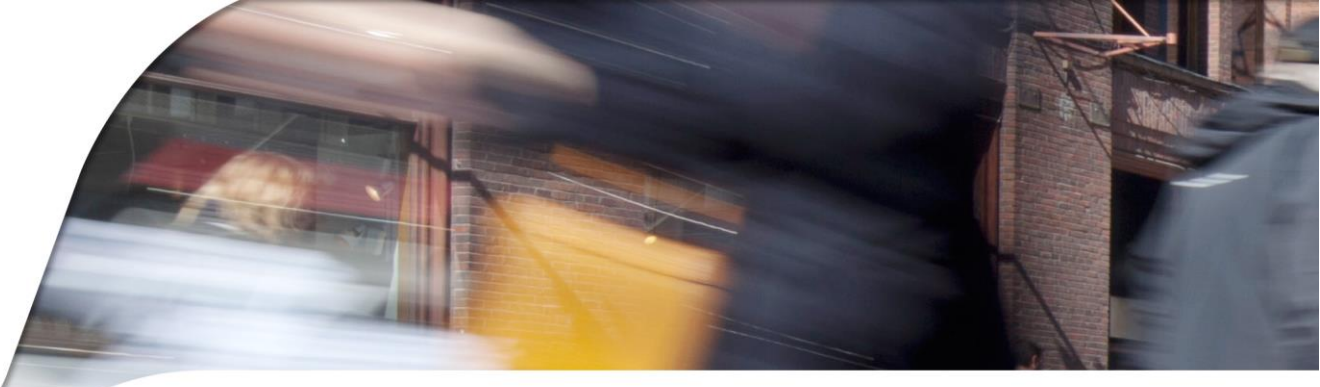


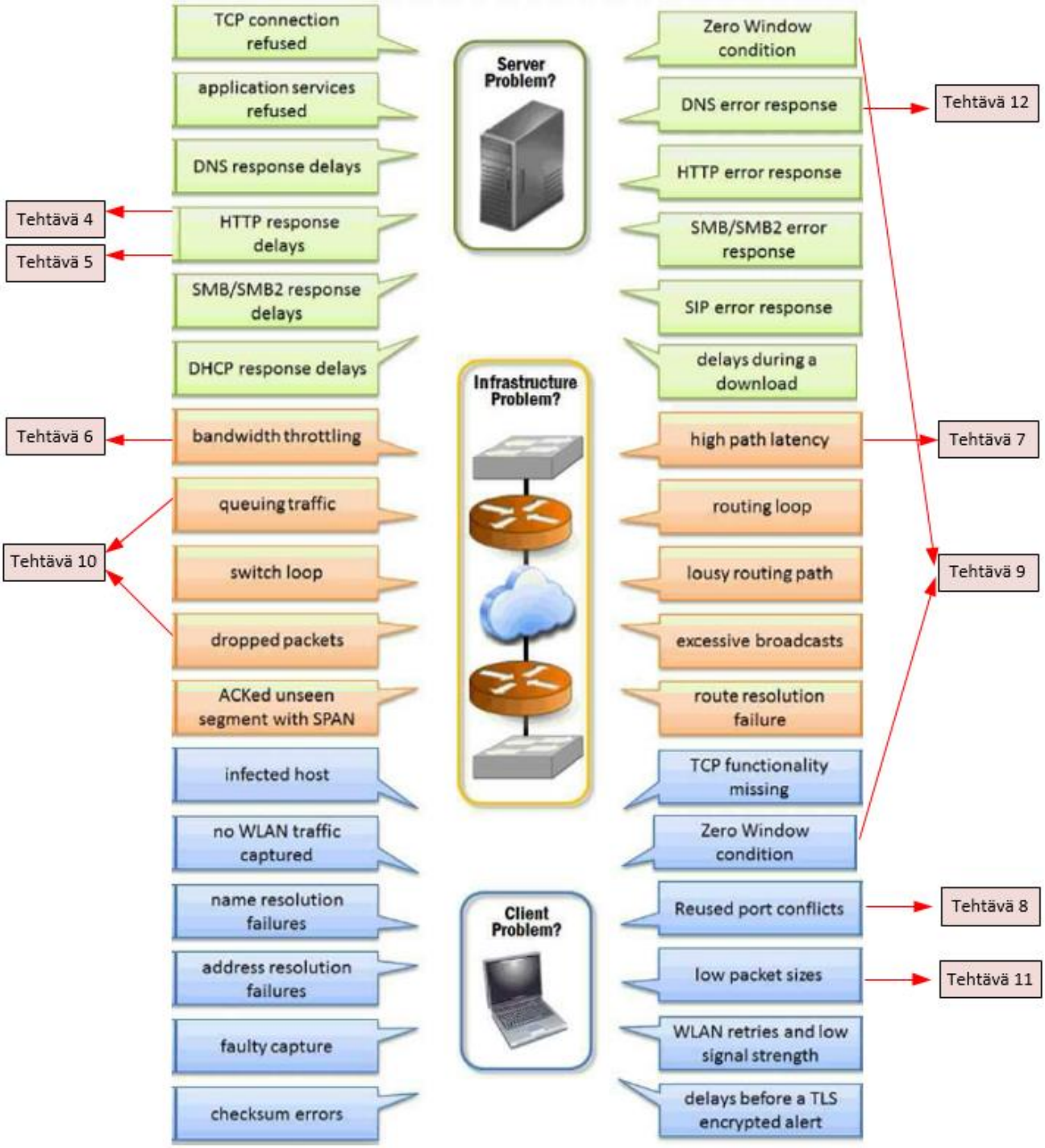

WIRESHARK

SISÄLLYSLUETTELO

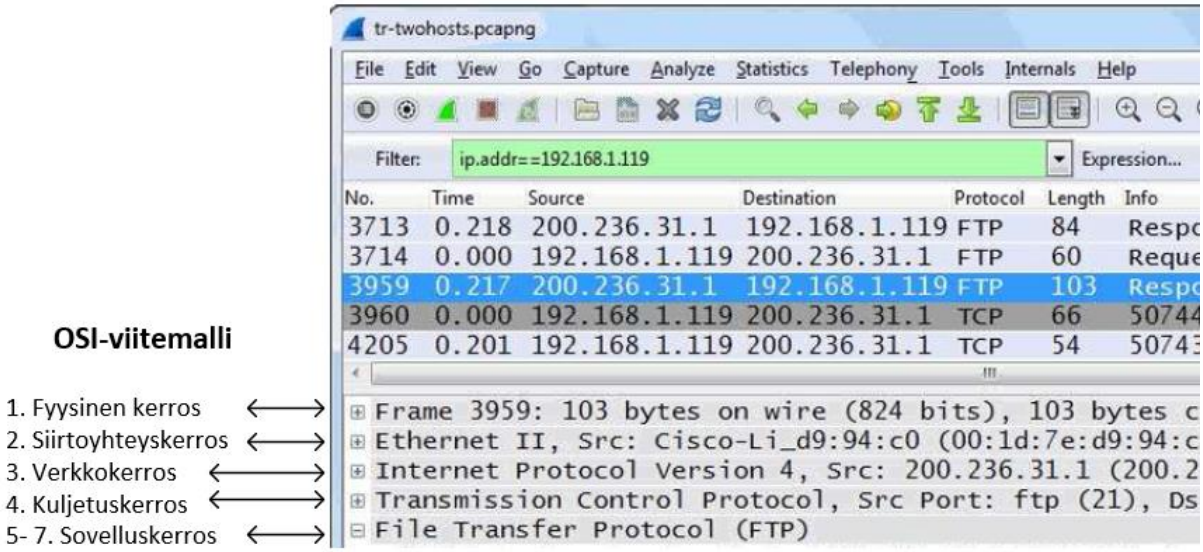
Wireshark ominaisuudet ja hyödyt

Wireshark (ent. Ethereal) on kätevä työkalu verkko-ongelmien analysoinnissa ja protokollien tutkimisessa. Ennen kaikkea Wiresharkin avulla IT-asiantuntija näkee mitä verkossa tapahtuu pakettitasolla. Parasta Wiresharkissa on sen mahdollistama "verkkovikojen" perusteellinen selvitys. Monesti kuulee yrityksissä ja organisaatioissa valitettavan, että "verkko on hidas" vaikka todellisuudessa vika on jossain muualla kuin verkkolaitteissa.

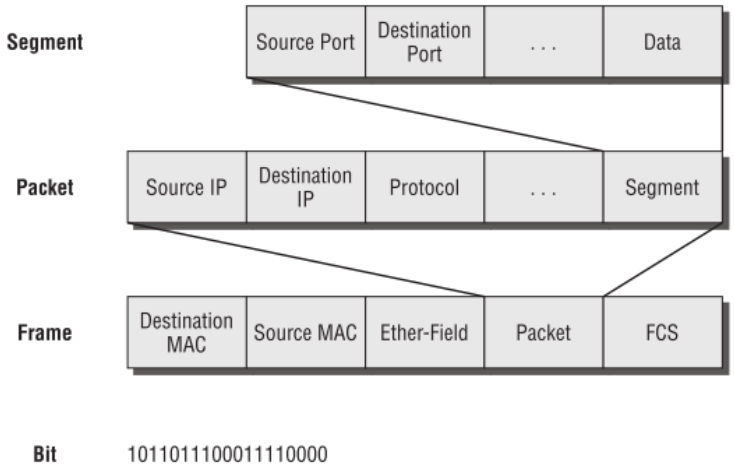
Alla olevassa kuvassa on esitetty erilaisia "verkkovikoja", joista vian oranssissa laatikoissa esiintyvät ongelmat todellisuudessa ovat verkkolaitteista (Infrastructure Problem?) johtuvia ongelmia, vihreissä laatikoissa olevat ongelmat palvelinpään ongelmia (Server Problem?) ja sinisissä laatikoissa olevat ongelmat työasemapään ongelmia (Client Problem?). Tässä oppimateriaalissa käydään läpi Wiresharkin laajoja ominaisuuksia verkkovikojen analysoinnissa ja paikantamisessa. Se miten eri tehtävissä selvitellään erilaisia ongelmia on esitetty alla.



Aluksi käydään kuitenkin läpi Wiresharkin perusominaisuuksia ja pakettinäkymää, jotta ne tulisivat ensin tutuksi. Allaolevassa kuvassa on esitetty miten yksittäisen Ethernet-kehykseen pääsee pureutumaan pakettilistanäkymässä ja miten siinä olevat tiedot on esitetty OSI-viitemallin (tai TCP/IP-mallin) mukaisessa järjestyksessä ylhäältä alas.



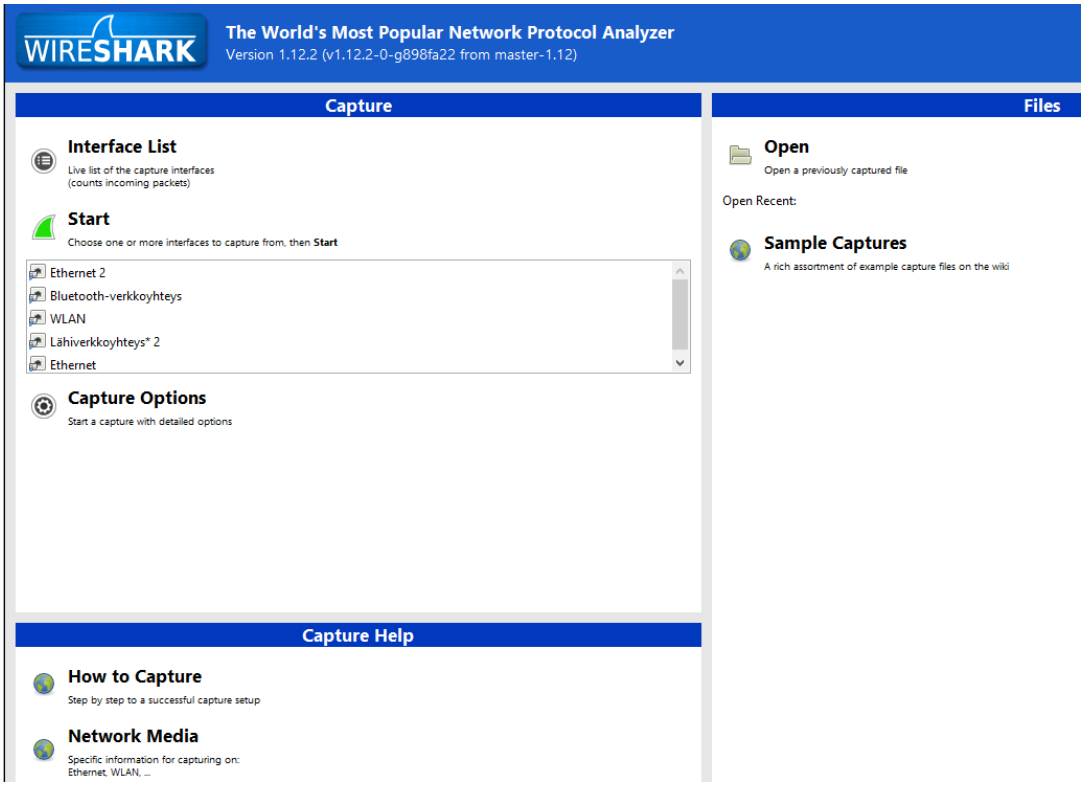
Wireshark esittää kaapatut (tai avatun trace-tiedoston) bitit, jotka muodostavat Ethernet-kehyksen. Ethernet-kehys pitää sisällään IP paketin, joka puolestaan pitää sisällään TCP/UDP segmentin tiedot, kuten lähde- ja kohdeporttinumerot (esim. TCP/80 = http)



Tehtävä 1: Verkkoliikenteen kuuntelu

Wireshark osaa kuunnella liikennettä joka saapuu tai lähtee tietokoneen verkkokortilta.

