Vežba 10 – Statistička analiza mrežnog saobraćaja upotrebom WinPcap (Windows) programske biblioteke

1. Statistički način rada mrežnog adaptera

WinPcap (Windows OS) pruža mogućnost praćenja statistike saobraćaja na mreži korišćenjem programibilnog modula za monitoring na nivou kernela. Sakupljanje podataka i računanje statistike se procesira na nivou kernela, tako da aplikacija direktno dobija rezultate statistike. Time se izbegava potreba za kopiranjem pojedinačnih paketa aplikaciji, što pozitivno utiče na opterećenje procesora i zauzeće memorije prilikom sakupljanja statistike.

Za računanje statistike koriste se komponenta za filtriranje i brojači. Korisniku se pruža informacija o broju paketa koji su zadovoljili uslov i njihovoj ukupnoj veličini u bajtima. Period dobavljanja statistike kao i izrazi koji se koriste u filterima su konfigurabilni.

Da bi se omogućila upotreba statističke analize, potrebno je otvoriti adapter i staviti ga u statistički režim rada (*mode*). Ovo se vrši pomoću funkcije *pcap_setmode()* gde se kao parametar funkcije prosleđuje MODE_STAT.

```
int pcap_setmode (pcap_t* device_handle, int mode);
```

Funkcija:	Opis:
pcap_setmode	Funkcija za postavljanje načina rada adaptera.
Parametri:	**Opis:**
pcap_t* device_handle	Deskriptor mrežnog adaptera.
int mode	Parametar za definisanje načina rada adaptera. MODE_CAPT : režim rada za hvatanje paketa (podrazumevani) MODE_STAT : statistički režim rada
Povratna vrednost:	**Opis:**
int	Ako se uspešno izvrši vraća 0. U suprotnom, vraća -1.

Primer:

```
/*-------------------------------------------------------------------------------*/
// Put the network adapter in statstics mode
if (pcap_setmode(device_handle, MODE_STAT) < 0)
{
	printf("\nError setting the mode.\n");
	pcap_close(device_handle);

	// Free the device list
	return;
}
/*-------------------------------------------------------------------------------*/
```

2. Statistička analiza

Prilikom statističke analize mrežnog saobraćaja, programer ima mogućnost da postavi filter koji će definisati koji podskup mrežnog saobraćaja se nadgleda i čiji statistički podaci se skupljaju. Ovo postavljanje filtera se vrši pre nego što adapter stavimo u statistički režim rada. Ukoliko se ne postavi nikakav filter, nadgledaće se celokupni mrežni saobraćaj.

Tipični koraci prikom statističke analize saobraćaja su:

1. Postavljanje filtera npr. želimo samo da nadgledamo *tcp* saobraćaj:

   ```
   pcap_compile(device_handle, &fcode, "tcp", 1, netmask);
   pcap_setfilter(device_handle, &fcode);
   ```

2. Stavljanje adaptera u statistički režim rada (*pcap_setmode()*):

   ```
   pcap_setmode(device_handle, MODE_STAT);
   ```

3. U okviru petlje (pomoću funkcije *pcap_loop()*) poziva se callback funkcija, na primer: *statistics_handler()*:

   ```
   pcap_loop(device_handle, 0, statistics_handler, (unsigned char*)&timestamp);
   ```

Pokazivač na strukturu *timeval* se prosleđuje funkciji *pcap_loop()* kao njen četvrti parametar. Ova struktura se koristi za smeštanje vremenske oznake („timestamp“) uzorka radi kasnijeg izračunavanja intervala između dva uzastopna uzorka.

Prilikom otvaranja adaptera definiše se parametar *timeout* koji nam služi da označimo vreme nakon koga se uhvaćeni paketi iščitavaju (to je 4. parametar pri pozivu funkcije *pcap_open_live()*). Ako se ovaj parametar postavi na vrednost npr. 1000 ms, to znači da se funkcija *statistics_handler()* poziva jednom u sekundi i da dobija statističke uzorke izračunate od strane drajvera svakih 1000 milisekundi. Ovi odbirci su smešteni u okviru 2. i 3. parametra callback funkcije, kao što je prikazano na Slici 1.

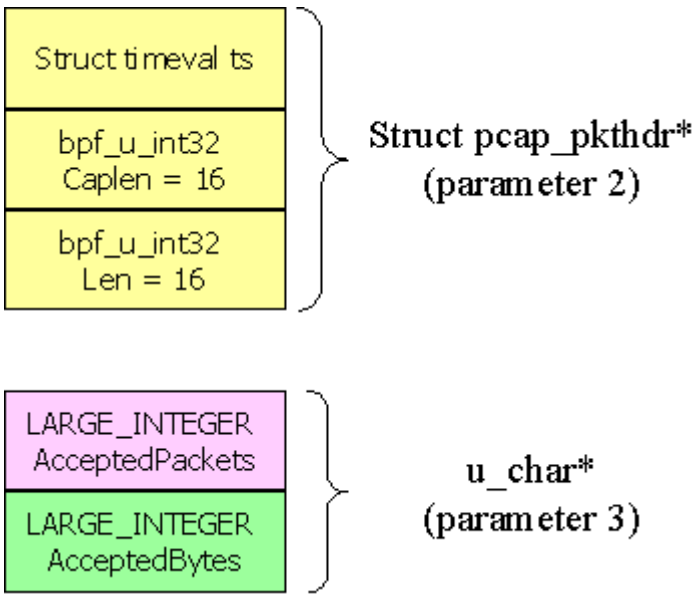
```
void statistics_handler(unsigned char* user_param, const struct pcap_pkthdr*
packet_header, const unsigned char* packet_data);
```

