What is a Network Protocol?

A protocol is a set of rules that govern the communications between computers on a network. These rules include guidelines that regulate the following characteristics of a network:

- access method,
- allowed physical topologies,
- types of cabling,
- speed of data transfer.

Protocols define the format, timing, sequence, and error checking used on the network.

In other manner, to provide shared access to information and resources, LANs must deliver information that is correct, in proper time sequential order, and understood by the recipient. To accomplish these functions, electrical circuits, error detection systems, error correction systems, information coding, data flow control, data formatting systems, and other hardware/software subsystems must all perform in a cooperative fashion following a set of rules or "protocols".

Protocols can be implemented either in hardware or software or a mixture of both. Typically, the lower layers are implemented in hardware, with the higher layers being implemented in software.

Protocols could be grouped into suites (or families, or stacks) by their technical functions, or origin of the protocol introduction, or both. A protocol may belong to one or multiple protocol suites, depending on how you categorize it. For example, the Gigabit Ethernet protocol IEEE 802.3z is a LAN (Local Area Network) protocol and it can also be used in MAN (Metropolitan Area Network) communications.

The OSI model provides a conceptual framework for communication between computers, but the model itself is not a method of communication. Actual communication is made possible by using communication protocols. In the context of data networking, a protocol is a formal set of rules and conventions that governs how computers exchange information over a network medium. A protocol implements the functions of one or more of the OSI layers.

A wide variety of communication protocols exist. Some of these protocols include LAN protocols,
WAN protocols, network protocols, and routing protocols. LAN protocols operate at the physical and data link layers of the OSI model and define communication over various LAN media. WAN protocols operate at the lowest three layers of the OSI model and define communication over the various wide-area media. Routing protocols are network layer protocols that are responsible for exchanging information between routers so that the routers can select the proper path for network traffic. Finally, network protocols are the various upper-layer protocols that exist in a given protocol suite. Many protocols rely on others for operation. For example, many routing protocols use network protocols to exchange information between routers. This concept of building upon the layers already in existence is the foundation of the OSI model.

**Protocols and OSI model**

The OSI model, and any other network communication model, provides only a conceptual framework for communication between computers, but the model itself does not provide specific methods of communication. Actual communication is defined by various communication protocols. In modern protocol design, protocols are "layered" according to the OSI 7 layer model or a similar layered model. Layering is a design principle which divides the protocol design into a number of smaller parts, each part accomplishing a particular sub-task and interacting with the other parts of the protocol only in a small number of well-defined ways.

**Layering benefits in networks**

- Layering allows the parts of a protocol to be designed and tested without a combinatorial explosion of cases, keeping each design relatively simple.
- Layering also permits familiar protocols to be adapted to unusual circumstances.

A multitude of different transport protocols are available on Windows, such as TCP, UDP, IPX, and SPX. Each protocol behaves differently. Some require a connection to be established before sending or receiving data. Others don't guarantee the reliability or integrity of the data.

**Protocol Key Elements**

A protocol defines what is communicated, how is it communicated and when is it communicated. The key elements of a protocol are:

- Syntax,
- Semantics,
Timing.

Let's explain these elements:

✓ Syntax

The term syntax refers to the structure or format of the data, meaning the order in which they are presented.

Example:
A simple protocol might expect the first 8 bits of data to be the address of the sender, the second 8 bits to be the address of the receiver, and the rest of the stream to be the message itself.

✓ Semantic

The term semantic refers to the meaning of each section of bits:

- How a particular pattern to be interpreted,
- What action is to be taken based on that interpretation?

Example:
Does an address identify the route to be taken or the final destination of a message?

✓ Timing

The term timing refers to two characteristics:

- When data should be sent,
- How fast they can be sent.
Example:
If a sender produces at 100 Mbps but the receiver can process data at only 1 Mbps, the transmission will overload the receiver and some data will be lost.

odef Some Protocol Types
A protocol provides either connection-oriented services or connectionless services and can be routable and non-routable. Let's see these types:

- Connection-oriented protocols
In connection-oriented services, a path is established between the two communicating parties before any data is exchanged. This ensures that there is a route between the two parties in addition to ensuring that both parties are alive and responding. In addition, most connection-oriented protocols guarantee delivery, which increases overhead as additional computations are performed to verify correctness.

