Requirements Specification

Requested by: Mr. Ken Swarner
Systems Administrator
Computer Science Department of Siena College

TCP/IP Packet Descriptor

Mirage Incorporated

"We are there…even if you cannot see us"
Mirageinc2003@yahoo.com

Prepared by: Paul Aiuto
Richard Connell
Lauren Englisbe, Team Leader
Jayme Gresen
Jeffrey Habiniak

October 30, 2003

Requirements Specification
Table of Contents

1.0 Introduction

1.1 Purpose and Origin of the Need for This System

The Internet plays an integral role in today's multimedia society. Much of the population has access to the medium and uses it regularly to pass information, without even having an idea of how it works. However, the transmission of such is a rich and layered field, deserving a closer look. Our project, the TCP/IP Packet Descriptor, aims to allow Internet users to explore certain protocols that allow for the passing of information, and to gain a closer understanding of the process.

1.2 Scope of the System Specified

We aim to create a user-intuitive TCP/IP Packet Descriptor for use in a classroom setting. We plan to design a graphical interface that allows a student to click on any field in question. Once a field is chosen, the program will display the definition and purpose of that field, an example of data values that may be contained within, what those data values may indicate, the data values shown in different radixes, and a link to a Request For Comment page if there is one for the chosen field.

1.3 Definitions, Acronyms, and Abbreviations

All definitions, acronyms, and abbreviations will be identified within the text and in the glossary beginning on page 13.

1.4 References to Related and Supporting Systems

Our TCP/IP Packet Descriptor will run on both Windows and Macintosh personal computers through an Internet web browser.

1.5 Overview of the System

Our Packet Descriptor, at its most basic definition, will illustrate through a legible and clear graphical user interface (GUI) the IP Packet and a selection of protocols that are contained within. The user will be able to in which radix (binary, hexadecimal, octal, decimal, or ASCII) they would like to view the data information that is contained within a protocol. Data values from previously captured packets will be displayed, and within a given field, the user will be able to choose a radix in which they would like to see that field's information.

2.0 General Description

2.1 Product Perspective

The TCP/IP Descriptor will be a graphical representation of the packets contained with the IP Protocol Suite. The descriptor will not be capturing packets as they move through the Internet, but will graphically display several packets and frames provided by the client. There is an emphasis on readability and clarity because this software may be used in a classroom setting involving a projection system. Therefore, different colors will be given to each field of the many protocols, making it easier to be distinguished.

2.2 Product Functions

The majority of the packets contained within the IP Protocol Suite are offshoots of the IP Protocol Data Unit; that is, those packets are actually contained within the IP PDU. For example, FTP is contained within TCP, and TCP in turn is contained within IP. The TCP/IP Descriptor will attempt to capture this hierarchical relationship and make that intuitive for the user.

Each frame and packet will be represented graphically. Different colors will be given to each field of the many protocols, making it easier to be clearly represented and viewed in a classroom setting. Additionally, the information will be readily available without much movement on the user's part. For example, a teacher may not have to look down from the projector to the computer screen to make sure he is in the right place, or to see what he is doing.

Information for each packet will be easily obtained by clicking on the fields within the packet. The software must be able to run and be viewed clearly on any web browser and must be tested thoroughly to make sure that this is satisfied.

2.3 User Characteristics

- Must be able to view all the information on the monitor without scrolling down
- Must be able to display the program on a 1024x768 pixel projector, and be able to be read the screen
- Will be able to get information just by clicking the mouse
- Is expected to have little or no knowledge of packets and protocols

2.4 General Constraints

In any software engineering project there are restraints that must be considered as one is working. These include size of team, knowing how to use the appropriate software at the right time, and the amount of time you actually have to do the project.

The size of the team is a very important constraint. If the teams were 25 to 30 people large the project will take a fraction of the time, as opposed to our group, which is made up of five people. With a large group there can be subgroups working on many different steps of the project. With the group that we currently have, all five of us are dividing up smaller sections of the project. Although a large group can be better it may also be harder to manage. This is an important consideration to think about when considering how to form a group.

The use of the appropriate software to develop our program is a constraint that we are all facing. In order to complete tasks correctly, orderly, and timely we need to learn various software packages that allow us to complete these tasks that were set forth. The major software packages that we are using are Visual Basic for prototyping, Dreamweaver for our web design, PHP, and C++ to implement our design. It is important that each member of the group knows how to use and be able to explain each step of the process in completing our software and know what part the software packages played in the completion.

