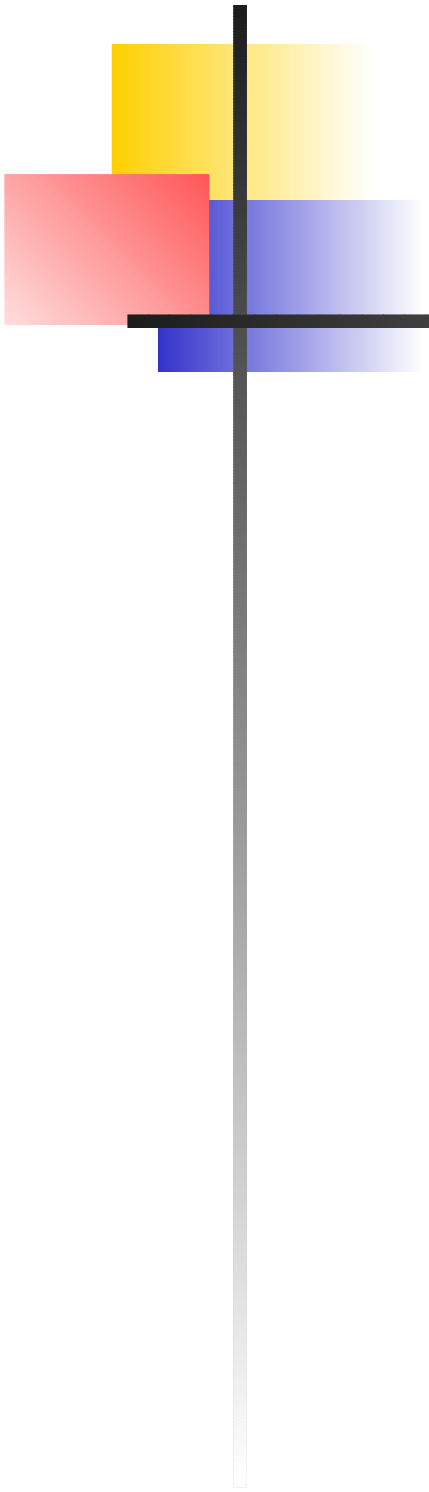


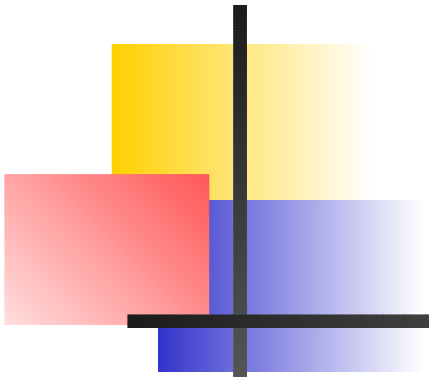
Data Transport over IP Networks

Derek Konigsberg

octo@logicprobe.org

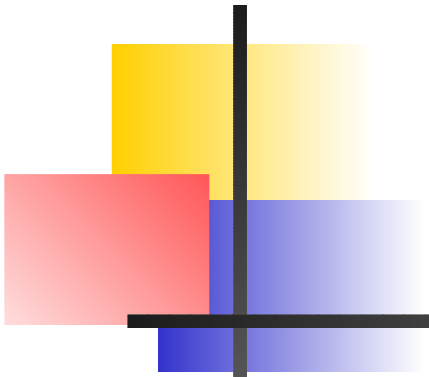
AITP

University of Central Florida



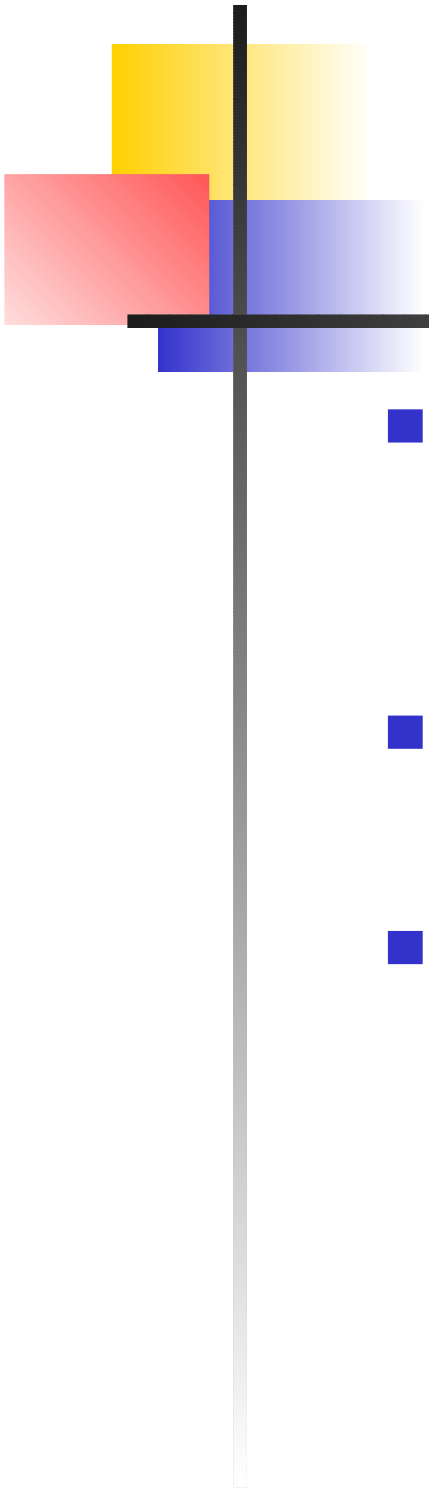
Introduction

- The TCP/IP protocol suite was created by DARPA for the creation of a resilient computer network
- At the foundation is the Internet Protocol (IP), which forms the foundation of the network
- On top of IP, we have protocols such as TCP and UDP, which are responsible for actually managing communications across the network
- Today, we will be focusing on these transport protocols



