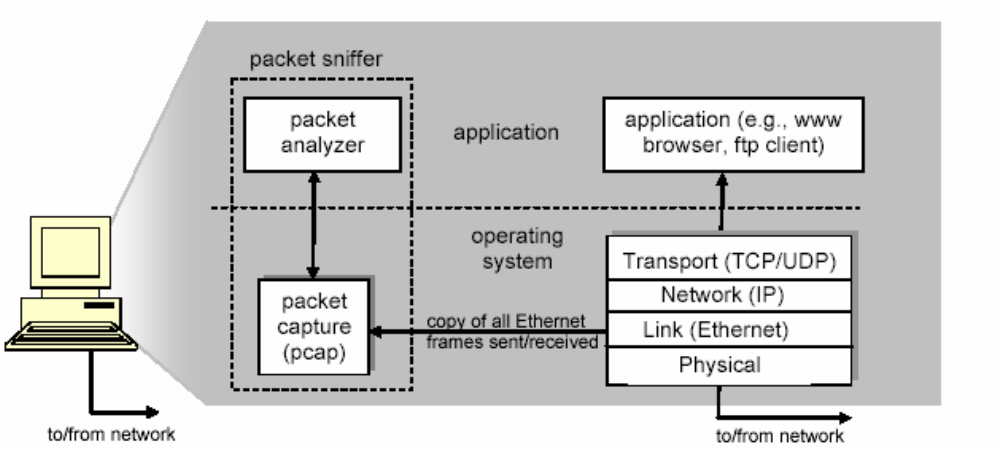
Lab 2: WIRESHARK

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 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 1. Struktura packet sniffer-a.

Packet sniffer, označen isprekidanim pravougaonikom na slici 1, je dodatak uobičajenom softveru na računaru i sastoji se od dva dijela:

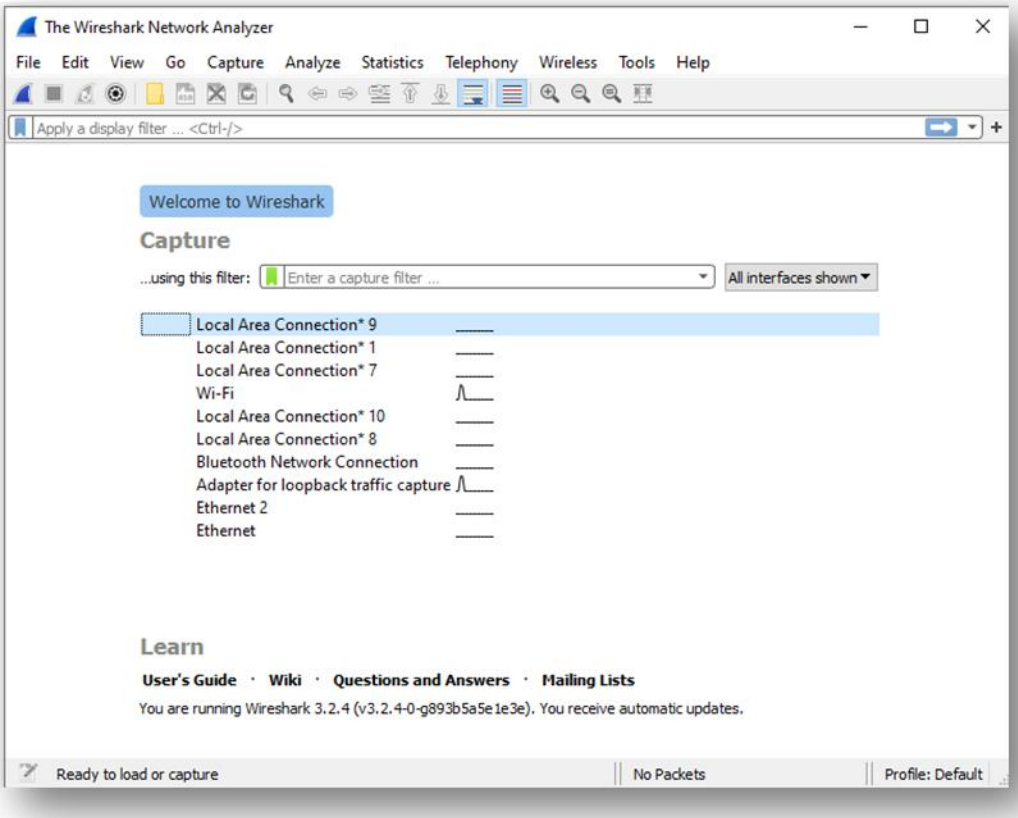
- ***packet capture library*** – zadužen je da prima kopiju svakog okvira nivoa linka (*link-layer frejma*) koji je primljen na računaru, ili se šalje sa računara. Inače, poruke se razmjenjuju pomoću protokola viših nivoa kao što su HTTP (*Hyper-Text Transfer Protocol*), FTP (*File Transfer Protocol*), 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 1 pretpostavljeno je da je nivo linka Ethernet dok su protokoli viših nivoa 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.

