Laboratorijska vježba broj 7

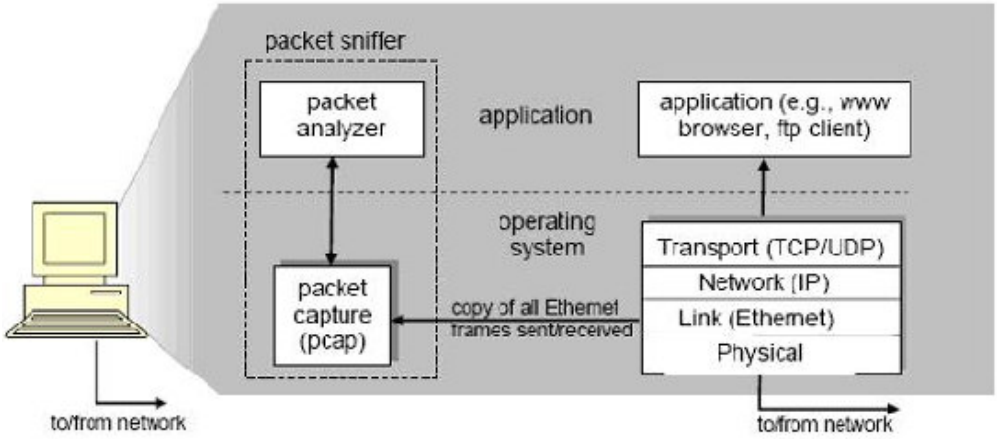
WIRESHARK – PACKET SNIFFER

Teorijska osnova vježbe:

Za bolje shvatanje mrežnih protokola najbolje je "posmatranje protokola u akciji, odnosno, posmatranje sekvenci poruka koje se razmjenjuju između dva entiteta i izazivanje protokola da izvršavaju određene akcije, a zatim posmatranje tih akcija i njihovih posljedica. Ovo se može uraditi pomoću simulacionih scenarija ili u "stvarnom" mrežnom okruženju kao što je Internet.

Osnovni alat za posmatranje poruka koje se razmjenjuju između izvršnih protokol entiteta naziva se ***packet sniffer*** (njuškalo paketa). Kao što ime kaže, *packet sniffer* hvata ("njuši") poruke koje se šalju ili primaju na računar, i prikazuje polja različitih protokola u ovim "uhvaćenim" porukama. *Packet sniffer* je sam po sebi pasivan program. On posmatra poruke koje su poslate ili primljene od strane aplikacija i protokola na računar, ali nikad ne šalje pakete sam. Slično, primljeni paketi nikada nisu eksplicitno adresirani na *packet sniffer*. Umjesto toga, packet sniffer prima kopije paketa koje su poslate/primljene na aplikacije i protokole koji se izvršavaju na računaru.

Slika 6.1 pikazuje strukturu *packet sniffer*-a, a na desnoj strani nalaze se protokoli (u ovom slučaju Internet protokoli) i aplikacije (kao što su web browser-i ili ftp klijent) koji se normalno nalaze na računaru.



Slika 6.1. Struktura packet sniffer-a

Packet sniffer, označen isprekidanim pravougaonikom na slici 6.1, je dodatak uobičajenom softveru na računaru i sastoji se od dva dijela.