OSI Model

- 7 - Application
- 6 - Presentation
- 5 - Session
- 4 - Transport (TCP, UDP)
- 3 - Network (IP, IPX)
- 2 - Data Link (Ethernet, Token Ring, FDDI)
- 1 - Physical (twisted pair, coax, fiber)

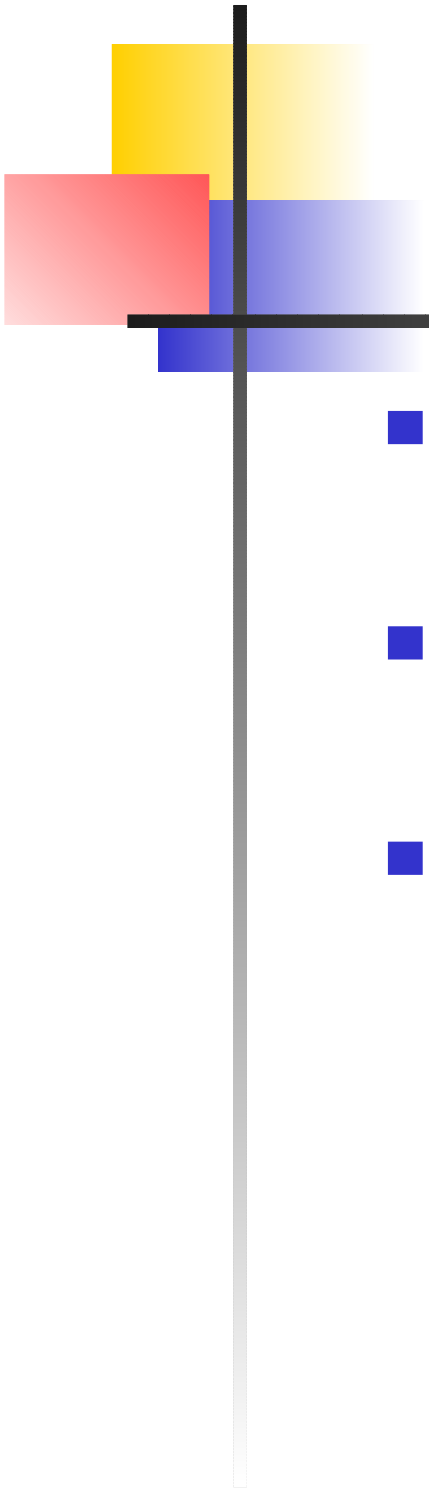


What does IP offer?

- IP is the protocol that takes our network packets from our computers and actually gets them across the network.
- IP is based on the concept of unordered best-effort delivery
- While IP forms the foundation of our network, it is not actually a protocol used directly by our applications for sending data

Types of Transport

- Connectionless unordered delivery
  - The most basic form of transport, optimised for raw speed
  - Useful for applications that can lose packets
- Reliable connection-oriented delivery
  - More complex transport, involving ordering, error correction, and flow control
  - Introduces the concept of a “session” for communication

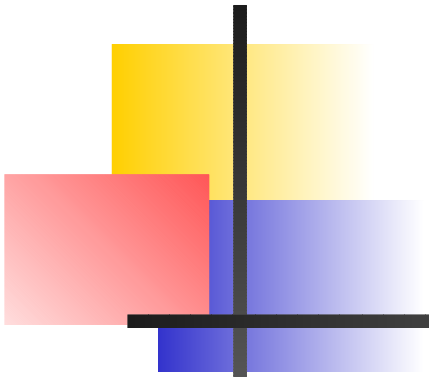


Why not IP directly?

- IP addresses only identify hosts on the network
- IP addresses do not identify applications running on those hosts
- Transport protocols introduce the concept of “port numbers,” which identify the actual client or server program trying to communicate

