High Level Design

Distributed Network Traffic Controller

Revision Number: 1.0
Last date of revision: 2/2/05

22c:198

Johnson, Chadwick Hugh

Change Record

Revision	Date	Author	Changes

Contents

1. Introduction

1.1. Why this High Level Design Document?

The purpose of this High Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

1.2. Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3. Definitions

- DNTC – Distributed Network Traffic Controller
- Fair Share – An administratively set data rate per time frame that is considered fair.
- Throttling – A reduction in maximum transfer rate of data.
- Open Pipe – A connection to the Internet with no throttling.
- Slow Pipe – A throttled version of an open pipe where all users of that gateway share that reduced pipe's rate.
- Firewall –Functionality that can allow or block certain ports and addresses.
- IPTables – A firewall built into the Linux kernel.
- IPForwarding / IPMasquerading – The ability to forward traffic.
- Postgres SQL Server – A database management system.
- Python – A possible programming language to interface between IPTables and Postgres.
- JDBC – A possible Java-based interface between IPTables and the Database.
- JSP – The language that will be used for displaying user history and administrative functionality.
- Tomcat – a free, open-source implementation of Java Servlet and JavaServer Pages technologies developed under the Jakarta project at the Apache Software Foundation.
- Apache - An open source Web server.
- ER – Entity Relation Diagram
- CBQ –Class-Based Queuing. Limits bandwidth at the IP/port level.
- Kernel – Core of an operating system, a kernel manages the machine's hardware resources (including the processor and the memory), and provides and controls the way any other software component can access these resources.
- DHCP – (Dynamic Host Configuration Protocol) – This is a protocol that lets network administrators centrally manage and automate the assignment of IP Addresses on the corporate network.
- Gateway –Bridges the gap between the internet and a local network.

1.4. Overview

The HLD will:

- present all of the design aspects and define them in detail
- describe the user interface being implemented
- describe the hardware and software interfaces
- describe the performance requirements
- include design features and the architecture of the project
- list and describe the non-functional attributes like:
    - security
    - reliability
    - maintainability
    - portability
    - reusability
    - application compatibility
    - resource utilization
    - serviceability

2. General Description

2.1. Product Perspective

The Distributed Network Traffic Controller will be comprised of several different components. Some of these components will be programmed, while others will be implementations of open-source programs. The language implemented will be dictated by it's purpose. The administrative and user interfaces will be using JSP to display the pages, and SQL to retrieve, insert, delete, and update the database. Either Python or JDBC will be used to submit SQL commands for the automated part of the project such as updating the user history and DHCP data. This setup will allow for multiple users to login and interact with the program at the same time. It will also be set up using two user levels. First is the basic user, which can only view their current average and their history. This page is automatically displayed based on their IP address. The second type of user is the Administrator. They have the ability to change information in the database such as settings and user history. This user level can only be attained by logging into the system.

2.2. Tools used

1. Jude, a Java based UML design program, is used to generate all of the diagrams used in analysis and design phases of the project.
2. The project will have a relational database backend that is SQL based. The actual software used is PostgreSQL.
3. Interfacing with the database to display information on the user's web browser will be done using JSP. It can connect to the database and parse it into viewable HTML code.
4. Tomcat compiles JSP pages into servlets to be displayed through Apache..
5. Apache - An open source web server that will display requested pages.

6. Automated interfacing with the database behind the scenes will be either Python or JDBC.
7. DHCP assigns IP addresses.
8. CBQ is a throttling mechanism based on classes.
9. After a Kernel recompile, IPTables allows for IPMasquerading. IPTables also acts as a built in firewall.
10. RedHat Fedora Core 2 is the development platform.

2.3. General Constraints

The Distributed Network Traffic Controller must be user friendly and as automated as possible. Administrators should not be required to do anything besides the initial setup, and users should not be required to know any of the workings. Without logging in, the user will only have the ability to view that IP's current average and history. After logging in, that user then has the ability to change settings and user histories.

2.4. Assumptions

This project is based on the idea of a Fair Share, and the goal is to make this idea a reality using Software Engineering practices. In doing so, many documents are created, and it is assumed that design flaws will be found early on. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting. Another assumption is that the current intended documentation will suffice to make this project count towards the Software Engineering Subtract. There is also an assumption that none of the work or hardware will be stolen or sabotaged. The final assumption is that a demonstration and presentation will be possible at the end of the semester.

2.5. Special Design aspects

One special design aspect is the understanding that without significant modification, this system will only work on IP based traffic. This means TCP and UDP will be counted, but it is not known how any other type of traffic will be handled by this system.