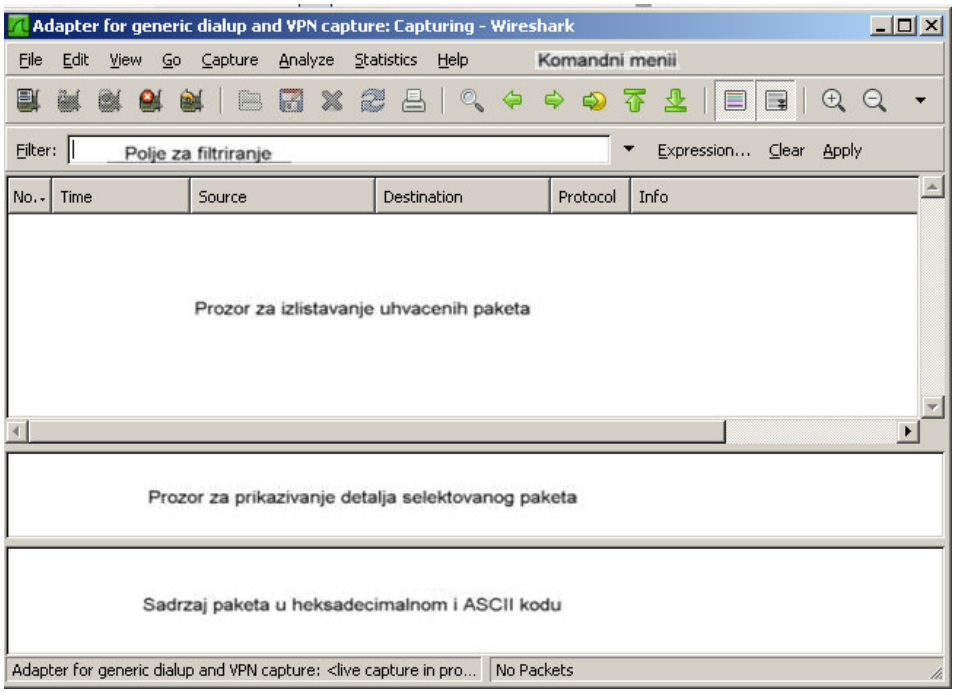
- Prvi dio čini ***packet capture library***. On je zadužen da prima kopiju svakog okvira nivoa linka (*link-layer frejma*) koji je primljen na računaru, ili se šalje sa računara. Inače, poruke se razmjenjuju pomoću protokola viših nivoa kao što su HTTP (*Hyper-Text Transfer Protocol*), FTP (*File Transfer Protocol*), TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), DNS (*Domain Name System*) ili IP (*Internet Protocol*). Svi oni su sadržani u okviru nivoa linka. Na slici 6.1 pretpostavljeno je da je nivo linka Ethernet i svi protokoli viših nivoa su obuhvaćeni u okviru Ethernet frejma. Prema tome, "hvatajući" sve okvire nivoa linka dobijaju se sve poruke koje su poslate do ili primljene od svih protokola i aplikacija koje se izvršavaju na računaru.

- Sledeća komponenta *packet sniffer*-a je ***packet analyzer***, koji prikazuje sadržaj svih polja u okviru poruke. Da bi to uradio, *packet analyzer* mora "razumjeti" strukturu svih poruka koje se razmjenjuju.

Startovanje Wireshark-a

Wireshark je besplatan mrežni *protocol analyzer* koji radi pod Windows, Linux/Unix i Mac operativnim sistemom.

Kada se pokrene Wireshark program, pojavljuje se Wireshark grafički korisnički interfejs prikazan na slici 6.2. Na početku, u prozorima neće biti nikakvih podataka.



Slika 6.2. Wireshark grafički korisnički interfejs

Wireshark interfejs ima pet osnovnih komponenti:

1. ***Command menus*** (komandni menii) **-** standardni padajući menii koji se nalaze na vrhu prozora. Najznačajni su:

- *File* - omogućava čuvanje "uhvaćenih" paketa ili otvaranje fajla koji sadrži prethodno uhvaćene pakete i napuštanje Wireshark aplikacije
- *Capture* - omogućava početak "hvatanja" paketa

2. ***Packet-listing window*** (prozor za prikazivanje uhvaćenih paketa) **-** prikazuje podatke za svaki "uhvaćeni" paket, uključujući broj paketa (koji mu dodjeljuje Wireshark; ovo nije broj paketa koji se nalazi u zaglavlju bilo kog protokola), trenutak u kojem je paket uhvaćen, adresu izvora i destinacije paketa, tip protokola i specifičnu informaciju o protokolu koja se nalazi u svakom paketu. Lista paketa može biti sortirana po bilo kojoj od ovih kategorija klikom na ime kolone. Polje za tip protokola izlistava protokol najvišeg nivoa koji je poslao ili primio ovaj paket npr. protokol koji je izvor ili krajnji cilj ovog paketa.