- ***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

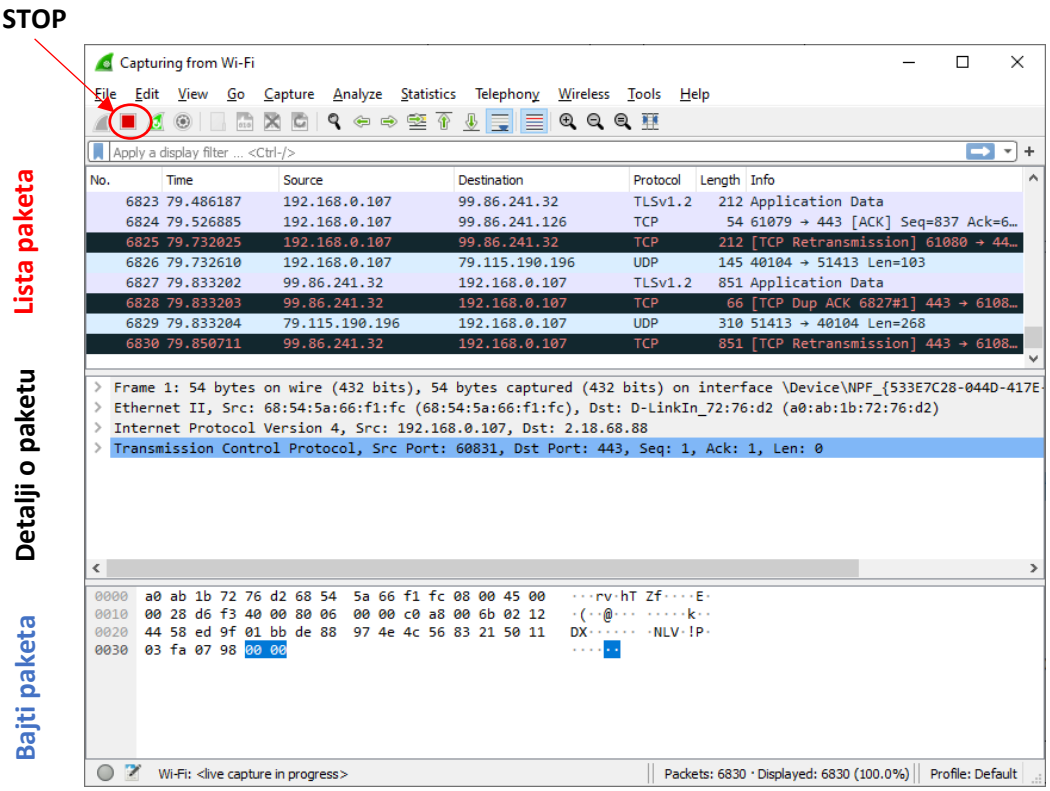
Wireshark je besplatan mrežni *protocol analyzer* koji radi pod Windows, Linux/Unix i Mac operativnim sistemom. Kada se pokrene Wireshark program, pojavljuje se Wireshark grafički korisnički interfejs prikazan na slici 2. Ukoliko koristite neki od Linux operativnih sistema, Wireshark može pokrenuti sa root privlegijama iz komandne linije:

```
$ sudo wireshark
```



Slika 2. Wireshark korisnički interfejs.