Port numbers

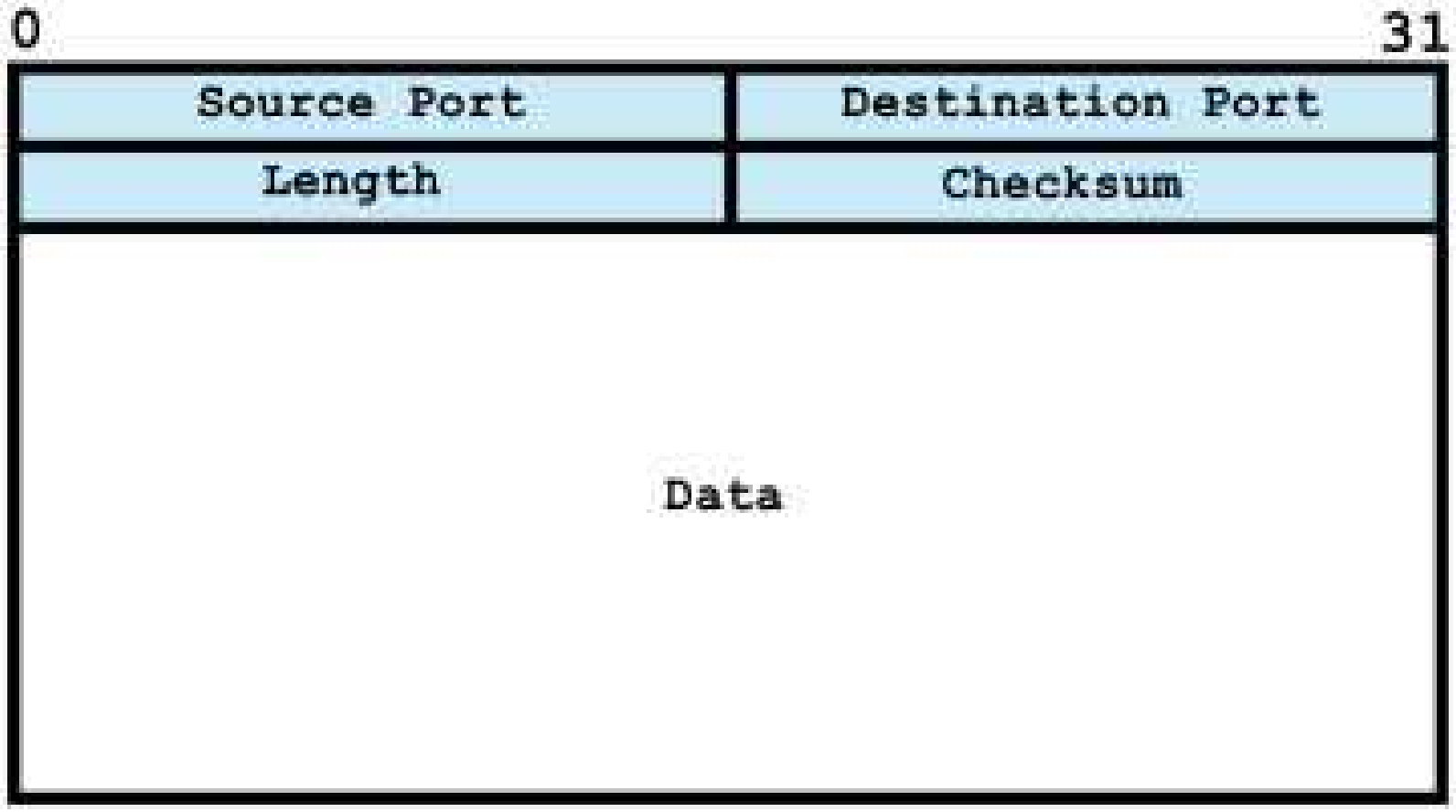
- “Source Port”
  - Identifies the client-end; often chosen when making a connection, and is usually arbitrary
- “Destination Port”
  - Identifies the server-end; chosen ahead of time, and must be known by the client in order to talk to the server
  - A list of well-known port numbers for common protocols can be found in the “/etc/services” file on any UNIX system



User Datagram Protocol

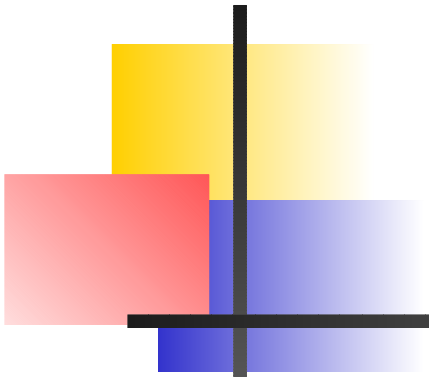
- The User Datagram Protocol, or UDP, is a very simple protocol on top of IP.
- UDP provides unreliable connectionless delivery
- Essentially, UDP is a very thin layer on top of IP, that does little more than identify the port numbers of the applications engaging in communication

UDP Packet Header



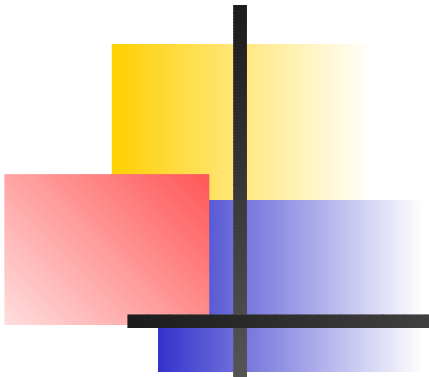
Common uses of UDP

- Video streaming
- Multi-player gaming
- Network filesystems (with reliability added by upper-layer protocols)



