TCP Technology and Testing Methodologies

By Hammadoun Dicko, Product Specialist, EXFO

As enterprises use more and more applications, such as Voice-over-IP (VoIP), Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP), service providers are now faced with the obligation to enforce stringent service level agreements (SLA). Furthermore, the typical SLA parameters such as throughput, latency, jitter and frame loss only cover the network performance up to the IP (Internet Protocol) layer and do not necessarily reflect the true user experience. How can service providers make sure that the end-user's most important applications make use of the full bandwidth?

TRANSMISSION CONTROL PROTOCOL

TCP is one of the two original components of the IP suite commonly referred to as TCP/IP. It provides connection-oriented, end-to-end communication services at an intermediate level between application programs and the IP. It offers reliable communication and guarantees orderly delivery to the upper layers for non-real-time applications such as email, FTP, FTTP, etc. The term connection-oriented means the two applications must establish a TCP connection before they can exchange data.

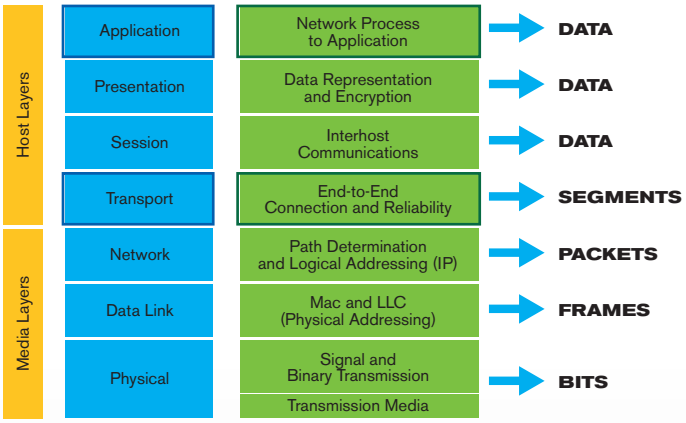

Figure 1. OSI reference model and nomenclature

HOW TCP OPERATES

The primary purpose of TCP is to provide reliable connection services between hosts. However, this becomes challenging on less reliable networks such as the Internet. This hurdle is overcome by the implementation of flow control, which ensures the integrity of each segment sent, and the congestion control mechanism for each byte stream, which allows the receiver to limit the amount of data a sender can transmit. To accomplish this, TCP provides the following:

Basic Data Transfer

TCP is able to transfer a continuous stream of bytes in each direction between applications by packaging the traffic into TCP segments, which are passed to the IP layer for transmission. TCP has the ability to decide when to block or forward data.

Reliability

TCP is able to recover from data that are damaged, lost, duplicated or delivered out of order by assigning a sequence number to each byte transmitted, and requiring a positive acknowledgement (ACK) from the far end. If the ACK is not received within the timeout interval, the data is retransmitted. In addition, the receiver uses the sequence number to rearrange segments that may be received out of order and eliminate duplicate segments. A checksum added to each transmitted segment is checked at the receiving end to discard damaged segments.

Flow Control

The receiver controls the amount of data the transmitter can send by returning a window size value with every ACK. The window size value indicates the number of bytes the sender may transmit before receiving further permission. In addition, the sequence numbers and receive windows behave like clocks that shift every time the recipient receives and acknowledges a new data segment. The sequence number loops back to zero, once it runs out of numbers. Figure 2 is a visual representation of the sequence numbers and its maximum values in the TCP.

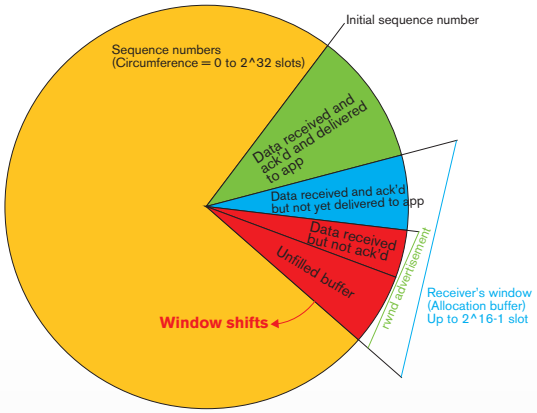

Figure 2. TCP window

Multiplexing

Many processes or communications can run within a single TCP host. A network socket uniquely identifies each connection by binding ports to processes. Consequently, multiple sockets can be used during a single exchange between two hosts, thus reducing the impact of high-latency networks and the window allocation buffer limit.

Connections

Reliability and flow-control mechanisms are possible because the status information is maintained for each data stream. The combination of the status, including sockets, sequence numbers and window sizes, is called logical connection, which is uniquely identified by a pair of sockets used by the sending and receiving parties.