1. **Käynnistä Wireshark**. Wiresharkilla kestää hetken käynnistyä. Kun se on käynnissä, työkalurivillä on ensimmäisenä nappina valikkona Capture-valikko, josta löydät listan verkkokorteista "inteface list". Tässä voit valita millä verkkokorteilla voit kaapata liikennettäsi. Options-napilla pääset säätelemään kaappauksen asetuksia ja **Start** aloittaa kaappaamisen suoraan.
2. **Paina start ja testaa kuinka Wireshark lähtee kuuntelemaan verkkokortin liikennettä.**
3. **Pysäytä kuuntelu hetken kuluttua ja sulje ohjelma.**
4. **Käynnistä ohjelma uudelleen**. Tällä kertaa käy tarkistelemassa "Capture Options"-napin kautta kurkkaamassa asetuksia. Tutustu "Capture Options" valikon asetuksiin

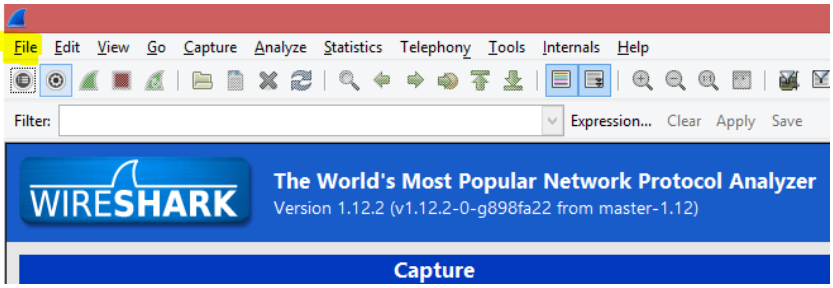


Wireshark kaappaa kaiken saapuvan ja tulevan Ethernet-kehykset verkkokortilta ellei toisin kaappausasetuksista määritetä. Wireshark ohjelmassa käytetään termejä kehys ja paketti ristiin rastiin. Pakettinäkymästä pääsee pureutumaan OSI-viitemallin 2. kerroksella välitettävät kehykset (engl. frames) ja 3. kerroksella reititettävät IP paketit.

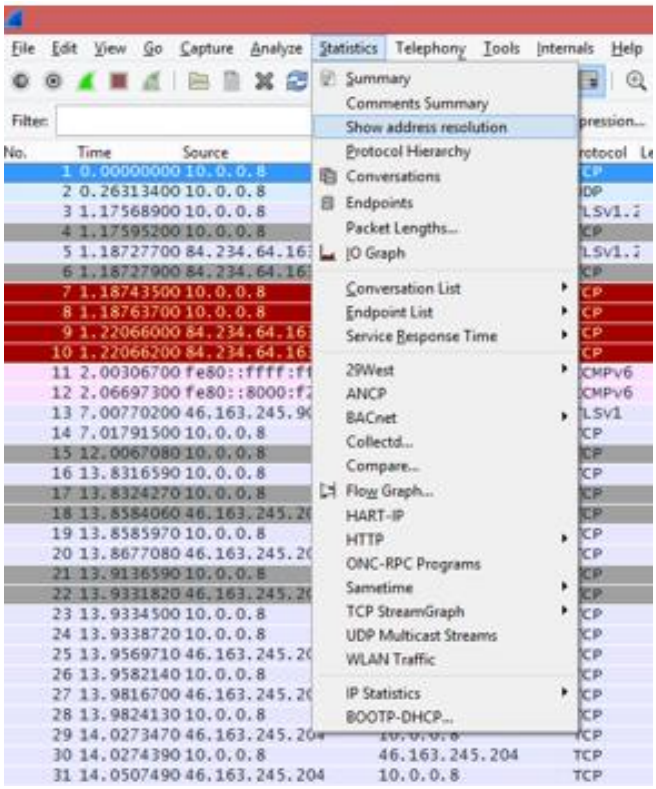
Tehtävä 2: Suodata pakettilistaus näyttämään vain tietty liikenne

Wiresharkin tehokas käyttö työssä vaatii näkyvillä olevan liikenteen rajauksen oleellisen liikenteeseen. Jotta mitään olennaista tietoa ei menetetä, on turvallisempaa suodataa näkymää kuin itse kaappausta.

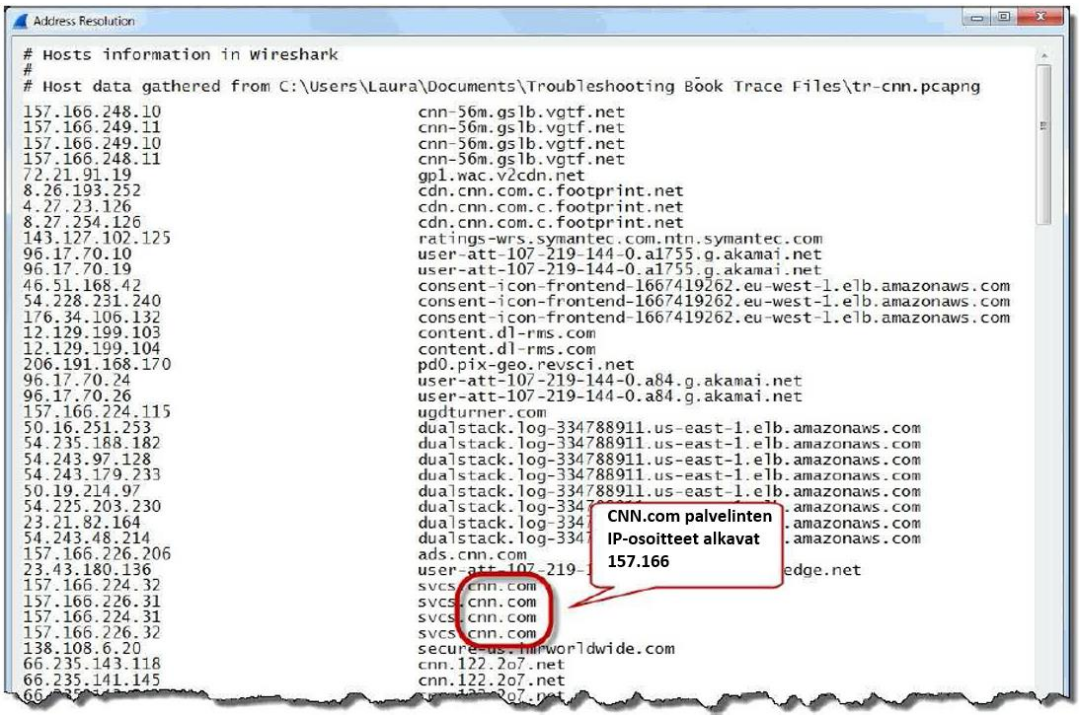
1. **Avaa Wiresharkilla trace-tiedosto tr-cnn.pccapng. ("File" -> "Open" -> tr-cnn.pccapng)**



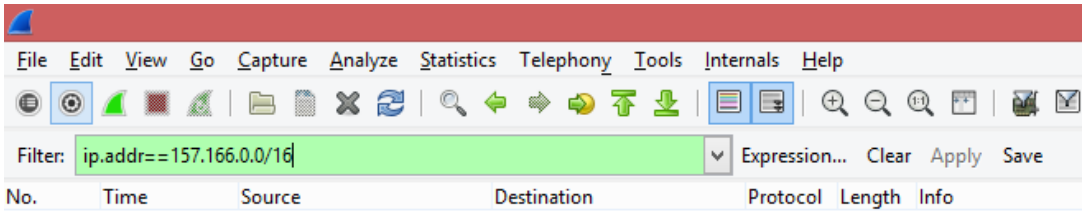
2. Tämä trace-tiedosto pitää sisällään lukuisia keskusteluja verkossa. Aloita tarkistelemalla kaikkien cnn.com nimisten DNS-nimien IP-osoitteita.
3. **Allaolevan kuvan mukaisesti: klikkaa "Statistics" -> "Show address resolution"**



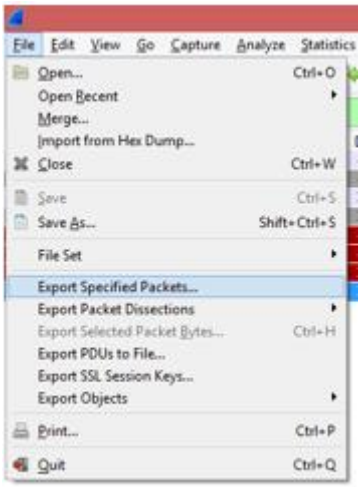
4. Tarkistele uudessa ikkunassa avautuvia cnn.com loppuisia nimiä ja huomaatko millä IP-osoitteella CNN. com palvelimet alkavat?



5. Hyvä! Eli ne alkavat 157.166.x.x **Luo seuraavaksi siis suodatin koskemaan vain verkkoa 157.166.0.0/16. Kirjoita "Filter" kohtaan *ip.addr==157.166.0.0/16* ja paina "Apply".** Huomaa myös kuinka alapalkissa "Packets" ja "displayed" arvot vaihtuvat koskemaan vain nyt filtteröityä liikennettä.

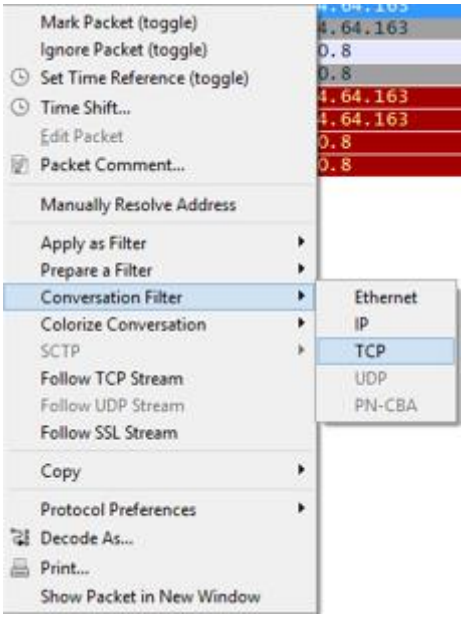


6. Nyt tallennetaan luotu näkymä. **Klikkaa "File"-valikkoa ja valitse "Export Specified Packets". Tallenna tiedosto nimellä tr-cnntraffic.pcapng. Paina "Save".**



7. Kokeillaan seuraavaksi pakettisuodattimen luontia näkymän rajaamiseksi TCP-keskusteluun. **Klikkaa pakettia #3**

hiireän oikealla napilla ja valitse "Conversation FIlter" -> "TCP". Pakettien pitäisi rajoittua nyt vain tuohon yhteen keskusteluun.



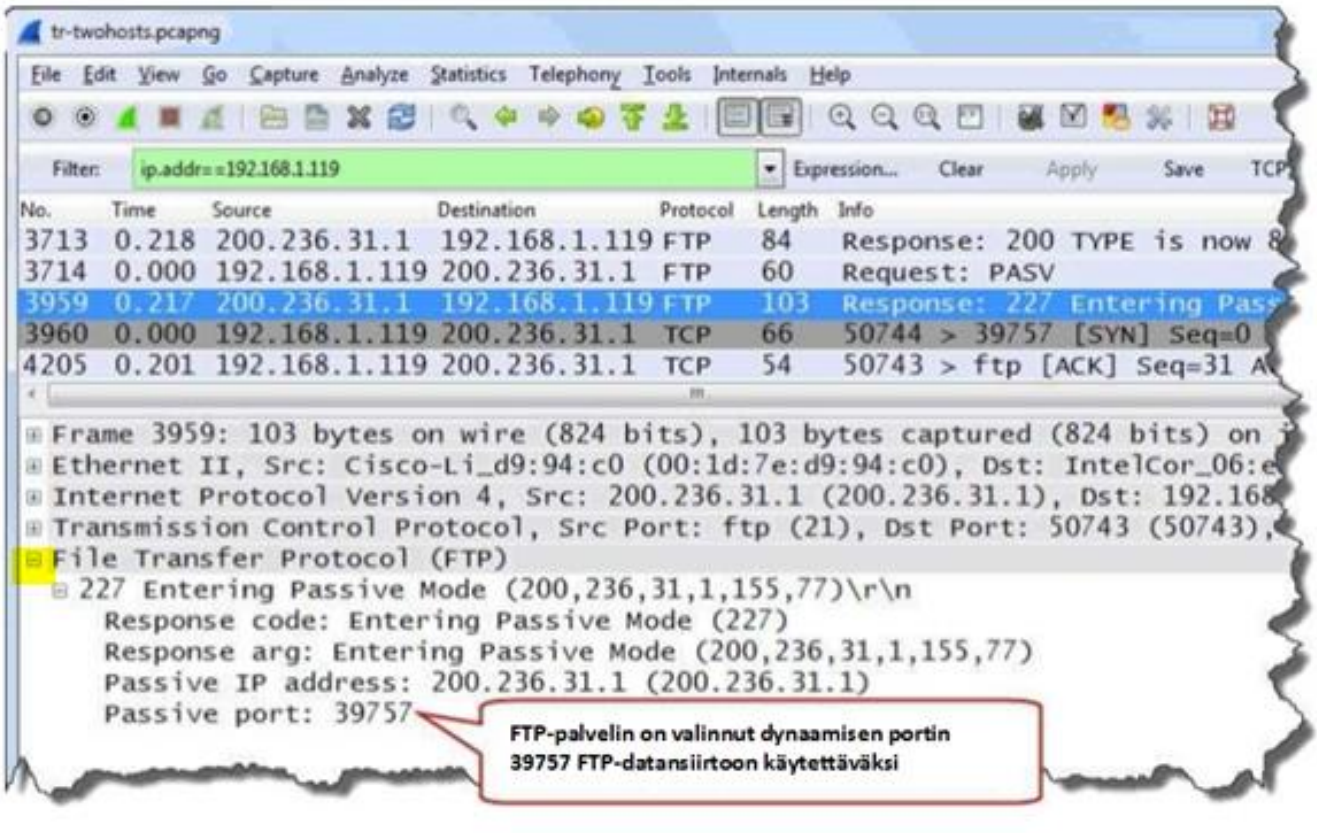
Tehtävä 3: Liikenteen suodatus porttinumeron perusteella