Parametri funkcije *statistics_handler()*:

1. `unsigned char* user_param` - za ovaj parametar se prosleđuje vremenska oznaka prethodnog uzorka paketa.
2. `const struct pcap_pkthdr* packet_header` - pokazivač na generičko zaglavlje. Iz generičkog zaglavlja koristi se vrednost polja *ts* koje nosi vremensku oznaku („timestamp“) za tekuće uzorke paketa.
3. `const unsigned char* packet_data` – pokazivač na prikupljene statističke podatke u vidu dva 64-bitna brojača (dva podatka tipa *long long*).
   - Prvi brojač (označen sa *AcceptedPackets* na Slici 1.) služi za čuvanje broja uhvaćenih paketa.
   - Drugi brojač (označen sa *AcceptedBytes* na Slici 1.) služi za čuvanje broja primljenih bajtova tokom zadnjeg vremenskog intervala.

Funkcija *statitstics_handler()* koristi informaciju o intervalu između dva uzorka da bi se za nadgledani saobraćaj mogla izračunati vrednost bita u sekundama i paketa u sekundama. Izračunavanje vrednosti intervala između dva uzastopna statistička uzorka vrši se oduzimanjem vremenske oznake tekućeg uzorka paketa i vremenske oznake prethodnog uzorka paketa.

Vrednost vremenske oznake („timestamp“) za tekuće uzorke paketa je sadržana u polju *ts* generičkog zaglavlja na koje pokazuje parametar *packet_header*. Vremenska oznaka prethodnog uzorka paketa prosleđuje se kao prvi parametar callback funkcije (parametar *state*).



Slika 1. Drugi i treći parametar callback funkcije u slučaju statističkog režima rada adaptera

Primer:

```
/*-------------------------------------------------------------------------------*/
void dispatcher_handler(unsigned char* state, const struct pcap_pkthdr*
packet_header, const unsigned char* packet_data)
{
	struct timeval *old_ts = (struct timeval *)state;
	unsigned int delay;
	LONGLONG bits_per_second, packets_per_second;
	struct tm ltime;
	char timestr[16];
	time_t local_tv_sec;

	// Calculate the delay in microseconds from the last sample.
	// This value is obtained from the timestamp
	//that is associated with the sample.
	delay=(packet_header->ts.tv_sec - old_ts->tv_sec) * 1000000 +
	packet_header->ts.tv_usec - old_ts->tv_usec;
	// Get the number of bits per second
	bits_per_second = (((*(LONGLONG*)(packet_data + 8)) * 8 * 1000000) / (delay));
	/*                                                      ^      ^
	                                                        |      |
	                                converts bytes in bits  --      |
	                              delay is expressed in microseconds --
	*/

	// Get the number of Packets per second
	packets_per_second = (((*(LONGLONG*)(packet_data)) * 1000000) / (delay));

	// Convert the timestamp to readable format
	local_tv_sec = packet_header->ts.tv_sec;
	localtime_s(&ltime, &local_tv_sec);
	strftime( timestr, sizeof timestr, "%H:%M:%S", &ltime);

	// Print timestamp
	printf("%s ", timestr);

	// Print the statistics
	printf("Bits per second=%I64u ", bits_per_second);
	printf("Packets per second=%I64u\n", packets_per_second);

	//store current timestamp
	old_ts->tv_sec = packet_header->ts.tv_sec;
	old_ts->tv_usec = packet_header->ts.tv_usec;
}
/*-------------------------------------------------------------------------------*/
```

ZADATAK

U prilogu materijala za vežbu dat je jedan primer implementacije statističke analize UDP saobraćaja sa mrežne kartice koja je postavljena u statistički režim (*mode*) rada.

1. Omogućiti da korisnik na početku izabere vrstu saobraćaja (po tipu protokola) za koji želi statističke podatke. Mogući izbor je IP, UDP, TCP i ARP saobraćaj.
    - U skladu sa izborom korisnika postaviti odgovarajući filterski izraz.

    (**0.4 bodova**)
2. Omogućiti da se funkcija *statistics_handler()* poziva svake 2 sekunde, tj. da se statistički podaci prikupljaju svake 2 sekunde. (**0.4 bodova**)
    - Za svaki statistički uzorak ispisati na ekranu koliko je paketa i koliko bajta primljeno.
    - Za svaki statistički uzorak pratiti saobraćaj izražen u broju bita u sekundi i broju paketa u sekundi.
3. Omogućiti da se nakon 30 sekundi statističke analize saobraćaja, na ekranu ispiše prosečna vrednost broja bita u sekundi i prosečna vrednost broja paketa u sekundi. (**0.4 bodova**)
4. Postaviti mrežnu karticu u režim rada (*mode*) za hvatanje paketa. (**0.4 bodova**)
5. Omogućiti hvatanje 20 paketa pomoću callback funkcije *packet_handler()*. (**0.4 bodova**)
6. Prilikom hvatanja paketa beležiti relevantne podatke kako bi se po završetku hvatanja dao prikaz saobraćaja izražen u broju paketa u sekundi i bita u sekundi. U tu svrhu potrebno je beležiti: (**0.4 bodova**)
    - Trenutak hvatanja prvog paketa.
    - Trenutak hvatanja zadnjeg paketa.
    - Ukupan broj primljenih bajtova (za sve uhvaćene pakete).
7. Ispisati na ekranu vrednost broja bita u sekundi i vrednost broja paketa u sekundi za nadgledani saobraćaj. (**0.6 bodova**)