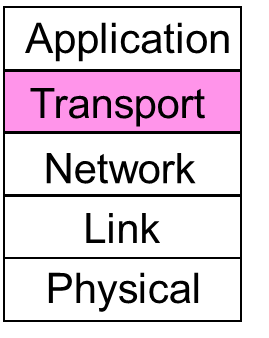
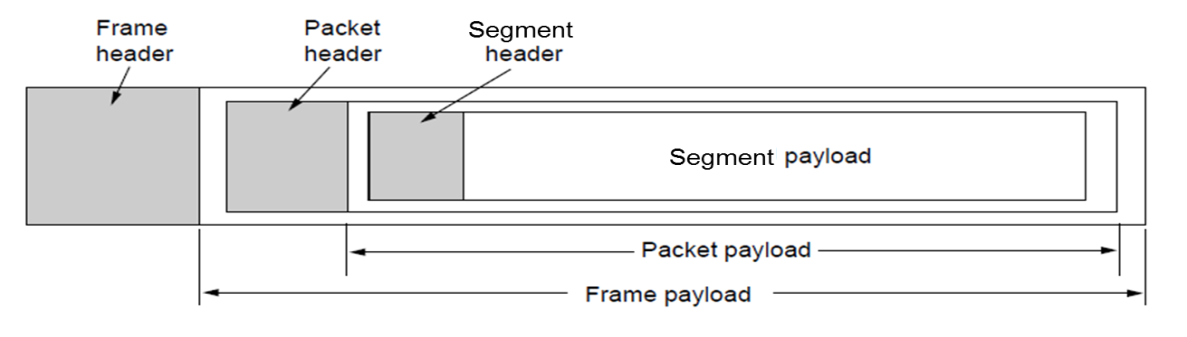
**Chapter 6: The Transport Layer**

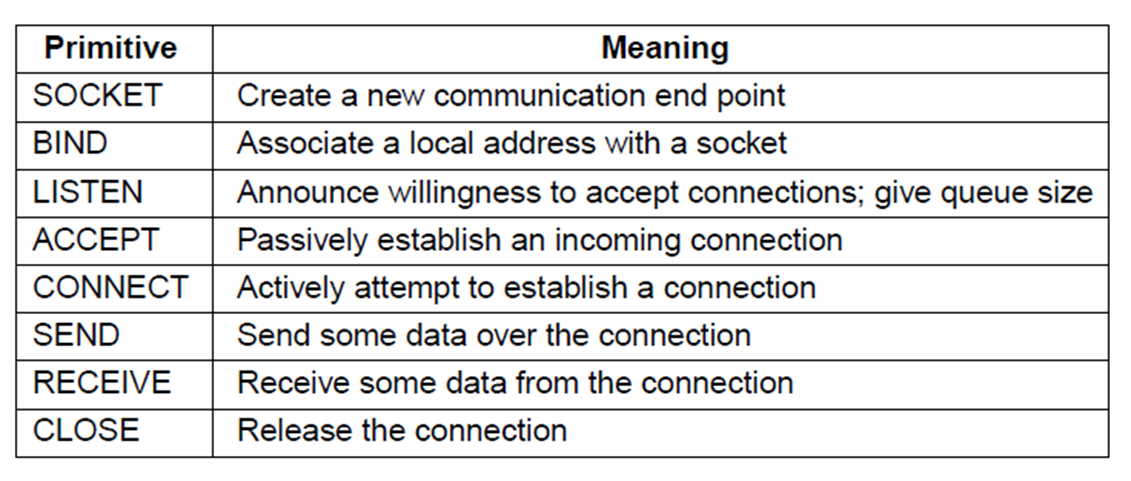


**Segments:** Transport layer breaks a message (such as a file) into segments which it sends in packets, which in turn are sent in frames.

.

**Berkeley sockets:** (or [BSD](http://en.wikipedia.org/wiki/Berkeley_Software_Distribution) sockets) is a [computing library](http://en.wikipedia.org/wiki/Library_%28computing%29) with an [application programming interface](http://en.wikipedia.org/wiki/Application_programming_interface) (API) for [internet sockets](http://en.wikipedia.org/wiki/Internet_socket) and [Unix domain sockets](http://en.wikipedia.org/wiki/Unix_domain_socket), used for [inter-process communication](http://en.wikipedia.org/wiki/Inter-process_communication) (IPC).

As the API has evolved with little modification from a [*de facto* standard](http://en.wikipedia.org/wiki/De_facto_standard) into part of the [POSIX](http://en.wikipedia.org/wiki/POSIX) specification. POSIX sockets are basically Berkeley sockets.



