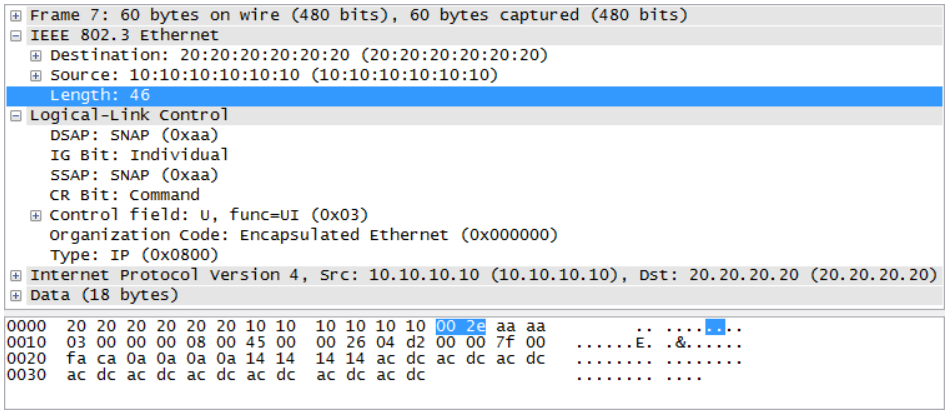

**Kuva 45.** Ensimmäinen Ostinatolla luotu IEEE 802.3-viesti.

Kun vertaa kuvassa 45 olevaa lähetettyä kehystä kuvassa 43 olleeseen IEEE 802.3 LLC-kehykseen, huomataan selvää samankaltaisuutta. Samalla voi verrata kuvassa 43 ylempänä olevaa Ethernet II-kehystä alempaan IEEE 802.3 LLC-kehykseen. Molemmissa on sama DID- sekä SID-kenttä, mutta erottavana tekijänä on näiden

jälkeen tuleva pituus-kenttä. Sama kenttä löytyy myös kuvassa 46 nimellä Length. Length kentässä tuleva arvo tulee, kun vähentää kehyksen koosta vastaanottaja-, lähettäjä-, pituus- sekä lopussa olevan tarkistussumma-kentän koot. Lähetetyn kehyksen minimikoko on 64 tavua, vastaanottaja- ja lähettäjä-kentän 6 tavua, pituuskentän 2 tavua sekä tarkistussumma-kentän 4 tavua. Näin pituuskentän arvoksi saadaan 64-6-6-2-4=46.

Tässä IEEE 802.3-kehyksessä lähetetyn LLC-sanoman sisältämät DSAP ja SSAP kertovat, että sovelluksen tunnisteena toimii TCP/IP. IG Bit kertoo sen, että kyseessä on suoraosoitus (Individual) ja CR Bit sen, että kyseessä on komento (Command). Control-kentässä oleva 0x03 kertoo sen, että kyseessä on yhteydetöntä dataa.

Kuvassa 46 on toinen IEEE 802.3-standardin sisältävä viesti, mutta mukaan on lisätty SNAP-protokolla.




**Kuva 46.** Toinen Ostinatolla luotu IEEE 802.3-viesti.

Kun vertaa keskenään kuvassa 43 olevaa IEEE 802.3 LLC- sekä IEEE 802.3 SNAP-kehyksiä huomataan, että ainoa ero LLCn sekä SNAPin välillä on ohjauskentän (CTL) jälkeinen SNAP-kenttä.