3. Design Details

3.1. Main Design Features

The main design features include five major parts: the architecture, the user interface design, external interface, the database, process relation, and automation. In order to make these designs easier to understand, the design has been illustrated in attached diagrams (ER, Use Case, and Screen Shots).

Screen Shot Breakdown:

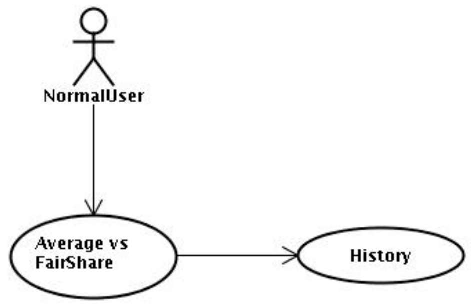
User

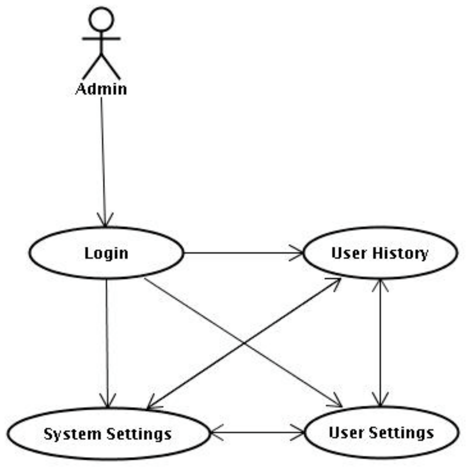
- indexGood.html – What a user would see if they were under their fair share.
- indexBad.html – What a user would see if they were over their fair share.
- indexUnlimited.html – What a user would see if they were set never to be throttled.
- history.html – What a user would see when they clicked on "View Your History" on their index.html page.

Administrator

- login.html – The administrative login screen.
- administrativeScreen-BrowseIPs.html – After clicking on "Browse all IPs" or typing in an IP address into the search, lists of IP's and their information displayed.
- administrativeScreen-UserHistory.html – After clicking on an IP address, this allows the administrator to change the history of an IP address.
- administrativeScreen-SystemSettings.html – After clicking on "System Settings", allows the administrator to specify settings for each of the gateways individually.

3.2. Application Architecture





3.3. Technology Architecture

3.3.1. Web Application Architecture

The front end of the program is a web application. Functionality will vary based user privileges if a user is logged in. Normal users are not required to log in, and can view their personal average and history. Administrators will have access administrative abilities based on permissions given to them.

3.3.2. Presentation Layer

Information will include the IP address, all information pertaining to that IP address. Administrative screens will have access to information pertaining to all IP addresses, and the ability to change histories and system settings.

3.3.3. Data Access Layer

The database will be accessible to all users, administrators, and automated services. A login will determine what parts of the database can be accessed and changed.

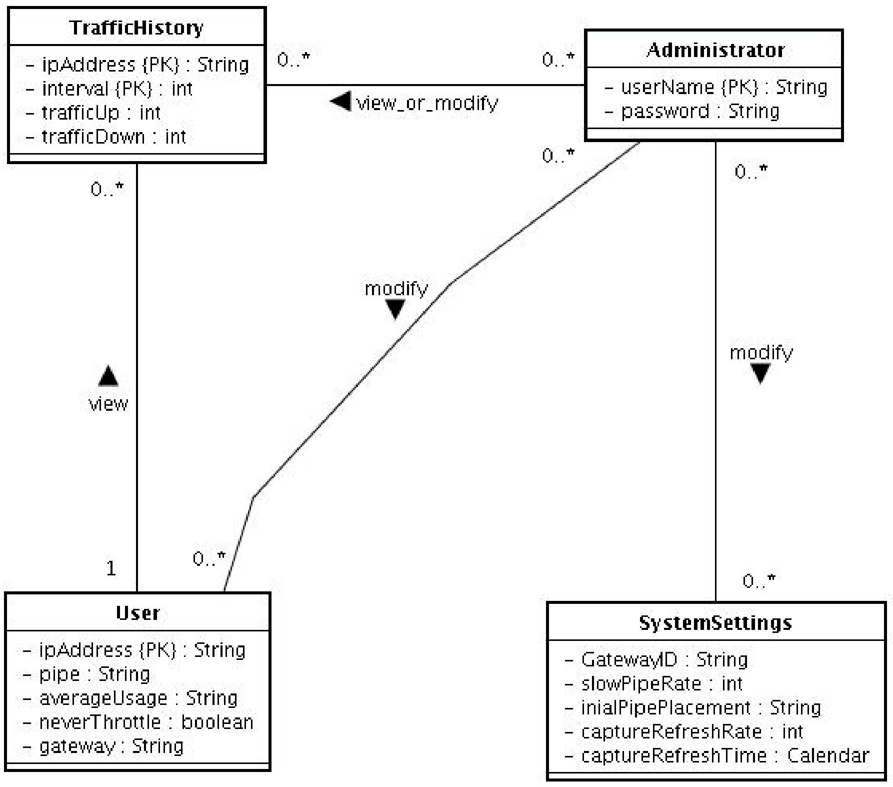
3.3.4. Tools Used

See section 2.2 for tools used in the design of this project.