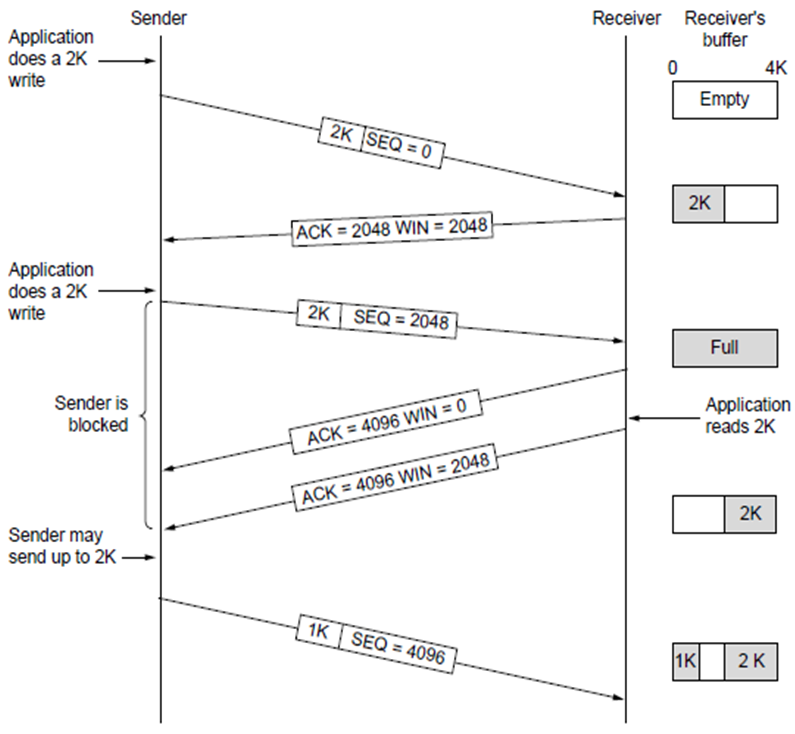
**Be familiar with the Sockets Interface handout on the BB site:**

**Flow control** manages buffering at sender/receiver

* + Issue is that data goes to/from the network and applications at different times
  + Window tells sender available buffering at receiver
  + Makes a variable-size sliding window

**Sliding Window:** the set of (sequence) numbers that the sender/receiver is permitted to send/receive at a given moment in time. Sender and receiver do not need to have the same

window sizes.



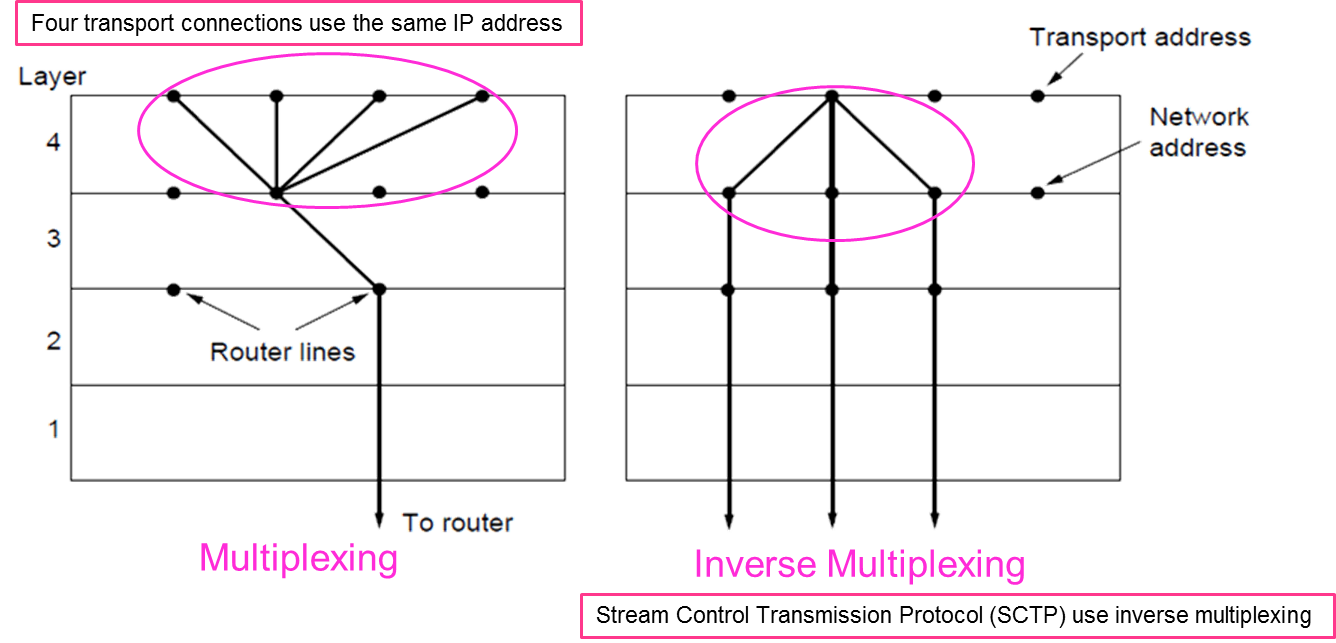
*Solution for silly window syndrome is that receiver should not send window updates for tiny buffer openings like “one byte available”. It should wait until a whole segment can be buffered to avoid this*.