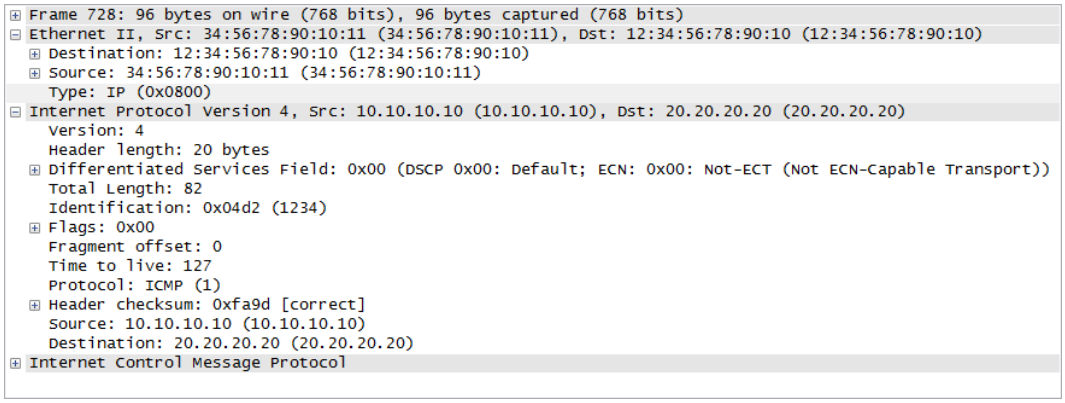
SNAP-protokolla lisätään LLC-protokollassa oleviin DSAP- ja SSAP-kohtiin sovelluksen tunnisteiksi. SNAPin sisältävän LLC-protokollan tunnisteeksi tulee 0xAA-AA-03, joista AA:t tulevat DSAP:sta sekä SSAP:sta. 03 tulee Control-kentän arvosta 03. SNAPin tunnisteeksi tulee 5 merkin mittainen otsikko

0x00-00-00 08-00. Tässä 00-00-00 tarkoittavat Ethernet-kehystä sekä lopussa oleva 08-00 Ethernet II-kehyksessä olevaa tyyppi-kenttää.

Huomion arvoista myös näissä kaikissa lähetyksissä on data-kentän pieneneminen. Protokollien lisääntyessä, mutta kehyksen koon pysyessä samana, data-kenttien ero suurimman ja pienimmän välillä on 8 tavua.

### 6.4 IP-protokollien simulointi

IPv4- ja IPv6-protokollien sisältävien viestien lähettäminen toteutettiin Ostinatolla. Kuten lähiverkkostandardeja sisältäviä viestejä lähettäessä, myös IP-protokollia simuloitaessa pyritään mahdollisimman paljon samanlaisten viestien lähettämiseen. Näin mahdolliset erot tulevat paremmin selville. Lähetyksessä käytettäväksi standardiksi valikoitiin Ethernet II, koska se hyväksyy kaikki tässä kohtaa tarvittavat lähetykset. Kuvassa 47 on Ostinatolla lähetetty IPv4-protokollan ja Ethernet II-standardin sisältävä viesti.




**Kuva 47.** IPv4-protokollan sisältävä viesti

Kuvassa 47 näkyvässä viestissä on muutama kohta, mikä kertoo että kyseessä on IPv4-protokollan sisältävä viesti. Koska viestiin on otettu mukaan myös Ethernet II-standardi, ensimmäisenä huomataan Type-kentässä oleva 0x0800, joka on IPv4-protokollan tunnus. Kenttä, johon tässä lähetyksessä pitää kuitenkin keskittyä, on lähettäjän ja vastaanottajan jälkeen tuleva IP-protokolla kenttä. Tässä kentässä määritellään tässä viestissä olleen IPv4-otsakkeen ominaisuudet. Kun vertaa kuvaa 47 luvussa 3.3 olleeseen kuvaan 5, huomataan, että teoria tukee käytäntöä.