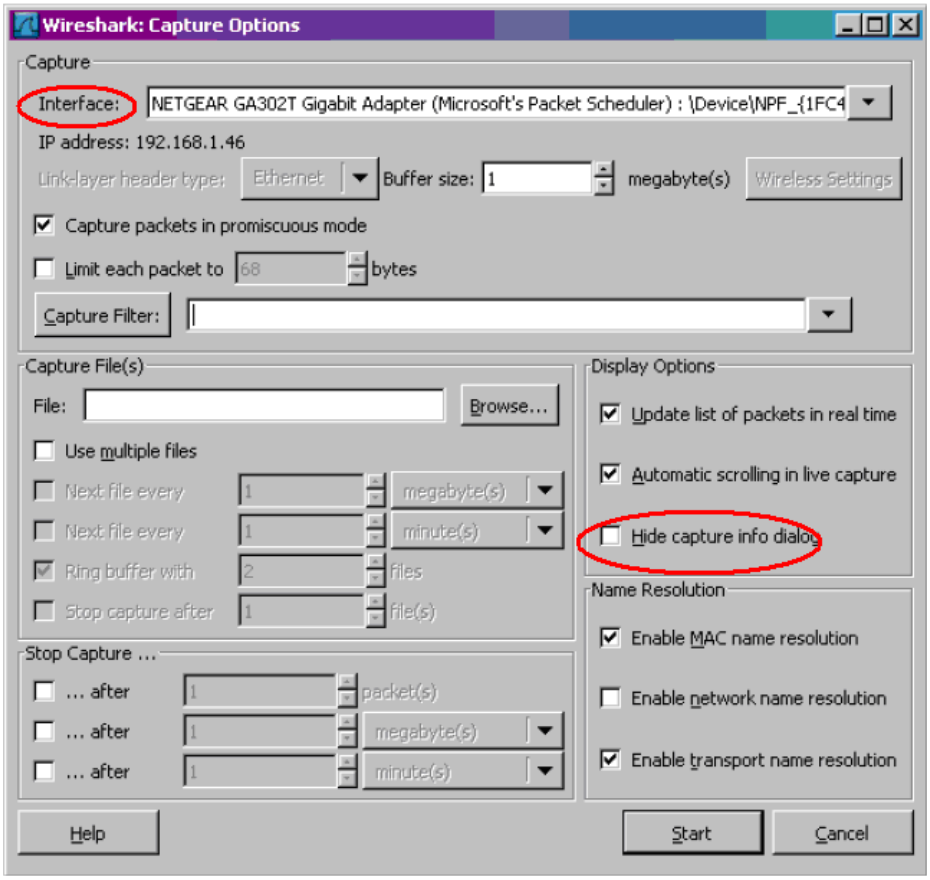
3. ***Packet - header details window*** (prozor za prikazivanje detalja selektovanog paketa) - obezbjeđuje detalje o paketu odabranom u *packet listing* prozoru. (Da bi odabrali paket u *packet listing* prozoru treba kliknuti na red u kome se nalaze podaci o tom paketu). Ovi detalji uključuju informaciju o Ethernet frejmu i IP paketu.

4. ***Packet - contents window*** **-** prikazuje ukupan sadržaj uhvaćenog paketa u ASCII i heksadecimalnom zapisu.

5. Pri dnu Wireshark grafičkog korisničkog interfejsa nalazi se ***Filter*** (polje za filtriranje) u koje se mogu unijeti ime protokola ili druge informacije, kako bi se u packet-listing prozoru (kao i packet–header i packet-contents prozorima) izdvojile samo one informacije koje nas zanimaju.

Testiranje Wireshark-a

- Startovati Internet explorer
- Startovati Wireshark softver, čime se otvara prozor kao na slici 6.2.
- Da bi započelo hvatanje paketa izabrati **Capture** padajući meni i izabrati **Options**. Pojaviće se prozor kao na Slici 6.3.

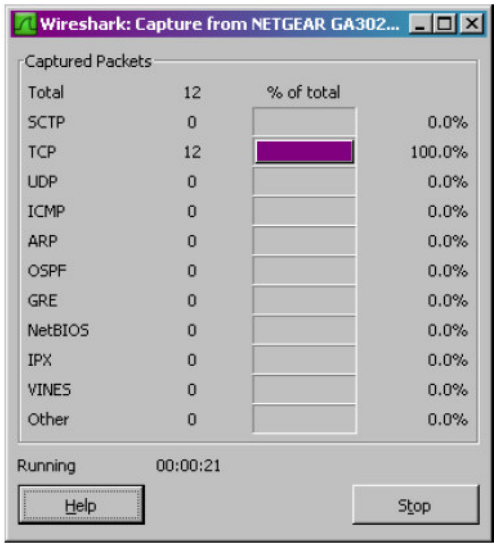


Slika 6.3. Wireshark Capture Options prozor

- Moguće je koristiti default vrijednosti u ovom prozoru, ali je bolje deselektovati „hide capture info dialog“ u *Display Options* polju

- U *Capture options* prozoru odabrati jednu od mrežnih kartica koje se nalaze na računaru (računar može da ima više mrežnih adaptera, npr. wireless i wired Ethernet). Nakon odabira mrežnog interfejsa kliknuti **Start**, čime počinje hvatanje paketa.