Transmission Control Protocol

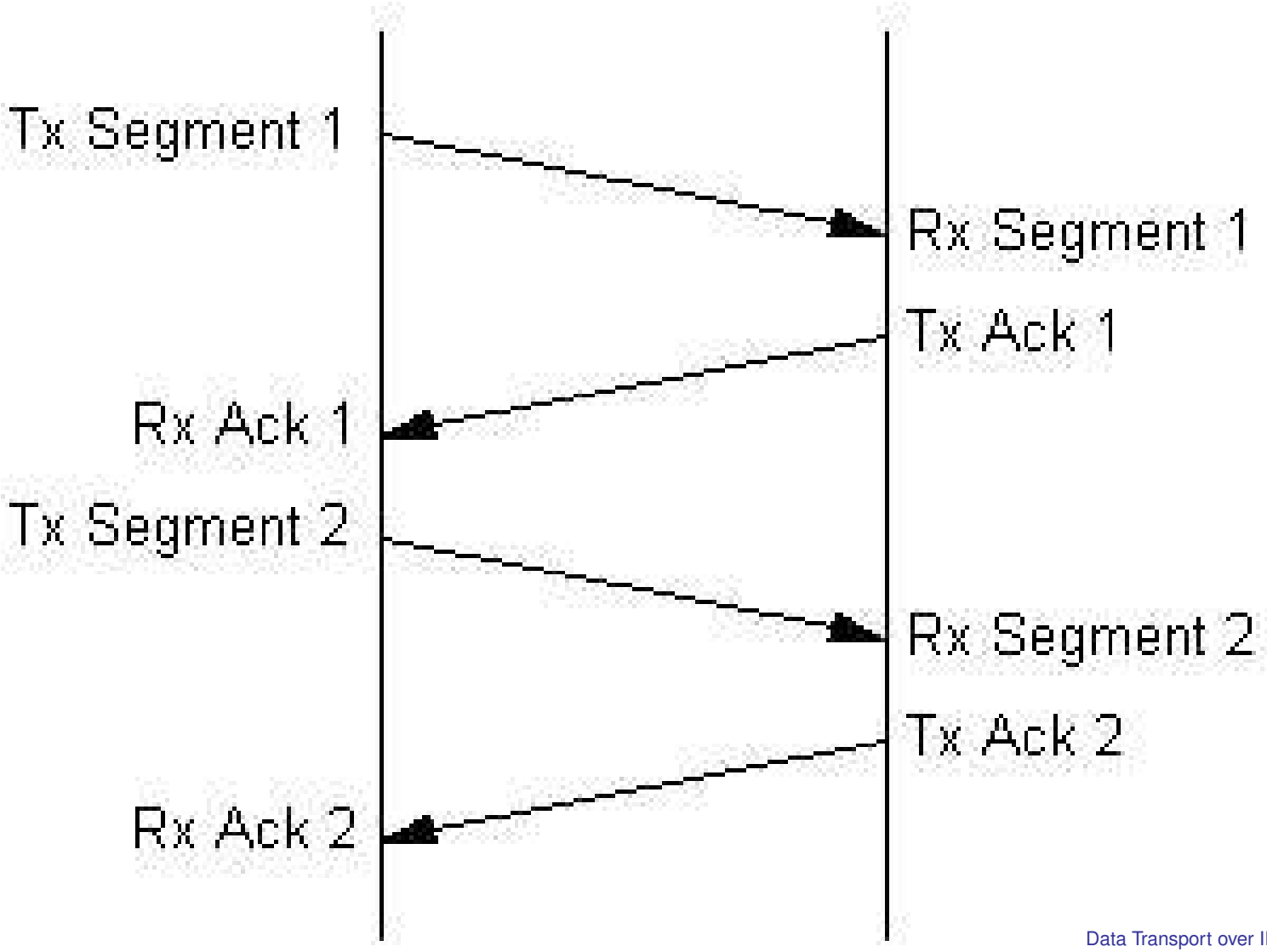
- The Transmission Control Protocol, or TCP, is designed to provide reliable stream transport over an unreliable network
- TCP provides the illusion of a “connection” across the network, between two applications
- It handles ordering, error recovery, and flow control



