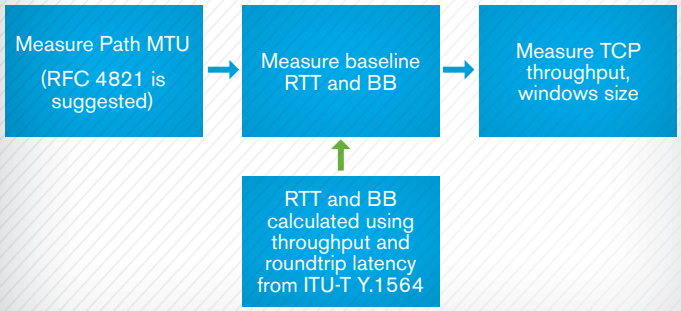
# TCP Technology Overview

## RFC 6349

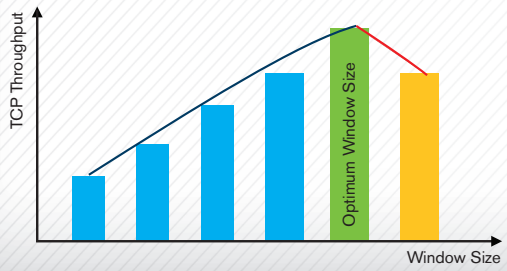
Describes a practical methodology for measuring end-to-end TCP throughput in a managed IP network using the following parameters:

- BB (Bottleneck bandwidth) - the lowest bandwidth along the complete path
- RTT (Roundtrip time)
- Send and receive socket buffers
- Minimum TCP receiver window size
- Path MTU (maximum transmission unit)




## TCP Throughput Measurement - RFC 6349

- TCP throughput determined for specific window sizes
- Starting window size = $\frac{\text{RTT x BB}}{8}$



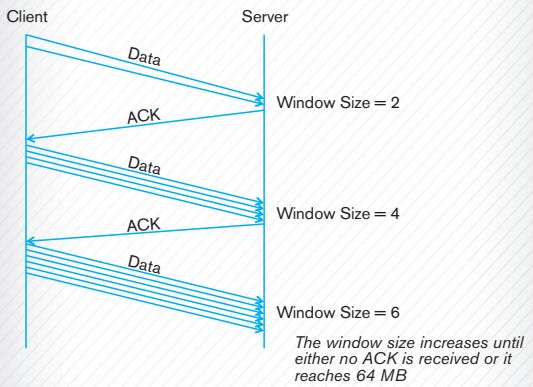

## EXactTCP

The TCP throughput test methodology:

- Is based on RFC 6349.
- Measures the maximum TCP throughput and its corresponding (optimum) window size.
- Uses TCP windows scaling as per RFC 1323.
- Is based on RFC 2581 for TCP flow control (slow start, congestion avoidance and fast retransmit).
- Is hardware-based and allows the full bandwidth to be used with a single TCP connection.