Wiresharkissa on kaksi erilaista tapaa suodattaa sovelluskohtaista liikennettä trace-tiedostossa. Suodatus onnistuu joko sovelluksen nimellä (jos wireshark tuntee sovelluksen) tai käytetyllä porttinumerolla. Jos halutaan suodattaa UDP-liikennettä voidaan liikenne suodattaa käyttäen pelkkää applikaationimeä (esim. "***tftp***" näyttää kaiken TFTP liikenteen)

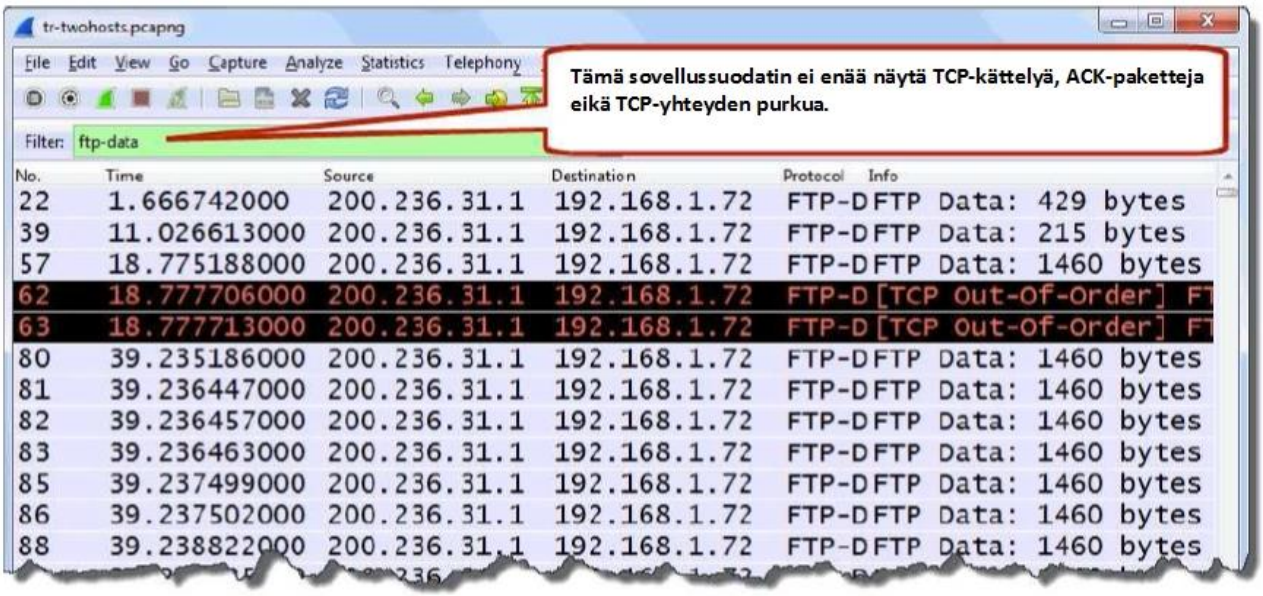
Jos taas kyseessä on TCP-liikenne, jota halutaan suodattaa, on parempi käyttää suodatuksessa porttinumeroita (esim. FTP:lle "***tcp.port==21***"), koska tällöin saadaan mukaan myös TCP:n kolmivaiheinen kättely, ACK-paketit ja TCP-yhteyden purkuun liittyvä liikenne.'

Allaolevassa tehtävässä kaksi työasemaa lataa tiedostoa nimeltä "*OOo_3.3.0_Linux_x86_langpack-rpm_en-US.tar.gz* "

1. **Avaa tiedosto *tr-twohosts.pcapng*.**
2. Suodatetaan näkymästä pois kaikki liikenne paitsi saapuva- ja lähtevä liikenne IP-osoitteesta 192.168.1.119. **Syötä suodatin (engl. Filter) kenttään *ip.addr==192.168.1.119***
3. Etsi vastaus PASV-komentoon (paketti 3959). **Valitse paketti**
4. **Laajenna File Transfer Protocol (FTP) osio paketin yksityiskohtaisessa näkymässä selvittääksesi mitä porttinumeroa FTP-palvelin tulee kuuntelemaan FTP data channel porttina (39757)**



5. **Vaihda suodattimeksi porttikohtainen suodatin: *tcp.port==39757* ja paina "Apply".** 28020 pakettia osuu tähän suodattimeen. Huomaatko kuinka pystyt analysoimaan porttikohtaisella suodattimella myös TCP-kättelyä, ACK-paketteja sekä TCP-yhteyden purkuun liittyviä paketteja.
6. Vaihda suodattimeksi sovelluskohtainen suodatin: ***ftp-data*** ja paina "**Apply**". Huomaatko kuinka et enää pystykään analysoimaan sovelluskohtaisella suodattimella enää TCP-kättelyä, ACK-paketteja sekä TCP-yhteyden purkuun liittyviä paketteja?
7. Paina **Clear** tyhjentääksesi näkymän suodatuksen.


tr-twohosts.pcapng
File Edit View Go Capture Analyze Statistics Telephony
Filter: ftp-data
Tämä sovellussuodatin ei enää näytä TCP-kättelyä, ACK-paketteja eikä TCP-yhteyden purkua.
No. Time Source Destination Protocol Info
22 1.666742000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 429 bytes
39 11.026613000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 215 bytes
57 18.775188000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 1460 bytes
62 18.777706000 200.236.31.1 192.168.1.72 FTP-D [TCP Out-Of-Order] FT
63 18.777713000 200.236.31.1 192.168.1.72 FTP-D [TCP Out-Of-Order] FT
80 39.235186000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 1460 bytes
81 39.236447000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 1460 bytes
82 39.236457000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 1460 bytes
83 39.236463000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 1460 bytes
85 39.237499000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 1460 bytes
86 39.237502000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 1460 bytes
88 39.238822000 200.236.31.1 192.168.1.72 FTP-D FTP Data: 1460 bytes

Tehtävä 4: HTTP-liikenteen viiveiden paikantaminen

Tehtävä 4 HTTP-liikenteen viiveiden paikantaminen. Tässä tehtävässä tarkstellaan HTTP pakettien välisiä viiveitä. Tämän mahdollistamiseksi luodaan yhden paketin lisätiedoista uusi sarake "time since request". Tiedoston alusta löytyy seuraavat paketit:

Paketit 1-3 ovat TCP handshake paketteja.

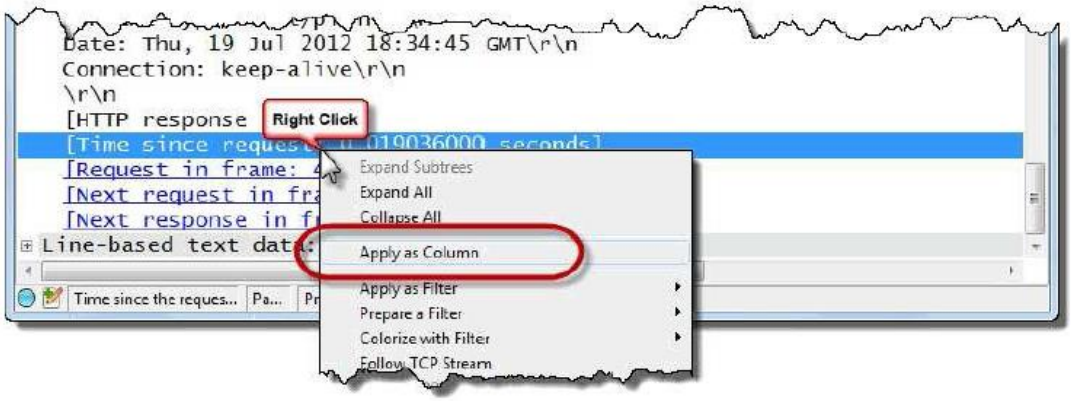
Paketti 4 on HTTP GET request tiedostolle nimeltä *minitri.flg*.

Paketti 5 on ACK GET request:lle.

Paketti 6 on HTTP 200 OK Response.

1. Avaa tiedosto **tr-httpdelta.pcapng.**
2. **Valitse Paketti 6 ja alhaalla näkyvässä paketin yksityiskohtaisessa näkymässä klikkaa Hypertext Transfer Protocol otsikon edessä näkyvää "+" -näppäintä laajentaaksesi HTTP-protokollaan liittyvät tiedot**

3. Selaa HTTP-kohdan pohjalle ja klikkaa hiiren oikealla näppäimellä kohtaa **[Time Since request: 0.019036000 seconds].** Valitse **"Apply as Column"**



4. Wireshark asettaa uuden sarakkeen pakettilistanäkymässä "Length" ja "Info" sarakkeiden väliin. Nimetään sarake uudestaan. **Klikkaa hiiren oikealla sarakkeen otsikkoa ja valitse "Edit Column Details". Kirjoita "HTTP Delta" kohtaan "Title"**

No.	Time	Source	Destination	Protocol	Length	HTTP Delta	Info
49	2.675406	216.115.74.202	24.6.173.220	HTTP	544	2.807332000	HTTP/1.1 200
83	0.823550	216.115.74.202	24.6.173.220	HTTP	543	0.958153000	HTTP/1.1 200
15	0.000005	96.17.148.114	24.6.173.220	HTTP	147	0.037208000	HTTP/1.1 200
105	0.000005	96.17.148.114	24.6.173.220	HTTP	90	0.035879000	HTTP/1.1 200
6	0.000800	96.17.148.114	24.6.173.220	HTTP	317	0.019036000	HTTP/1.1 200
96	0.000860	96.17.148.114	24.6.173.220	HTTP	317	0.016734000	HTTP/1.1 200
61	0.000778	96.17.148.115	24.6.173.220	HTTP	317	0.015392000	HTTP/1.1 200
63	0.015217	96.17.148.115	24.6.173.220	HTTP	287	0.015217000	HTTP/1.1 304

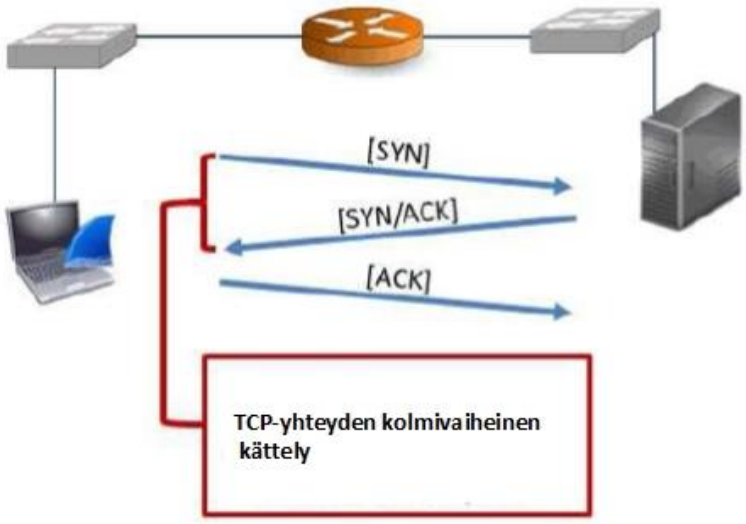
5. **Tuplaklikkaa nyt "HTTP Delta" saraketta pakettilistanäkymässä** ja katso kuinka paketit järjestyvät näkymässä suurimmasta viiveestä pienimpään. Huomaa kuinka paketin numero 49 viive on lähes kolme sekuntia.

Tehtävä 5: HTTP-aikojen mittaaminen (TCP-dissektori kokoamisasetus)

Kaikki Wiresharkin oletusasetukset eivät ole parhaita mahdollisia vianselvitykseen. Esimerkiksi *"Allow subdissector to reassemble TCP streams"* asetus on oletuksena päällä.

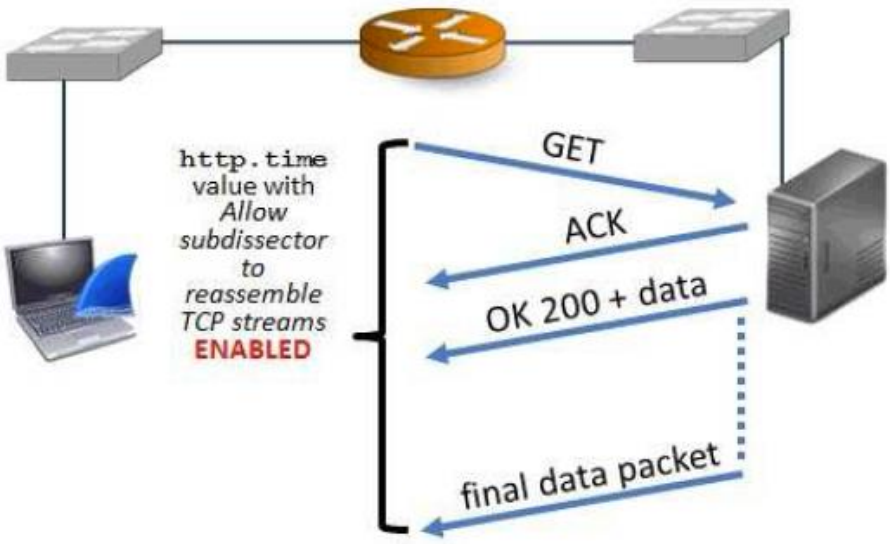
Tämän asetuksen on hyvä antaa olla päällä, jos halutaan mitata eri tiedostojen HTTP-latausaikoja. Tämä asetus on hyvä ottaa pois käytöstä, kun halutaan mitata HTTP Response aikoja ja kun halutaan tietää kuinka pitkää palvelimella meni ensimmäisessä vastauksessa. Tässä tehtävässä haluamme ottaa asetuksen pois käytöstä.

