Vežba 8 – Rad sa datotekama upotrebom Libpcap (Raspbian) – WinPcap (Windows) programske biblioteke

1. Otvaranje datoteke

Da bi sačuvali pakete koje hvatamo sa mreže, prvo je potrebno otvoriti datoteku u kojoj ćemo čuvati te pakete. Za tu namenu služi funkcija *pcap_dump_open()*. Ova funkcija osim što otvara fajl vrši i njegovo povezivanje sa adapterom sa koga se hvataju paketi. Naravno, neophodno je da pre poziva ove funkcije adapter bude otvoren za hvatanje paketa.

Format sačuvanih fajlova je .pcap. Ovaj format obezbeđuje čuvanje sadržaja uhvaćenih paketa u binarnom obliku. On se standardno koristi od strane mnogih mrežnih alata uključujući i WinDump, Ethereal i Snort.

```
pcap_dumper_t* pcap_dump_open (pcap_t* device_handle, const char* filename);
```

Funkcija:	Opis:
pcap_dump_open	Funkcija za otvaranje fajla u koji će se sačuvati uhvaćeni paketi.
Parametri:	**Opis:**
pcap_t* device_handle	Deskriptor adaptera koji je otvoren za hvatanje paketa na mreži.
const char* filename	Ime fajla koji se otvara.
Povratna vrednost:	**Opis:**
pcap_dumper_t *	Ako se uspešno izvrši, vraća pokazivač na deskriptor fajla u koji se čuvaju paketi. Vraća NULL ako se desi greška. U tom slučaju može se pozvati *pcap_geterr()* radi ispisa greške. U suprotnom, vraća se vrednost NULL a u parametru *error_buffer* se nalazi poruka o grešci.

2. Snimanje paketa u datoteku

Upis paketa u datoteku se vrši pomoću funkcije *pcap_dump()* koja se izvršava pri pozivu callback funkcije *pcap_handler()*. Parametri *pcap_dump()* funkcije u potpunosti odgovaraju parametrima *pcap_handler()* funkcije (preslikavanje je 1-1).

```
void pcap_dump(unsigned char* user, const struct pcap_pkthdr* packet_header,
const unsigned char* packet_data);
```

Funkcija:	Opis:
pcap_dump	Funkcija za čuvanje paketa u fajl.
Parametri:	**Opis:**
unsigned char* user	Za ovaj parametar se prosleđuje parametar tipa pcap_dumper_t koji je vratila funkcija pcap_dump_open. Naravno, potrebno je da se prethodno kastuje u pokazivač tipa (unsigned char*).
const struct pcap_pkthdr* packet_header	Pokazivač na generičko zaglavlje koje drajver za hvatanje paketa prikači na svaki uhvaćeni paket.
const unsigned char* packet_data	Pokazivač na početak podataka u paketu, uključujući i zaglavlja protokola.
Povratna vrednost:	**Opis:**
void	Nema povratne vrednosti.

Da bi se ispravno snimili podaci i sam fajl zatvorio, potrebno je nakon snimanja u fajl zatvoriti adapter sa funkcijom pcap_close().

```
void pcap_close (pcap_t* device_handle);
```

Funkcija:	Opis:
pcap_close	Funkcija za zatvaranje fajlova koji su pridruženi deskriptoru adaptera device_handle i oslobađanje zauzetih memorijskih resursa.
Parametri:	**Opis:**
pcap_t* device_handle	Deskriptor adaptera sa koga se hvataju/čitaju paketi
Povratna vrednost:	**Opis:**
void	Nema povratne vrednosti.

Primer:

```
/*-------------------------------------------------------------------------*/
int main()
{
	pcap_t* device_handle;
```

```
        ...
        // Open the dump file
        pcap_dumper_t* file_dumper = pcap_dump_open(device_handle, "example.pcap");

        if (file_dumper == NULL)
        {
                printf("\n Error opening output file\n");
                return -1;
        }
        ...

        // Start the capture
        pcap_loop(device_handle, 10, packet_handler, (unsigned char*) file_dumper);

        // Close the file associated with device_handle and deallocates resources
        pcap_close(device_handle);

        return 0;
}

// Callback function invoked by libpcap for every incoming packet
void packet_handler(unsigned char* fd, const struct pcap_pkthdr *
        packet_header, const unsigned char *packet_data)
{
        // Save the packet on the dump file
        pcap_dump(fd, packet_header, packet_data);
}
/*-------------------------------------------------------------------------*/
```