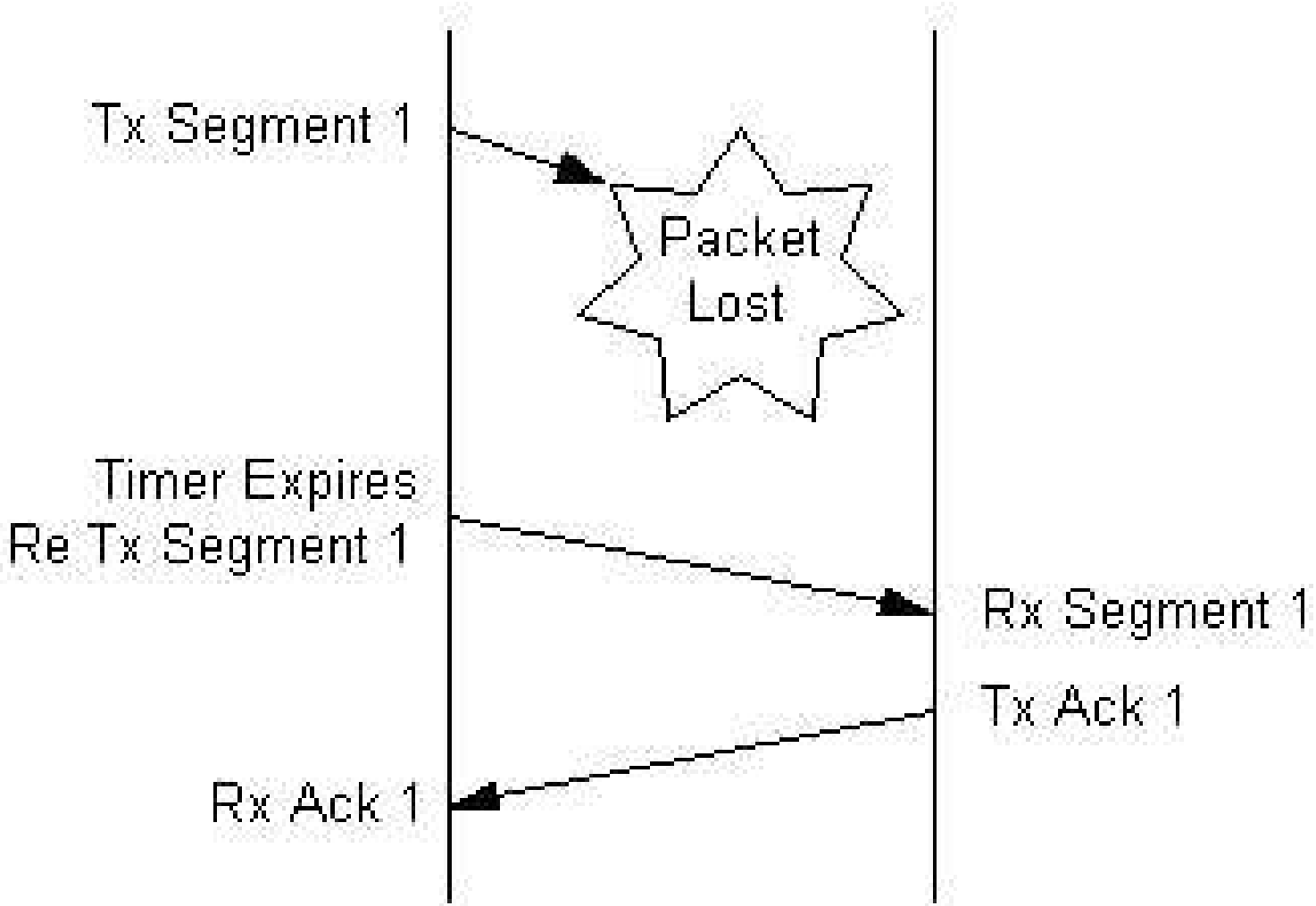
Reliability

- The concept of reliable transfer is based on a technique known as positive acknowledgement (ACK) with retransmission.
- The sender keeps track of each packet it sends, and waits for an ACK before ending the next packet
- The sender also has a timer, and will retransmit if the timer expires before receiving an ACK.