Nakon pokretanja Wireshark-a, možete direktno na radnoj površini odabrati mrežne interfejs(e) sa kojih želite da snimate saobraćaj. Kada započne hvatanje paketa, pojaviće se glavni Wirehshark prozor, kao što je prikazano na slici 3. Ovaj prozor sadrži broj paketa različitih tipova koji su uhvaćeni, i sadrži top taster koji omogućava prekid hvatanja paketa.



Slika 3. Glavni Wireshark prozor.

Tri ključne sekcije glavnog prozora su:

1. **Lista paketa** (sekcija na vrhu). Za svaki paket se prikazuju sledeće informacije:
    - Redni broj paketa (u odnosu na ukupan broj uhvaćenih paketa)
    - Trenutak u kojem je paket uhvaćen, pretpostavljajući da je vrijeme hvatanja prvog paketa 0.0
    - Izvorišnu i destinacionu IP adresu paketa
    - Protokol najvišeg reda koji je pridružen paketu.
    - Pregled ključnih informacija koje se prenose u paketu.
2. **Detalji o individualnom paketu** (sekcija u sredini). Ovdje se prikazuju detaljne informacije o paketu odabranom u gornjoj sekciji. Informacije su podijeljene po nivoima protokola paketa.
3. **Bajti indvidualnog paketa** (sekcija na dnu). Prikazuje heksadecimalni i ACII sadržaj paketa.

Analiza i statistike

Wireshark vodi evidenciju o brojnim statistikama, što korisniku olakšava analizu uhvaćenog saobraćaja. Ovo je naročito korisno ukoliko se radi o velikom broju paketa. Istražite sledeće opcije iz **Statistics** menija:

- Summary
- Protocol Hierarchy
- Conversations
- Flow Graph
- HTTP
- Packet Length
- TCP Stream Graph

Filtri

Kada se podaci prikupljaju tokom dužeg vremenskog perioda (stotine hiljada paketa), često je potrebno izdvojiti određenu grupu paketa (npr. pakete između određenog para hostova, ili one koji koriste određeni protokol). Wireshark filtri se mogu primijeniti za ove svrhe u toku snimanja saobraćaja - tako da se pikazuju samo oni paketi koji zadovljavaju specificirane kriterijume (*capture* filteri), ili nakon što je snimanje saobraćaja završeno – u cilju analize samo određenog skupa paketa koji zadovoljavaju tzv. *display* filtre.

Display fitri olakšavaju analizu paketa. Mogu se definisati u *input* polju iznad sekcije sa listom paketa. Na primjer, filter:

Može se koristiti za prikaz paketa koji imaju navedenu izvorišnu ili destinacionu IP adresu 10.10.1.171:

```
ip.addr==10.10.1.171
```

Sledeći filtar prikazuje samo pakete koji imaju IP adresu 10.10.1.127 i nemaju TCP broj porta 8080:

```
ip.addr==10.10.1.127 && !tcp.port==8080
```

Sledeći filtar prikazuje samo ICMP pakete:

```
icmp
```

Filtar koji prikazuje samo pakete razmijenjene sa web serverima (pod pretpostavkom da web serveri koriste port 80):

```
tcp.port=80
```

Detaljne informacije o Wireshark filtrima možete naći u Wireshark dokumentaciji.

Zadaci za samostalni rad:

1. Snimanje saobraćaja

Pomoću Wireshark-a snimiti pakete koji se prenose prilikom pretraživanja web stranice http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html. Istražite protokole koji se koriste u svakom paketu, vrijednosti polja u zaglavljima i veličinu paketa.

2. Osnovna GET/"odgovor" interakcija

Od saobraćaja koji je uhvaćen u prethodnom poraku isfiltrirati samo HTTP pakete. Odgovorite na sledeća pitanja:

- 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 su date u sekundama).
- Koja je IP adresa gaia.cs.umass.edu servera?
- Koja je IP adresa računara na kojem radite?

