Project Plan

Distributed Network Traffic Controller

Revision Number: 1.5
Last date of revision: 2/2/05

22c:198

Johnson, Chadwick Hugh

Change Record

Revision	Date	Changes
1.1	1-28-05	ER Added, small items added to project functionality
1.2	1-28-05	Better defined problem, “fair share” and throttling
1.3	1-31-05	Added tomcat and apache to resources, adjusted ER diagram
1.4	2-1-05	Change ER to store gateway-user relationship, and administrative ability to edit Users, and combined permissions and administrator. Moved ER diagram to High Level Design Document. (It really belongs there instead)
1.5	2-2-05	Removed VMWare and VirtualPC as software resources and removed Perl as a language.

Project Scope

Any Internet connection that allow multiple people to connect their personal computers run the risk of over saturation, often as a result of peer-to-peer file sharing programs like bittorrent. This over saturation of a connection can cause all Internet traffic to slow, and in some cases, come to almost a standstill. Options such as blocking ports for peer-to-peer file sharing programs are ineffective in many cases because of the large number of peer-to-peer options and some even allow for connections through common ports like 80. Another lacking with this way of handling the problem is that everyone is treated the same way, irregardless of whether they have abused their privilege or not. Currently there are pieces of hardware and software that allow for traffic histories and set limits on total amount of traffic they are allowed to create, but these are often expensive.

I propose a free and open-source distributed system that makes use of a "fair share". A fair share would be a set amount of traffic created over a given period of time. This amount would be administratively set, and should have the ability to be changed due to varying views on what is fair. This fair share could then be used to determine if a user should be placed in a "slow pipe" or an "open pipe". The slow pipe is some fraction of the total possible speed, and is a throttled version of the open pipe. Should the slow pipe not be fully utilized, or the open pipe has a greater demand, additional bandwidth is automatically shifted to the open pipe. Everyone in the slow pipe will share a specified bandwidth with everyone else in the pipe.

This system will accept requests for an IP address, and will maintain the association between that MAC address and the newly assigned IP for an administratively specified time, such as a semester. All internet traffic will automatically be forwarded, so there will be no pause between the request for an IP and when it will be open for use. These gateways will be capable of monitoring and storing of traffic based on IP address, and will be distributed to help remove the bottleneck of a single gateway. This setup will also make it easier for different subnets to have different slow-pipe settings.

The users have the ability to:

- view their history
- view a graphical representation of their average usage. This will make it so they can understand their placement in their current pipe. This page is automatically generated by pointing a browser at any gateway IP address.

The administrators have the ability to:

- view and edit the usage of any user
- change the definition of a fair share
- change the size of the small pipe per gateway
- specify traffic capture refresh time or rate
- specify number of instances the average usage is based on
- specify if an IP is throttled
- change their login password
- create administrators
- specify what those new administrators are able to do
- specify if new users are initially placed on a slow or open pipe
- specify when all collected user data should be purged, such as in a new semester

Due on time constraints, possible additional administrator functionality may include:

- block ports
- give weights to common ports and addresses
- add or remove gateways from the system.

Since this project will follow Software Engineering principles, there will be many work products delivered along with the final application. These work products include:

1. Project Plan – Goes into detail about what the project is supposed to accomplish, along with an ER diagram for the database.
2. High Level Design – Describes how the things in the project plan work, and includes the DDL for the Database.
3. Test Plans – A description of the various techniques used to test the final project.
4. Distributed Network Traffic Controller + instructions on use – The actual program written. Chances are this will actually be several individual programs that all work together to
5. Test Results – Results collected from the test plans which include bugs found and performance results.
6. *Instructions on how to create project from the start and all code.
7. Short Report containing assessments of data collected
8. Concluding overview of whole project

Project Schedule

- Week 1+2 Project Plan, High Level Design, Test Plans
- Week 3 Get together hardware and set up systems
- Week 4 DHCP + IP forwarding
- Week 5+6 Database setup + Throttling
- Week 7+8 Administrative Screens
- Week 9+10 Setup and integration of second gateways
- Week 11 User screens
- Week 12+13 Attempt optional components if time allows
- Week 14 Testing
- Week 15 Assessment of data collected and overview of whole project.

Required Resources

- **Operating Systems**
  - **Fedora 2 Core Final**
  - **Windows XP Professional**
- **Software**
  - **Tomcat**
  - **Apache**
  - **CBQ**
  - **IPTables / IPForwarding / IPMasquerading**
  - **DHCP**
  - **Postgres SQL Database**
  - **Jude Community 1.4.3**
- **Computers - Minimum**
  - **Two computers for Gateways**
  - **Two computers for Databases**
  - **One computer for a web server**
  - **One or two computers for testing**

- **Hardware**
  - **Network Cables**
  - **Two Switches**
- **Languages (Possible)**
  - **HTML**
  - **JSP**
  - **Java**
  - **Python**

Feasibility

Time constraints are the major limitation for this project. This is a large project with several individual modules. To ensure everything is done properly, testing will be done at every stage of development. This project is very feasible and can be completed under current known constraints.