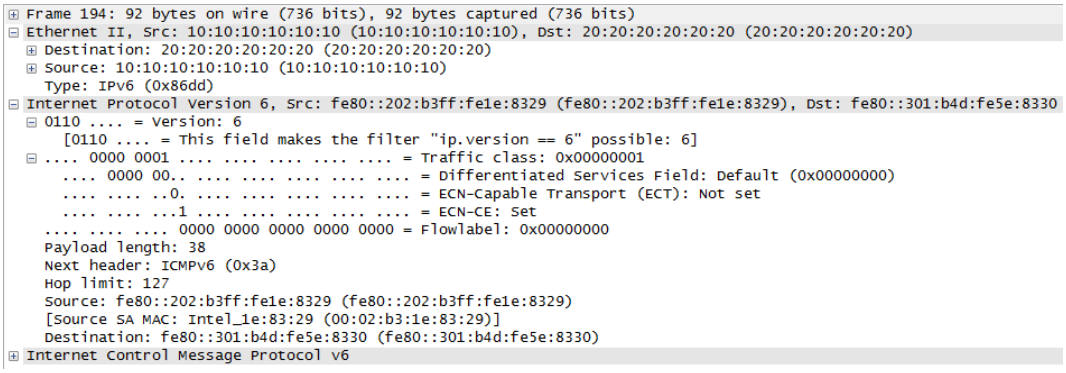
Kaikki kuvassa 5 olevat kentät löytyvät myös lähetetyn viestin IPv4-otsakkeesta. Lähetettävän IPv4-otsakkeen arvot saa asettaa mieleisekseen Ostinatossa.

Header Length-kohdassa oleva arvo 20 on viestissä olevan IP-otsakkeen koko. Ostinatossa pystyy lähettäessä muuttamaan tämän kohdan arvoa, mutta tähän lähetykseen käytettiin valmiiksi ollutta arvoa 5, joka on samalla myös pienin mahdollinen arvo. Tähän kohtaan tullut arvo lasketaan lausekkeella 5x32=160-bittiä/8=20 tavua. Tässä lausekkeessa oleva arvo johtuu pituus-kentän 32-bittisyydestä. Seuraavana oleva jakolasku 160/8 taas johtuu siitä, että 8 tavua on 1 bitti.

Differentiated Services Field-kohdan arvo 0x00 kertoo sen, että lähetetyllä otsakkeella ei ole mitään erikoisarvoja vaan se on oletusarvoinen. Total Length-kohta kertoo sekä IP-otsakkeen että sen jälkeen tulevan datan koon, joka tässä tapauksessa on 20+62=82 tavua. Pienimmillään arvo voi olla joko 20 tavua (otsake 20+data 0) ja suurimmillaan 65,535 tavua. Identification-kohtaan annettu arvo 0x04d2 on mahdollista fragmentoitumista varten.

Protokolla-kohdassa näkyy IP-otsaketta seuraava mahdollinen data-kohdassa näytettävä protokolla. Ennen lähetystä Ostinatossa valittiin L4-kohdassa protokollaksi ICMP. Muita mahdollisia Ostinatossa valittavia protokollia olisivat olleet esim. IGMP, TCP tai UDP. Viimeisenä ennen hyötykuormia tulevat vastaanottajan ja lähettäjän IP-osoitteet.

Kuvassa 48 on Ostinatolla lähetetty IPv6-protokollan ja Ethernet II-standardin sisältävä viesti.




