

CS636 Programming Project #1

A Packet Dump Function

Due Date

The project is due before midnight Sunday, January 29th. You will receive instructions on how to submit the code electronically.

Purpose

To understand packet formats, header nesting, network byte order, and possible ways to structure protocol processing code.

What You Will Do

Write a C function, *pktdump*, that formats and prints fields of a packet. Your function must be able to handle the following protocols: ARP, IPv4, IPv6 (base header), UDP, TCP, and DHCP. The function takes two arguments defined as follows:

Argument	Meaning
<code>char *pktptr</code>	A pointer to the start of a network packet
<code>int dbytes</code>	The number of data bytes to dump

The idea is to build a diagnostic tool that can be used to examine incoming or outgoing packets. The format must be easily readable by a human. You can choose to follow existing conventions (e.g., use a format similar to the *tcpdump* program) or invent your own. In practice, many packet dump facilities try to squeeze the information onto a single line of output because a typical scenario consists of dumping all incoming packets. Therefore, if the output from each packet takes many lines, it will scroll by quickly and be difficult to spot the packets for which you are searching.

Structure Of The Code

We will discuss the design in class. The short version is: the code you will be given has a C struct that includes multiple protocols, and the code you will write uses a separate struct for each protocol.

Extra Credit

Build a *packet filter* that allows a caller to specify a set of protocols to select, and arrange for the filter to skip packets that do not match the selection criteria and only dump packets that meet the criteria. For example, the selection criteria might allow a caller to specify IPv6 only, or IPv4 and UDP with UDP limited to source port 7.