Tämän tehtävän trace-tiedoston alusta löytyy TCP-kättely (paketit 1-3).



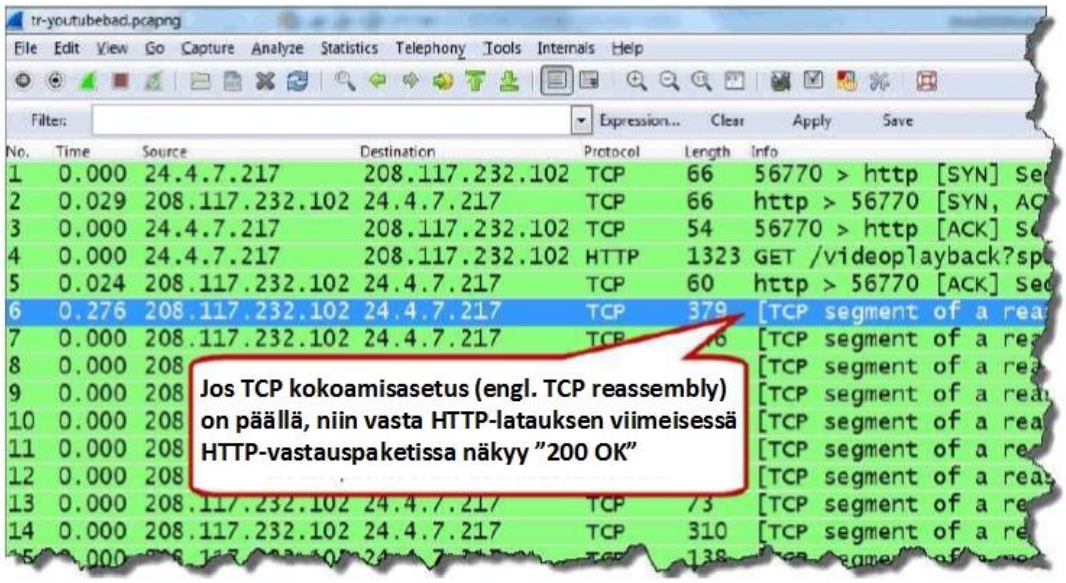
Tämän jälkeen paketit 4-6 ovat alla olevan kuvan mukaisesti:

- Paketti 4 on HTTP GET pyyntöpaketti (request) työasemalta
- Paketti 5 on ACK paketti palvelimelta
- Paketti 6 on HTTP 200 OK vastauspaketti palvelimelta



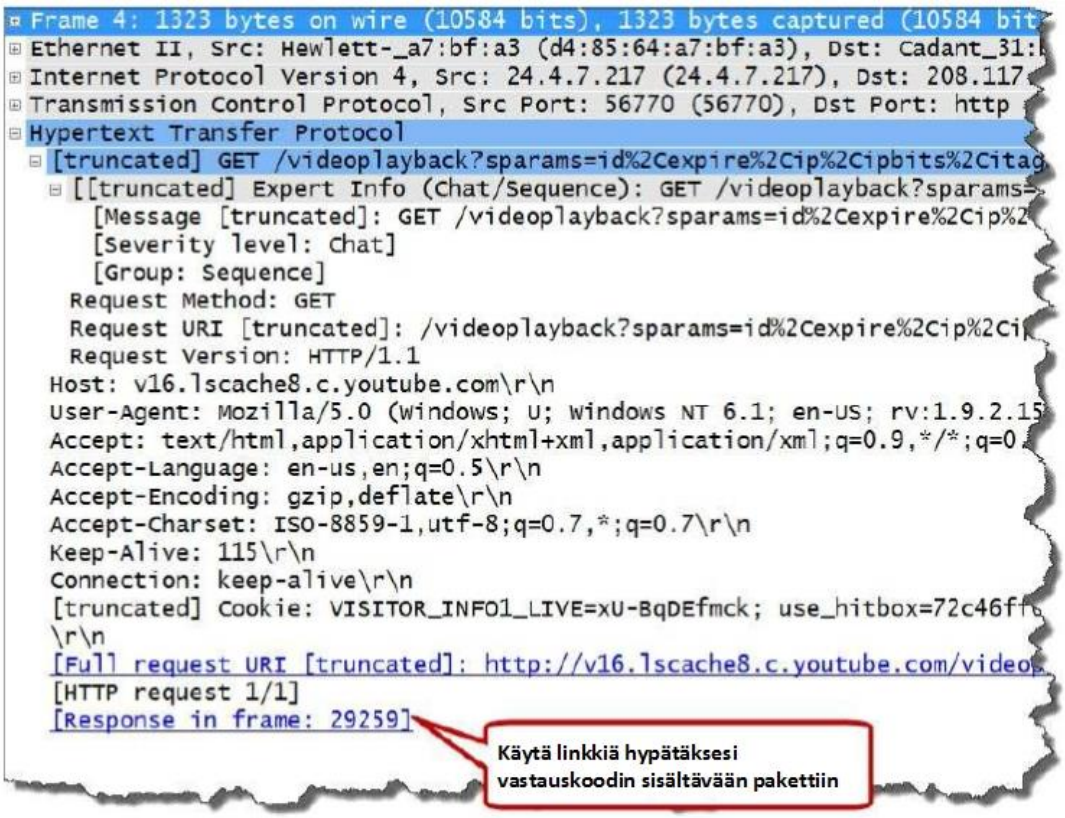
Valitettavasti Response code tietoa ei näy Info sarakkeessa paketille 6, koska *"Allow subdissector to reassemble*

TCP streams" asetus on päällä. Responce code tieto on näkyvissä vasta paketissa 29259, joka pakettina sisältää viimeiset tavut pyydetystä tiedostosta.



1. Avaa tiedosto **tr-youtubebad.pcapng.**
2. **Valitse paketti, klikkaa hiiren oikealla kohtaa Hyper Text Transfer Protocol otsikkoa ja valitse "Expand Subtrees".**

- Selaa HTTP-näkymän pohjalle ja tuplakilkkaa hyperlinkkiä pakettiin numero 29259 (kuva alla)

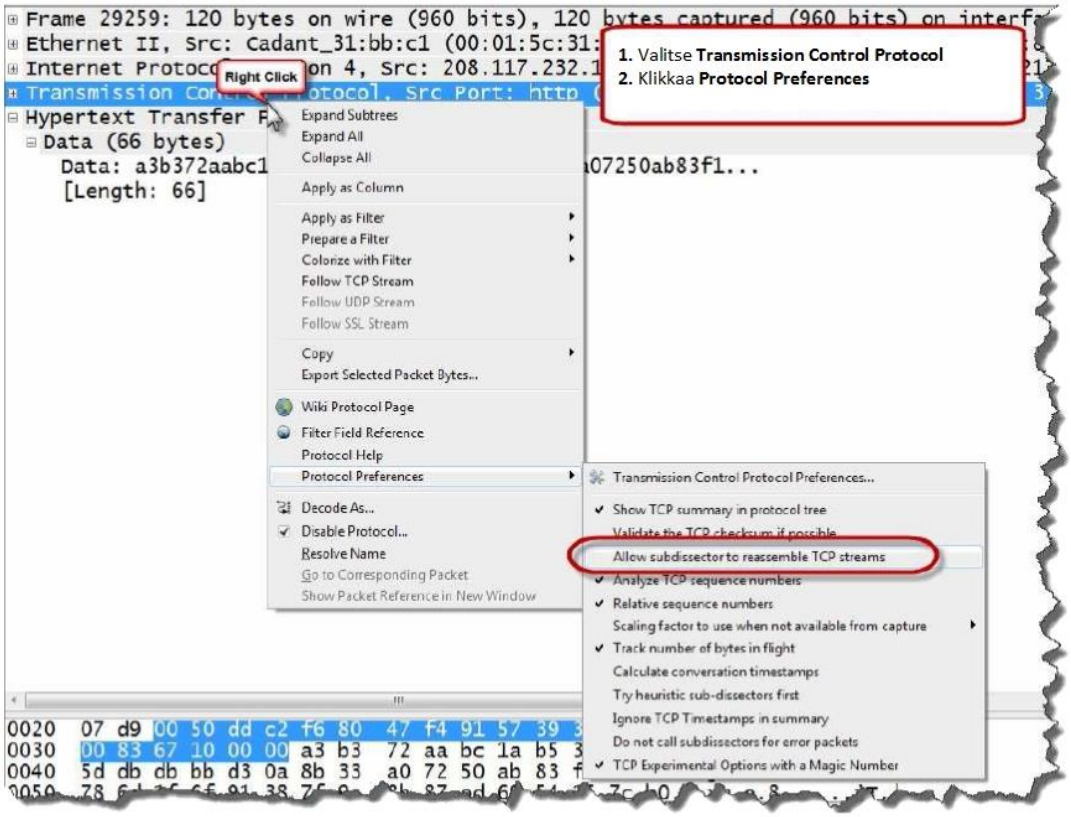


3. Selaa HTTP otsakkeen loppuun paketissa 29259. **"Time Since Request"** kenttä kertoo, että HTTP vastausaika oli

yli 276 sekuntia.

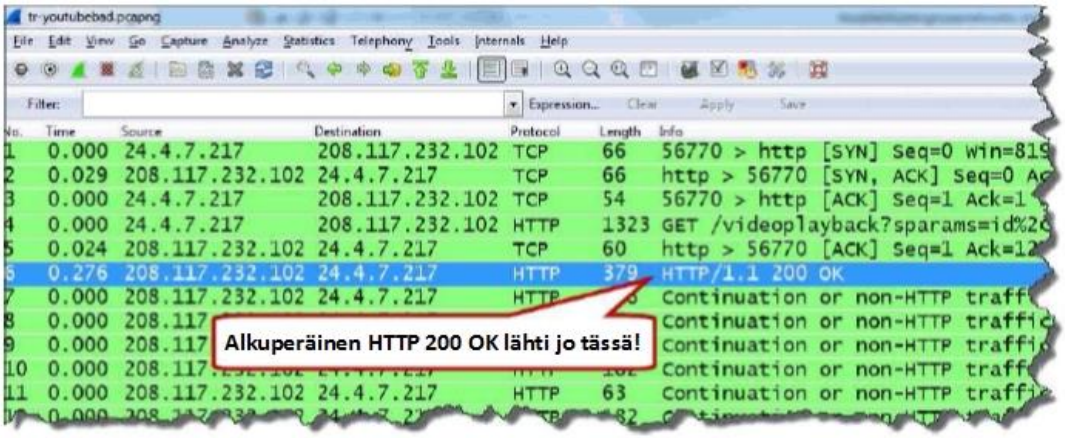
- Tämä ei ole oikein. HTTP-vastausaika lasketan HTTP-pyyntöpaketista HTTP-vastauspakettiin, joka sisältää "200 OK" vastauksen. Wireshark merkitsee viimeisen paketin latauksesta vastauspaketiksi silloin, kun *Allow subdissector to reassemble TCP streams"* asetus on päällä.

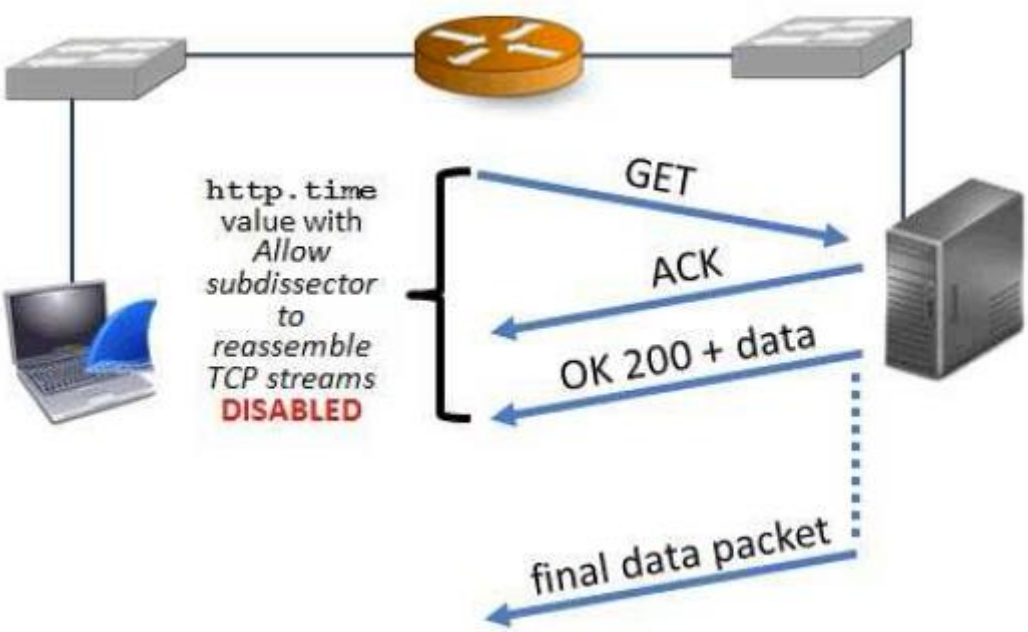
4. Etsitään oikea HTTP-vastausaika. Minkä tahansa paketin paketin lisätietokentässä klikkaa hiiren oikealla näppäimellä kohtaa "**TCP Header"**, valitse **"Protocol Preferences"** ja paina pois päältä **"allow subdissector to reassemble TCP streams"** asetus.