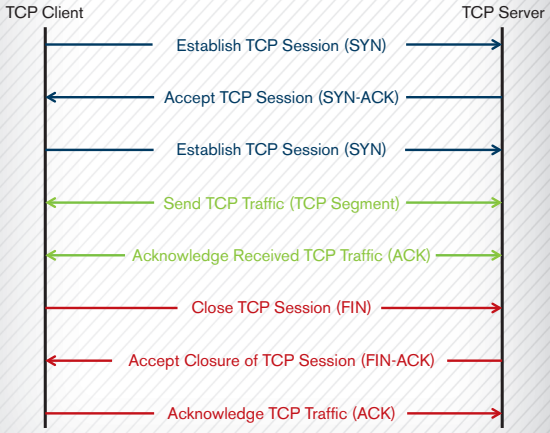
ExacTCP

## TCP Window Scaling

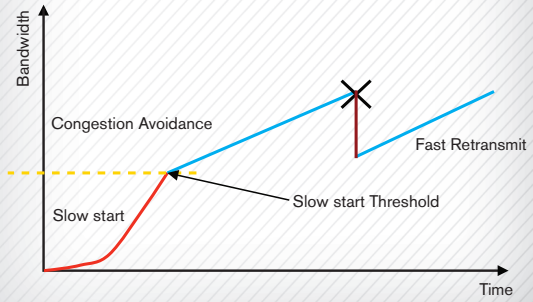



*The window size increases until either no ACK is received or it reaches 64 MB*

## TCP Communication Mechanism




## TCP Transmission Mechanism



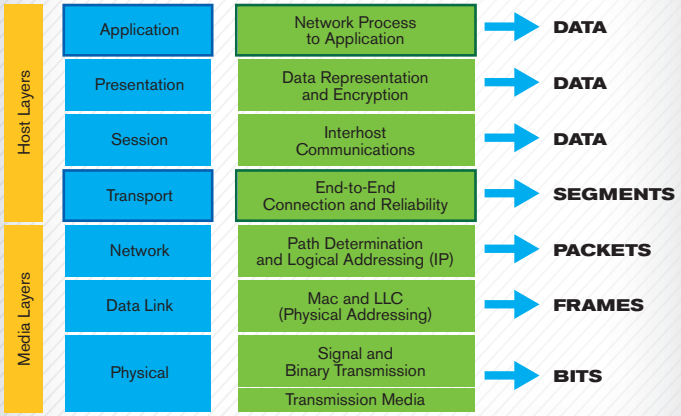

EXFO

# TCP Technology Overview

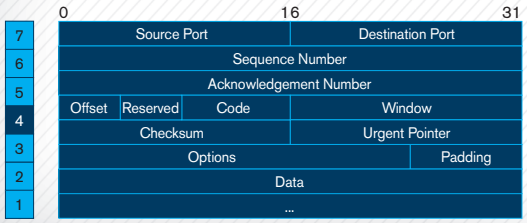
## Definition

| | |
|---|---|
| Transmission Control Protocol (TCP) | Layer-4, connection-oriented protocol based on client-server communication<br>Defined in RFC 793<br>TCP ensures reliable and transparent data transfer between networked end-points; end-to-end error detection and recovery as well as data flow control; segmentation and reassembly of user data and higher-layer protocols. |
| Roundtrip Time (RTT) | The time it takes to send a packet to a remote host and receive the response |
| Window Size | The amount of traffic that can be transmitted without receiving an acknowledgement (ACK) from the other side |
| Bandwidth Delay Product (BDP) or Capacity | The theoretical maximum window size:<br>**BDP = Bandwidth x RTT**<br>The actual maximum window size is 64 MB |

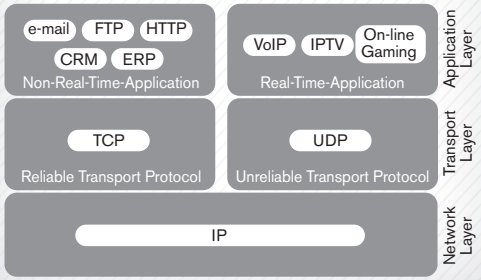
## OSI Model



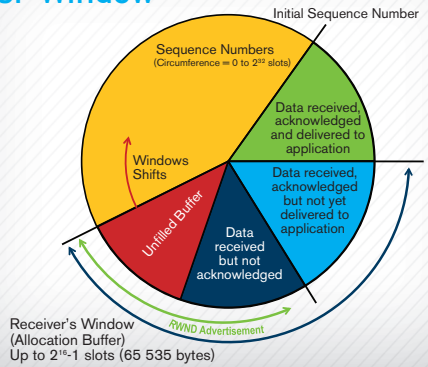

## TCP Segment




## TCP/UDP Applications




## TCP Window




| | |
|---|---|
| Source port (16 bits) | Identifies the sending port |
| Destination port (16 bits) | Identifies the receiving port. |
| Acknowledgment number (32 bits) | If the ACK flag is set, then the value of this field is the next sequence number that the receiver is expecting. This acknowledges receipt of all prior bytes (if any). The first ACK sent by each end acknowledges the other end's initial sequence number itself, but no data. |
| Data offset (4 bits) | The offset from the start of the TCP segment to the actual data. |
| Reserved (3 bits) | For future use and should be set to zero. |
| Code (6 bits) | Contains control bits:<br>URG: Urgent pointer field significant<br>ACK: Acknowledgment field significant<br>PSH: Push function<br>RST: Reset the connection<br>SYN: Synchronize sequence numbers<br>FIN: No more data from sender |
| Window size (16 bits) | The size of the receive window, which specifies the number of window size units (that the sender of this segment is currently willing to receive). |
| Checksum (16 bits) | The 16-bit checksum field is used to check errors in the header and data. |
| Urgent pointer (16 bits) | If the URG flag is set, then this 16-bit field is an offset from the sequence number indicating the last urgent data byte. |
| Options (variable 0–320 bits, divisible by 32) | The length of this field is determined by the data offset field. Options have up to three fields:<br>› Option-Kind: Indicates the type of option, and is the only field that is not optional.<br>› Option-Length: Indicates the total length of the option.<br>› Option-Data: Contains the value of the option. |
| Padding | The TCP header padding is used to ensure that the TCP header ends and data begins on a 32-bit boundary. The padding is composed of zero. |
| Data | The actual data to be carried. |

EXFO