Reliable transfer



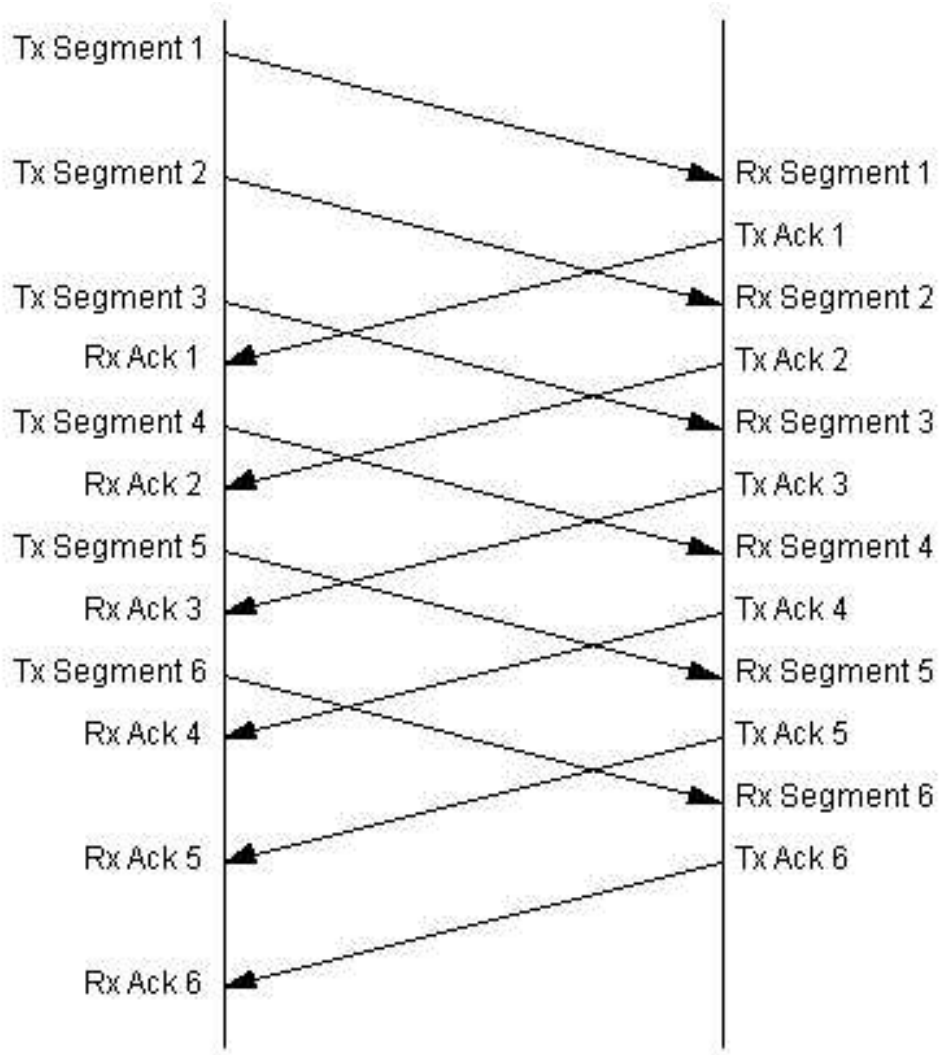
Retransmission



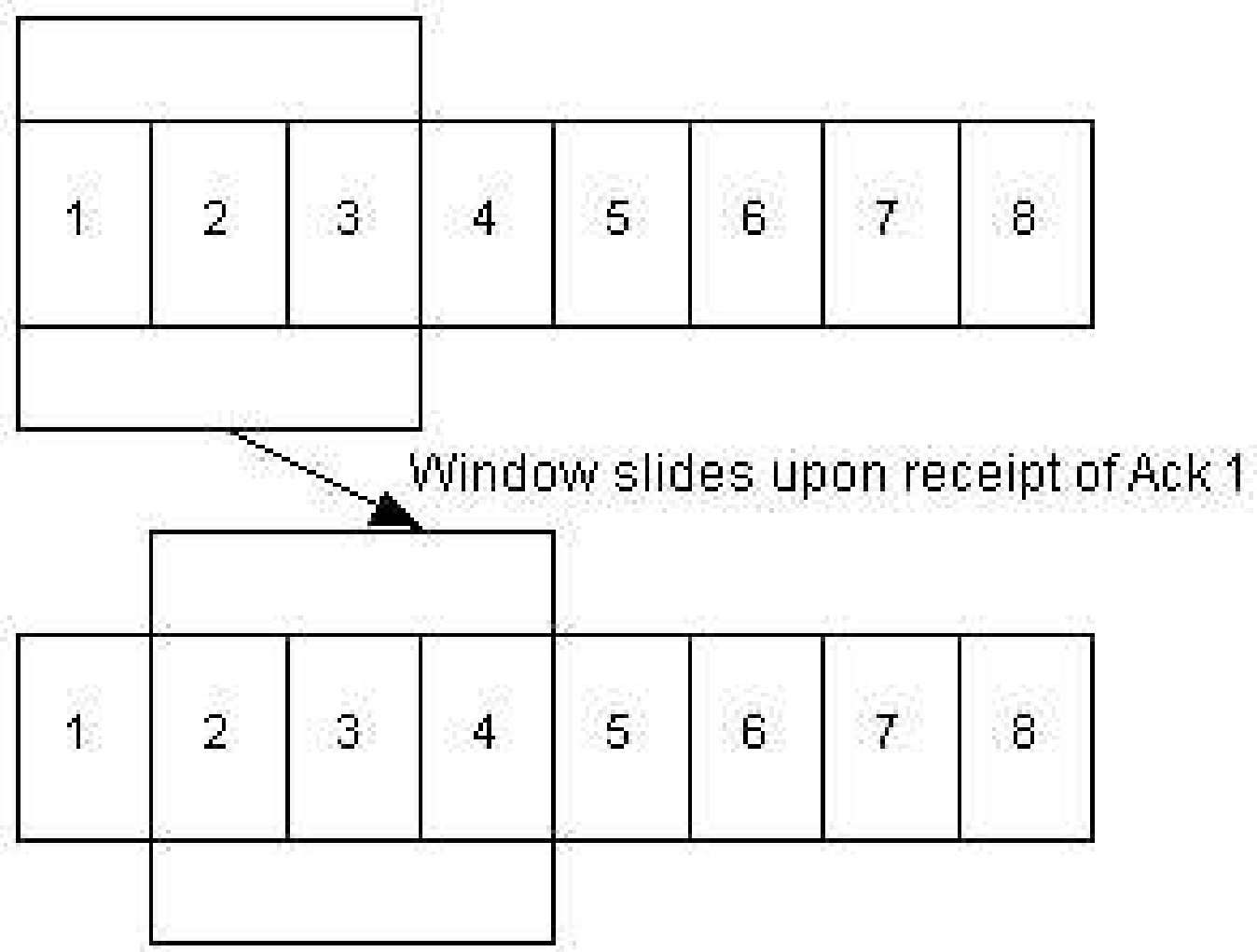
Sliding windows

- We just described a form of communication known as “stop and wait.”
- This is not very efficient, especially when there is high latency or packet loss
- With sliding windows, we no longer have to wait for an ACK on each packet before sending the next one
- Instead, we define a “window” of packets that are sent one after another, and deal with ACKs and retransmission later

