Practical Exercises in Computer Networks

UDP service model check, Datagram loss experiment (WIP)

One of the most important properties of the IP service model represents the fact that the IP network can loose IP packets. The UDP transport protocol adds a multiplexing level to IP, in unicast IP we communicate a host with a host, with UDP we *can* communicate an application running on a host with another application running on another host thereby extending the basic host-to-host communication to a process-to-process model (In fact, in most modern operating systems this is conducive to a thread-to-thread communication). Nevertheless, the promoted UDP-over-IP thread-to-thread model does not overcome the packet loss problem, which now is a datagram loss problem. In this experiment we want to check practically that UDP datagrams can be lost due to a variety of causes, specifically, in this experiment we want to observe the effect of overflowing or overrunning the UDP socket queue of a UDP receiver.

UDP adds a *namespace* for processes in a host

When a Java application opens a UDP socket associated with a UDP port, a bidirectional association is created between the Java thread and the Socket descriptor within the operating system. From that moment on, incoming traffic at that UDP port will be delivered to the UDP socket incoming queue. The Java thread consumes bytes from the incoming queue by invoking methods of DatagramSocket (Actually, those bytes are encapsulated into DatagramPacket objects). What happens if the thread does not consume packets for some arbitrary amount of time? In general, depending on the ratio of the speed at which the thread consumes and the network delivers more traffic, the queue will fill up and eventually overflow, empty or anywhere in between (For ergodic or stationary processes, Little's Law governs the queue length at any time).

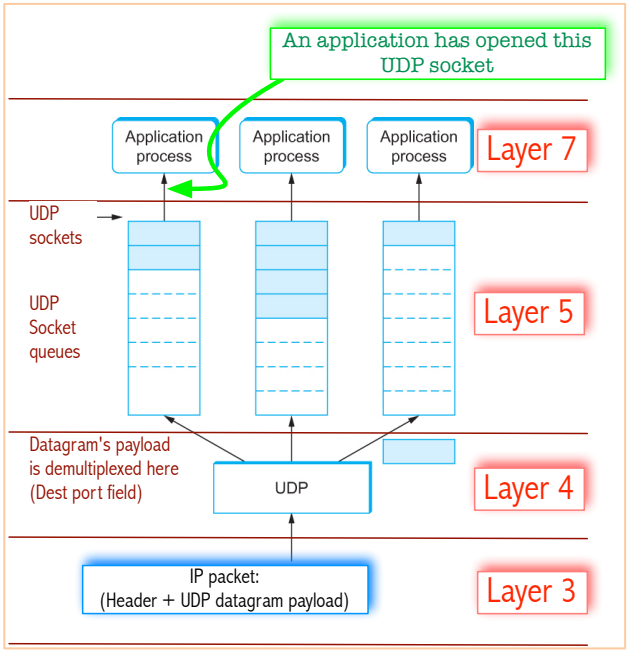

Fig. 1. UDP Socket queues

In this experiment, we will write a Java C/S pair of programs for proving this in a practical way. We will start with the same two Java programs used in **LAB 2** (Introduction to Java Network Programming, paloalto.unileon.es/cn), their names are **UDPPingClient.java and UDPPingServer.java**. The exercise consists of swamping the server with some arbitrary amount of packets while the server is somewhat stopped, not consuming any incoming information from the UDP socket. The decisions regarding for how long the server stops, at which rate the client will send, how to sequence the packets so that the server can make for sure that packets are received in sequence and that therefore no loss is taking place, or otherwise, how many packets have gotten lost due to queue overrun.