The time allotted to complete the project is the most important constraint. This is the deadline. It is crucial for a team to follow a schedule and adhere to it in order to make the deadlines for the project. If deadlines are not kept it will give the company a bad reputation and restrict the team from getting other clients. It may also create legal trouble because of a breach of contract that you and the client have set forth.

2.5 Assumptions and Dependencies

We are assuming that our team will remain together in the organized efficient manner that we are currently working. It is also an assumption that everyone in the group will continue to put forth 100% of their effort into the project. Also we assume that we will continue to receive information and feedback from our client, Mr. Swarner. We will depend on this information about requirements to be reliable and continuous. This information is vital to designing the software in the manner the client desires.

3.0 Specific Requirements

3.1 Functional Requirements

The TCP/IP Descriptor software will serve one user case. This program will be used for instructional/investigative purposes pertaining to TCP/IP Packets; however, the program may also be used by anyone with access to the website. The following is a listing of functional requirements that the system will perform for the user:

- Will contain information for various protocols
- Produce Graphical User Interface (GUI) to colorfully and clearly display the components of specified protocols
- Display on a 1024x768 pixel screen
- Certain menus should be visible on every page to allow user to go back to protocol (i.e. if user wishes to look at a different protocol, there should be a menu to provide for that option)
- Clearly show how different protocols are related to one another through our hierarchical conception of the IP Protocol Suite
- Ability to change either an independent field or an entire protocol into a different radix (i.e. binary, octal, hexadecimal)
- Ability to disply different protocols (i.e. Simple Mail Transfer Protocol (SMTP), PING)
- Produces message box when user clicks on a field for more explanation
    - Purpose of field
    - Options for pattern
    - Bit pattern form
    - Minimum/Maximum length
- Display Request For Comments (RFC) number for relevant fields and give link

3.2 External Interface Requirements

This will be determined in the Design phases of the Software Development Process.

3.3 Design Constraints

- The TCP/IP Packet Descriptor must be web-based and will be hosted on the Ares Siena College computer science web server
- Must be viewable with Microsoft Internet Explorer 5 or higher and Netscape Navigator 7.1.
- Will be using Macromedia Dreamweaver for the GUI design, mated with PHP for functionality
- Must be clearly presentable on 1024x768 classroom projector screens or computer monitors
- Must use clear and distinguishable colors for boxes, fields, letters, numbers and values
- When boxes "blow out," they should not block the field which it references
- Must have minimum amount of scrolling for better presentation

3.4 Attributes

Since the program will be based on the Oraserv Computer Science server at Siena College, we can assume that it is stable and can be accessed at any time. We are not working with sensitive data, so security does not seem to be an issue.

