Announcements

• Today:
  – In-class exercise on designing your own protocol
  – Open Q&A on Project 1
Protocol specifications

• Precise ways of communicating syntax and semantics of protocols to developers

• Postel’s Law:
  – “Be conservative in what you do, be liberal in what you accept from others”

• Meaning:
  – Be extremely careful when you implement your side of the protocol, but try to be accommodating when you interpret their side
Common specifications: RFCs

• Internet “Request for Comments”
• Defines protocols, protocol headers, and behavior of Internet-enabled software
• Often for binary protocols
### IPv6 Header Format

<table>
<thead>
<tr>
<th>Version</th>
<th>Traffic Class</th>
<th>Flow Label</th>
<th>Payload Length</th>
<th>Next Header</th>
<th>Hop Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Source Address**

**Destination Address**

**Version**
- 4-bit Internet Protocol version number = 6.

**Traffic Class**
- 8-bit traffic class field. See section 7.

**Flow Label**
- 20-bit flow label. See section 6.

**Payload Length**
- 16-bit unsigned integer. Length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets. (Note that any
TCP

TCP Header Format

0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Source Port | Destination Port |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sequence Number |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Acknowledgment Number |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Data | U | A | P | R | S | F |
| Offset | Reserved | R | C | S | | Y | I |
| | | | G | K | H | T | N | N |
| | | | | | | | | | Window |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Checksum |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Urgent Pointer |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Options |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Padding |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| data |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Today’s exercise

• Design
  – Break into groups of 3 to 5
  – Define the “wire” format for a video game protocol as in the above IPv6 and TCP examples
  – Put into a document

• Implement sketch
  – Write just the code that reads/writes messages into classes or C structs

• Critique
  – We’ll put up a few designs and discuss
For binary protocols

NAME

fread, fwrite - binary stream input/output

SYNOPSIS

#include <stdio.h>

size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);

size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);
Design decisions

• Framing:
  – Fixed size?
  – Length + field?
    • E.g., (5,"hello")
  – Delimiter-based?

• Encoding:
  – Binary?
  – Text?
Video game server example

• Create_User
  – Client->Server:
    • [Username, Full name, avatar (256x256 image)]
  – Server->Client: [Unique ID]

• Get_HighScore
  – Client->Server: [N] (N = #scores requested)
  – Server->Client:
    • [(userID_1, score1), (userID_2, score2), ...]