CSC / ECE 573 Internet Protocols, Fall 2005

Project Part I

Due Date

- Wednesday, September 28, at 11:45 PM

Instructions

- The project can be worked on individually, or in groups of 2.
- Put your name (both names if joint with another person), "Project, Part I", and date at the top of the first page.
- Do not plagiarize; that means, do not copy content from any source without permission from the instructor, and if permitted, acknowledge the source.
- Submit this part to both persons' lockers if doing jointly. They will be identical in this case, and have the name of both persons at the top of page 1.

Description

In the project you will implement and extend a standard Internet protocol. By doing so, you will learn to read and understand one standard protocol specification in detail, to analyze it, and to implement and test it. You cannot do this for all protocols we look at in this course, but you can at least do it for one.

The requirements and instructions for this first part of the project are as follows. There is no required length for this assignment (Part I), but at a standard font size and spacing, two pages should be adequate.

1. *Choose which protocol you will implement*
   The choices are listed below. I have attempted to weed out choices that would be significantly easier or harder than the "average" difficulty of the protocols listed below.
2. *Choose a partner, if you wish to work on this in a group of 2*
   If you don't already have a partner, I suggest you use the message board to locate one. It's hard to know if they will be compatible with you, but at least find out if their experience, expectations, schedule, etc. are suitable for working together. I will ask partners to evaluate how much each contributed to the project, so don't look for someone who will give you a "free ride".
3. *Identify what sources of information you will use*
   This includes the RFC, and additional websites, books (including our textbook), or documentation you think will be helpful.
4. *Identify what platform and programming language you will use*
   The only hardware choice is the PC (sorry, Macintosh fans).
   The OS choices are Windows or Linux.
   The programming language choices are C, C++, or Java.
5. *Describe how you will test your program*
   Identify how many machines you will need, what level of access will be required, and what things you will measure
6. *Identify what functions you will *not* implement*
   Peruse the RFC(s) and identify functionality that may no longer be in use (some of these RFCs are 25-30 years old, after all), and/or that you think will be too hard or of limited value (e.g., some protocols may have 100 or more message codes - you don't really need to do every single one). If you trivialize the assignment, we'll have to do some negotiation.
7. *List several candidate new functions (extensions) that you believe are worthwhile to add to the protocol specification*
   I would like you to think about what is *not* in the protocol, that would be convenient or useful. This project requirement will exercise your ability to design a small-scale protocol function, and to specify it in the form of an Internet Draft or RFC. The list you provide here could be of varying levels of difficulty, but don't suggest something truly trivial, or breathtakingly grandiose (on the other end of the scale). By listing possibilities, you are

not committing to any of them at this time; that will come later. I may only ask you to specify (but not implement) an extension.

Choices for the project are the following:

- Implement TCP at the application layer. You will write an application that sends data, such as a file transfer, across the Internet using UDP. I will implement or use an existing network emulator that drops, reorders, duplicates, corrupts, and delays segments. Your application level code should do the things that TCP does: duplicate elimination, payload checksum, acknowledgment and retransmission, and flow and congestion control.
- Implement the routing protocol RIPv2 at the the application layer. You will not have access to actual routers. Rather, you will run multiple instances of the router protocol on a single machine. Each instance emulates a router, and will read a separate configuration file that represents the interfaces that router has. The instances will then contact each other to exchange routing information (distance vectors), and update their forwarding tables, in the specified way.
- Implement a new protocol described in a recent (last 2-3 years) research paper from either Infocom or SIGCOMM. You will need to also write an RFC-style specification for your protocol.
- Implement a new version of TCP, also at the application layer, but suitable for use in interplanetary communication. Expect loss rates, bandwidths, and latencies that are representative of conditions in space.
- Implement a DNS server at the application layer, along with an application program that sends DNS queries to the server. The server should be able to load a zone from a master file, and interact with other DNS servers. The server need not support Inverse Queries, which are not widely used.
- Implement IPSec Authentication Header functionality. Key exchange can be done manually (no need to implement IKE).
- Implement basic VoIP functionality, with SIP invites and replies, RTP headers, RTCP feedback (it is allowed to plan on only one receiver, and not design for multicast applications), an elastic buffer at the receiver, and playback (of audio). No compression / decompression is required.
- If you have another Internet protocol you really, really want to implement, talk with me to see if it meets the requirements. 8-)

Created on September 14, 2005
Last Modified September 23, 2005
Maintained by Douglas S. Reeves