5. Nyt klikkaa "mene ensimmäiseen pakettiin" nappia 

- Huomaatko kuinka paketti 6 oikeasti sisältääkin 200 OK vastauksen? Tarkkaile http vastausaikaa paketissa 6. Se on vähän yli 300 ms. Aikamoinen ero alkuperäiseen Wiresharkin raportoimaan 276 sekuntiin.



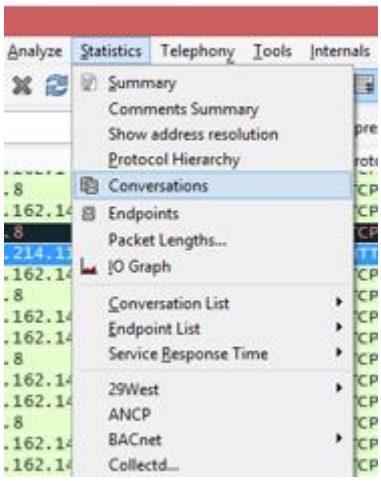

http.time
value with
Allow
subdissector
to
reassemble
TCP streams
DISABLED
GET
ACK
OK 200 + data
final data packet

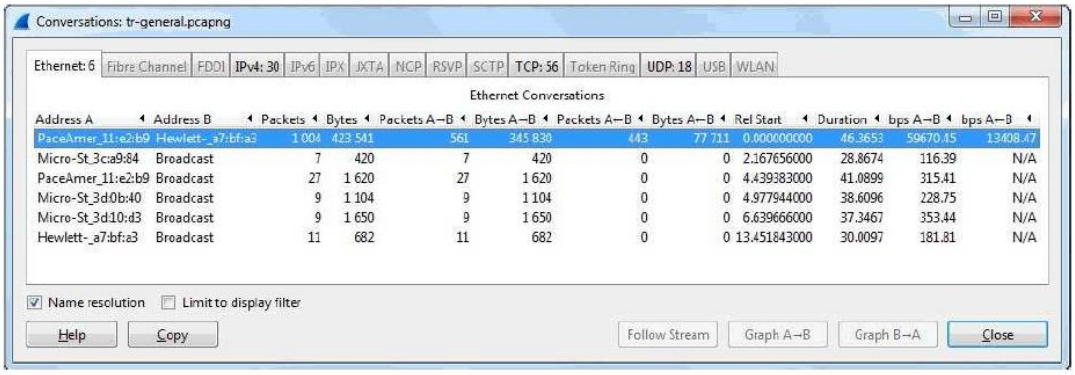
Tehtävä 6: Etsi isoimmat kaistansyöjät verkosta

Joskus saastuneet työasemat tai joskus niin sanottu "viihdekäyttö" saattaa kuormittaa yrityksien ja organisaatioiden verkkoja kohtuuttomasti ja luoda vikatilanteita.

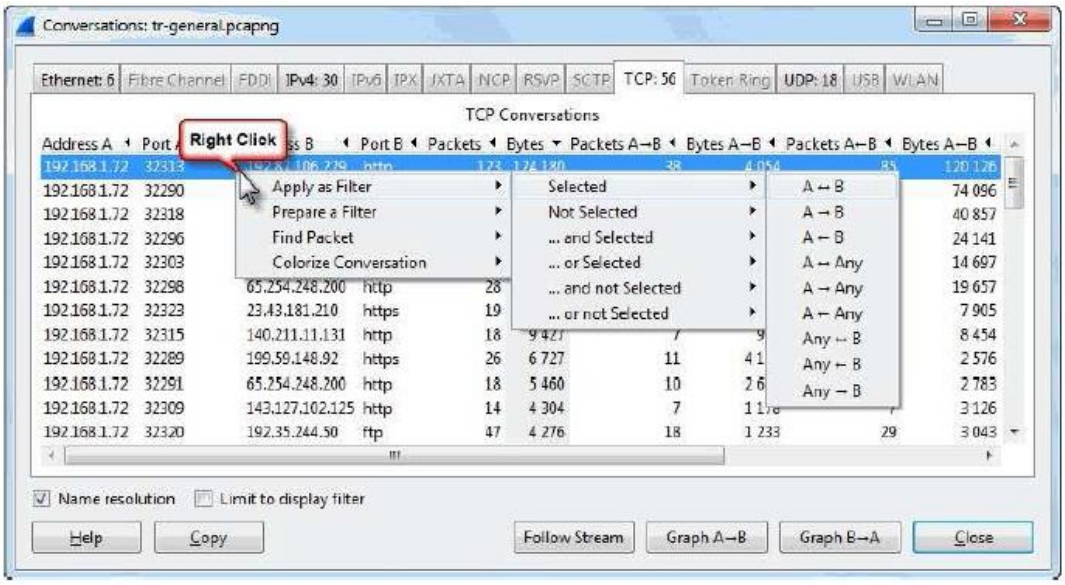
Käytä Wirsharkin keskustelut (engl. conversations)-ikkunaa selvittääksesi verkkoa eniten kuormittavat lähettäjät ja vastaanottajat. Luokitus voi tapahtua MAC-, IP-osoiteen tai vaikkapa porttinumeron perusteella. Tässä tehtävässä etsitään **Byte Count** arvon perusteella verkon suurin kuormittaja.

8. Avaa tiedosto **tr-general.pcapng.**
9. Valitse **Statistics | Conversations**. Tässä ikkunassa näet erityyppisten keskustelujen lukumäärät trace-tiedostossa





10. Meitä kiinnostaa erityisesti näiden keskustelujen tavumäärät (byte count). Klikkaa **TCP** painiketta ja sitten klikkaa **Bytes** saraketta kahdesti järjestääksesti keskustelut suurimmasta pienimpään. Isoin keskustelu verkossa on käyty 192.168.1.72 portin 32313 ja 192.168.106.229 portin 80 välillä (listattu http:na)
11. Klikkaa hiiren oikealla tätä ylintä keskustelua (engl. top conversation) ja valitse Apply as Filter | Selected A <--> B. Wireshark ottaa käyttöön suodattimen ja näyttää keskustelun 123 pakettia.


Conversations: tr-general.pcapng
Ethernet: 6
Fibre Channel
FDDI
IPv4: 30
IPv6
IPX
JXTA
NCP
RSVP
SCTP
TCP: 56
Token Ring
UDP: 18
USB
WLAN
TCP Conversations
Address A
Port A
Address B
Port B
Packets
Bytes
Packets A→B
Bytes A→B
Packets A←B
Bytes A←B
Right Click
192.168.1.72 32313
192.168.1.72 32290
192.168.1.72 32318
192.168.1.72 32296
192.168.1.72 32303
192.168.1.72 32298 65.254.248.200 http 28
192.168.1.72 32323 23.43.181.210 https 19
192.168.1.72 32315 140.211.11.131 http 18 9 427 7 9
192.168.1.72 32289 199.59.148.92 https 26 6 727 11 4 1
192.168.1.72 32291 65.254.248.200 http 18 5 460 10 2 6
192.168.1.72 32309 143.127.102.125 http 14 4 304 7 1 1
192.168.1.72 32320 192.35.244.50 ftp 47 4 276 18 1 233 29 3 043
Apply as Filter
Prepare a Filter
Find Packet
Colorize Conversation
Selected
Not Selected
... and Selected
... or Selected
... and not Selected
... or not Selected
A ↔ B
A → B
A ← B
A ↔ Any
A → Any
A ← Any
Any ↔ B
Any ← B
Any → B
Name resolution
Limit to display filter
Help
Copy
Follow Stream
Graph A→B
Graph B→A
Close

Tehtävä 7: Käytä aikasaraketta paikantaaksesi latenssiongelmat

Latenssi tarkoittaa tietoliikennetekniikassa tavallisimmin sitä aikaa, mikä paketilta kuluu matkaan lähettäjältä vastaanottajalle ja takaisin eli paketin edestakaiseen matkaan kuluvaa aikaa. Yksinkertaisessa lähiverkossa tyypillinen latenssi on alle millisekunnin luokkaa.

Latenssi ei sisällä sitä aikaa, joka vastaanottajalta kuluu paketin prosessointiin. Useimmissa järjestelmissä meno-paluu viive voidaan mitata ICMP-protokollaa hyödyntävällä ping-toiminnolla. Ping-palvelu ei prosessoi saapuvaa pakettia mitenkään, vaan palauttaa lähettäjälle vastauksen heti paketin saavuttua.

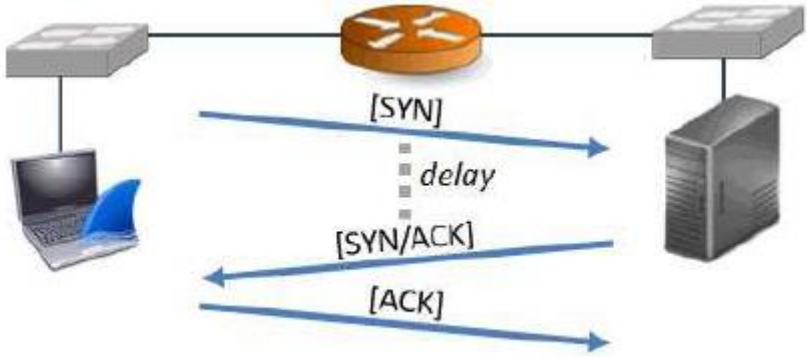
Latenssiongelmien selvittelyssä toimivat seuraavat metodit riippuen siitä, missä liikennettä kaapataan:

Liikenteen kaappaus työaseman päässä:

- Suuret viiveet TCP-kättelyssä lähtevän SYN-paketin ja saapuvan SYN/ACK paketin välillä (tcp.time_delta).

Liikenteen kaappaus palvelimen päässä:

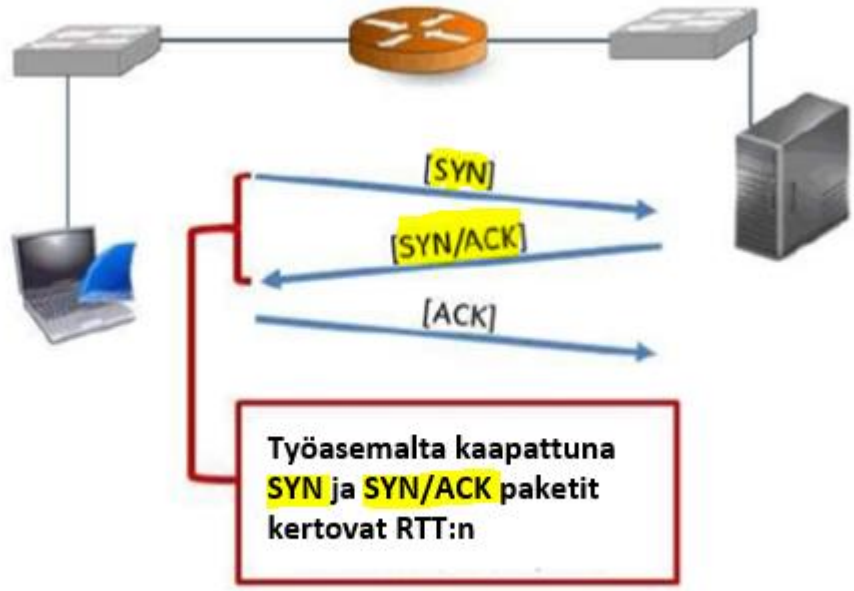
- Suuret viiveet TCP kättelyssä SYN/ACK ja ACK viestien välillä (tcp.time_delta).



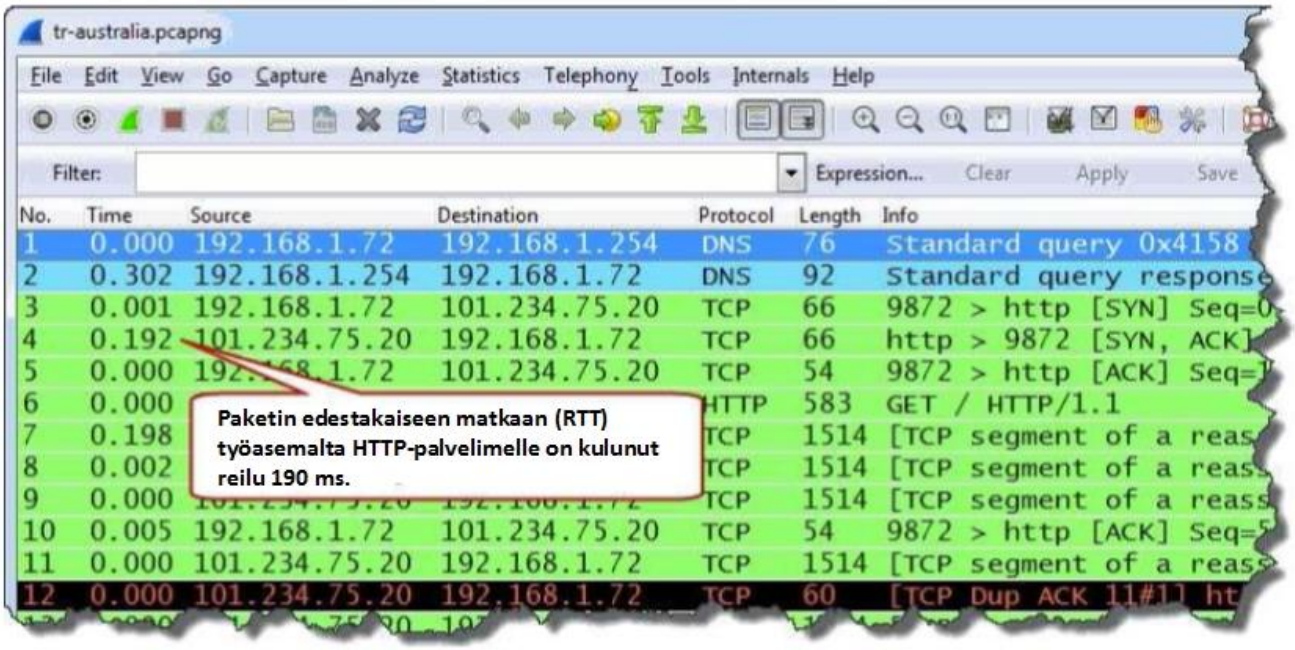
Paketit saavat aikaleiman sillä hetkellä, kun niitä kaapataan. Oletuksena Wireshark asettaa pakettilistanäkymän aikasarakkeen asetukselle "sekuntia kaappauksen aloittamisesta" (engl.Seconds Since Beginning of Capture). Tämän lisäksi oletuksena käytetään nanosekunteja riippumatta siitä tukevatko pakettien aikaleimaukset tällaista tarkkuutta.

