Project 1: Develop a packet analyzer

Due date: **Friday, January 28**
Project Writeup base on one from Dr. Kwon

1. Analyzing the Captured Packets

Write an application that reads a set of packets and produces a detailed summary of those packets. Your packet analyzer should run as a shell command. The syntax of the command is the following:

```
% java pktanalyzer datafile
```

The pktanalyzer program will extract and display the different headers of the captured packets in the file datafile. First, it displays the ethernet header fields of the captured frames. Second, if the ethernet frame contains an IP datagram, it prints the IP header. Third, it prints the packets encapsulated in the IP datagram. TCP, UDP, or ICMP packets can be encapsulated in the IP packet.

2. Constraints

The pktanalyzer program must:

1. Be Java only unless approved by the Instructor
2. Not depend on external libraries
3. Must use binary operations (&, |, ^, etc.) instead of string methods

3. Submission

Zip up all your source files with a readme.txt file explaining how to compile and execute your program. Submit the zip file through mycourses.rit.edu in the Project 1 drop box.

4. Sample captured packets

- TCP
- UDP
- ICMP

You can hexdump these binary files using a command like hexdump in Linux. For example,

```
% hexdump -C new_udp_packet1.bin
00000000  00 00 0c 07 ac 01 c0 14  3d d5 72 8b 08 00 45 00  |........=.r...E.|
00000010  00 3d 54 33 40 00 40 11  9e ec 81 15 42 55 81 15  |.=T3@.@.....BU..|
00000020  03 11 8b 30 00 35 00 29  9c 1d f3 8e 01 00 00 01  |...0.5.)........|
00000030  00 00 00 00 00 00 04 70  6c 75 73 06 67 6f 6f 67  |.......plus.goog|
00000040  6c 65 03 63 6f 6d 00 00  01 00 01                 |le.com.....|
0000004b
```

5. Output

Here are some examples of the output:

```
% java pktanalyzer pkt/new_icmp_packet2.bin
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet size = 60 bytes
ETHER:  Destination = c0:14:3d:d5:72:8b,
ETHER:  Source      = 00:1d:a1:38:58:00,
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 40 bytes
IP:   Identification = 54321
IP:   Flags = 0x00
IP:         .0.. .... = OK to fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
```

```
IP:   Time to live = 244 seconds/hops
IP:   Protocol = 1 (ICMP)
IP:   Header checksum = 0x05a2
IP:   Source address = 198.20.99.130
IP:   Destination address = 129.21.66.85
IP:   No options
IP:
ICMP:  ----- ICMP Header -----
ICMP:
ICMP:  Type = 8 (Echo request)
ICMP:  Code = 0
ICMP:  Checksum = 0x60b1
ICMP:

% java pktanalyzer pkt/new_tcp_packet1.bin
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet size = 266 bytes
ETHER:  Destination = 00:00:0c:07:ac:01,
ETHER:  Source      = c0:14:3d:d5:72:8b,
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 252 bytes
IP:   Identification = 32492
IP:   Flags = 0x4
IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 6 (TCP)
IP:   Header checksum = 0x4a9e
IP:   Source address = 129.21.66.85
IP:   Destination address = 172.217.0.46
IP:   No options
IP:
TCP:  ----- TCP Header -----
TCP:
```

```
TCP:  Source port = 52566
TCP:  Destination port = 443
TCP:  Sequence number = 424114169
TCP:  Acknowledgement number = 2974881950
TCP:  Data offset = 8 bytes
TCP:  Flags = 0x18
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP:        .... 1... = Push
TCP:        .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 1832
TCP:  Checksum = 0x75b0
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
TCP:  Data: (first 64 bytes)
TCP:  cd56 01bb 1947 77f9 b151 189e 8018 0728       '.V...Gw..Q.....('
TCP:  75b0 0000 0101 080a 0007 eb7b 5410 dc48       'u...........T..H'
TCP:  1703 0300 c300 0000 0000 0000 713e fc3b       '............q>.;'
TCP:  b6ce fab3 6791 9e87 b9ac 681c 2696 dc87       '....g.....h.&...'

% java pktanalyzer pkt/new_udp_packet1.bin
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet size = 75 bytes
ETHER:  Destination = 00:00:0c:07:ac:01,
ETHER:  Source      = c0:14:3d:d5:72:8b,
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 61 bytes
IP:   Identification = 21555
IP:   Flags = 0x4
IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 17 (UDP)
IP:   Header checksum = 0x9eec
IP:   Source address = 129.21.66.85
IP:   Destination address = 129.21.3.17
IP:   No options
IP:
UDP:  ----- UDP Header -----
UDP:
UDP:  Source port = 35632
UDP:  Destination port = 53
UDP:  Length = 41
UDP:  Checksum = 0x9c1d
UDP:
UDP:  Data: (first 64 bytes)
UDP:  8b30 0035 0029 9c1d f38e 0100 0001 0000       '.0.5.)..........'
UDP:  0000 0000 0470 6c75 7306 676f 6f67 6c65       '.....plus.google'
UDP:  0363 6f6d 0000 0100 01                        '.com............'
```