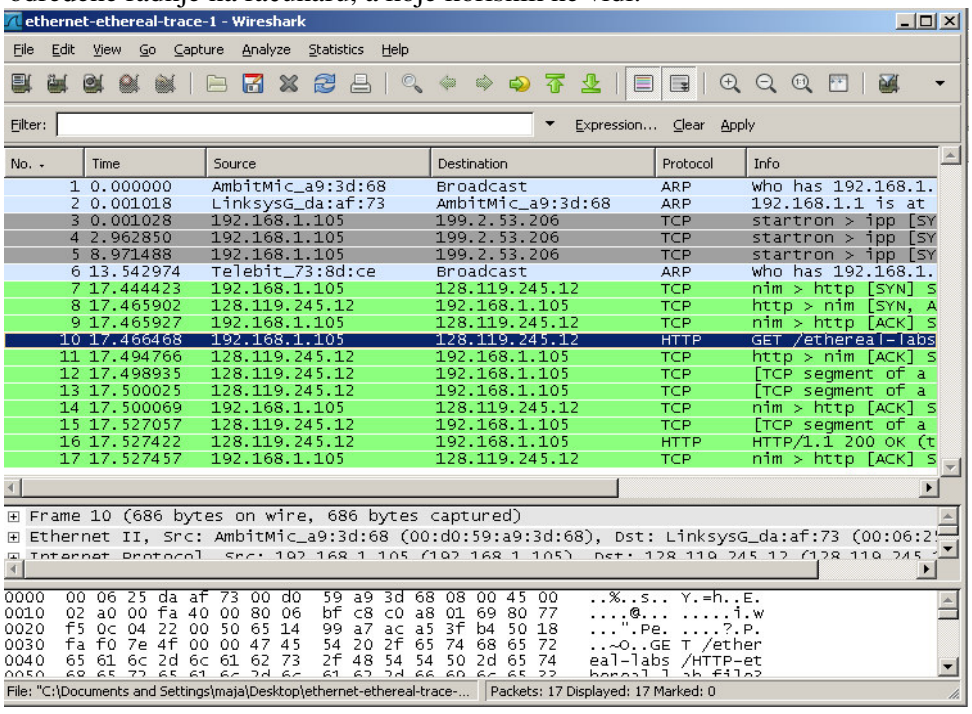
- Kada započne hvatanje paketa, pojaviće se *packet capture summary* prozor, kao što je prikazano na slici 6.4. Ovaj prozor sadrži broj paketa različitih tipova koji su uhvaćeni, i sadrži taster *Stop* koji omogućava prekid hvatanja paketa.