Alla olevan tehtävän trace-tiedosto pitää sisällään web-surffausistunnon, joka on napattu työasemalta. Tehtävässä on tarkoituksena muokata aikasarakkeen oletusasetuksia ja selvittää kolmivaiheisen TCP-kättelyn kahden ensimmäisen paketin välinen aika.

1. **Avaa tiedosto tr-australia.pcapng.**
2. **Päätellään paketin edestakaiseen matkaan kulunut (engl. round trip time / RTT) aika vertaamalla SYN ja SYN/ACK pakettien välistä aikaerotusta.**



3. Tämä trace-tiedosto alkaa DNS-kyselyllä ja vastauksella. TCP-yhteyden luonti alkaa paketilla 3. **Valitse View | Time Display Format | Seconds Since Previous Displayed Packet.**
4. **Vaihda tarkkuus: View | Time Display Format | Milliseconds: 0.123.**

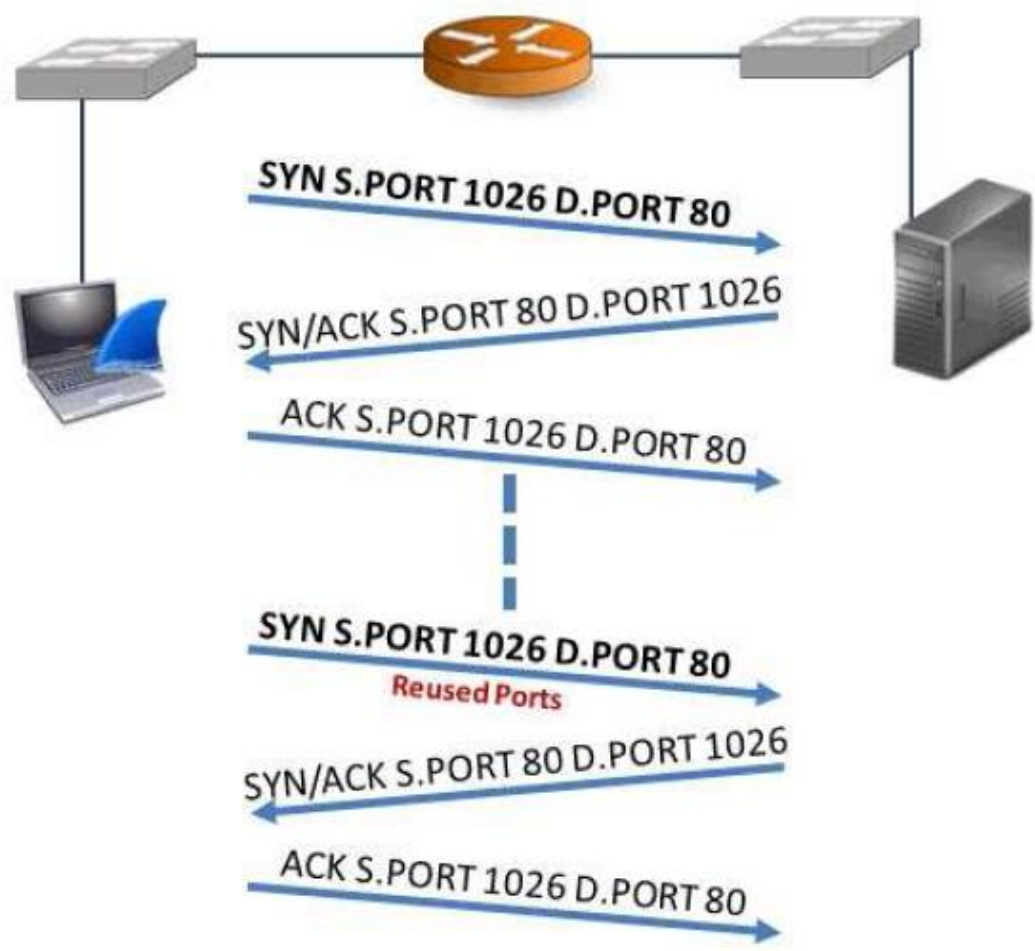


Nyt nähdään, että Round Trip Time on ollut 192 millisekuntia (ms). Onko 192 ms. sitten paljon vai vähän? Se riippuu siitä mikä yleensä on kyseiselle liikenneyhteydelle normaalia. Jos esim. Suomessa MPLS-verkkojen sisällä tyypillinen viive on muutamia millisekunteja, niin 192 ms. on aika paljon.

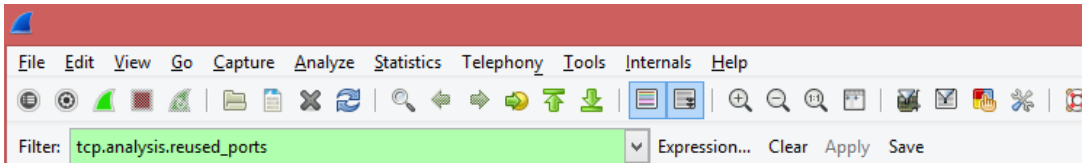
Mitä isoille viiveille sitten voi tehdä? Jos viive johtuu verkkolaitteista (esim. palomuurin suorituskykyongelma), viiveelle voi tehdä jotain.. Jos viive puolestaan johtuu verkkoinfrastruktuurin ulkopuolisista asioista (matkalla internetin halki), ei ole juuri muuta tehtävissä kuin ottaa yhteys internet-palveluntarjoajaan.

Tehtävä 8: Laske suodattimella paketit, joissa uudelleen käytetty porttinumero

Uudelleen käytetyt TCP-porttinumerot eivät normaalitilanteessa ole ongelma, jos avatut yhteydet on terminoitu. Jos kuitenkin yhteyksiä jää "roikkumaan" muodostuu tästä ongelma palvelimen päässä, kun edellistä yhteyttä ei ole terminoitu (joko TCP FIN:lla tai TCP RST:lla). Tässä tehtävässä selvitetään nopealla suodattimen käytöllä onko päällekkäisiä porttinumeroita käytetty. Wireshark osaa tunnistaa tällaiset paketit erikseen.

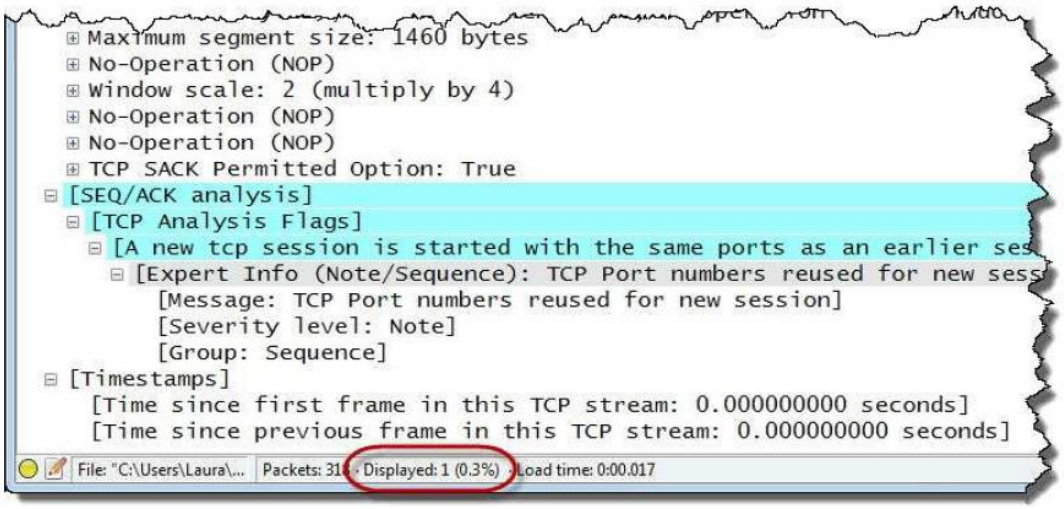


1. **Avaa tr-reusesports.pccapng.**
2. **Syötä suodattimeksi tcp.analysis.reused_ports. Wiresharkin tilannerivi näyttää, että suodattimella on löytynyt yksi osuma.**



3. **Laajenna [SEQ/ACK analysis] kohta paketista, jossa porttinumero on uudelleen käytetty. Huomaa Wiresharkin vaalean siniseksi väritettämä alue. Tämä väri tarkoittaa asiantuntijoille tarkoitettuja**

huomioita.

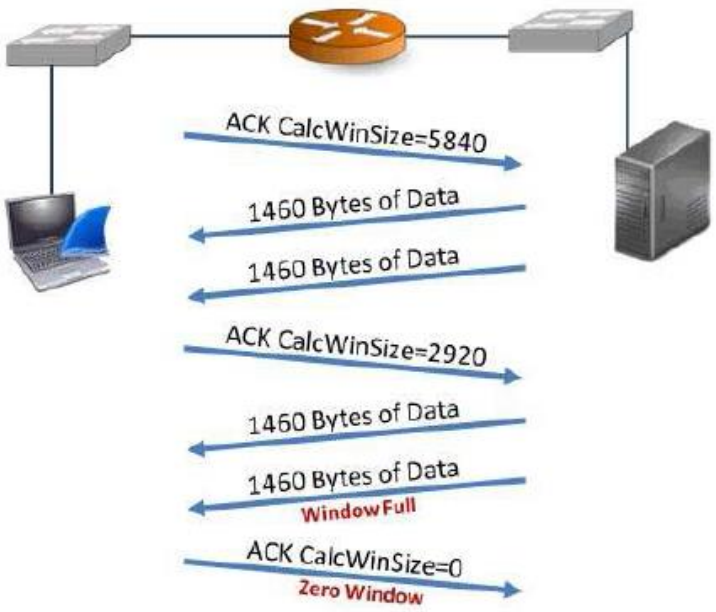


4. Tyhjennä suodatin.

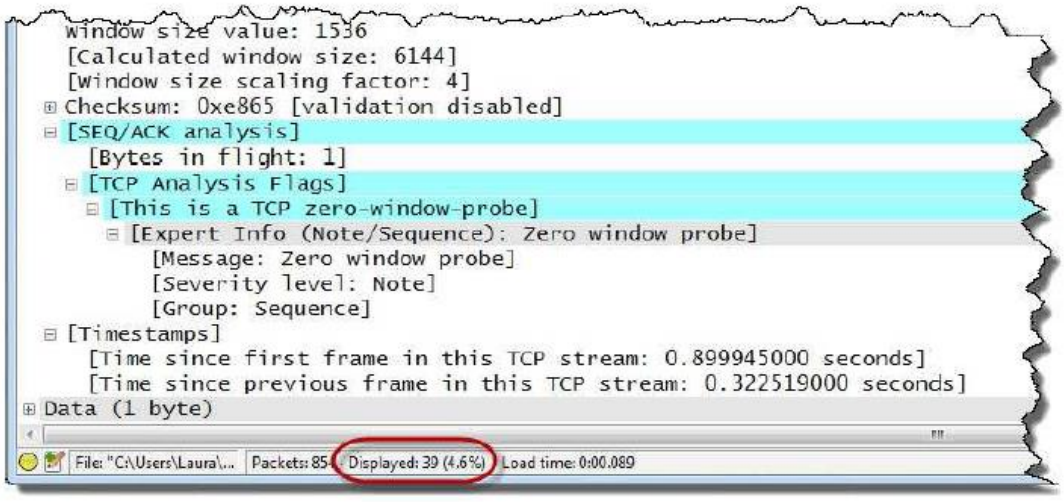
Tehtävä 9: Nollakokoinen TCP-vastaanottoikkuna

TCP-vastaanottoikkuna määrittää käyttäjän päätelaitteella, montako tavua lähettäjä voi lähettää saamatta kuittausta. Yleensä suuret vastaanottoikkunat parantavat suorituskykyä suurten viiveiden ja suuren kaistanleveyden verkoissa.

Joskus käy niin, että vastaanottajan sovellus ei saa purettua vastaanottopuskurista tarpeeksi nopeasti dataa ja tällöin mainostettava ikkunan koko voi tippua nollaan. Tämä taas saattaa johtua esim. sovelluksen vikatilasta tai liian pienestä keskusmuistimäärästä koneella. Wireshark osaa tunnistaa "Zero Window" paketit erikseen asiantuntijoita varten.