**Kuva 48.** IPv6-protokollan sisältävä viesti

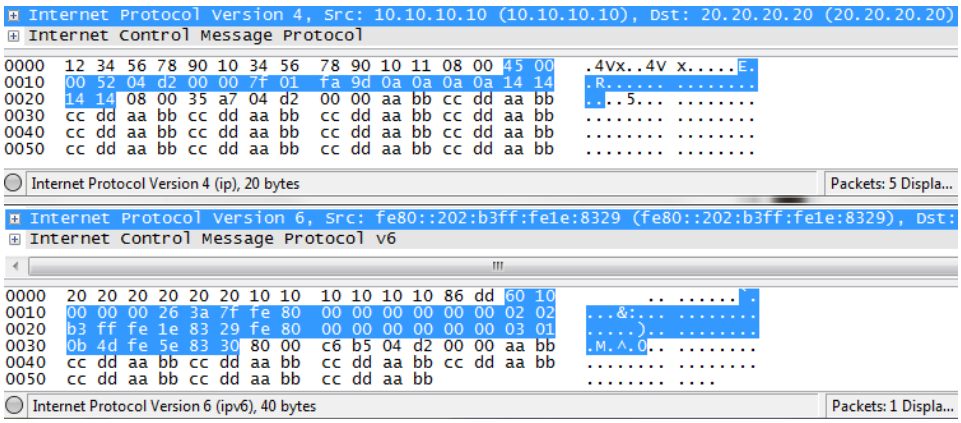
Ensimmäinen kohta kuvassa 48, joka kertoo lähetyksen oleva IPv6-protokollaa, on tyyppi-kentässä oleva arvo 0x86DD, joka on IPv6-protokollan tunnus. Kuten IPv4-protokollan avauksessakin, on tässä hyvä verrata lähetettyä kehystä kappaleessa 3.3 olevaan kuvaan 6. Huomataan, että tässäkin kohdassa teoria tukee käytäntöä. Seuraavaksi käsiteltävän IPv6-otsakkeen kaikkia arvoja pystyy asettamaan mieleisekseen Ostinatossa.

Ensimmäinen kohta IPv6-otsakkeessa on versio-kohta missä kerrotaan bittikuviolla 0110, että kyseessä on IPv6-protokolla. Tätä bittikuviota hyväksikäyttämällä on mahdollista suodattaa Wiresharkin avulla pelkkää IPv6-protokollaa sisältävää liikennettä. Seuraavaksi tuleva Traffic class-kohta vastaa IPv4-protokollassa ollutta Differentiated Services Field-kohtaa. Tässä ensimmäiset 6 bittiä vastaa IPv4-protokollassa olevaa DSCP-protokollaa ja viimeiset 2 bittiä vastaavat IPv4-protokollassa olevaa ECN-protokollaa.

Payload length-kohdan arvo tulee IPv6-otsaketta seuraavan hyötykuorman koosta, joka tässä tapauksessa on 38 tavua. Next Header-kohta toimii samalla tavalla kuin IPv4-protokollassa oleva Protocol-kohta, ilmoittaen seuraavan otsakkeen/protokollan. Tässä tapauksessa seuraavaksi otsakkeeksi on valittu Ostinatossa streamin asetuksista ICMPv6. Hop Limit-kohta kertoo, kuinka monen aliverkon tai linkin läpi paketti voi kulkea. Samalla se vastaa IPv4-protokollassa olevaa Time to Live-kohtaa. Jos tämä viesti kulkisi oikeassa verkossa, saisi se kulkea 127

aliverkon tai linkin läpi ennen kuin reititin poistaa sen liikenteestä. Jokaisen siirtymän jälkeen tätä kohtaa vähennetään yhdellä. Kuten IPv4-protokollassa, myös IPv6-protokollassa viimeisenä ennen hyötykuormaa tulevat lähettäjän ja vastaanottajan IP-osoitteet.

IPv4- ja IPv6-otsakkeita keskenään verrattaessa huomaa näissä olevat suurimmat erot. IPv6-protokollassa ei ole otsakkeen pituuden määritystä. Koon puolesta IPv4-otsakkeen sisältävät paketit ovat pienempiä kuin IPv6-otsakkeen sisältävät paketit. Kuvassa 49 on tämä esitettynä.

