User Datagram Protocol (UDP)

Srinidhi Varadarajan

UDP: The User Datagram Protocol

- UDP is another transport protocol in the TCP/IP suite
- UDP provides an unreliable datagram service
  - Packets may be lost or delivered out of order
  - Users exchange datagrams (not streams)
  - Connection-less
  - Not buffered -- UDP accepts data and transmits immediately (no buffering before transmission)
  - Full duplex -- concurrent transfers can take place in both directions

UDP Datagram Format

0 4 8 16 24 31

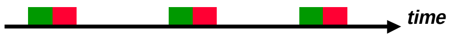
Source Port	Destination Port
Message Length	Checksum
Data	

UDP Header Fields

- *UDP Destination Port*: identifies destination process
- *UDP Source Port*: optional -- identifies source process for replies, or zero
- *Message Length*: length of datagram in bytes, including header and data
- *Checksum*: optional -- 16-bit checksum over header and data, or zero

UDP Versus TCP (1)

- Choice of UDP versus TCP is based on:
  - Functionality
  - Performance
- Performance
  - TCP's window-based flow control scheme leads to bursty bulk transfers (not rate based)
  - TCP's "slow start" algorithm can reduce throughput
  - TCP has extra overhead per segment
  - UDP can send small, inefficient datagrams

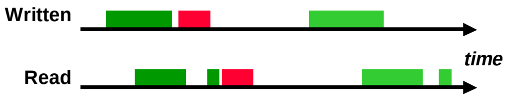


UDP Versus TCP (2)

- Reliability
  - TCP provides reliable, in-order transfers
  - UDP provides unreliable service -- application must accept or deal with
    - Packet loss due to overflows and errors
    - Out-of-order datagrams
- Multicast and broadcast
  - Supported only by UDP
  - TCP's error control scheme does not lend itself to reliable multicast
- Data size
  - UDP datagrams limited to IP MTU (64KB)

UDP Versus TCP (3)

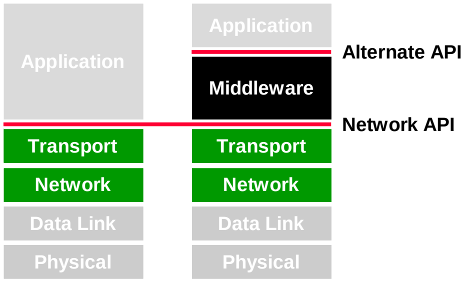
- Application complexity
  - Application-level *framing* can be difficult using TCP because of the Nagle algorithm
  - Nagle algorithm controls when TCP segments are sent to use IP datagrams efficiently
  - But, data may be received and read by applications in different units than how it was sent



UDP versus TCP (4)

- Which is used for ...
  - HyperText Transfer Protocol (HTTP)?
  - File Transfer Protocol (FTP)?
  - Telnet?
  - Post Office Protocol (POP)?
  - Remote WHO (rwho)?
  - MBONE audio/video?
  - Real Player?
  - Network File System (NFS)?

"Middleware" Model (1)



"Middleware" Model (2)

- Higher-level services can be provided for application development by so called "middleware"
  - May be built above the operating system (and run on various operating systems)
  - May be part of the operating system
- Possible functions: messaging, distributed object management, directory services,
  user-defined data types, composite data types, remote procedure calls (RPC), alternate communication abstractions

You should now be able to ...

- Distinguish between services, interfaces, and implementations related to protocols
- Identify the relationship between the TCP/IP protocol suite and the OSI Reference Model
- Identify the functions of the key protocols in the TCP/IP protocol suite
- Identify the differences between transport services provided by TCP and UDP
- Match application needs to services provided by TCP and UDP