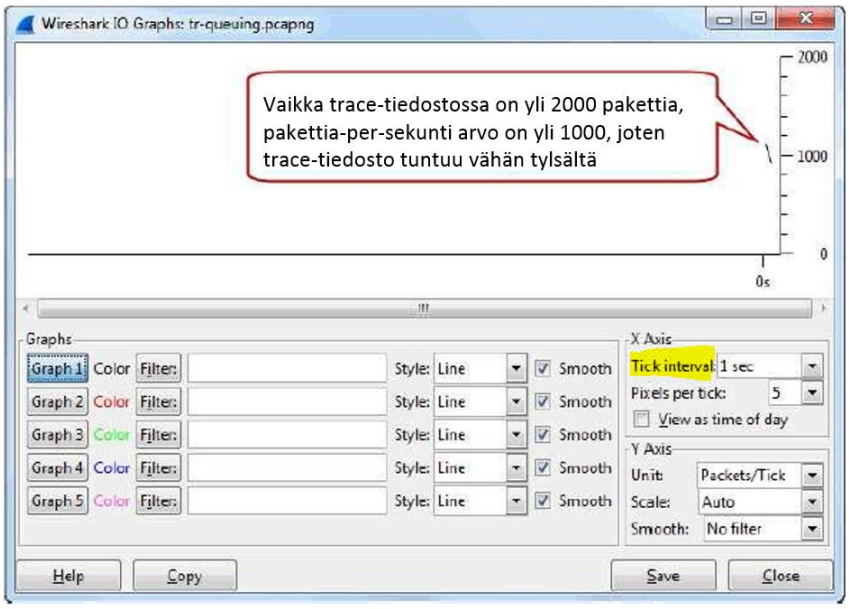
1. **Avaa tr-winzero-print.pccapng.**
2. **Syötä suodattimeksi tcp.analysis.zero_window_probe || tcp.analysis.zero_window_probe_ack ja paina "Apply" .Wiresharkin tilannerivi näyttää, että suodattimella on löytynyt 39 osumaa.**



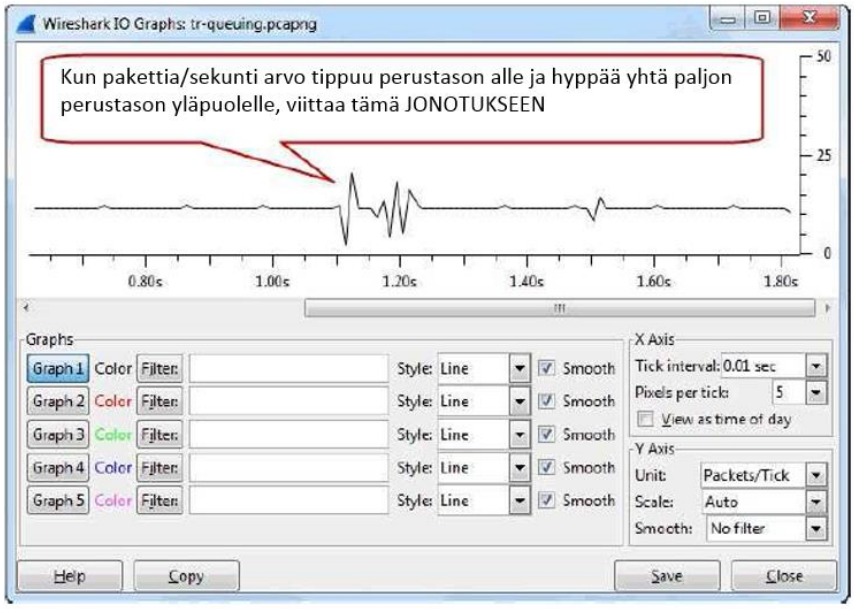
3.Tyhjennä suodatin. Huomaa, että tässä ei ole kyseessä verkko-ongelma vaan työasemaongelma.

Tehtävä 10: Jonotusviiveiden ja pakettihävikin tunnistaminen

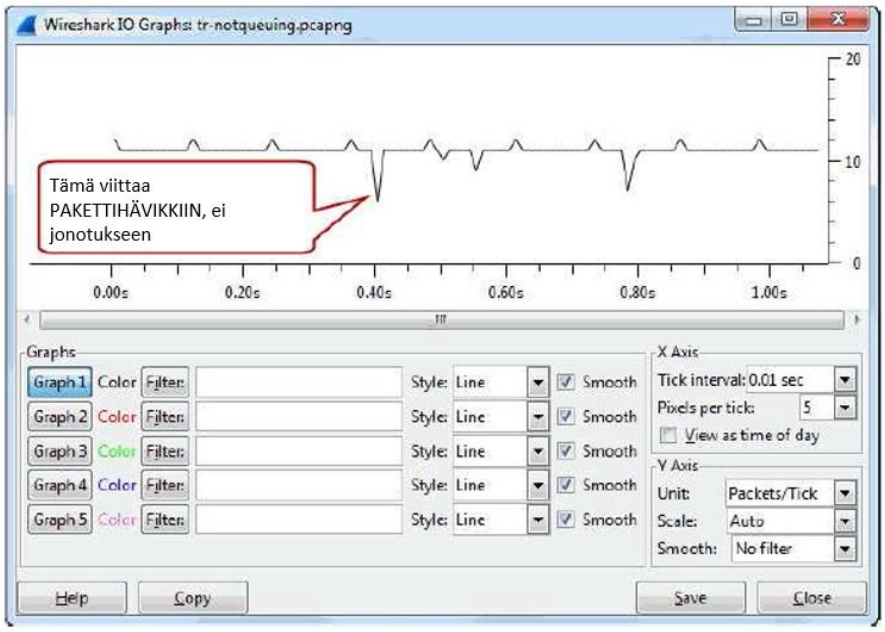
1. **Avaa tr-queuing.pccapng.**
2. **Valitse Statistics | IO Graph.** Näkyviin tulee pieni viivan tynkä. Trace-tiedostossa on 2016 pakettia ja paketin lähetys tahti on yli 1100 pakettia/sek.



3. "Tick interval" arvoa (X-akseli) täytyy muuttaa, jotta kuvaan saataisiin järekä. Vaihdetaan arvio 1 sekunnista 0,01 sekuntiin.



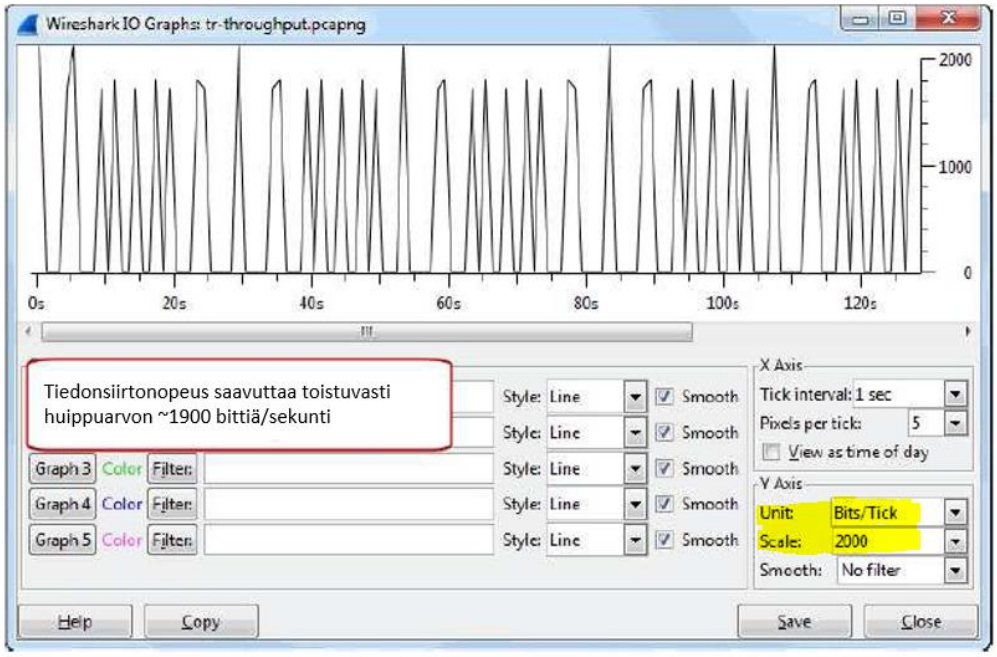
4. Nyt jonotus on helpompi havaita. Kun liikennettä lähetetään tasaiseen tahtiin, saadaan aikaiseksi ns. lähtötilanne ("baseline") sille kuinka nopeasti paketteja on lähetetty. Kun arvo tippuu alhaisemmaksi kuin lähtötilanteessa ja nouse saman määrän lähtötilanteen yläpuolelle, niin voimme päätellä, että liikenne on matkan varrella jäänyt jonottamaan. Mistä tiedämme, että tässä ei ole kyseessä pakettihävikki? Mennään kohtaan 5.
5. **Jätä IP Graph ikkuna auki ja avaa Wiresharkin pääikkunasta trnotqueuing.pcapng.**



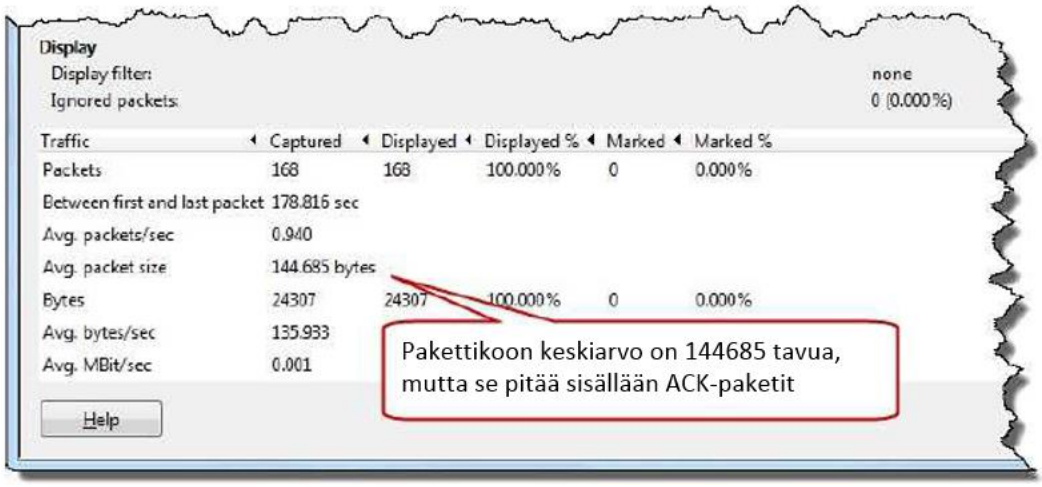
6. Tällä kertaa graafissa näkyy selkeä pudotus ilman samankokoista nousua lähtötilanteen yläpuolelle. Tämä viittaa vahvasti pakettihävikkiin (engl. packet loss).

Tehtävä 11: Verkon huono suorituskyky liian pienten pakettikokojen vuoksi

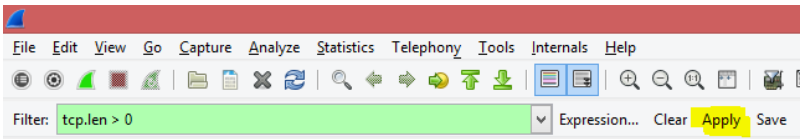
1. **Avaa tr-throughput.pccapng.** Selaa trace-tiedostoa ja katso length-sarakkeen arvoa. Tässä tiedostossa on paljon pieniä paketteja. Ensiksi karsitaan pois alhainen MSS asetus syynä pienille paketeille.
2. **Tarkistele "TCP Option" kohtaa paketeissa 1 ja 2.** Huomaatko kuinka molemmat koneet mainostavat MSS arvoja toisilleen.
3. **Valitse Statistics | IO Graph. Aseta Y-akseli arvoon Bits/Tick. Vaihda Y akselin skaala 2000:n.** Onpas hidas verkkoyhteys!



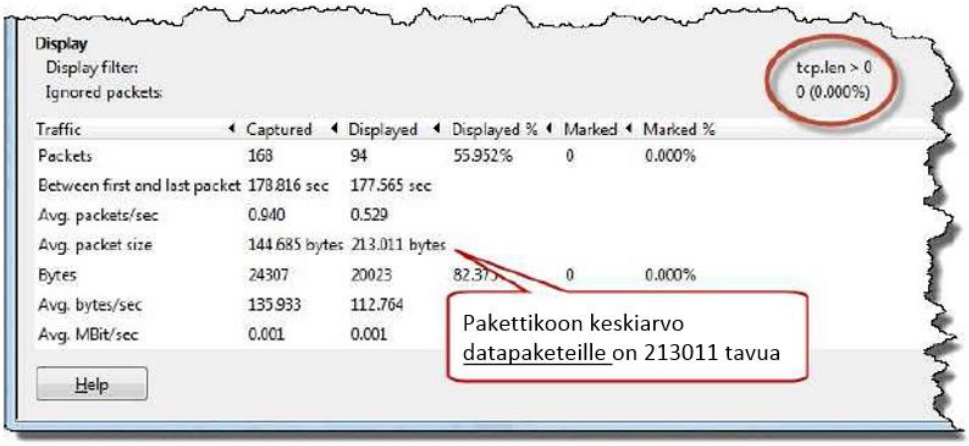
4. **Klikkaa Close. Valitse Statistics | Summary.** Tässä näet pakettikoon keskiarvon trace-tiedostossa.



