

*Transport Layer*  
*UDP & TCP*

Info 341 Networking and  
Distributed Applications

**INFO 461** Cooperative Software Design

want to work in the software industry as a **developer, tester, manager, or designer?**

you'll implement and deploy 2 versions of a web application in small teams, learning

software architecture	debugging
team coordination	testing
version control	verification
issue tracking	analytics

this is a great chance to put your web development skills into practice!

## Info 447 - Computer Supported Cooperative Work CSCW

INFO 447 Explores a range of topics to understand Collaboration and Collaborative Technologies

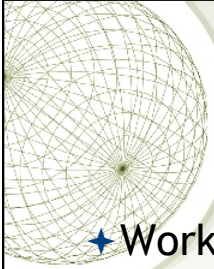
How do people collaborate using technology?

What is a collaborative technology?

Email, Word, Skype, Twitter, Facebook, Wikipedia?


How do we maintain awareness of the collaborative activity of others?

For Winter 2012 the course will explore CSCW as it happens in Wikipedia through readings, discussion, and concrete exploration of collaborative activity in Wikipedia. As part of the class students will participate in Wikipedia, and explore collaborative activity through logs and dumps of Wikipedia.

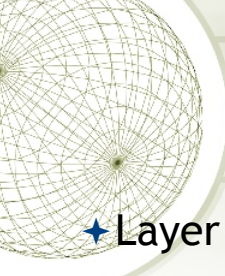


## Context

- ★ Working up the layers
- ★ IP is Layer 3 - Internetwork
  - ★ Addressing, fragmentation, reassembly, flow control, etc
- ★ Layer 4 - Transport

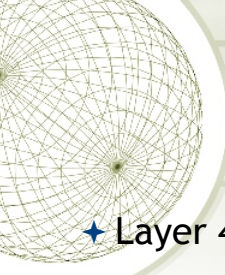


Internet Reference Model



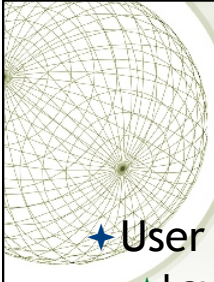
## Internetwork Layer

- ★ Layer 3
  - ★ Delivery is machine to machine
  - ★ Need a technique that allows end-to-end delivery
    - ★ Delivery from one application to another application




## Transport Layer

- ★ Layer 4 - Transport Layer
  - ★ Provide end-to-end delivery of packets
    - ★ Requires some added overhead to track source and destination applications
  - ★ Two transport layer protocols
    - ★ UDP - User Datagram Protocol
    - ★ TCP - Transmission Control Protocol



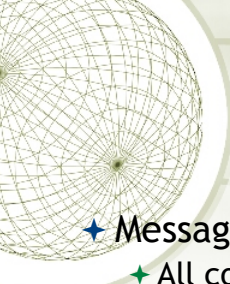
## *UDP Datagram Transport*

- ★ **User Datagram Protocol**
  - ✦ Layer 4 protocol in TCP/IP
  - ✦ End-to-end
  - ✦ Connectionless
  - ✦ Message-oriented
  - ✦ Best-effort
  - ✦ Arbitrary Interactions
  - ✦ Operating system independent



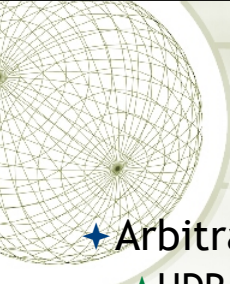
## *UDP*

- ★ **End-to-end**
  - ✦ UDP delivers from one application at the transport layer to another
- ★ **Connectionless**
  - ✦ There is no need to set up a connection
  - ✦ Simply send the data to the remote application
  - ✦ Very low overhead to send data



## UDP

- ★ **Message-oriented**
  - ★ All communication is by single discrete messages
    - ★ The contrast is a message “stream”
  - ★ Structure and semantics of message is set up by programmer and maintained
- ★ **Best-effort delivery**
  - ★ UDP will make the best-effort to deliver, but delivery is not guaranteed
  - ★ Can lose a whole message at a time!

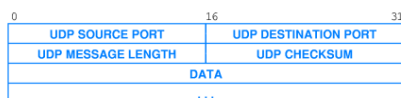


## UDP

- ★ **Arbitrary Interactions**
  - ★ UDP supports any type of machine interactions
    - ★ 1-1, 1-many, many-1, many-many

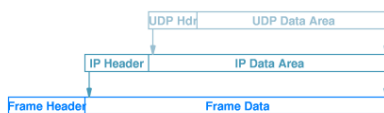
## UDP Datagram

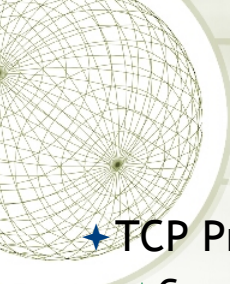
- ★ UDP Inserts simple header data
  - ◆ Source port, Destination port
  - ◆ Applications at the source and destination