Sliding window transmission



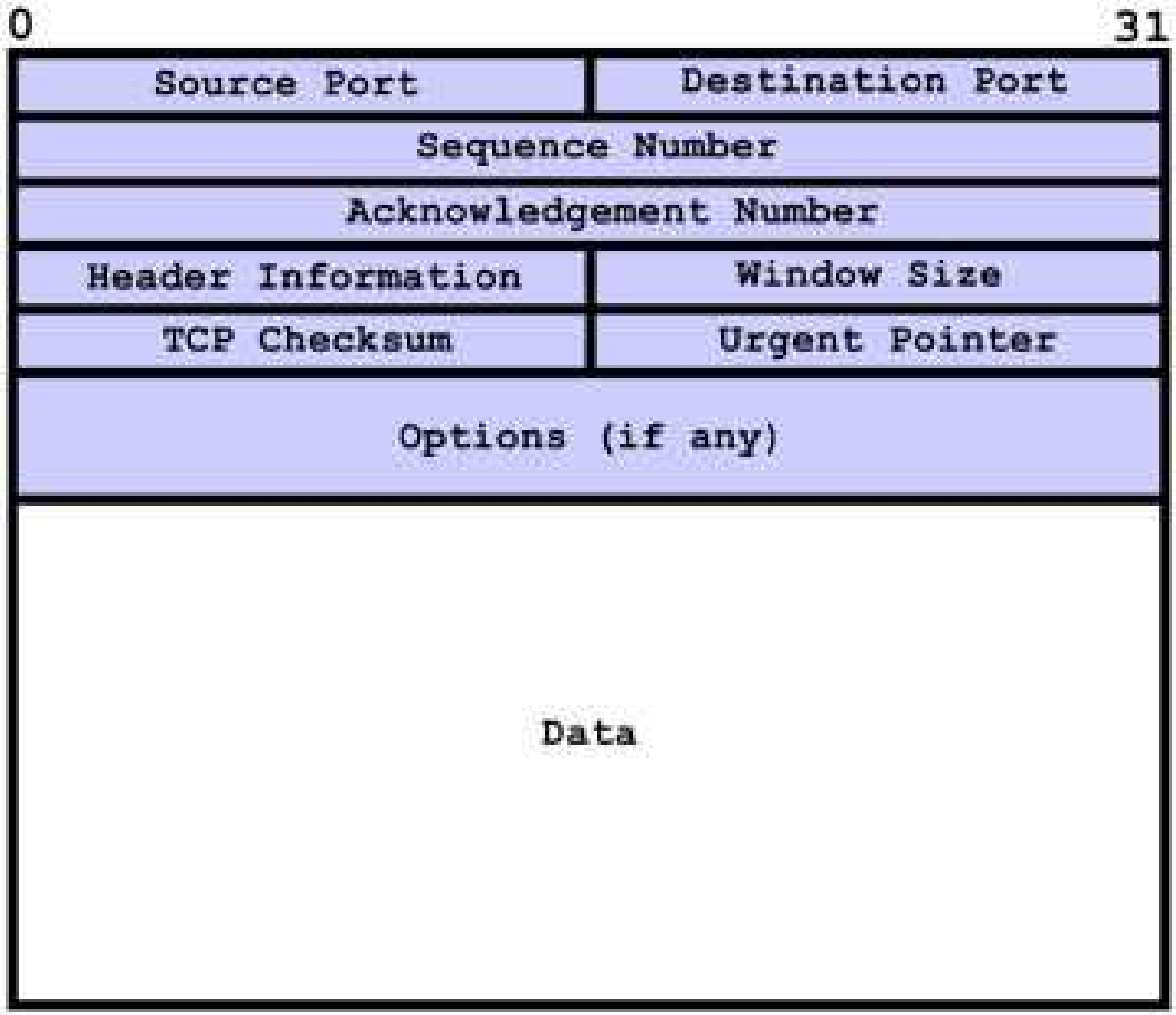
The window slides



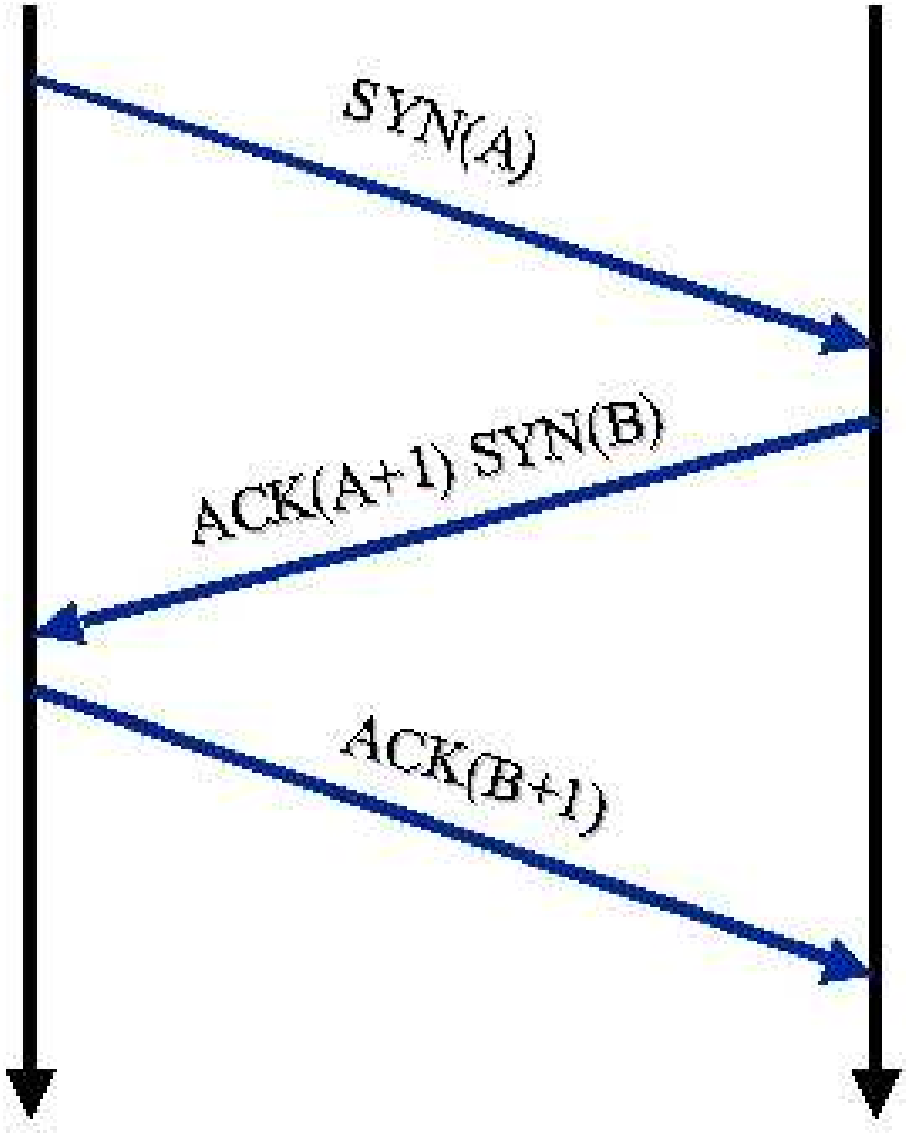
How does TCP work?

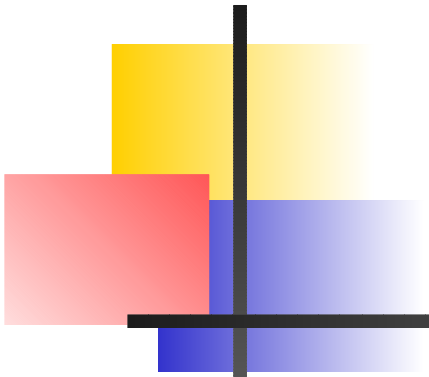
- TCP itself is a protocol, which specifies a means for reliable connection-oriented communications
- Unlike UDP, TCP uses the connection, not the port number, to identify a session
- Also, the unit of transfer in a TCP session is known as a “segment”
- TCP uses a form of sliding windows and segment sequencing

TCP Packet Header



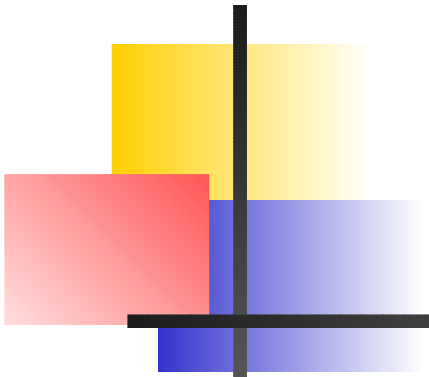
Opening a TCP connection





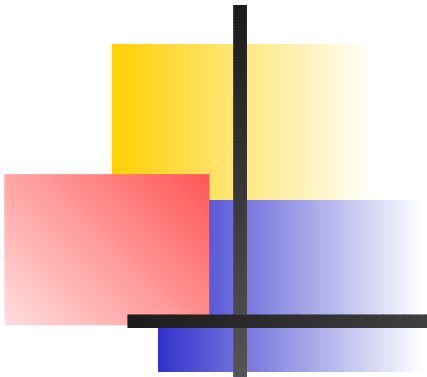
Flow control

- TCP uses variable sized sliding windows to manage data transfer
- This allows the receiver to control the rate of data flow, to avoid being overrun



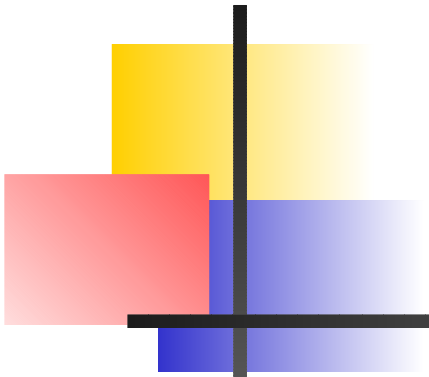
Final thoughts on TCP

- The Internet is not a very clean network, and latency can vary wildly from packet to packet
- TCP implementations are actually quite sophisticated, and have to deal with flow control, congestion control, timeouts, and retransmission
- These are handled with complex algorithms which are beyond the scope of this presentation



Conclusion

- We have seen an overview of how transport protocols function across IP networks
- Having a basic understanding of these concepts is very important, because it gives one a better understanding of how networks actually function



Questions?

Slides produced with Prosper and LaTeX

`http://prosper.sourceforge.net`