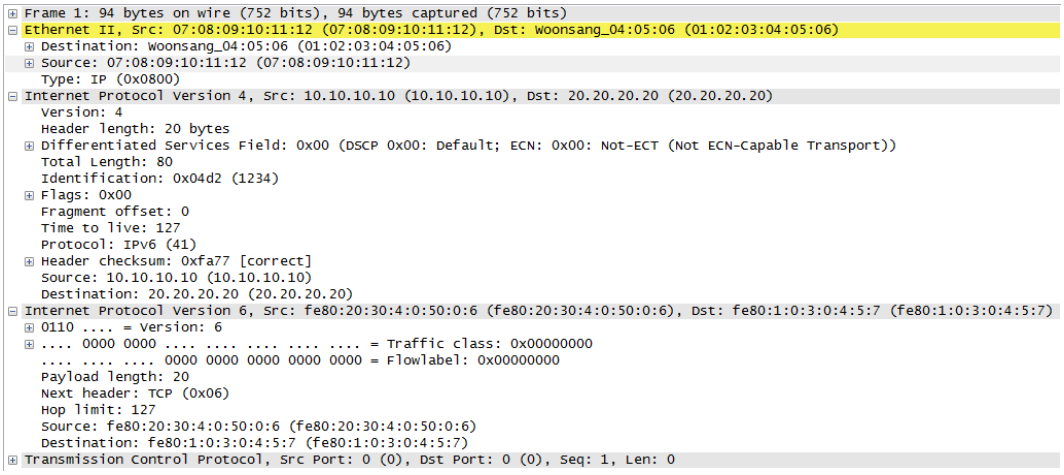



**Kuva 49.** Kokojen vertailu

Kuten kuvasta 49 huomaa, on IPv6-otsake 2 kertaa niin suuri kuin IPv4-otsake. On kuitenkin hyvä muistaa, että sen minkä IPv6 häviää koossa, voittaa se esim. osoitteiden suurella määrällä (IPv4-protokollassa 32-bittinen osoitteisto, IPv6-protokollassa 128-bittinen osoitteisto) sekä tarkistussumman puuttumisella. Myös hyötykuorman mahdollisessa koossa voittaa IPv6-protokolla IPv4-protokollan (IPv6-protokollassa mahdollisuus yli 4 000 000 000 oktettiin verrattuna IPv4-protokollan 65 535 oktettiin). Tätä ei kuitenkaan pysty simuloimaan Ostinatolla.

Viimeisenä IP-protokollien simuloinnissa keskityttiin tunnelointiin, mistä valikoitui 6over4-tunnelointi.

Kuvassa 50 näkyy Ostinatolla lähetetty Ethernet II-standardin sisältävä 6over4-kehys.




**Kuva 50.** 6over4-tunneloinnin simulointi

Kuten kuvassa 50 näkyy, 6over4-tunneloinnissa on kyse siitä, että IPv4-otsakkeen perään, ennen suoritettavaa protokollaa (tässä tapauksessa TCP-protokollaa), asetetaan IPv6-otsake. MAC-osoitteiden jälkeen tuleva Type-kohta kertoo kehyksen tyypiksi IPn. IPv4-otsakkeessa oleva Total length-kohta oleva arvo 80 kertoo kaikkien mukana olevien protokollien koon (IPv4=20, IPv6=40 ja TCP=20). Protocol-kohta kertoo seuraavaksi tulevan protokollan, joka tässä tunneloinnissa on IPv6-protokolla. IPv6-protokollaa seuraavan TCP-protokollan taas löytää IPv6-otsakkeen sisältä löytyvästä Next header-kohdasta.

