Network Fundamentals

In the OSI model, control is passed from one layer to the next, starting at the application layer in one station and proceeding to the bottom layer. It is then passed over the channel to the next station and back up the hierarchy. The OSI model takes the task of internetworking and divides that up into what is referred to as a vertical stack that consists of the following seven layers:

Application Presentation Session Transport Network Data Link Physical

A layer should only be created where the definite levels of abstraction are needed.The function of each layer should be selected as per the internationally standardized protocols.The number of layers should be large so that separate functions should not be put in the same layer. At the same time, it should be small enough so that architecture doesn't become very complicated.In the OSI model, each layer relies on the next lower layer to perform primitive functions. Every level should able to provide services to the next higher layer Changes made in one layer should not need changes in other lavers.

TCP/UDP helps you to determine how a specific computer should be connected to the internet and how you can transmit data between them. It helps you to create a virtual network when multiple computer networks are connected

Socket connection occurs via protocol. Internet Protocol (IP) is a low-level routing protocol that separates data into small packets and sends it to an address across the network. But in this there is no guarantee of the delivery of the packets to its destination. For a reliable transmit of data, there is another higher-level protocol known as Transmission Control Protocol (TCP) which manages to string together these packets robustly

Stream Sockets allow processes to communicate using TCP. Stream sockets provide a two-way, reliable, ordered, non-replicated stream of data with no record boundaries. It is connection-oriented socket. Once a connection is established, data can be read and written from these sockets as streams of bytes. The socket type is SOCK_STREAM

TCP/IP-style networking is appropriate for most networking needs. It provides a serialized, predictable, reliable stream of packet data. This is not without its cost, however. TCP incorporates many state-of-the-art algorithms to deal with congestion control on crowded networks, as well as pessimistic expectations about packet loss. This leads to a somewhat inefficient way to transport data. Datagrams provide an alternative

It's been a while since I used Wireshark, as I often work from the command line unless I have a specific need for a graphical interface, so I apologize in advance if anything is out of date.

It sounds like you're running wireshark as a regular user (not root), without your network card having "promiscuous mode" enabled, or both. This is assuming you're running it in Linux - while I personally haven't run Wireshark under Windows, I would assume that it would require administrative rights as well. This is also assuming you're trying to run Wireshark on an ethernet interface. If you're running it on a wireless adapter, you should check that monitor mode (select capture => interfaces => options to review this) is enabled.