**Pipelining** can be implemented with different choices for error control and buffering. (W=2BD)

Best window size w depends on the **Bandwidth delay product** (BD)

BD = (link bandwidth) x (one-way propagation delay)

* + **Multiplexing:** connections share a network address
  + **Inverse multiplexing:** addresses share a connection



**Packet switching:** Hosts send packets into the network; packets are forwarded by routers. Routers treat packets as messages, receiving (storing) them and then forwarding them based on how the message is addressed.

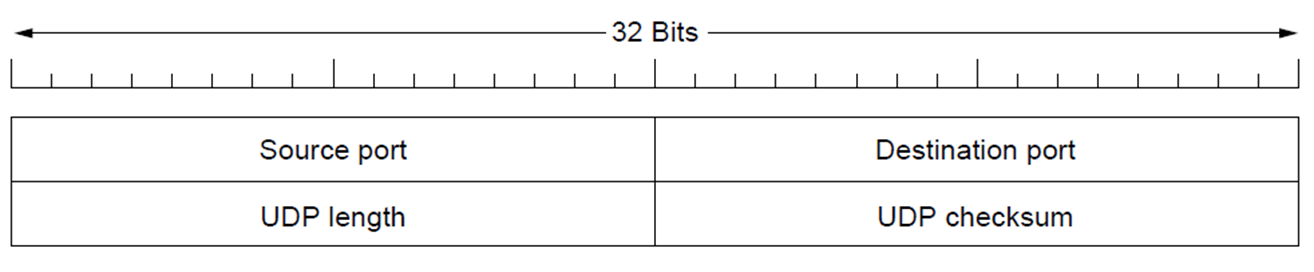
TCP = **Connection-oriented**

UDP = **Connectionless**

Ports and

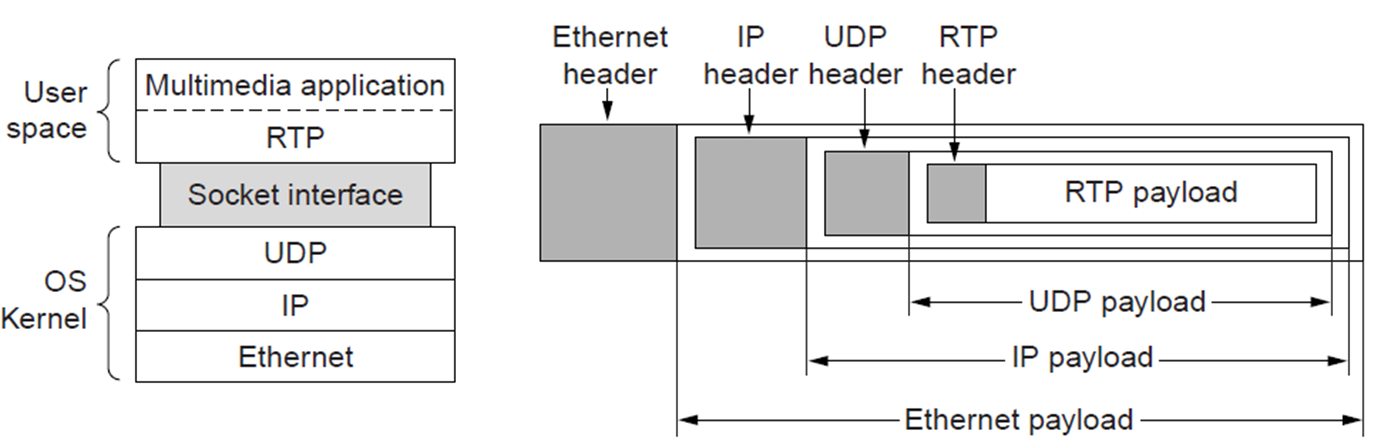
**Sockets:** A transport endpoint is a socket, identified by the combination of the IP address and the port number.

**UDP:** (User Datagram Protocol) connectionless data transport.



**RPC:** (Remote Procedure Call) connects applications over the network with the familiar abstraction of procedure calls

**RTP:** (Real-time Transport Protocol) provides support for sending real-time media over UDP. It is often implemented as part of the application, and can be thought of as being layer 4.5.RTP header contains fields to describe the type of media and synchronize multiple media streams into a single stream of UDP packets.