Slika 6.4. Wireshark Packet Capture Window

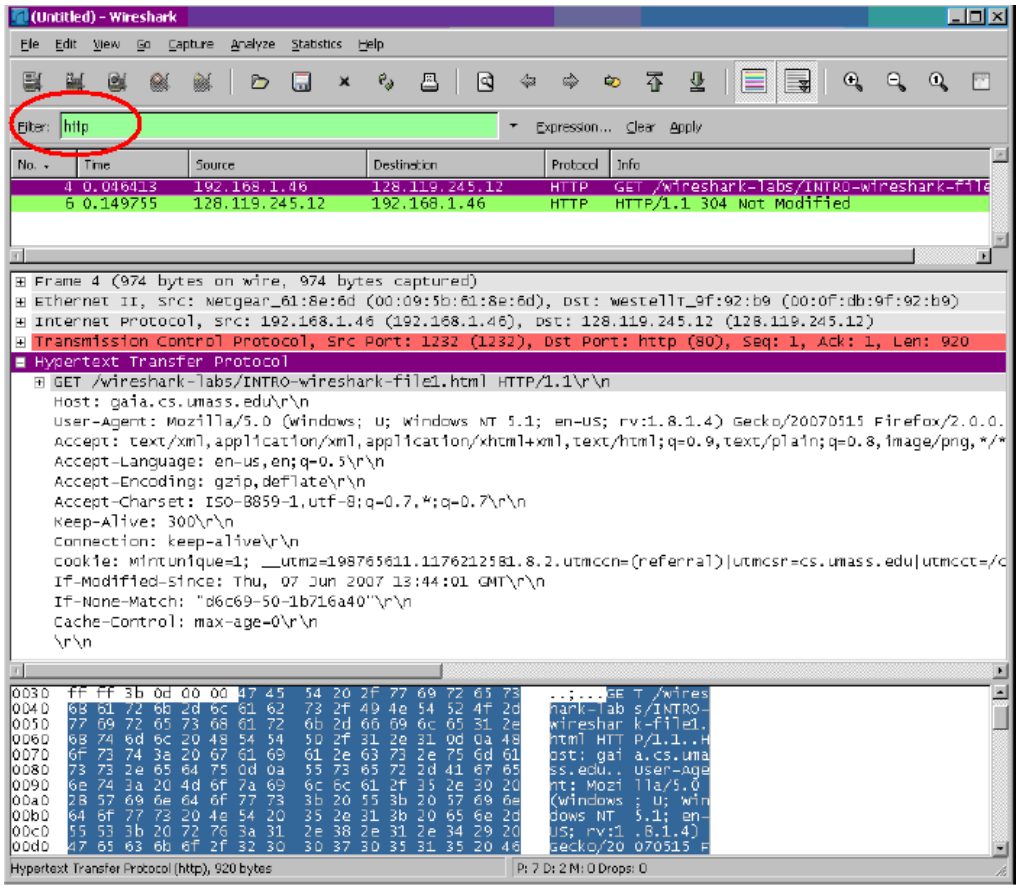
Dok Wireshark obavlja svoj posao, ukucati URL: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html. Da bi se prikazala ova strana, računar će kontaktirati HTTP server i razmijeniti HTTP poruke, koje mu omogućavaju download-ovanje web strane. Ethernet frejmove koji sadrže ove HTTP poruke uhvatiće Wireshark.

- Nakon što browser prikaže INTRO-Wireshark-file1.html stranicu, stopirati Wireshark hvatanje paketa odabirom tastera STOP u *Wireshark capture* prozoru. Ovo će prouzrokovati nestanak Wireshark capture prozora i glavni Wireshark prozor će prikazati sve uhvaćene pakete. Sada imamo podatke o paketima koji sadrže sve poruke razmijenjene između računara i drugih mrežnih entiteta (Slika 6.5). HTTP poruka razmijenjena sa gaia.cs.umass.edu web serverom bi se trebala pojaviti negdje u listi uhvaćenih paketa. Međutim, biće prikazani i mnogi drugi tipovi paketa. Iako je jedina sprovedena radnja download-ovanje web stranice, evidentno je da postoji još mnogo drugih protokola koji obavljaju određene radnje na računaru, a koje korisnik ne vidi.



Slika 6.5. Spisak svih uhvaćenih paketa

• Ukucati "**http**" (bez navodnika i malim slovima – imena svih protokola se pišu malim slovima u Wireshark-u) u *filter* prozoru pri dnu glavnog Wireshark prozora. Zatim kliknuti ***Apply***. Sada će jedino HTTP poruke biti prikazane u packet-listing prozoru (Slika 6.6).



Slika 6.6. HTTP poruke razmijenjene između računara i servera

• Izabrati prvu http poruku prikazanu u packet-listing prozoru. To bi trebala da bude HTTP GET poruka koja je poslata sa računara gaia.cs.umass.edu HTTP server-u. Kada se izabere HTTP GET poruka, Ethernet frejm, IP paket, TCP segment i zaglavlje HTTP poruke će biti prikazani u packet-header prozoru.

Pitanja:

1. Navesti sve protokole koji se pojavljuju u koloni protokola u *packet-listing* prozoru prije filtriranja.
2. Koliko vremena protekne od trenutka kada se pošalje HTTP GET poruka do trenutka kada se primi HTTP odgovor? (Po default-u, vrijednosti za vrijeme u koloni vremena u packet-listing prozoru su date u sekundama).
3. Koja je IP adresa gaia.cs.umass.edu servera?
4. Koja je IP adresa računara na kojem radite?

Wireshark Lab: HTTP

U ovom dijelu vježbe ispitivaće se nekoliko aspekata HTTP protokola: osnovna GET/"odgovor" interakcija, format HTTP poruke, pozivanje velikih HTML fajlova, kao i pozivanje HTML fajlova sa ugrađenim objektima.