3.4. Standards

Database – relational
Inputs – entered through text field and stored in database.
Security – username and password are required for access to the system.
Quality – by keeping the interface simple and direct, quality should be kept at a maximum.

3.5. Database design



3.6. Files

This product will not use a large number of files. DHCP uses a file to maintain IP to MAC association. Tomcat uses JSP pages. A file will be used to store all of the usernames and passwords and all other attributes specified for those users. This file will be accessed at login. It can be modified by the administrator at any time. Another file will store the database. This file will be accessed and modified by all users with proper permission.

3.7. User Interface

The user interface is a very simple plain layout with little to no graphics. It will display information very clearly for the user and will primarily output information to the user through HTML pages. Administrative screens are use mainly for input through text fields in HTML pages. Screen shots have been provided to demonstrate the user and administrative interface.

3.8. Reports

The reports will display the user’s average usage up to the last time the system calculated it and their history. Reports can also be created by an administrator of any user’s traffic.

3.9. Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong. An error will be defined as anything that falls outside the normal and intended usage.

3.10. Interfaces

There are four main interfaces for this project. First, the traffic interface, which consists of the IP assignment and all traffic that goes through the gateways. Second is the interface for the automated services which are sent from the gateway to the database. The third interface is the user and administrative interface Tomcat. The final interface will be between the Tomcat and the database.

3.11. Help

Help will come in the form of all the documentation created prior to coding, which explain the intended uses. Should time allow, detailed instructions will be written on how to create and implement the system with the intentions of publishing as an Open Source solution.

3.12. Performance

Performance is going to be very important for this project. For everything to run smoothly for this project, the gateways will have to be able to update data on the database and refresh the IPTables before it is supposed to do so again. This is likely to be the most processor intensive aspect of the project. The gateways will also need to supply requested pages to the users at a reasonable speed. The database server will need to keep up with all database requests and transactions.

3.13. Security

Because security is not the prime focus of this project, only the minimal aspects of security will be implemented. A username and password will be required to log into an administrative interface and database. For now, all data will be sent in plain text. Verification of user to IP or MAC address is also outside the scope of this project. For now, there will also be no log of failed attempts of an administrator logging in.

3.14. Reliability

A redundant database server will be implemented so that if the main database server stops responding, the gateways will automatically start using the other server. The mechanism used for syncing these two databases has not yet been fully established. Likely candidate solutions include:

- Each gateway updating to both servers.
- An archive field being implemented and the redundant server constantly searching the tables for new data.
- A full periodic backup for the entire database.
- A trigger being used on the main database where all data is automatically copied to the secondary database.

3.15. Maintainability

Very little maintenance should be required for this setup. An initial configuration will be the only system required interaction after system is put together. The only other user maintenance would be any changes to settings after setup, and any specified special cases where user settings or history need to be changed. Physical maintenance on the system's parts may be required, and would result in temporary loss of data or Internet. Upgrades of hardware and software should have little effect on this project, but may result in downtime.

3.16. Portability

This system should have the ability that, once it is together, the entire system should be able to be physically moved to any location. Code and program portability should be possible between kernel-recompiled Linux distributions. For everything to work properly, all components should be compiled from source.

3.17. Reusability

The code written and the components used should have the ability to be reused with no problems. Should time allow, and detailed instructions are written on how to create this project, everything will be completely reusable to anyone.

3.18. Application compatibility

The different components for this project will be using Java or Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python or Java code to ensure proper transfer of information.

3.19. Resource utilization

When any task is performed, it will likely use all the processing power available until that function is finished. The gateways are likely to use their processors the hardest when they are refreshing the IPTables, but this will depend largely on how many IP addresses are being implemented for IPMasquerading.

3.20. Major Classes

There are a total of five major classes: Administrators, SystemSettings, TrafficHistory, Users. The relationships between these major classes are:

A user can view their history
An administrator can view or modify a user’s traffic history
An administrator can modify system settings
An administrator can modify a user