The end-to-end TCP communication between two devices is carried out in many steps:

1. Establish a connection between two end-points.
2. Manage the exchange of information, making sure that packets are delivered without error, and retransmitting them, if necessary.
3. Reorder and remove all duplicate segments received.
4. Provide flow control between the two end-points through a window size value sent with every ACK.
5. Disconnect from the device once the exchange of information is complete.

An example of the flow-control mechanism used by TCP is shown in figure 3. This figure shows the communication from Host A to Host B, although TCP has the capacity to handle full-duplex connection or concurrent data streams in both directions.

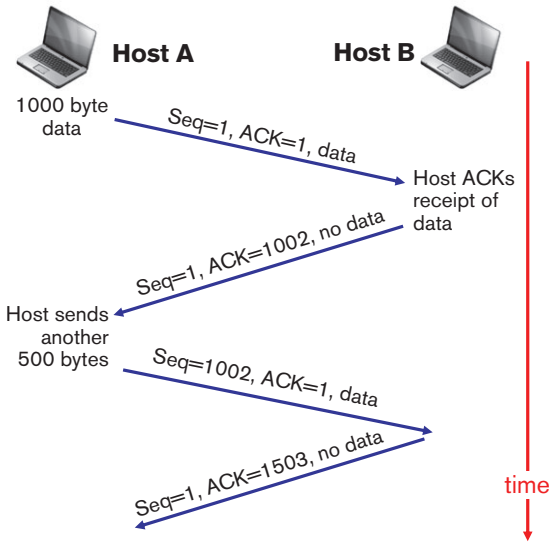

Figure 3: Flow-control mechanism between two hosts

TCP THROUGHPUT

From a purely theoretical perspective, the maximum amount of data that would fill the pipe or the buffer space required at the sender and receiver to obtain a maximum throughput on a TCP connection is known as the "bandwidth-delay product" (BDP). It refers to the amount of unacknowledged data that TCP must handle in order to keep the pipeline full. Bandwidth-delay product is the product of the data link capacity and the roundtrip delay.

BDP or Capacity (bits) = Bandwidth (bits/sec) x Roundtrip Time (seconds)

As the formula shows, the theoretical TCP bandwidth that a circuit is able to carry from one end to the other will be affected by high delays (RTT). In addition, the high bandwidth-delay product value, also known as long fat network (LFN), is a major problem in TCP circuits because the protocol can only attain optimal throughput when the sender transmits a large quantity of data before being acknowledged. If the quantity of data sent does not achieve this large value, then the link is not being kept busy and the protocol is operating below peak efficiency.

IMPORTANCE OF TCP THROUGHPUT TESTING

TCP will transport application data from one end to the other without error and in perfect order, but it will add overhead, which will cause a delay in the transmission.

Without going into great detail, certain TCP protocol parameters can be configured to influence a device's capacity to transfer information efficiently across the network. These parameters are: transmission window size, segment size and retransmission timeout. Furthermore, roundtrip delay and frame loss are the most important factors in TCP link operation.

Other factors will also affect an application's ability to transmit data across a network. These factors include the application being used, the type of TCP/IP stack and the performance of the computers/servers running the applications.

TCP TESTING METHODOLOGIES

When turning up and troubleshooting data services or mobile backhaul, service providers typically use layer 2 or 3 test methodologies, which are based on ITU-T Y.1564. EtherSAM can simulate all types of services that will run on the network while validating their configuration. It also simultaneously qualifies all key SLA parameters and validates the quality of service (QoS) mechanisms provisioned in the network in order to prioritize the different service types. The result is more accurate validation as well as faster deployment and troubleshooting.

This methodology is perfect for assessing network performance. Metrics such as throughput, frame loss, latency and jitter provide a comprehensive snapshot of network quality, which is the basis of all SLAs. This methodology is also ideal for service activation as it validates the configuration and quality of each service used in a network. This is enough for most service providers because they are only responsible for the service at the delivery point. However, Y.1564 will only provide a general idea of the network performance for applications running on TCP, it will not assess the end-user's quality of service. A TCP throughput test is required in these cases.

End-users tend to base the TCP throughput on the bandwidth statistics provided by the computers/servers that are running their applications, or use software tools that run on computers/operating systems to emulate TCP traffic. The latter will lead them to the conclusion that the service provider is at fault because their measurement shows that their maximum throughput is nowhere near the bandwidth they purchased. Unfortunately, not all operating systems were created equal. Some have a locked TCP/IP stack and use the basic windowing scheme as defined for TCP, which is 65 535 bytes. These software tools are also only as good as the computers they run on. Poor computer performance will reflect poor performance in the measurement, and therefore will provide a false view of network performance.

RFC 6349

RFC 6349 offers a practical recommendation for measuring end-to-end TCP throughput in a predictable, managed IP network. The results are based on throughput as a theoretical achievable amount of data per unit of time, when TCP is in the equilibrium state. Essentially, the recommendation assumes that the IP network has appropriate TCP adjustments in the IP hosts and applications to operate efficiently, so that the bottleneck bandwidth (BB) is attained. As a result, TCP throughput during the transient stages of the TCP connection, such as Slow Start, cannot be predicted.