## 6.5 VLAN-viestien simulointi

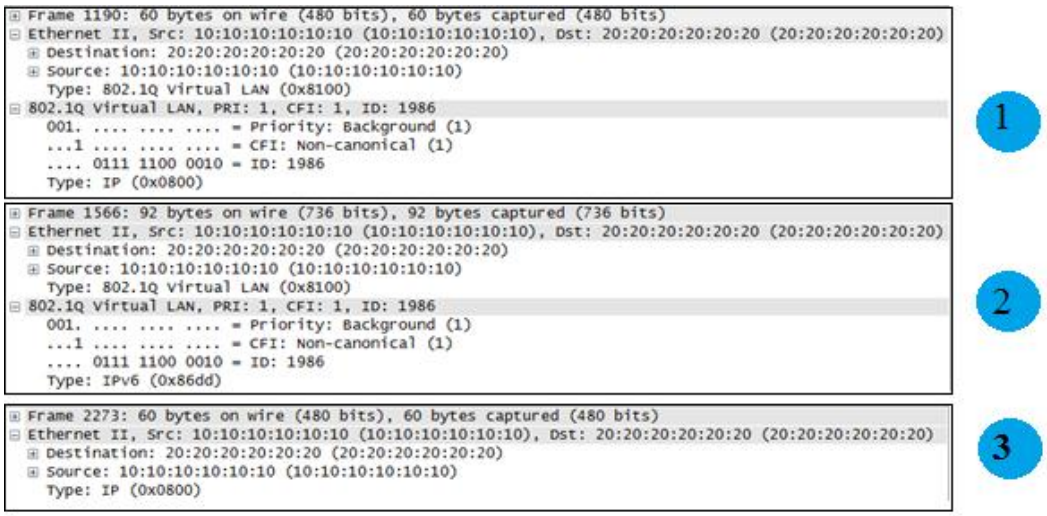
VLAN-tunnisteen sisältävien viestien lähettäminen toteutettiin Ostinatolla. Kuten myös aiemmissa kohdissa, myös tässä kohdassa pyritään 2 samanlaisen viestin lähettämiseen. Tässä luvussa emme keskity eri tapoihin toteuttaa virtuaalisia lähiverkkoja, vaan keskitymme vain niissä liikkuviin viesteihin.

Kuvassa 51 on myöhempää käyttöä varten tässä luvussa tutkittu VLAN-tunniste.

| Protokollatunniste | Prioriteetti | CFI | VLAN ID |
|---|---|---|---|
| 16-bittiä | 3-bittiä | 1-bitti | 12-bittiä |

**Kuva 51.** VLAN-tunniste

VLAN-viestit simuloitiin sekä IPv4- että IPv6-protollilla, jotta myös näiden protokollien mahdollisesti aiheuttamat erot nähtäisiin. Kuvassa 52 näkyy lähetyt VLAN-viestit.




**Kuva 52.** Simuloidut VLAN-viestit

Numerolla 1 merkitty kehys on IPv4-protokollan sisältävä VLAN-tunnistettu viesti. Numerolla 2 merkitty kehys on IPv6-protokollan sisältävä VLAN-tunnistettu viesti ja viimeisenä oleva numerolla 3 merkitty kehys on IPv4-protokollan sisältävä VLAN-tunnisteeton viesti.

Kuten kuvassa 51, on myös kuvassa 52 olevassa simuloiduissa VLAN-viesteissä Type-kohta (protokollatunniste). Tässä kohdassa oleva 802.1Q Virtual LAN kertoo, että lähetetty kehys sisältää VLAN-tunnisteen. Priority-kohta (prioriteetti) varaa VLAN-tunnisteeseen varatusta bitti-määrästä 3-bittiä (001. …. …. ….). Simuloiduissa viesteissä tätä kohtaa ei muutettu, samoin kuin ei seuraavaa CFI-kohtaakaan. CFI-kohta varaa VLAN-tunnisteesta 1-bitin (…1 …. …. ….). Viimeisenä tuleva ID-kohta varaa jäljelle jääneet 12-bittiä (…. 0111 1100 0010).

Kun näitä kehyksiä vertailee keskenään huomataan, että eri protokollan sisältävän viestin erottaa toisesta vain VLAN-viestin sisällä olevan Type-kohdan arvon avulla. Jos VLAN-viesti sisältää IPv4-protokollan, on Type-kohdan arvo 0x0800 (IP) ja jos VLAN-viesti sisältää IPv6-protokollan, on Type-kohdan arvo 0x86dd (IPv6).

### 6.6 Työohje

Tämä harjoitus tehdään pääasiassa käyttämällä Ostinatoa luomaan tarvittavaa liikennettä ja kaappaamalla sitä Wiresharkilla. Tarvitsette harjoituksen tekemiseen yhden tietokoneen.