Osnovna GET/"odgovor" interakcija

Potrebno je uraditi sledeće:

- Startovati web browser (duplim klikom miša na ikonu **Internet explorer**-a, ili **Start > Programs > Internet explorer**).
- Startovati Wireshark packet sniffer, ali još ne započinjati hvatanje paketa. U *Filter* prozoru unijeti "http" (bez navodnika), kako bi se samo uhvaćene HTTP poruke kasnije prikazale u *packet-listing* prozoru.
- Sačekati malo više od jednog minuta, a zatim otpočeti Wireshark hvatanje paketa.
- U browser unijeti adresu: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html. Prikazuje se vrlo jednostavan HTML fajl koji se sastoji od samo jednog reda.
- Zaustaviti Wireshark hvatanje paketa.

Wireshark prozor bi trebao da bude sličan onom prikazanom na slici 6.6. Ukoliko niste u mogućnosti da pokrenete Wireshark na aktivnoj Internet konekciji, možete downloadovati gotovi packet trace fajl sa adrese http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip i ekstraktovati fajl http-ethereal-trace-1.

Primjer na slici 6.6 prikazuje da su u packet-listing prozoru uhvaćene dvije HTTP poruke: GET poruka (od browser-a do gaia.cs.umass.edu web servera) i odgovor servera browser-u. Packet-contents prozor prikazuje detalje selektovane poruke (u ovom slučaju HTTP GET poruke, koja je označena u packet-listing prozoru). Pošto se HTTP poruka nalazi u TCP segmentu, koji je je u okviru IP paketa, a koji je opet u okviru Ethernet frejma, Wireshark prikazuje, takođe, Frejm, Ethernet, IP i TCP informacije o paketu.

Potrebno je minimizovati broj prikazanih podataka koji ne pripadaju HTTP protokolu. U tom cilju provjeriti da li se u prozoru za prikazivanje detalja uhvaćenih paketa, lijevo od prikaza informacija za Frejm, Ethernet, IP i TCP nalazi znak plus u kvadratiću (što znači da postoje skrivene, neprikazane informacije), kao i da li se ispred HTTP linije nalazi znak minus (koja ukazuje na to da su sve informacije o HTTP poruci prikazane).

(*Napomena:* Treba ignorisati bilo koji HTTP GET ili odgovor za favicon.ico).

Pitanja:

Posmatrajući informacije u HTTP GET i "odgovor" poruci, odgovoriti na sljedeća pitanja.

5. Da li vaš browser podržava HTTP verziju 1.0 ili 1.1? Koju verziju HTTP-a podržava server ?
6. Koji jezik (ako postoji neki) browser pokazuje da može biti prihvaćen od strane servera?
7. Koji je status kod vraćen od servera ka vašem računaru?
8. Koji format podataka za slike, a koji za aplikacije, može biti prihvaćen od strane browsera?
9. Kada je HTML fajl koji je pozvan posljednji put modifikovan na serveru?
10. Koliki sadržaj u bajtima se vratio browser-u?

HTTP CONDITIONAL GET/"odgovor" interakcija

Mnogi web browser-i vrše keširanje (smještanje poslednjih pozivanih stranica u memoriju) objekata. U tom slučaju se vrši conditional GET kada se poziva HTTP objekat. Prije početka izvršavanja sljedećih koraka, provjeriti da li je keš vašeg browser-a prazan. (odaberite ***Tools-***

>Internet Options->Delete File*;* na ovaj način biće izbrisani svi keširani fajlovi iz vašeg browser-a.) U narednom dijelu:

- Startovati web browser i provjeriti da li je obrisan browser keš, kao što je prethodno opisano.
- Startovati Wireshark packet sniffer.
- Ukucati sljedeći URL u browser:
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html
- Browser bi trebao da prikaže vrlo jednostavan HTML fajl od pet redova.
- Kliknuti na refresh dugme u browser-u (ili brzo ukucati ponovo u browser isti URL)
- Nakon što se stranica ponovo učita zaustaviti Wireshark hvatanje paketa, i ukucati "http" u display-filter-specification prozoru, tako da samo uhvaćene HTTP poruke budu kasnije prikazane u packet-listing prozoru.