- Connectionless protocols
On the other hand, a connectionless protocol makes no guarantees that the recipient is listening. A connectionless service is similar to the postal service: the sender addresses a letter to a particular person and puts it in the mail. The sender doesn't know if the recipient is expecting to receive a letter or if severe storms are preventing the post office from delivering the message.
Routable protocols

If a protocol is routable, a successful communication path can be set up (either a virtual connection-oriented circuit or a data path for datagram communication) between two workstations, no matter what network hardware lies between them. For example, machine A is on a separate network from machine B. A router linking the two networks separates the two machines. A routable protocol realizes that machine B is not on the same network as machine A; therefore, the protocol directs the data to the router, which decides how to best forward it so that it reaches machine B.

Routable protocols are communications protocol that contains a network address as well as a device address. It allows packets of information to be forwarded from one network to another. This means information can be routed anywhere—it uses both an exterior and interior routing system to transmit data.

Non-routable protocols

A non-routable protocol is not able to make such provisions; the router drops any packets of non-routable protocols that it receives. The router does not forward a packet from a non-routable protocol even if the packet's intended destination is on the connected subnet. NetBEUI is the only A Three-Layer Model.

Non-routable protocols, which regulate the transfer of data, always use interior routing systems as a way to transmit this data. A non-routable protocol is "a communications protocol that contains only a device address and not a network address. It does not incorporate an addressing scheme for sending data from one network to another." This means that these protocols are very specific about where information can routed, and it must be within the interior network. Some people may find this type of protocol very limited.
The main power of the Internet lies in two protocols:

- the Transmission Control Protocol (TCP),
- the Internet Protocol (IP).

Because of this, the Internet suite of protocols is referred to as the TCP/IP protocol suite (Fig. 1). TCP is very close to the definition of transport layer of the OSI-RM. It provides a connection-oriented delivery of data between two distributed applications in the Internet. TCP provides this reliable service on the top of an unreliable IP.

IP is responsible for the routing and switching of datagrams through the Internet. We summarize these functions in the following table.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>Provides a connection-oriented delivery of data between two distributed applications in the Internet.</td>
</tr>
<tr>
<td>IP</td>
<td>Responsible for the routing and switching of datagrams through the Internet.</td>
</tr>
</tbody>
</table>
Here is a list of acronyms used in Figure 1:

- FTP: File Transfer Protocol
- HTTP: Hyper Text Transfer Protocol
- SMTP: Simple Mail Transfer Protocol
- SNMP: Simple Network Management Protocol
- TCP: Transmission Control Protocol
- UDP: User Datagram Protocol
- IP: Internet Protocol
- ICMP: Internet Control Message Protocol
A comparison between TCP and IP

In the following figure we compare the main characteristics between TCP and IP:

Fig. 2: A comparison between TCP and IP

A standard practice is to show the protocol stack for the TCP/IP suite and compare it with the OSI-RM. Such a stack is shown in the Figure 3.

TCP/IP Compared to the OSI Model

To compare TCP/IP to OSI model, we first give the two models as illustrated in figure 3.

Fig. 3: TCP/IP compared to the OSI Model
• Application layer roughly corresponds to Session, Application, and Presentation layers of OSI Model
• Transport layer roughly corresponds to Transport layers of OSI Model
• Internet layer is equivalent to Network layer of OSI Model
• Network Interface layer roughly corresponds to Data Link and Physical layers of OSI Model

❖ The Transmission Control Protocol (TCP)

TCP provides a connection-oriented, full-duplex, reliable, streamed delivery service using IP to transport messages between two processes.

Reliability is ensured by:
• Connection-oriented service
• Flow control
• Error detection
• Error control
• Congestion avoidance algorithms;

The data part of the IP packet is the TCP packet. TCP takes care of the imperfections of IP and protocols below IP. As mentioned earlier, the IP protocol may result in packets arriving out-of-sequence at the destination. This is due to the possibility that each IP packet could be taking different route to destination. TCP provides the capability of re-arranging the packets back in sequence. IP packet may get lost and TCP would request a duplicate copy of the lost packet. Besides, error checking in IP is provided only for the packet header. The TCP provides for the error checking of the complete packet. If the reliability functions (in-sequence delivery, error checking and congestion control) are embedded in IP, then the IP would become very reliable. However, the data transfer rate will suffer because IP header is processed at every node. The TCP is an end-to-end protocol and therefore is processed only at the sending and destination computers.