Työhön kuuluu seuraavat asiat:

Broadcast-, Multicast- ja Unicast-liikenteen luominen ja kaappaaminen.

Ethernet II- ja IEEE 802.3-kehysten sisältävien viestien luominen, kaappaaminen ja vertaaminen.

1. IPv4- ja IPv6-protokollien sisältävien viestien luominen, kaappaaminen ja vertaaminen. Myös 6over4-viestin luominen, kaappaaminen ja analysoiminen.
2. Tunnisteellisen VLAN-viestin ja tunnisteettoman VLAN-viestin luominen, kaappaaminen ja vertaaminen.

HUOM! Ryhmänne numero tulee näkyä lähettäjän MAC-osoitteessa kaikissa Ostinatossa lähetettävissä viesteissä (esim. jos olette ryhmä 4, niin MAC-osoitteen tulee loppua numeroihin 04). Samoin ryhmän numero tulee näkyä 2.b:ssä Mcast- ja TfGen-ohjelmiin asetettavassa UDP-portissa (esim. jos olette ryhmä 4, niin portin numeroksi tulee 4444).

Yksityiskohtaiset ohjeet:

1. Avatkaa tietokone ja kirjautukaa sisään järjestelmänvalvojana. Avatkaa Ostinato ja Wireshark.
2. Broadcast, Multicast & Unicast

   a. Luokaa broadcast-viesti käyttämällä Ostinatoa ja kaapatkaa se suotimella, jonka selvititte esitehtävissä.
   b. Luokaa multicast-viesti käyttämällä TfGen- ja Mcast-ohjelmia ja kaapaatkaa se suotimella, jonka selvititte esitehtävissä. Luokaa samanlainen viesti käyttämällä Ostinatoa.
   c. Kaapatkaa Unicast-liikennettä suotimella, jonka selvititte esitehtävissä. Tarvittavan liikenteen voi luoda esim. katsomalla videota Yle:n Areena- tai Nelosen Ruutu.fi-palvelusta.
   d. Tallentakaa saamanne kaappaukset.
3. IEEE 802.3 & Ethernet II
   a. Luokaa viesti, joka sisältää IEEE 802.3 LLC-protokollan ja kaapatkaa se.
   b. Luokaa täysin samanlainen viesti, joka sisältää Ethernet II-protokollan ja kaapatkaa se.
   c. Asettakaa kaappaukset rinnakkain ja vertailkaa näitä toisiinsa. Mikä erottaa Ethernet II:n sisältävän viestin sekä IEEE 802.3 sisältävän viestin toisistaan?
   d. Tallentakaa saamanne kaappaukset.
4. IPv4 & IPv6
   a. Luokaa viesti, joka sisältää IPv4-protokollan ja kaapatkaa se.
   b. Luokaa täysin samanlainen viesti, joka sisältää IPv6-protokollan ja kaapatkaa se.
   c. Miten nämä eroavat toisistaan?
   d. Luokaa viesti, joka sisältää IP6over4-tunneloinnin. Analysoikaa sitä. Miten eroaa normaalista viestistä?
   e. Tallentakaa saamanne kaappaukset.
5. VLAN
   a. Luokaa tunnisteellinen (tag) VLAN-viesti ja kaapatkaa se.
   b. Luokaa tunnisteeton (untagged) VLAN-viesti ja kaapatkaa se.
   c. Vertailkaa näitä toisiinsa. Miten ne eroavat toisistaan?
   d. Tallentakaa saamanne kaappaukset.

