Internet: Transport layer protocols

The Transport layer in the DOD reference model (see Internet is really simple number 5/2003) provides the ability to communicate from one application program to another. In this section, we will look at two protocols in the Transport Layer: Transport Control Protocol (TCP) and User Datagram Protocol (UDP). All packets exchanged between computers on the network are thanks to one of the two protocols.

The Transport layer in the DOD reference model (see Internet is really simple number 5/2003) provides the ability to communicate from one application program to another. In this section, we will look at two protocols in the Transport Layer: Transport Control Protocol (TCP) and User Datagram Protocol (UDP). All packets exchanged between computers on the network are thanks to one of the two protocols.

1.**Transport Control Protocol (TCP)**

In two protocols in Transport layer, TCP is the most used protocol. This is connection-oriented protocol, there are 5 characteristics of reliable delivery service. That is:

+ Stream orientation: When two application programs (user processes) transmit large amounts of data, we see this data as a series of bits, divided into 8-bit octets, which they We often call bytes. The stream delivery service destination host name is sent to the destination correctly with the same sequence of octets that the machine sends.

+ Virtual circuit connection: Performing stream transmission is similar to making a phone call. Before the transmission can begin, both the application program sends and the application program receives interaction with the operating system, notifying the desire to obtain stream transmission. Conceptually, an application program will make a "call" that must be accepted by the other end, ie establish a connection - or virtual circuit - to transmit and receive data correctly.

+ Transmission with buffers: Application programs send a stream of data over a virtual circuit by repeating the transfer of data octets to the protocol software. When transmitting data, each application program uses any transmission unit size it finds convenient, maybe only one octet. At the receiving end, the automatic transfer protocol software data follows the exact order in which they are sent, making them available for use with the receiving application program, as soon as they are received and checked. look up. The protocol software is free to divide the data stream into data packets independent of the unit that the application program transmits. To make the transmission more efficient and minimal on the network, the settings often gather enough data from the data stream to place in the appropriate large datagram before transmitting it over the Internet.

+ Stream unstructured: It is important that the stream TCP service does not identify structured data streams. For example, the employee payroll program, there is no way for the stream service to mark the border between employee records, or to determine where the data stream is employee data. Application programs that use stream services must understand the stream content and unify the stream format before starting the connection.

+ Two-way connection: Connections are provided by the TCP stream service to allow simultaneous transmission from both directions. This connection is called duplex (full deplex). From the perspective of an application process, a two-dimensional context consists of two independent data streams 'running' in opposite directions, and without interaction or collision. The stream service allows an application process to terminate "flow" in one direction while data continues to "run" the other way, making the connection one-way (half duplex). The main advantage of bidirectional connectivity is that the basic protocol software can send control information to a stream back to the source in the reverse datagrams in the opposite direction. This reduces traffic on the network.

Format TCP segment

The unit transmitting between TCP software on two machines is called segment. Segments are exchanged to establish a connection, to transmit data, to send acknowledgments, to notify window sizes (to optimize data transmission and reception). and to close the connection.

Each segment is divided into two parts, the head and the data part. The first part, named TCP header (TCP header), conveys control information and other necessary identifiers. The two most important areas in the TCP header are SOURCE PORT and DESTINATION PORT that contain TCP port values ??to identify application programs at both ends. Whenever the TCP receives packets (called packets) from IP, TCP removes the IP header and reads the TCP header of the segmet result. When TCP reads DESTINATION PORT, it will look in the file containing the service information to send data to the program corresponding to that port number.

SEQUENCE NUMBER area determines the location in the sequence of data bytes in segmet of the sender. The ACKNOWLEDGEMENT NUMBER area determines the number of octets that the source is waiting to receive next. Note that SEQUENCE NUMBER refers to the amount of data in the same direction as the segment, while the ACKNOWLEDGEMENT NUMBER value refers to the amount of data in the opposite direction to the segment.

The HLEN region contains an integer to determine the length of the segment header, calculated by multiples of 32 bits. HLEN values ??are needed because the OPTIONS area has a variable length, depending on which choices are included. Thus, the size of the TCP header is also changed depending on the choices that have been taken. RESERVED region is reserved for future use.

There are segms that only convey acknowledgment, there are other segments that transmit data. There are also segments that transfer requests to establish or close a connection. TCP software uses the FLAG area to determine the purpose and content of the segment. The TCP software also informs how much data it is willing to receive each time a segment is sent by describing its buffer size in the WINDOW area.

Picture 1 of Internet: Transport layer protocols

Three-way Handshake and sliding window (Sliding Window)

At the beginning of each TCP session, the computer sends and receives data to perform a 3-way handshake.Each step uses a segment with only TCP headers without data.First, the computer sends data sent to the receiver of a segment with the following information: The synchronous flag (set in the Flag area) is set to on (on) state, the Sequence number for the segment to be sent later and the value of the size of its window size.Next, the computer receives the data that will respond by a segment with the information: The synchronous flag is also on, the

sequence number is set to the value of the segment it expects to receive from the computer to send data and click data buffer size.Finally, the computer sending the data will send an acknowledgemnet to the sequence number to which the computer receives the data expected in the second step.Through this procedure, 2 computers will be ready for data transmission and reception.In the 3-step handshake procedure, each computer also controls the size of the buffer to send data in accordance with the size of the data buffer buffer area.TCP will receive data from the Application Layrer above it, dividing the data into segments and attaching to each TCP header segment.TCP sends only segments that fit its buffer size and sends its data and starts a timer for the segment to be sent.If the timer expires (time out) and the computer receives data that has not yet responded to the confirmation, the computer sends the data and sends the segment again.When TCP receives acknowledgment for the sent segment, it will continue to send pending segments.After all data has been sent and confirmed, TCP closes the current session.

User Datagram Protocol (UDP)

Like TCP, UDP also transmits data between applications. UDP is a connectionless protocol, it does not test data like TCP. UDP provides low-cost datagram delivery because it does not contain control information in the UDP header.

Applications need to select UDP or TCP when a data delivery service is needed. Although UDP provides a less reliable service (does not guarantee packet order, error control or data flow), its simplicity allows applications to interact directly with the IP protocol.

UDP packet format

Picture 2 of Internet: Transport layer protocols

As can be seen in the figure, the UDP header does not contain any control information at only 8 bytes in size.Because there is no control information, UDP is less reliable than TCP.Applications written to use UDP as a transport protocol must perform a number of procedures for checking data delivery.However, some other UDP applications may not need to perform any testing.For example, most email applications use the UDP protocol.They simply send data without performing a test to ensure that the message has arrived intact at the destination computer.

Applications that use UDP

Although UDP does not provide reliable delivery, there are many types of applications suitable for using it as a transport protocol. In any case, TCP becomes too complicated, too slow or simply unnecessary, application developers can use UDP as an alternative. Applications that use UDP can be applications that themselves have a method of checking data delivery or applications that match the query / response model (query / response).

Picture 3 of Internet: Transport layer protocols