TCP is connection-oriented. This means that the protocols on two communicating computers set up a connection before actual data transfer takes place. This is a very desirable property of TCP. However, if the actual data message is short, then connection setting up and error checking adds too much overhead. Besides, if the TCP at the receiving computer finds a packet in error, it requests the TCP at the sending station to retransmit a copy of the packet. This is not a good error recovery mechanism for delay sensitive applications. Due to these two inefficiencies of TCP, the Internet protocol suite includes a connectionless protocol at the same level. It's called the user datagram
protocol (UDP). UDP is a connectionless protocol and does not require the use of retransmission of packets. However, the capability of error control is included as an option in UDP.

Used Addresses in TCP/IP

The used Addresses in TCP/IP are (Fig. 5):

- Port address,
- Internet address,
- Physical address

The buffer size of these addresses is illustrated in the following table:

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Buffer Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port Address</td>
<td>16</td>
</tr>
<tr>
<td>Internet Address</td>
<td>32</td>
</tr>
<tr>
<td>Ethernet Address</td>
<td>48</td>
</tr>
</tbody>
</table>

Fig. 5: Three types of addresses used in TCP/IP
Connection-oriented service

TCP performs data communication in full-duplex mode, that is both the sender and receiver processes can send segments simultaneously. For connection establishment in full-duplex mode, a four-way protocol can be used. However, the second and third steps can be combined to form a three-way handshaking protocol.

Reliable Communication

To ensure reliability, TCP performs flow control, error control and congestion control.

Flow control:
TCP uses byte-oriented sliding window protocol, which allows efficient transmission of data and at the same time the destination host is not overwhelmed with data. The receiver has a buffer size of 8 K bytes. After receiving 4 K bytes, the window size is reduced to 4 Kbytes. After receiving another 3 K bytes, the window size reduces to 1 K bytes. After the buffer gets empty by 4 K bytes, the window size increases to 7 K bytes. So it may be noted that the window size is totally controlled by the receiver window size, which can be increased or decreased dynamically by the destination. The destination host can send acknowledgement any time.

Error Control:
Error control in TCP includes mechanism for detecting corrupted segments with the help of checksum field. Acknowledgement method is used to confirm the receipt of uncorrupted data. If the acknowledgement is not received before the time-out, it is assumed that the data or the acknowledgement has been corrupted or lost. It may be noted that there is no negative acknowledgement in TCP. To keep track of lost or discarded segments and to perform the operations smoothly, the following four timers are used by TCP:

- Retransmission; it is dynamically decided by the round trip delay time.
- Persistence; this is used to deal with window size advertisement.
- Kep-alive; commonly used in situations where there is long idle connection between two processes
- Time-waited; it is used during connection terminations

Congestion control:
To avoid congestion, the sender process uses two strategies known as slow-start and additive increase, and the send one is known as multiplicative decrease. To start with, the congestion window
size is set to the maximum segment size and for each segment that is acknowledged, the size of the congestion window size is increased by maximum segment size until it reaches one-half of the allowable window size. Ironically, this is known as slow-start, although the rate of increase is exponential. After reaching the threshold, the window size is increased by one segment for each acknowledgement. This continues till there is no time-out. When a time-out occurs, the threshold is set to one-half of the last congestion window size.

**TCP/IP Topology**

The TCP/IP protocol suite can be used on stand-alone LANs and WANs or on complex internetworks created by gluing many networks together. Figure 6 illustrates stand-alone network links. Any hosts that are equipped with TCP/IP can communicate with one another across a LAN, point-to-point line, or wide area packet network.

Networks are joined into an internetwork by means of IP routers. Figure 7 shows an internetwork that was created by connecting the stand-alone networks together via IP routers.

*Fig.6: Stand-alone networks.*
Fig. 7: Gluing networks together with routers.