Comp 8005 Computer Systems Technology February 2010.

Data Communication Applications

Final Project

Due: April 5, 1730 hrs.

Objective:

- To design and develop a network application that uses advanced TCP/IP programming techniques.
- To design and implement a minimum-functionality "Port Forwarder" using C/C++ for the Linux platform.
- A maximum-functionality port forwarder could include a proxy server, caching, etc.

Description:

Port forwarding is a mechanism that facilitates the forwarding of network traffic (connections) from one machine to another. The most common use of this technique is to allow an access to an externally available service (such as a web server), which is running on a server in a private LAN. In this way, remote client systems are able to connect to servers offering specific services within a private LAN, depending on the port that is used to connect to the service.

Your Mission:

Design and implement a port forwarding server that will forward incoming connection requests to specific ports/services from any IP address, to any user-specified IP address and port. For example, an inbound connection from 192.168.1.5 to port 80 may be forwarded to 192.168.1.25, port 80, or to 192.168.1.25, port 8005.

Constraints

- Your forwarder must provide the following **minimum** functionality:

- Forward any **IP: port** pair to any other user-specified **IP: port** pair.
- The application must support multiple inbound connection requests, as well as simultaneous two-way traffic.
- **OOB** traffic will be supported by your application.
- Only TCP connections will be forwarded by the basic implementation.
- You are required to provide a detailed test case that will document the complete functionality of the port forwarding application. For example, beyond the basic functionality tests you may want to test and see how well your application performs under a heavy load, i.e., heavy throughput from multiple clients.
- Your application will read the IP: port combinations to forward to from a separate configuration file.
- As always additional work and features will be considered for **bonus marks**.

To be Submitted:

1. Detailed design work (State transition diagrams or equivalent, and pseudocode) for the implementation.
2. A brief user manual that will allow a user to understand and use your application.
3. A detailed test plan and document showing the results of all your test cases.
4. A disk containing the source code listings, all documentation, and an executable version of your application.

- You are also required to demonstrate your working programs during the lab the day the assignment is due.

- Ensure that you clearly explain testing procedures for your programs and provide test programs if necessary.