## UDP Header Encapsulation

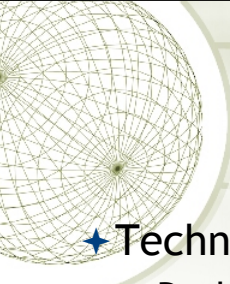
- ★ UDP header is inserted in addition to IP header and frame header
  - ◆ Encapsulation





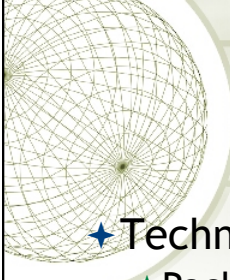
## *TCP Reliable Delivery*

- ★ TCP Properties
  - ★ Connection Oriented
  - ★ Point-to-point Communication
  - ★ Reliable Delivery
  - ★ Full-Duplex
  - ★ Stream
  - ★ Reliable connection setup and shutdown



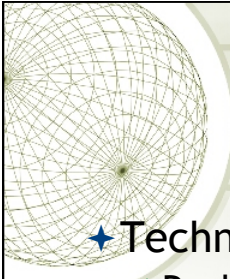
## *TCP Reliable Delivery*

- ★ Techniques to handle
  - ★ Packet loss, retransmission
  - ★ Adaptive Retransmission
  - ★ Flow control
  - ★ Connection Initiation



## TCP Reliable Delivery

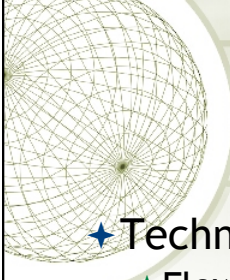
- ★ Techniques to handle
  - ✦ Packet loss, retransmission
    - ✦ How does this work?



## TCP Reliable Delivery

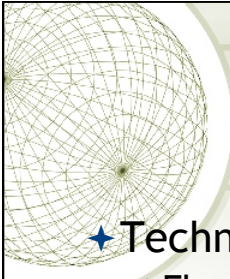
- ★ Techniques to handle
  - ✦ Packet loss, retransmission
    - ✦ How does this work?
  - ✦ Adaptive Retransmission
    - ✦ Does the same timer work for all connections?
      - ✦ What if the connection is on the same LAN
      - ✦ What if the connection is to a LAN far, far away?





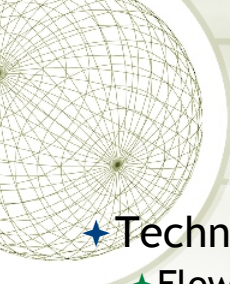
## TCP Reliable Delivery

- ★ Techniques to handle
  - ◆ Flow control
    - ◆ What was our example of flow control before?



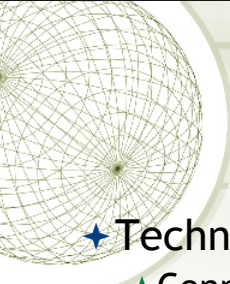
## TCP Reliable Delivery

- ★ Techniques to handle
  - ◆ Flow control
    - ◆ What was our example of flow control before?
      - ◆ Sliding Window buffer
    - ◆ Sliding window is “message” oriented
      - ◆ A chunk of data at once
    - ◆ What if we have a stream of octets?



## TCP Reliable Delivery

- ★ Techniques to handle
  - ✦ Flow control
    - ✦ What was our example of flow control before?
      - ✦ Sliding Window buffer
    - ✦ Sliding window is “message” oriented
      - ✦ A chunk of data at once
    - ✦ What if we have a stream of octets?
      - ✦ Different amounts of octets at different times
      - ✦ Use a “window” (buffer) of octets and let the sender know the size

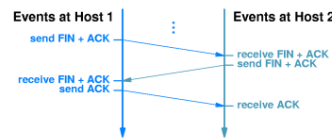


## TCP Reliable Delivery

- ★ Techniques to handle
  - ✦ Connection Initiation
    - ✦ 3-way handshake
      - ✦ A client sends a SYN to the server
      - ✦ The server replies with a SYN + ACK
      - ✦ The client sends an ACK back to the server

## TCP Reliable Delivery

- ★ Connection initiation
- ✦ 3-way handshake also used to close a connection



## SYN Attack

- ★ Connection initiation can be a source of problems ...
- ✦ 3-way handshake
  - ✦ A client sends a SYN to the server
  - ✦ The server replies with a SYN + ACK
  - ✦ The client never responds

## SYN Attack

- ✦ Connection initiation can be a source of problems ...
  - ✦ 3-way handshake
    - ✦ A client sends a SYN to the server
    - ✦ The server replies with a SYN + ACK
    - ✦ The client never responds
  - ✦ Server must set up some buffers and get ready for the communication
  - ✦ How long should the server wait and hold those buffers?
  - ✦ DDOS - Distributed Denial of Service Attack

## TCP Segment

- ✦ Segment - because the data is a stream of octets
- ✦ Data flows in two directions (full duplex)
  - ✦ Sequence Number for outgoing data
  - ✦ Acknowledgement Number and Window for incoming data

