

Project

*Assigned: March 27, 2009**Due: April 24, 2008*

In this project, you will have an opportunity to explore a specific issue in depth. You can implement a method from the textbook, or from a paper in our reading list. You are also encouraged to find papers from other sources and use them as a basis for the project. The topic is open, but must be somehow related to what we discussed in the class. You can either do it by yourself, or form a group of two. The results of a group project should reflect an appropriate amount of efforts.

There are three deliverables for the project.

- *A progress report.* It will be due on April 17. You should email me a one-page progress report detailing what you have done and what you plan to do.
- *A final technical report.* It should describe the problem, the method, and the performance results (with a total of 3-8 pages). Implementation of an existing method with some significant testing data is the minimal requirement. You are encouraged to either make some improvement over an existing method, or perform analysis of and provide some insights into the results.
- *A working system.* I will try to arrange a time for you to demonstrate your system. You should submit your codes by the due date. One of the possible environments you can use is the UK Emulab (<http://uky.emulab.net/>), which can allocate multiple machines for you to run your system. You can get an Emulab account for this project. Please talk to the instructor about the details.

Here are some ideas of possible systems you may implement.

- **Evaluation of Control Overhead of Servers.** You can implement a server using multi-process, multi-thread, and `select()`. The key part of this project is to create scenario (client requests) to show the difference in performances of these methods for transferring control in the server.
- **Fast IP Look-up Based on Binary Search.** Implement the fast IP look-up algorithm based on binary search on prefix length. You can use the sample data from <http://www.cs.purdue.edu/homes/kompella/teaching/sp08/cs636/p2/> for testing. (Also assume that the input data will be in this format.) Make sure that your program will work for other sets of data.
- **Optimization of Variable-stride Tries.** Implement the optimization algorithm for the variable stride tries for prefix lookup. You can refer to Srinivasan and Verghese's 1999 paper published in *ACM Transactions on Computer Systems* for details.
- **Packet Classification using Equivalenced Cross-Producting.** Refer to Gupta and McKeown's paper in SIGCOMM 1999 and implement a packet classification algorithm based on equivalence cross-producting. You need to get some traces (by `tcpdump` and/or find online) and sample rules to test your algorithm.
- **Packet Scheduling Algorithms.** You can implement *Deficit Round Robin* and *Virtual Clock* algorithms for packet scheduling. Generate traffic to compare and analyze how they schedule the packets compared with the GPS model.
- And remember that you can always define your own project.