EXPERIMENT - XVII
PACKET CAPTURING AND FILTERING APPLICATION

April 14, 2020

ADITHYA D RAJAGOPAL
ROLL NO : 9
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING TRIVANDRUM

AIM

To develop a packet capturing and filtering application using raw sockets.

THEORY

Packet

A packet is a basic unit of communication over a digital network. When data has to be transmitted, it is broken down into similar structures of data before transmission, called packets, which are reassembled to the original data chunk once they reach their destination.

libpcap

libpcap allows us to capture or send packets from a live network device or a file. This package is aimed at Debian based Linux distributions but may also work on Mac OSX. Not intended for Windows, but WinPcap is a port that is available. Compiling a pcap program requires linking with the pcap lib. You can install it in Debian based distributions with
sudo apt-get install libpcap-dev

ALGORITHM

The below provided algorithm is an example for the program in cpp.

Algorithm 1 Algorithm for the packet capture

1: START
2: Check for network devices.
3: List the devices and allow the user to select one of the network devices.
4: Use pcap_open_live for live capture of packets one at a time.
5: Use pcap_loop to process the packets which includes calling the
6: corresponding callback function.
7: The ethernet header is examined and ether_type value is compared against known constants for IP and other similar types.
8: Corresponding required values are printed.
9: STOP

SOURCE CODE

```
#include<cstdio>
#include<iostream>
#include<pcap.h>
#include<sys/socket.h>
#include<pcap/pcap.h>
#include<netinet/in.h>
#include<netinet/if_ether.h>
#include<arpa/inet.h>
using namespace std;
int ipc=0;
int arpc=0;

void printit(u_int8_t *etherheader,int n)
{
 printf("%02x:%02x:%02x:%02x:%02x:%02x\n",etherheader[0],etherheader[1],
        etherheader[2],etherheader[3],etherheader[4],etherheader[5]);
}

void callBack(u_char *args,const struct pcap_pkthdr *header,const u_char *packet)
{
 u_int8_t *src,*dest;
 struct ether_header *hdr=(struct ether_header *)packet;
 src=hdr->ether_shost;
 dest=hdr->ether_dhost;
 if(ntohs(hdr->ether_type)==ETHERTYPE_IP)
 {
  cout<<"Ethernet type hex "<<std::hex<<ntohs(hdr->ether_type)
      <<" decimal "<<std::dec<<ntohs(hdr->ether_type)<<" is an IP packet \n";
  cout<<"Received "<<++ipc<<" IP packets\n";
 }
 else if(ntohs(hdr->ether_type)==ETHERTYPE_ARP)
 {
  cout<<"Ethernet type hex "<<std::hex<<ntohs(hdr->ether_type)
```

```
      <<" decimal "<<std::dec<<ntohs(hdr->ether_type)<<" is an ARP packet\n";
  cout<<" Received "<<++ipc<<" ARP packets\n";
 }
 cout<<"Destination address : ";
 printit(dest,ETHER_ADDR_LEN);
 cout<<"Source address : ";
 printit(src,ETHER_ADDR_LEN);
 cout<<endl;
}

int main()
{
 char errbuf[PCAP_ERRBUF_SIZE];
 pcap_t *handle;
 pcap_if_t *devs,*dev;
 int i,ch;
 struct bpf_program fp;
 if(pcap_findalldevs(&devs,errbuf)==-1)
 {
  cout<<errbuf<<"\n";
  exit(0);
 }
 cout<<"Select the interface of your choice:\n";
 for(dev=devs,i=0;dev;dev=dev->next)
 printf("%d. %s\n",i++,dev->name);
 cout<<"Enter your choice:";
 scanf("%d",&ch);
 for(dev=devs,i=-1;dev;dev=dev->next)
 if((++i)==ch)
 break;
 handle=pcap_open_live(dev->name,BUFSIZ,0,0,errbuf);
 if(handle==NULL)
 {
  cout<<"Could not open device "<<dev->name<<":"<<errbuf<<"\n";
  exit(0);
```

```
  }
  pcap_loop(handle,10,callBack,NULL);
  return 0;
}
```

OUTPUT

```
root@adithya:~/s6/np# g++ p17.cpp -lpcap
root@adithya:~/s6/np# ./a.out
Select the interface of your choice:
0. wlp1s0
1. any
2. lo
3. enp2s0
4. bluetooth0
5. nflog
6. nfqueue
7. usbmon1
8. usbmon2
Enter your choice:0
Ethernet type hex 800 decimal 2048 is an IP packet
Received 1 IP packets
Destination address : 01:00:5e:00:00:fb
Source address : 3c:6a:a7:58:4c:0b

Destination address : 33:33:00:00:00:fb
Source address : 3c:6a:a7:58:4c:0b

Ethernet type hex 800 decimal 2048 is an IP packet
Received 2 IP packets
Destination address : 00:1e:a6:21:75:c6
Source address : 3c:6a:a7:58:4c:0b

Ethernet type hex 800 decimal 2048 is an IP packet
Received 3 IP packets
Destination address : 3c:6a:a7:58:4c:0b
Source address : 00:1e:a6:21:75:c6

Ethernet type hex 800 decimal 2048 is an IP packet
Received 4 IP packets
Destination address : 00:1e:a6:21:75:c6
Source address : 3c:6a:a7:58:4c:0b

Ethernet type hex 800 decimal 2048 is an IP packet
Received 5 IP packets
Destination address : 3c:6a:a7:58:4c:0b
Source address : 00:1e:a6:21:75:c6

Ethernet type hex 800 decimal 2048 is an IP packet
Received 6 IP packets
Destination address : 00:1e:a6:21:75:c6
Source address : 3c:6a:a7:58:4c:0b

Ethernet type hex 800 decimal 2048 is an IP packet
Received 7 IP packets
Destination address : 3c:6a:a7:58:4c:0b
Source address : 00:1e:a6:21:75:c6

Ethernet type hex 800 decimal 2048 is an IP packet
Received 8 IP packets
Destination address : 00:1e:a6:21:75:c6
Source address : 3c:6a:a7:58:4c:0b

Ethernet type hex 800 decimal 2048 is an IP packet
Received 9 IP packets
Destination address : 3c:6a:a7:58:4c:0b
Source address : 00:1e:a6:21:75:c6

root@adithya:~/s6/np#
```

RESULT

Packet capturing and filtering application has been developed using raw sockets in C++.