ECE 5600 Project (Phase 5)

OBJECTIVE

1. Be familiar with UDP protocol.
2. IP fragmentation and defragmentation.

BACKGROUND

The User Datagram Protocol belongs to the transport layer of the TCP/IP Protocol stack. Datagrams are unreliable, connectionless packets of data that are delivered to a particular port on a computer with a particular IP address. Even though UDP is connectionless, all datagrams have an originating port and a destination port. Some of the ports have well known functions. For example, port 7 is an echo port. If a datagram is delivered to port 7, an identical datagram is sent back to the originating port of the originating computer.

In terms of actual UDP services, one of the services is: Send User Datagram. To transmit a UDP datagram, a computer completes the appropriate fields in the UDP header and forwards the data together with the header for transmission by the IP network layer.

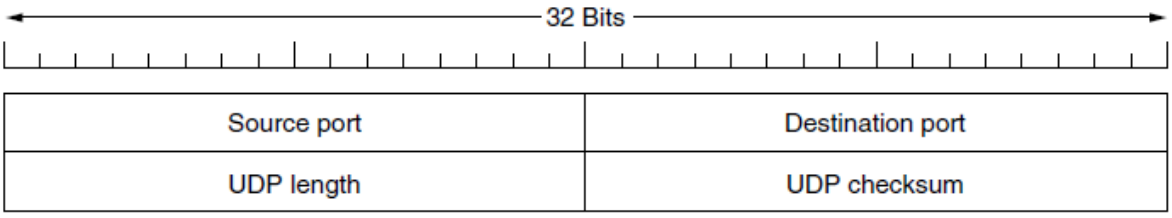

Figure 1. The UDP header.

The UDP header consists of four fields each of 2 bytes in length:

Source Port: UDP packets from a client use this as a service access point (SAP) to indicate the session on the local client that originated the packet. UDP packets from a server carry the server SAP in this field.

Destination Port: UDP packets from a client use this as a service access point (SAP) to indicate the service required from the remote server. UDP packets from a server carry the client SAP in this field.

The source port is primarily needed when a reply must be sent back to the source. By copying the Source port field from the incoming segment into the Destination port field of the outgoing segment, the process sending the reply can specify which process on the sending machine is to get it.

UDP length: The number of bytes comprising the combined UDP header information and payload data. The UDP length field includes the 8-byte header and the data. The minimum length is 8 bytes, to cover the header. The maximum length is 65,515 bytes, which is lower than the largest number that will fit in 16 bits because of the size limit on IP packets.

UDP Checksum: A checksum to verify that the end to end data has not been corrupted by routers or bridges in the network or by the processing in an end system. The algorithm to compute the checksum is the Standard Internet Checksum algorithm. This allows the receiver to verify that it was the intended destination of the packet, because it covers the IP addresses, port numbers and protocol number, and it verifies that the packet is not truncated or padded, because it covers the size field. Therefore, this protects an application against receiving corrupted payload data in place of, or in addition to, the data that was sent. In the cases where this check is not required, the value of 0x0000 is placed in this field, in which case the data is not checked by the receiver. When performing this computation, the Checksum field is set to zero and the data field is padded out with an additional zero byte if its length is an odd number.

The pseudoheader for the case of IPv4 is shown in Fig. 2. It contains the 32-bit IPv4 addresses of the source and destination machines, the protocol number for UDP (17), and

the byte count for the UDP segment (including the header).

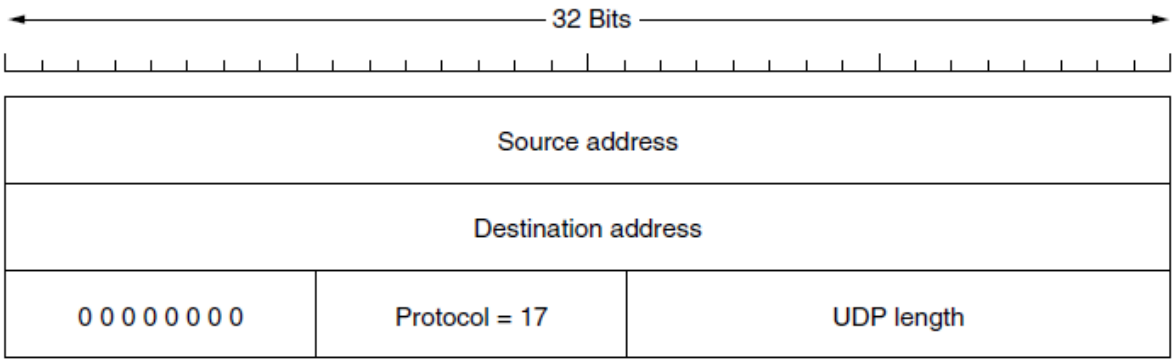

Figure 2. The IPv4 pseudoheader included in the UDP checksum.

It attaches the UDP header (including a checksum) and passes the packet to IP for delivery. In IP, datagrams may be as large as 65536 bytes. Since the MTU of Ethernet is around 1500 bytes, large datagrams have to be fragmented and defragmented.

In this phase of the project, you will be implementing the Send User Datagram service as well as the echo protocol for UDP. I will provide you with a Linux program (udp_echo) that will send random datagrams to port 7 of your UDP implementation and expect identical datagrams in return.

PRE-LAB READING

Chapter 6, P 541-546.

PROJECT PROCEDURE

The following steps outline this phase of this project:

1. Write the code in your IP implementation to do fragmentation and defragmentation.
2. Implement the Send User Datagram service. Compute the UDP header fields and pass the header and datagram to IP for delivery. Send a datagram of at least 1600 bytes and not more than 2048 bytes to port 7 of another computer in the lab and verify that a copy of your datagram is returned to you. Copy and paste the UDP

packets up through the UDP header into a file (there should be at least four: 2 outgoing and 2 incoming). Do not paste the 1600+ bytes of data please!

3. Project Due: Dec 15, 2017.