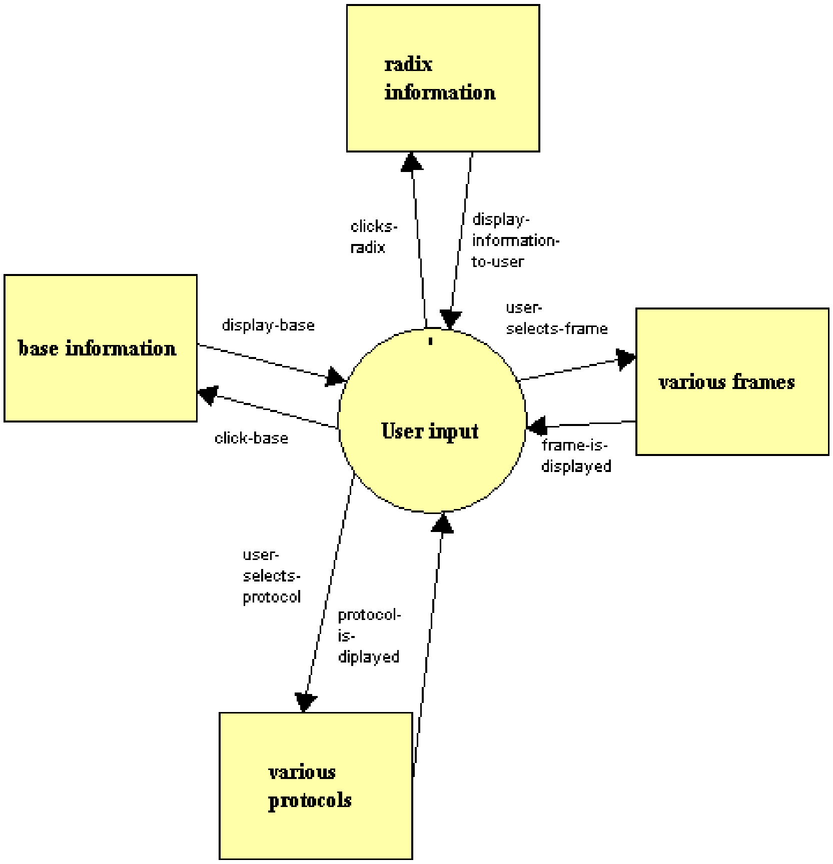
3.5 Other Requirements

The future desirable abilities for the TCP/IP Packet Descriptor are to make it robust enough for future Software Engineering teams to expand upon the program as the client desires.

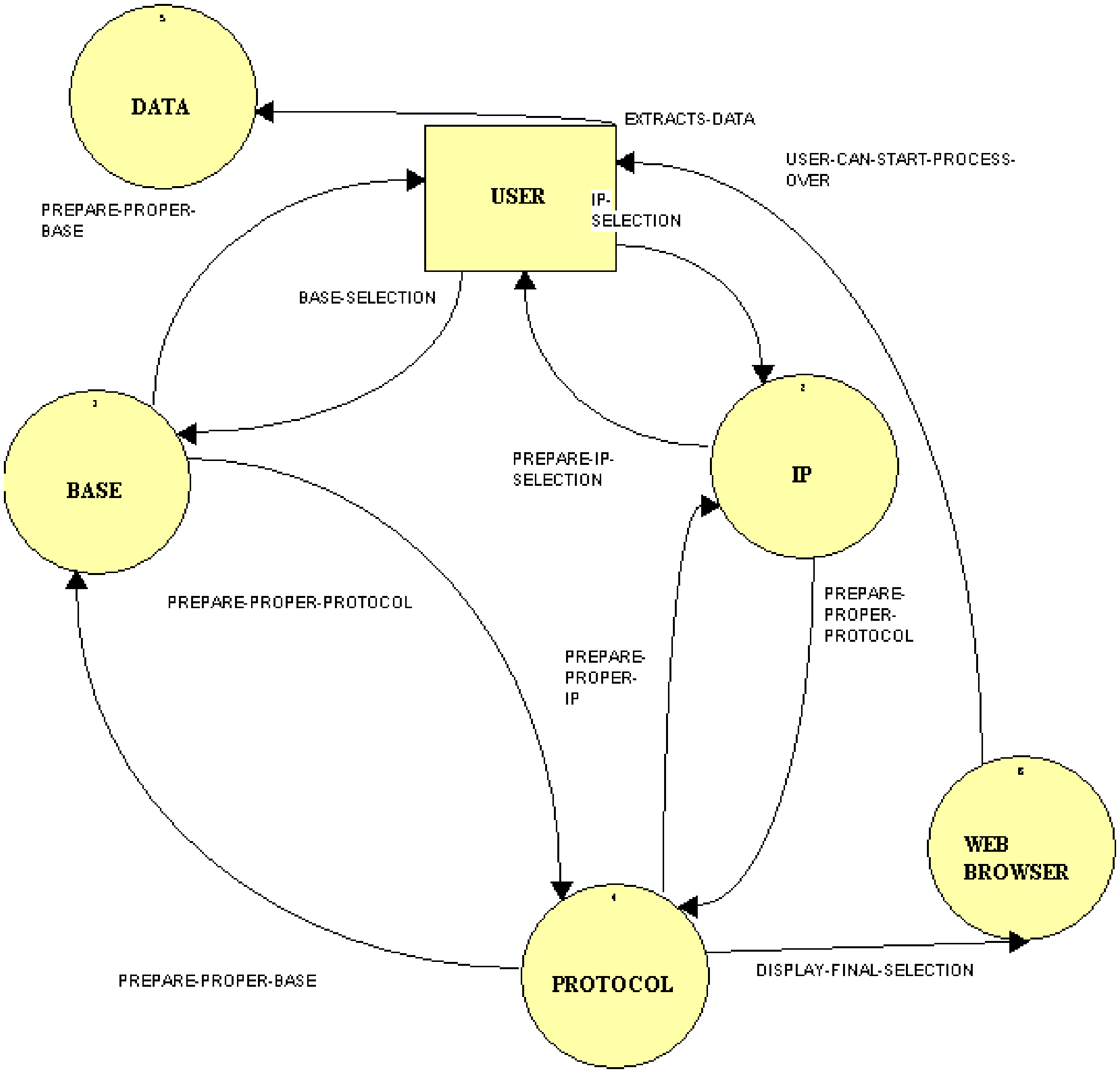
4.0 Appendix

4.1 Data Flow Diagrams

Level 0 Diagram:



