Name: ____

Experiment 1.7

Create a Bare-bones MicroACE

Purpose

Learn how to create a basic MicroACE.

Background Reading And Preparation

Read chapters 22 through 24 of Network System Design to learn about MicroACEs and the basics of how they are programmed. Read the Intel IXP1200 Programmer's Reference Manual to learn the microcode of the IXP1200.

Overview

Construct a MicroACE that counts frames.

Procedure And Details (checkmark as each is completed)

______ Obtain a basic MicroACE from your lab instructor that initializes the ACE but drops all frame that come to it. The StrongARM component should also include a mechanism for receiving commands from applications. Learn how the code operates. Try compiling and running the MicroACE.

______ Modify the microengine component so that instead of dropping all frames it receives from an Ethernet port, it raises them as exceptions to the StrongARM component.

______ Alter the StrongARM component so that it keeps count of the number of frames raised as exceptions, and prints that number when issued a command from a command-line application. Have the StrongARM component drop all frames after counting them.

______ Test your ACE by connecting the IXP1200 to an Ethernet LAN and generating traffic on that LAN. Check that the MicroACE counts packets correctly.

Optional Extensions (checkmark options as they are completed)

______ Create your own cross-call mechanism to allow arbitrary applications to retrieve the frame counts from your MicroACE.

______ Extend the MicroACE so that each frame is counted by the microengines instead of the StrongARM. The frame counts should be kept in a table in memory shared with the StrongARM core so the StrongARM can print the frame counts upon application request.

______ Extend the microengine-based counting mechanism even further so that it also counts different types of frames such as IP, ARP, TCP, TCP destined for a particular port, IP fragments, etc...

Notes