The recommendation uses three subsets to compute the ideal and expected TCP throughputs:

- **Path MTU detection**: The maximum transmission unit (MTU) of an end-to-end network must be identified because it can vary between hops. This is done to avoid packet fragmentation during all subsequent tests. The easiest way to accomplish this is through ICMP commands; however, network managers tend to disable ICMP. Consequently, RFC 4821 is used to determine the MTU in a path between a source and a destination node.
- **RTT**: The real roundtrip time (RTT) is required to provide the window-size estimation that will be used in the TCP throughput test.
- **TCP throughput**: This test is based on the estimated window sizes and the MTU.

The RFC 6349 is very theoretical and has the following important limitations when it comes to field testing:

- Testing has to be performed at varying time intervals at different times throughout the day to provide a better characterization of TCP throughput since it does not support testing over long periods of time (based on file transfer).
- TCP throughput measurements will be meaningless on networks showing high packet loss (5% or higher) and/or high jitter (150 ms).

JPerf

JPerf is a network testing tool that creates TCP and UDP data streams to measure network throughput. It is based on a client-to-server approach that tests each direction of a circuit, where the server generates traffic while the client receives it. It provides throughput, latency and average duration, but does not measure the window size.

Since JPerf is a software-based application written in C++, it uses the TCP/IP stack of the operating system whose TCP receiver window size may be locked up to its maximum value of 65 535 bytes. This can negatively impact the value of the TCP throughput. Furthermore, the test is measured all the way to layer 7 of the OSI model, where the application resides, which can provide erroneous results because additional overhead may be required.

In some cases, JPerf gives an idea of the TCP throughput value, but this is definitely not a tool that should be used to accurately measure it. Any solution that is based on JPerf will inherit the same flaws.

EXacTCP

EXacTCP is based on TCP Reno, as defined in RFC 793/1122/1323/2581, and performs TCP throughput measurements based on RFC 6349. This TCP test methodology can provide accurate measurements of TCP metrics, such as TCP throughput, RTT and optimal window size, thanks to its hardware-based implementation, which does not rely on any communication stack found in PC operating systems or servers. It also addresses the limitations of RFC 6349.

The TCP throughput measurements are based on the TCP window scale options described in RFC 1323. This means that a single stream can be used to provide the TCP throughput, RTT and window size measurements. Consequently, it fills a circuit at full bandwidth with TCP traffic when the roundtrip time or the transmission bandwidth is too large for standard TCP implementation.

Characteristics	Benefits
TCP throughput measurement up to line rate	This allows service providers to validate the TCP performance for customers who use their high-bandwidth services.
Optimal window-size identification	This lets service providers help their customers configure their equipment to obtain maximum TCP throughput.
Repeatable TCP throughput test	This helps service providers determine their customers' TCP performance at varying times, which is important since TCP performance can vary over time.
Long-term testing	This allows service providers to determine their customers' TCP throughput over a long period of time and verify its stability.

Table 1. EXacTCP benefits

This methodology is easier to use than software-based solutions because the person performing the test does not have to calculate the number of sessions required for the configuration and which TCP port to use on each. From a results perspective, the user does not have to average multiple test results to validate that the circuit is capable of transporting TCP application traffic. Furthermore, having only one TCP test session provides repeatability. If the network conditions (frame loss, roundtrip time, etc.) are the same, the TCP throughput test should yield the same results.

CONCLUSION

To summarize, TCP is used by most non-real-time applications to deliver mission-critical information from one end to the other of a network. Since the TCP protocol must validate that the information was transmitted without any errors, it has a built-in functionality to limit its capacity in high-latency or high-bandwidth networks.

Since applications come in all shapes and sizes and run on a wide range of computers/servers, the TCP/IP implementation and configuration will vary from one end-user to another. This makes it very difficult for service providers to prove that they are fulfilling their customers' requirements.

While service activation tests, such as those based on Y.1564, are adapted to service turn-ups and most troubleshooting, and are the only tests required most of time, they only provide a general idea of the state of network performance for applications running on TCP.

The only way to optimally test TCP throughput is to use a solution that complies with and addresses the shortcomings of TCP standards, namely RFC 6349. EXacTCP is the only standards-based TCP throughput solution that provides accurate, all-inclusive results in one test.

EXFO Headquarters > Tel.: +1 418 683-0211 | Toll-free: +1 800 663-3936 (USA and Canada) | Fax: +1 418 683-2170 | info@EXFO.com | www.EXFO.com

EXFO serves over 2000 customers in more than 100 countries. To find your local office contact details, please go to www.EXFO.com/contact.

  Printed in Canada 13/05