3. Čitanje paketa iz datoteke

Kada imamo sačuvane pakete unutar fajla, potrebno nam je i da znamo kako da iščitamo sadržaj *.pcap fajla. Za otvaranje fajla služi nam funkcija *pcap_open_offline()*, a zatim se sekvencijalno iščitavanje paketa vrši pomoću *pcap_loop()* ili *pcap_next_ex()*. Ovaj postupak za iščitavanje paketa iz fajla je veoma sličan hvatanju paketa sa mreže. Pre poziva funkcije za iščitavanje može se postaviti filter (pomoću funkcije *pcap_setfilter()*) kako bi se definisao željeni podskup mrežnog saobraćaja koji je potrebno čitatit iz fajla.

```
pcap_t* pcap_open_offline (const char* filename, char* error_buffer);
```

Funkcija:	Opis:
pcap_open_offline	Funkcija za otvaranje fajla radi čitanja paketa.
Parametri:	**Opis:**
const char* filename	Specificira ime fajla koji se otvara.
char* error_buffer	String koji sadrži tekst greške ukoliko se ona desi. Ima sadržaj samo ukoliko se funkcija ne izvrši uspešno (vrati NULL pokazivač).
Povratna vrednost:	**Opis:**
pcap_t*	Vraća deskriptor otvorenog fajla iz koga se čitaju paketi. Ukoliko se desi greška, vraća NULL.

Primer:

```
/*-------------------------------------------------------------------------*/
int main()
{
	pcap_t* device_handle;
	char error_buffer [PCAP_ERRBUF_SIZE];

	// Open the capture file

	if ((device_handle = pcap_open_offline("example.pcap",// Name of the device
						error_buffer // Error buffer
	)) == NULL)
	{
		printf("\n Unable to open the file %s.\n", "example.pcap");
		return -1;
	}

	// Check the link layer. We support only Ethernet for simplicity.
	if(pcap_datalink(device_handle) != DLT_EN10MB)
	{
		printf("\nThis program works only on Ethernet networks.\n");
		return -1;
	}

	// Read and dispatch packets until EOF is reached
	pcap_loop(device_handle, 10, dispatcher_handler, NULL);

	// Close the file associated with device_handle and deallocates resources
	pcap_close(device_handle);

	return 0;
}

void dispatcher_handler(unsigned char* user, const struct pcap_pkthdr *
packet_header, const unsigned char* packet_data)
{
	// Print packet timestamp and packet length
	printf("%ld:%ld (%ld)\n", packet_header->ts.tv_sec,
		packet_header->ts.tv_usec,
		packet_header->len);

	// Print the packet
	for (int i=0; (i < packet_ header->len); i++)
	{
		printf("%.2x ", packet_data[i]);
		if ( (i+1) % 16 == 0)
			printf("\n");
	}

	printf("\n\n");
}
/*-------------------------------------------------------------------------*/
```

ZADATAK

Napisati libpcap/WinPcap aplikaciju koja omogućava razvrstavanje prethodno uhvaćenih paketa po datotekama u zavisnosti od tipa protokola koji paketi koriste.

- Učitatati presretene pakete koji se nalaze uskladišteni u datoteci *example.pcap* koja je data u prilogu vežbe. (**0.6 bodova**)
- Razvrstati pakete prema protokolu koji implementiraju:
    a. ARP – *implementirano u primeru programskog koda datog u materijalima za vežbu*,
    b. ICMP, (**0.4 bodova**)
    c. UDP i (**0.4 bodova**)
    d. TCP. (**0.4 bodova**)
- Razvrstane pakete potrebno je sačuvati u zasebne datoteke:
    a. arp_packets.pcap – *implementirano u primeru programskog koda datog u materijalima za vežbu*,
    b. icmp_packets.pcap, (**0.4 bodova**)
    c. udp_packets.pcap i (**0.4 bodova**)
    d. tcp_packets.pcap. (**0.4 bodova**)