

Network Programming - Spring 2003

Test #2

My Name is: _____

Instructions: *You need to do any one of the three problems (only one!).* For each problem you are given a *functional* description of a network service. Your job is to *design* the network service including such decisions as whether the service should be peer-to-peer or client/server, what transport layer protocol(s) should be used, the application-level protocol, and the design of client and server programs. Keep in mind that you need to convince us that you understand the issues, and that given enough time you could come up with a feasible, efficient, scalable design and implementation. Make sure you think the entire service/system through (that what you propose is feasible, complete, and makes sense).

Grading: Your grade depends on the following (somewhat vague!) formula:

- You get points for:
 - Convincing us that you understand the service and the critical issues related to designing a usable implementation.
 - Discussion or list of the pros/cons of various approaches/alternatives.
 - Meaningful comparisons to existing services/systems that you are an expert on (case studies, projects).
 - You get points for spelling TCP correctly.
- You lose points for:
 1. Anything you state that is wrong.
 2. Statements that don't help in any way to illustrate anything relevant to the problem or related to this course. For example: "I'd design a 3-D graphics system that could spin the corporate logo on the client desktop!". We will interpret this kind of stuff as "filler", and we assume that the presence of "filler" is a sign of weakness...

Network Services: The services listed below are intentionally *general*. Make (and state in writing) any assumptions you want to make that don't change the service significantly. Feel free to stray a bit from any particular service, but make sure you are not oversimplifying the service!

- **Spam Detection/Filtering Service** – you set up a service on The Internet that will provide detection and filtering of junk email. There are many possible approaches, consider only those that can work with existing email clients/servers (don't propose a completely new email system for the Internet – just a system that current users can subscribe to that will detect and filter out SPAM). The system must include some mechanism where users can provide filtering information (for example, they could tell the system to assume anything that has the keyword "brittney" in the subject line is junk that should be filtered). Focus on the network service, not on "how to detect spam by looking at email content or headers".
- **Workflow Routing** - Office automation extended to support virtual offices anywhere on the Internet. Workflow routing involves (as a simple example) the automated forwarding of various documents to the appropriate people. You fill out a form requesting the purchase of a new stapler. The form is automatically *sent* to someone who gets a quote on the stapler. Once this is done the form is routed to your manager for approval, then on to his manager, then to the purchasing department and so on. This service involves the actual transfer of documents, but also requires some general mechanism for creating rules on where documents should go and when. There are lots of business processes that can be automated this way (be creative!).
- **Distributed Packet Sniffer** – A *sniffer* is a program that looks at every frame sent on a wire and can record the actual data (the frame), or can look for specific kinds of frames (could look for only frames holding TCP segments, or could look for only those TCP segments that are part of an HTTP conversation, etc). A sniffer can only work with the frames it can see (typically at all the frames *on a wire*). This network application involves the creation of a distributed sniffer that can operate over multiple wires, and can be used for applications like network management, fault detection and network intrusion detection. Supporting switched networks is a special case – you don't need to consider this (but you can).