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

- Da li vaš pretraživač podržava HTTP verziju 1.0 ili 1.1? Koju verziju HTTP-a podržava server?
- Koji jezik (ako postoji neki) pretraživač pokazuje da može biti prihvaćen od servera?
- Koji je status kod vraćen od servera ka vašem računaru?
- Koji format podataka za slike, a koji za aplikacije, pretraživač može prihvatiti?
- Kada je zahtijevani HTML fajl poslednji put modifikovan na serveru?
- Koliko bajta sadržaja je preuzeto?

3. HTTP CONDITIONAL GET/"odgovor" interakcija

Mnogi web pretraživači vrše keširanje (smještanje poslednjih pozivanih stranica u memoriju) objekata. U tom slučaju se šalje conditional GET kada se povlači HTTP objekat. Prije početka izvršavanja sljedećih koraka, provjeriti da li je keš vašeg pretraživača prazan. Ukoliko nije, ispraznite ga. U narednom dijelu:

- Startovati web pretraživač i provjeriti da li je obrisan keš;
- Startovati Wireshark packet sniffer;
- Ukucati sljedeći URL u pretraživač: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html. Pretraživač bi trebao da prikaže vrlo jednostavan HTML fajl od pet redova;
- Kliknuti na *refresh* dugme u pretraživaču;
- Nakon što se stranica ponovo učita zaustaviti Wireshark hvatanje paketa, i ukucati "http" u display-filter prozoru, tako da samo uhvaćene HTTP poruke budu kasnije prikazane u listi paketa.

Odgovorite na sledeća pitanja:

- Pregledati sadržaj prvog HTTP GET zahtjeva od pretraživača do servera. Da li postoji "IF-MODIFIED-SINCE" linija u tom HTTP GET zahtjevu?
- Pregledati sadržaj odgovora servera. Da li je server eksplicitno vratio sadržaj fajla? Šta se može zaključiti?
- Pogledati sadržaj drugog HTTP GET zahtjeva od pretraživača do servera. Da li postoji "IF-MODIFIED-SINCE:" linija u ovom HTTP GET zahtjevu? Ako postoji, koje informacije prate "IF-MODIFIED-SINCE:" zaglavlje?
- 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.

4. Povlačenje velikih dokumenata

U primjerima do sada pozivani dokumenti bili su jednostavni i kratki HTML fajlovi. Postavlja se pitanje šta se dešava kada se preuzima veliki HTML fajl. Potrebno je uraditi sledeće:

- Startovati web pretraživač i ispraznit njegov keš;
- Startovati Wireshark;
- Ukucati sledeći URL u pretraživač: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html. Pretraživač bi trebalo da prikaže prilično dugačak *US Bill of Rights* sadržaj;
- Zaustaviti Wireshark hvatanje paketa i unijeti "http" u display-filter prozor, tako da se prikažu samo HTTP poruke

U analiziranom sluča ju 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 u Wireshark-u. Bitno je naglasiti da nema nikakve "Continuation" poruke u HTTP-u.

Odgovoriti na sledeća pitanja:

- Koliko je HTTP GET poruka je web pretraživač poslao?
- Koliko TCP segmenata koji sadrže podatke je potrebno da se prenese svaki od HTTP odgovora?
- Koja je veličina fajla koji server šalje pretraživaču? Kolika je veličina pojedinačnih TCP segmenata?

5. Povlačenje HTML dokumenta sa ugrađenim objektima

Istražite šta se dešava kada pretrživač preuzima HTML fajl sa ugrađenim objektima (u narednom primjeru ugrađeni objekti su slike) koji su smješteni na drugom serveru.

Potrebno je pratiti sledeće korake:

- Startovati web pretraživač i isprazniti njegov keš;
- Startovati Wireshark;
- Ukucati sljedeći URL u web pretraživač: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html. Pretraživač bi trebalo da prikaže 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 samo njihovi URL-ovi.

Odgovoriti na sledeća pitanja:

- Koliko je HTTP GET poruka je vaš web pretraživač poslao? Na koju Internet adresu su poslati ovi GET zahtjevi?
- Da li je vaš pretraživač preuzeo dvije slike serijski ili su preuzete sa dva web sajta paralelno? Objasnite.

6. U Wiresharku istražite: filtre, Stream Graphs (TCP), statistike, hijerhije protokola.