5. Tämä keskiarvo pitää sisällään ACK-paketit. Otetaan suodatin käyttöön, jotta voimme päätellä keskiarvo pelkkien datapakettien koolle.
6. **Klikkaa OK Summary ikkunassa. Kirjoita tcp.len > 0 kohtaan Filter ja klikkaa Apply**



7. **Valitse Statistics | Summary.** Tässä näet pakettikoon keskiarvon trace-tiedostossa. **Katso Displayed-saraketta.** Tämä sarake näyttää, että datapakettien koon keskiarvo on 213011 tavua.



8. Miksi pienempiä paketteja käytetään? Tässä tehtävässä tarkistettiin, että se ei johdu TCP-yhtyeden asettamista rajoitukista. Todennäköisesti tämä onkin sovelluksen asetus. Täytyy tutkia sovelluksen asetuksia, jotta voitaisiin tietää onko tämä tahallista vai konfiguraatio-ongelma.

Tehtävä 12: "Verkkovika" paljastuukin nimipalveluviaksi.

DNS on pyyntö/vastaus-pohjainen sovellus. Asiakaskone lähettää palvelimelle nimenselvityskyselyn ja me näemme vastauksen saapuvan hyväksyttävässä ajassa. Toisin kuin muut yleiset sovelluksen DNS voi käyttää UDP:ta (yleiset nimenselvityskyselyt) tai TCP:tä tiedonsiirrossa (Zone-siirtoihin)

Wireshark DNS dissektori (packet-dns.c) tunnistaa yleisimmät DNS virheilmoitukset, jotka on listattu alle.

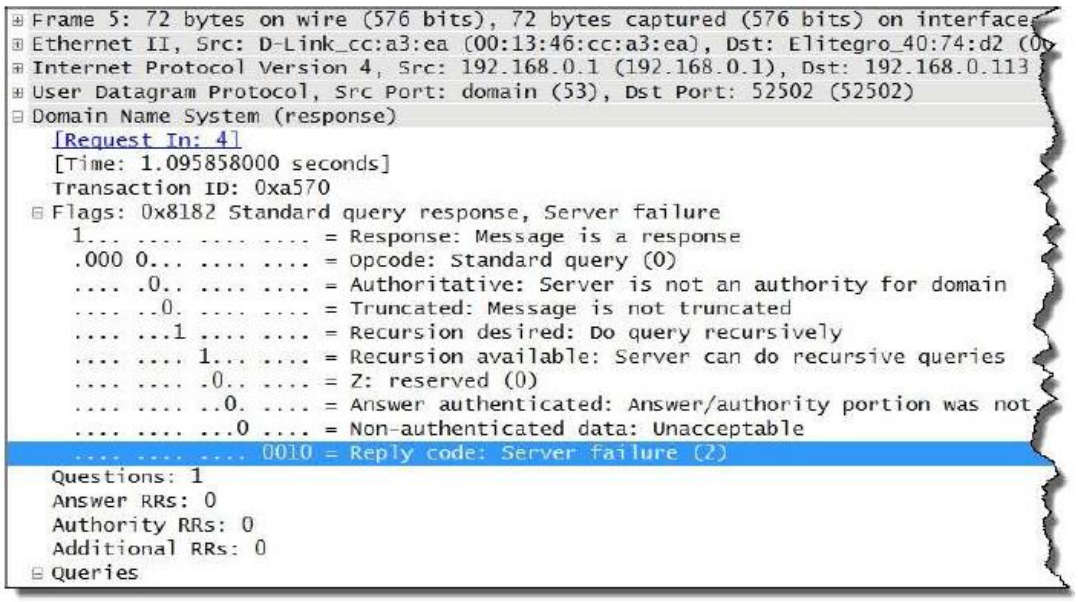
Reply Code 0: NoError (No Error)
Reply Code 1: FormErr (Format Error)
Reply Code 2: ServFail (Server Failure)
Reply Code 3: NXDomain (Non-Existent Domain)
Reply Code 4: NotImp (Not Implemented)
Reply Code 5: Refused (Query Refused)
Reply Code 6: YXDomain (Name Exists when it should not)
Reply Code 7: YXRRSet (RR Set Exists when it should not)
Reply Code 8: NXRRSet (RR Set that should exist does not)
Reply Code 9: NotAuth (Server Not Authoritative for zone (RFC 2136))
Reply Code 9: NotAuth (Not Authorized (RFC 2845))
Reply Code 10: NotZone (Name not contained in zone)
Reply Code 16: BADVERS (Bad OPT Version (RFC 6891))
Reply Code 16: BADSIG (TSIG Signature Failure (RFC 2845))
Reply Code 17: BADKEY (Key not recognized)
Reply Code 18: BADTIME (Signature out of time window)
Reply Code 19: BADMODE (Bad TKEY Mode)
Reply Code 20: BADNAME (Duplicate key name)
Reply Code 21: BADALG (Algorithm not supported)

Tässä tehtävässä keskitytään kahteen yleisimpään DNS virheilmoituksiin:

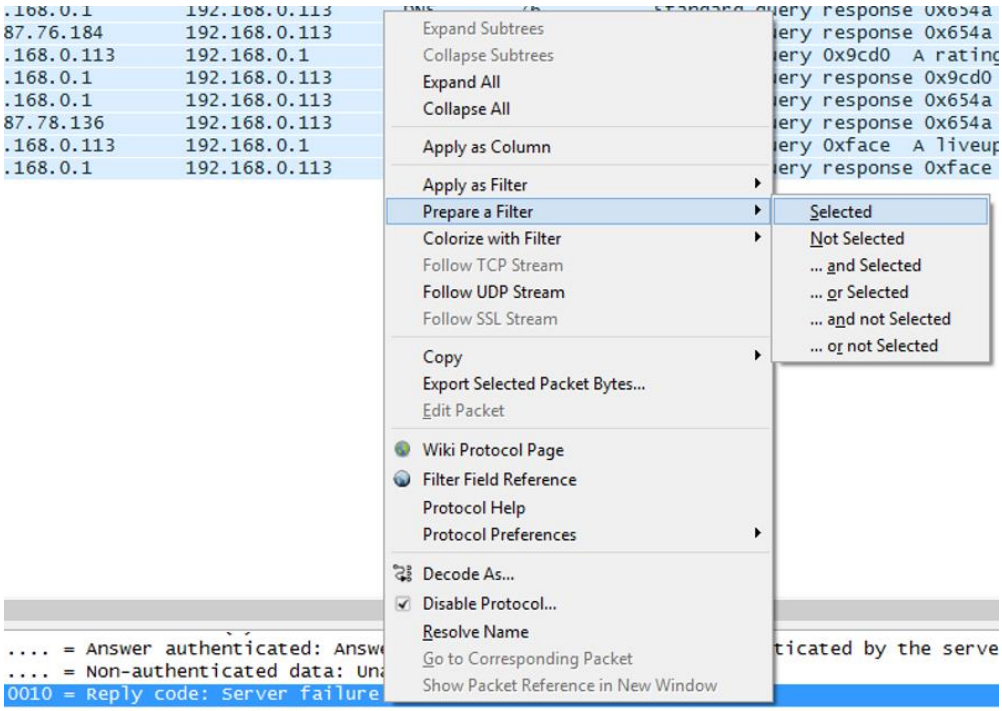
- Palvelinongelmaan (Reply Code 2: Server Failure)
- Virheelliseen DNS-nimeen (Reply Code 3: Non-Existent Domain)

Palvelinongelmat viittavaat siihen, että vastaava DNS palvelin ei saa vastauksia toiselta DNS-palvelimelta. Ongelma sijaitsee siis työasemalle vastaavasta nimipalvelimesta kauempana. Nimivirheet puolestaan viittaavat siihen, että nimeä ei saaada selvitettyä IP-osoitteeksi. Tässä jälkimmäisessä tapauksessa siis itse DNS-prosessi on toiminut oikein, mutta yksikään DNS palvelin ei voinut palauttaa pyydettyä tietoa.

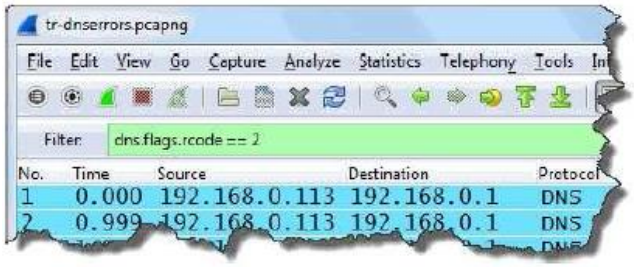
1. **Avaa tr-dnserrors.pccapng.**
2. **Klikkaa pakettia 5**. Tämä on ensimäinen DNS-vastaus tässä trace-tiedostossa. Info-sarakkeesta näet, että tämä on ensimmäinen palvelinvirheviesti (Server Failure reply)
3. **Klikkaa hiiren oikealla napilla DOmain Name System (response) riviä ja valitse Expand Subtrees.** DNS vastauskoodikenttä on Flags kohdan sisällä tässä DNS-vastausviestissä.



4. **Klikkaa hiiren oikealla napilla Packet Details ruudussa kohtaa Reply Code ja valitse Prepare a Filter | Selected**

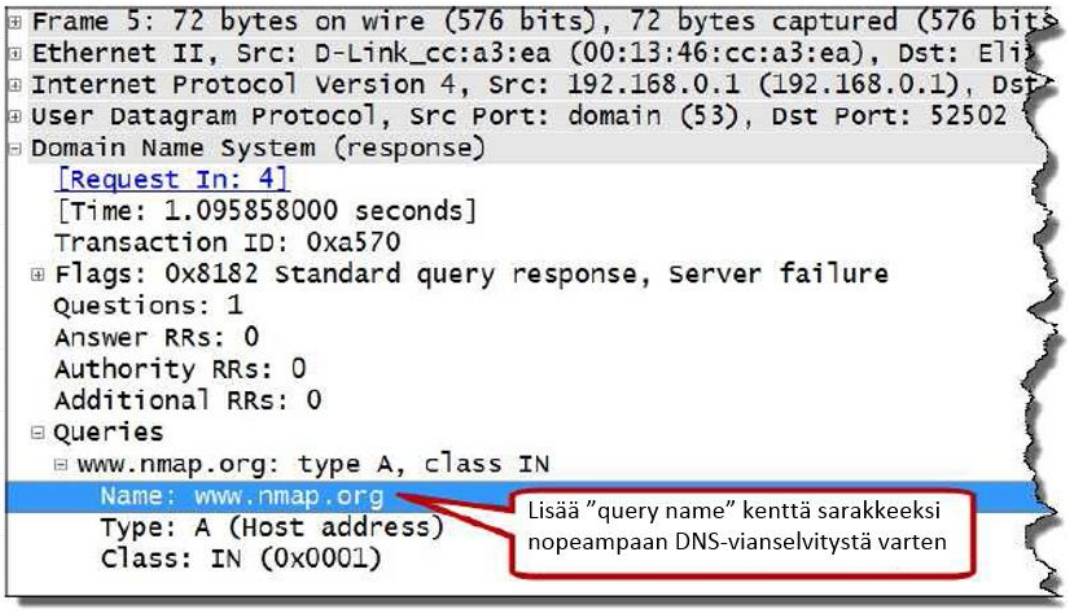


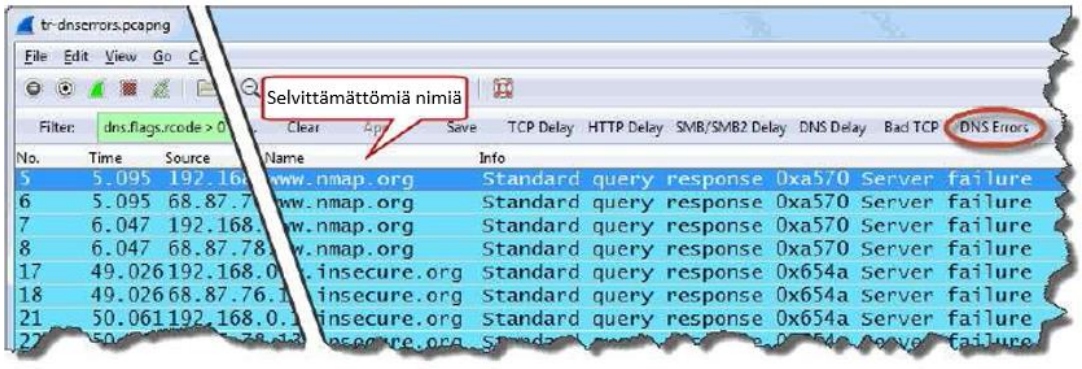
Tämä suodatin löytäisi vain "Server Failures" tyylisiä DNS virheitä. Haluamme luoda napin, joka löytää minkä tahansa tyyppisiä DNS virheitä. Näissä DNS virhepaketeissa on korkeampi kuin 0 arvo **dns.flags.rcode** kentässä.



5. Muokkaa näkymäsuodatinta niin, että siinä lukee **dns.flags.rcode > 0 ja klikkaa sitten Save nappia. Syötä tälle nimeksi DNS Errors ja paina OK.**

6. **Paina nyt uutta DNS Errors nappia** löytääksesi kahdekan DNS-virhettä tässä trace-tiedostossa.
7. Päätelläksesi nopeasti mitkä nimet saivat nämä vastaukset aikaan: laajenna Queries kohta Packet Details ruudulla Klikkaa hiiren oikealla napilla **Name** kenttää ja valitse **Apply as Column**





8. Tässsä tapauksessa näyttää siltä, että upstream DNS-palvelin meidän paikalliselta DNS-palvelimelta ei vastaa rekursiivisiin DNS-kyselyihin. Työasema ei saa nimeä www.nmap.org tai nimeä www.insecure.org selvitettyä, koska upstream DNS-palvelimella on ongelmia.