Pitanja:

11. Pregledati sadržaj prvog HTTP GET zahtjeva od browser-a do servera. Da li postoji "IF-MODIFIED-SINCE" linija u tom HTTP GET zahtjevu?
12. Pregledati sadržaj odgovora servera. Da li je server eksplicitno vratio sadržaj fajla? Šta se može zaključiti?
13. Pogledati sadržaj drugog HTTP GET zahtjeva od browser-a do servera. Da li postoji "IF-MODIFIED-SINCE:" linija u ovom HTTP GET zahtjevu? Ako postoji, koje informacije prate "IF-MODIFIED-SINCE:" zaglavlje?
14. Koji je HTTP status kod i fraza vraćena od servera kao odgovor na ovaj drugi HTTP GET? Da li je server eksplicitno vratio sadržaj fajla? Objasniti.

Pozivanje velikih dokumenata

U primjerima do sada, pozivani dokumenti bili su jednostavni i kratki HTML fajlovi. Postavlja se pitanje šta će se desiti kada se download-uje veliki HTML fajl. Potrebno je uraditi sledeće:

- Startovati web browser, provjeriti da li je keš vašeg browser-a obrisan, kao što je ranije objašnjeno.
- Startovati Wireshark packet sniffer
- Ukucati sljedeći URL u vaš browser:
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html
  Vaš browser bi trebalo da prikaže prilično dugačak US Bill of Rights
- Zaustaviti Wireshark hvatanje paketa i unijeti "http" u display-filter-specification prozor, tako da se prikažu samo HTTP poruke

U prikazanom packet-listing prozoru nalazi se HTTP GET poruka, koju prati odgovor na HTTP GET zahtjev, a koji se sastoji od više paketa. HTTP odgovor se sastoji od status linije, koju prati linija zaglavlja, praćena praznom linijom koja je praćena tijelom entiteta – jezgro (entity body). U slučaju HTTP GET poruke, jezgro u odgovoru je čitav zahtijevani HTML fajl. U analiziranom slučaju HTML fajl je prevelik da bi stao u jedan TCP paket. Zbog toga se jedna poruka HTTP odgovora razbija na nekoliko djelova uz pomoć TCP-a, tako da se svaki dio sadrži u okviru odvojenih TCP segmenata. Svaki TCP segment Wireshark snima kao odvojen paket, pa na činjenicu da je jedan HTTP odgovor izdjeljen na više TCP paketa ukazuje "Continuation" fraza prikazana od strane Wireshark-a. Treba zapamtiti da **nema nikakve "Continuation" poruke u HTTP-u!**

Pitanja:

15. Koliko je HTTP GET poruka zahtjeva poslato od strane web browser-a?
16. Koliko TCP segmenata koji sadrže podatke je potrebno da se prenese jedan HTTP odgovor?
17. Koja je veličina fajla koji server šalje browser-u? Kolika je veličina pojedinačnih TCP segmenata?

HTML dokument sa ugrađenim objektima

Pošto je objašnjeno kako Wireshark prikazuje pakete uhvaćenog saobraćaja za veliki HTML fajl, može se pogledati šta se dešava kada browser download-uje fajl sa ugrađenim objektima, tj. fajlove koji uključuju druge objekte (u narednom primjeru to su image fajlovi) koji su smješteni na drugom serveru(ima).
Potrebno je pratiti sledeće korake:

- Startovati web browser, a zatim provjeriti da li je keš browser-a obrisan, kao što je ranije objašnjeno.
- Startovati Wireshark packet sniffer
- Ukucati sljedeći URL u vaš browser: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html . Browser prikazuje kratak HTML fajl sa dvije slike. Ove dvije slike su vezane za HTML fajl. To znači da slike nisu sadržane u HTML fajlu, već su URL-ovi za ove slike smještani u download-ovanom HTML fajlu. Browser će morati pozvati ove logo-e iz naznačenih web sajtova. Logo izdavača je pozvan sa www.awl.com web sajta. Slika omota tražene knjige je smještena na manic.cs.umass.edu serveru.
- Zaustaviti Wireshark hvatanje paketa i ukucati "http" u display-filter-specification prozoru, tako da samo HTTP poruke budu prikazane.

Pitanja:

18. Koliko je HTTP GET poruka zahtjeva poslato od strane vašeg browser-a? Na koju Internet adresu su poslati ovi GET zahtjevi?
19. Da li je vaš browser download-ovao dvije slike serijski ili su download-ovane sa dva web sajta paralelno? Objasnite.