# 7 YHTEENVETO

Opinnäytetyö onnistui mielestäni asetetun tavoitteen mukaisesti. Aluksi suunnitelmissa oli kokonaisen testiympäristön luominen käyttäen reitittimiä ja erilaisia koneita luomaan tarvittavaa liikennettä. Lopulta kuitenkin päädyttiin luomaan kevyt sekä helppo simulointiympäristö, joka ei vaadi käytettävältä laitteistolta paljoa vaan ohjelmat saa toimimaan myös vanhemmilla koneilla. Uskon ja toivon tämän helpottavan myös opiskelijoiden tarttumista näihin ohjelmiin hanakammin, sillä heidän ei tarvitse uusia laitteistoaan vain simuloidakseen näillä ohjelmilla haluamaansa liikennettä. Näen myös mahdollisena käyttökohteena erilaiset tietoverkkolaitteet, joita halutaan testata esim. lähettämällä tietyn protokollan sisältäviä viestejä.

Opinnäytetyön pohjalta tehty harjoitustyö täyttää myös mielestäni sille asetetut tavoitteet antaen opiskelijalle mahdollisuuden kokeilla käytännössä aiemmin opittua teoriaa. Työssä vaadittuja protokollia, standardeja sekä erilaisia tietoverkoissa olevia lähetystapoja pystytään simuloimaan ilman suurempaa vaivaa. Tämän pitäisi helpottaa oppimisprosessia, sillä suurin osa ajasta ei mene erilaisten asetusten tai koneiden kanssa taistelemiseen, vaan opiskelija saa keskittyä rauhassa aikaansaatujen tulosten analysoimiseen.

Mahdollisina kehityskohteina näen erilaisten protokolla-ketjujen luomisen tällä simulointiympäristöllä sekä erilaisten laitteiden lisäämisen, joilla voidaan luoda aiemmin simuloituja viestejä. Näistä protokolla-ketjuista näen yhtenä mahdollisuutena esimerkiksi DHCP-keskustelun simuloimisen. Tämä ei kuitenkaan ole tällä hetkellä mahdollista, sillä Ostinatosta puuttuu vielä DHCP-protokolla kokonaan. Mutta sivuilla olevan tiedon mukaan sen pitäisi lähitulevaisuudessa saapua tähän ohjelmaan. Näen yhtenä mahdollisena laajennusmahdollisuutena myös mobiilidatan, jota voitaisiin jollain tavalla kaapata Wiresharkilla ja analysoida.

# LÄHTEET

## Kirjat

Granlund, K. 2007. 1. painos. Tietoliikenne. Porvoo. WS Bookwell

Hakala, M., Vainio, M. 2005. 1. painos. Tietoverkon rakentaminen. Porvoo. WS Bookwell.

## Elektroniset julkaisut

Carpenter B., C. Jung. 1999. Transmission of IPv6 over IPv4 Domains without Explicit Tunnels. Viitattu 26.1.2012. http://tools.ietf.org/html/rfc2529

Crawford, M., Fermilab. 1998. Transmission of IPv6 Packets over Ethernet Networks. Viitattu 26.1.2012. http://tools.ietf.org/html/rfc2464

Deering, S. 1989. Host Extensions for IP Multicasting. Viitattu 26.1.2012. http://tools.ietf.org/html/rfc1112

Harju, J. 2005. Verkkotekniikan jatkokurssi. Multicast. Viitattu 26.1.2012. http://www.cs.tut.fi/kurssit/TLT-2600/t2005-06/TLT-2600-VTJ-luento-multicast.pdf

Hinden, R., Nokia, Deering, S. & Cisco Systems. 2006. IP Version 6 Addressing Architecture. Viitattu 11.3.2012. http://tools.ietf.org/html/rfc4291#section-2.7

Hornig, C. 1984. A Standard for the Transmission of IP Datagrams over Ethernet Networks. Viitattu 26.1.2012. http://tools.ietf.org/html/rfc894

Mogul, J. 1984. Broadcasting Internet Datagrams. Viitattu 26.1.2012. http://tools.ietf.org/html/rfc919

Ramakrishnan, K., Floyd S. & Black, D. 2001. The Addition of Explicit Congestion Notification (ECN) to IP. Viitattu 25.3.2012. http://tools.ietf.org/html/rfc3168