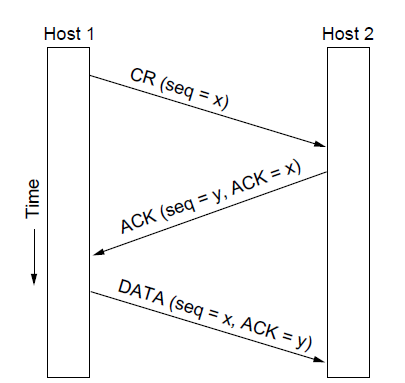


**RTCP:** RTP Control Protocol (RTCP) is a sister protocol of the [Real-time Transport Protocol](http://en.wikipedia.org/wiki/Real-time_Transport_Protocol) (RTP. RTCP provides [out-of-band](http://en.wikipedia.org/wiki/Out-of-band_signaling) statistics and control information for an RTP session. It partners with RTP in the delivery and packaging of multimedia data, but does not transport any media data itself. The primary function of RTCP is to provide feedback on the [quality of service](http://en.wikipedia.org/wiki/Quality_of_service) (QoS) in media distribution by periodically sending statistics information to participants in a streaming multimedia session.

**TCP:** provides applications with a reliable byte stream between processes; it is the workhorse of the Internet. Applications using TCP see only the byte stream and not the segments sent as separate IP packets.

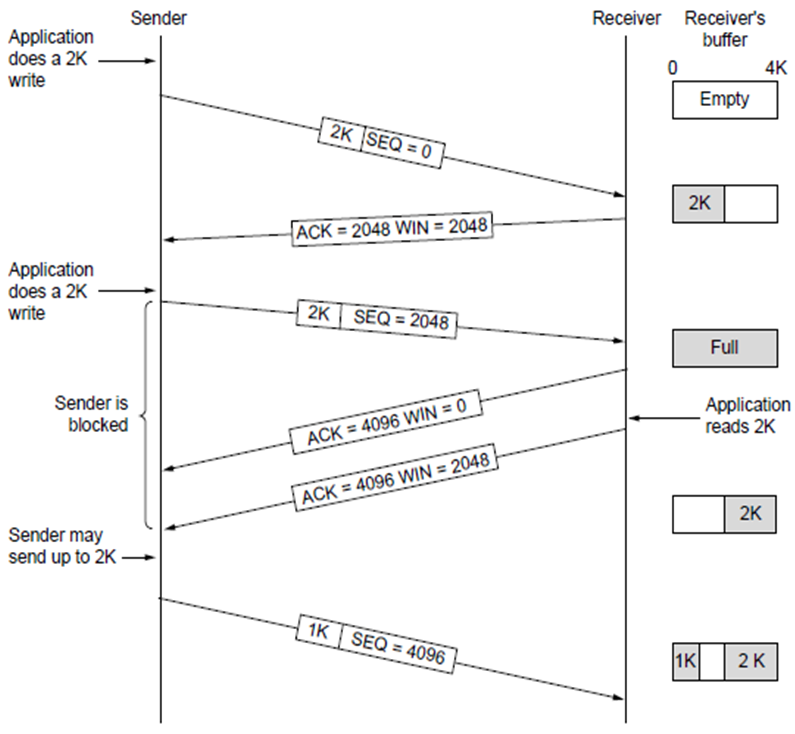
**Segmentation:** a technique for increasing outbound [throughput](http://en.wikipedia.org/wiki/Throughput) of high-[bandwidth](http://en.wikipedia.org/wiki/Bandwidth_%28computing%29) network connections by reducing [CPU](http://en.wikipedia.org/wiki/Central_processing_unit) overhead. It works by queuing up large [buffers](http://en.wikipedia.org/wiki/Buffer_%28computer_science%29) and letting the [network interface card](http://en.wikipedia.org/wiki/Network_interface_card) (NIC) split them into separate packets.

**TCP byte stream:** A byte Stream is a series of [bytes](http://en.wikipedia.org/wiki/Byte). It is also a [communication](http://en.wikipedia.org/wiki/Communication) channel down which one entity can send a sequence of bytes to the entity on the other end. Such channel is often bidirectional, but sometimes unidirectional. In almost all instances, the channel has the property that it is reliable; i.e. exactly the same bytes emerge, in exactly the same order, at the other end. [TCP](http://en.wikipedia.org/wiki/Transmission_Control_Protocol) [communications protocol](http://en.wikipedia.org/wiki/Communications_protocol) transports a byte stream without synchronous timing.



*TCP connection establishment (three-way handshake)*

**TCP sliding window:** TCP adds flow control to the sliding window as before. (ACK + WIN is the sender’s limit.)



**TCP timer management:** TCP estimates retransmit timer from segment RTTs(Round Trip Times.) TCP tracks both average and variance (for Internet case) and then timeout is set to average plus 4 x variance.