Topic 1: Application Layer - Socket Programming

Questionnaire 1

Q1.

TCP is a connection-oriented protocol that implies a connection-establishment procedure between client and server known as *three-way handshake*. In a TCP connection:

A. `connect()` returns in the client and `accept()` returns in the server when the three-way handshake, which takes place within the application layer, completes.

B. the three-way handshake is completely invisible to the client and server programs.

C. `accept()` returns in the server when the three-way handshake completes, and the original socket passed to `accept()` is associated to the connection.

D. the server initiates a three-way handshake by calling `socket()`, `bind()`, `listen()` and `accept()` functions.

Q2.

A socket can be informally defined as a "door" between application process and the end-to-end transport-layer protocol (TCP or UDP). Considering this socket definition,

A. the TCP server would need *n* sockets to support *n* simultaneous connections, each from a different client host.

B. the UDP server creates a socket by calling the `socket()` function, and can receive datagrams from different UDP clients on this single socket.

C. a socket of type `SOCK_DGRAM` denotes a reliable byte-stream service, such as that provided by UDP.

D. a socket is uniquely identified by an internet address and an end-to-end protocol (TCP or UDP).

Q3.

The Internet makes two transport protocols available to applications, TCP and UDP. There are some fundamental differences between TCP and UDP sockets, namely:

A. TCP server does not include a welcoming socket, which differs from UDP server.

B. An application that invokes UDP as its transport protocol can rely on UDP to deliver messages sent, although they may arrive out of order, whereas TCP ensures the delivery of a stream of bytes sent through a TCP socket in the proper order.

C. UDP preserves boundary between messages, where each message read with `recvfrom()` corresponds to a single `sendto()`, which differs from a byte stream sent through a TCP socket, where bytes read with `read()` may correspond to several `write()`.

D. TCP provides reliable data transfer through the invocation of `read()` and `write()` by the communicating processes, whereas UDP provides reliable data transfer through the invocation of `recvfrom()` and `sendto()` by the communicating processes.

Q4.

Consider the `bind()` function in socket programming (using TCP or UDP).

A. In order to establish a TCP connection, the client supplies the server's address to `connect()` and the server specifies its own address to `bind()`.

B. A TCP client cannot call `bind()` to specify its local address/port.

C. When a TCP or UDP server invokes `bind()`, the server's port number is bound to the client's socket.

D. The `bind()` function assigns a port number in host byte order to the server's socket.

Q5.

The *Domain Name System* (DNS) is used to map between hostnames and IP addresses. Consider the resolver functions `gethostbyname()` and `gethostbyaddr()`.

A. A UDP client cannot check the source IP address and port number of a received datagram by using the address returned by `recvfrom()` in the call to `gethostbyaddr()`.

B. The `gethostbyname()` function looks up a hostname and, if successful, returns a pointer to a `struct hostent` that contains exactly one IP address for the host.

C. TCP client/server applications are not allowed to use the `gethostbyaddr()` function.

D. The `gethostbyaddr()` function takes a binary IP address and tries to find the corresponding hostname.