Name: _____

Experiment 1.3

Build a Packet Analyzer

Purpose

To learn how to build a packet analyzer on a network processor.

Background Reading And Preparation

Review Ethernet, IP, and TCP headers (chapters 9, 20, and 24 in Computer Networks and Internets and IETF RFCs 791 and 793). Also read the the documentation for the simplified network processor API.

Overview

Build an application that captures packets from the network and analyzes them.

Procedure And Details (checkmark as each is completed)

_____ Write an application that reads a set of packets and analyzes them. When a user enters the command "send_command summary"† while logged into the network processor, the application should produce the following summary:

Layer 2 Summary (Ethernet)

- Total number of frames processed
- Average frame size (excluding header)
- Number and percentage of broadcast frames
- Number and percentage of multicast frames
- Number and percentage of unicast frames
- Number and percentage of each of the top five frame types

Layer 3 Summary (IP)

- Total number of datagrams processed
- Average datagram size (excluding header)
- Number and percentage of datagram fragments
- Number and percentage of datagrams sent to network broadcast address
- Number and percentage of datagrams sent to limited broadcast
- Number and percentage of datagrams carrying TCP
- Number and percentage of datagrams carrying UDP
- Number and percentage of datagrams carrying ICMP

Layer 4 Summary (TCP)

Total number of TCP segments processed
Average segment size (excluding header)
Number and percentage of acknowledgements (no data)
Number and percentage of data segments
Number and percentage of SYN/FIN segments
Number and percentage of each of top five destination ports
Number and percentage of each of top five source ports

_____ Use two computers and a network processor to test the application. Connect the two computers and the network processor to a private LAN. Generate traffic between the computers using applications such as telnet, ping and traceroute. Use tcpdump or an equivalent program to capture the traffic and check that your application produces the correct output.

_____ Modify the application so that it can display the headers of the packets it analyzes. When a user enters the command "send_command verbose" the application should print out the field contents of each header it parses (i.e. Ethernet, IP and TCP) in each packet it subsequently receives. Disable this feature when the user enters the command "send_command quiet".

Optional Extensions (checkmark options as they are completed)

_____ Modify the application to take arguments that specify which packets to examine. When a user enters a command of the form "send_command filter <pattern>" the application only analyzes (and prints if in verbose mode) packets matching the pattern. For example, the pattern "-ip" may cause the application to only analyze and print IP packets. It should also be possible to limit processing to frames sent by a specified computer. When issued the command "send_command nofilter" all packets should be analyzed.

_____ Extend your program to allow boolean combinations of pattern options. For example it should be possible to limit processing to "frames carrying IP that also contain TCP, or ARP frames."

Notes

†See the simplified API manual for instructions on how to program your ACE to handle input from *send_command* program.