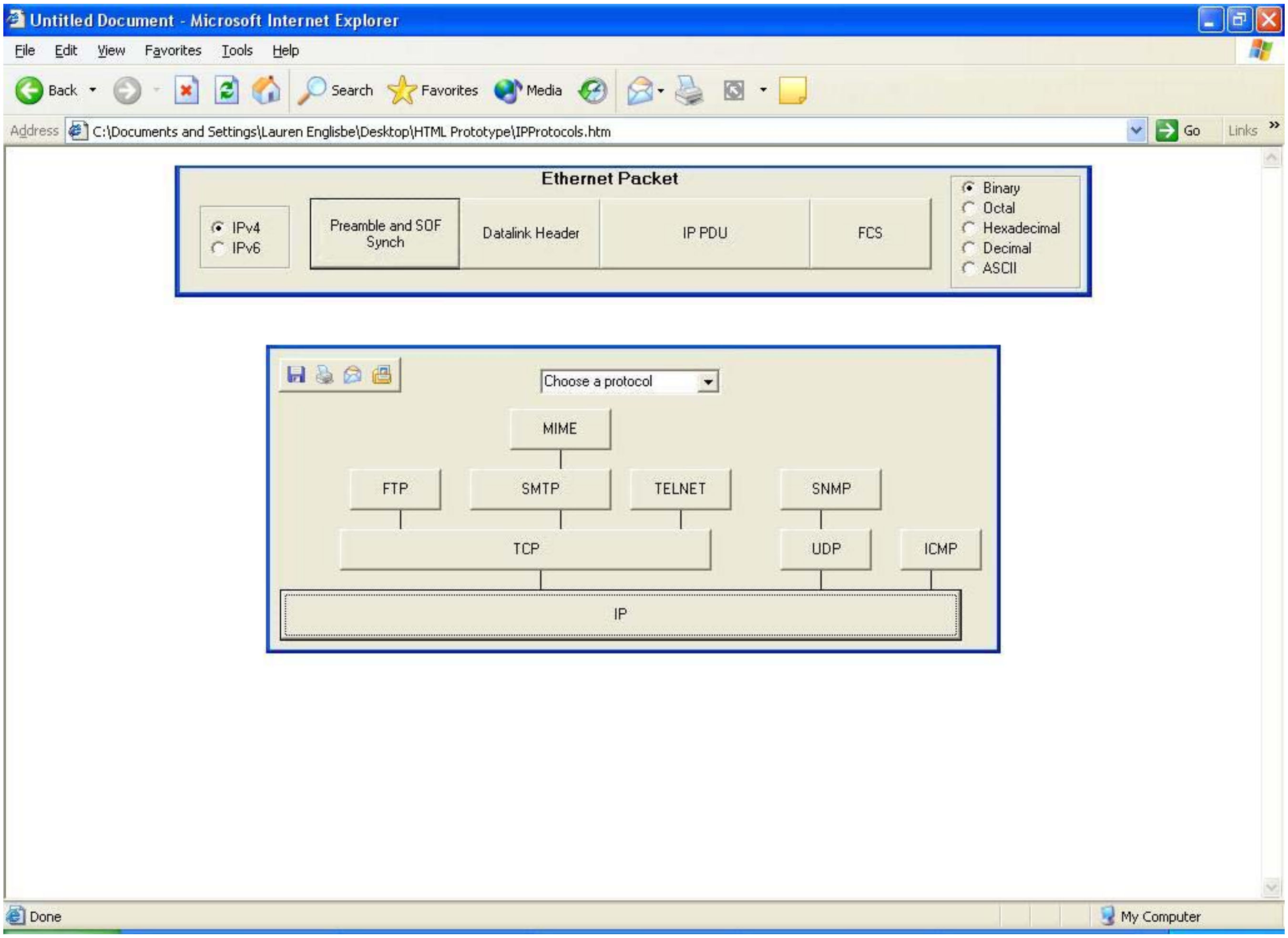
Context Diagram:



4.2 The Prototype

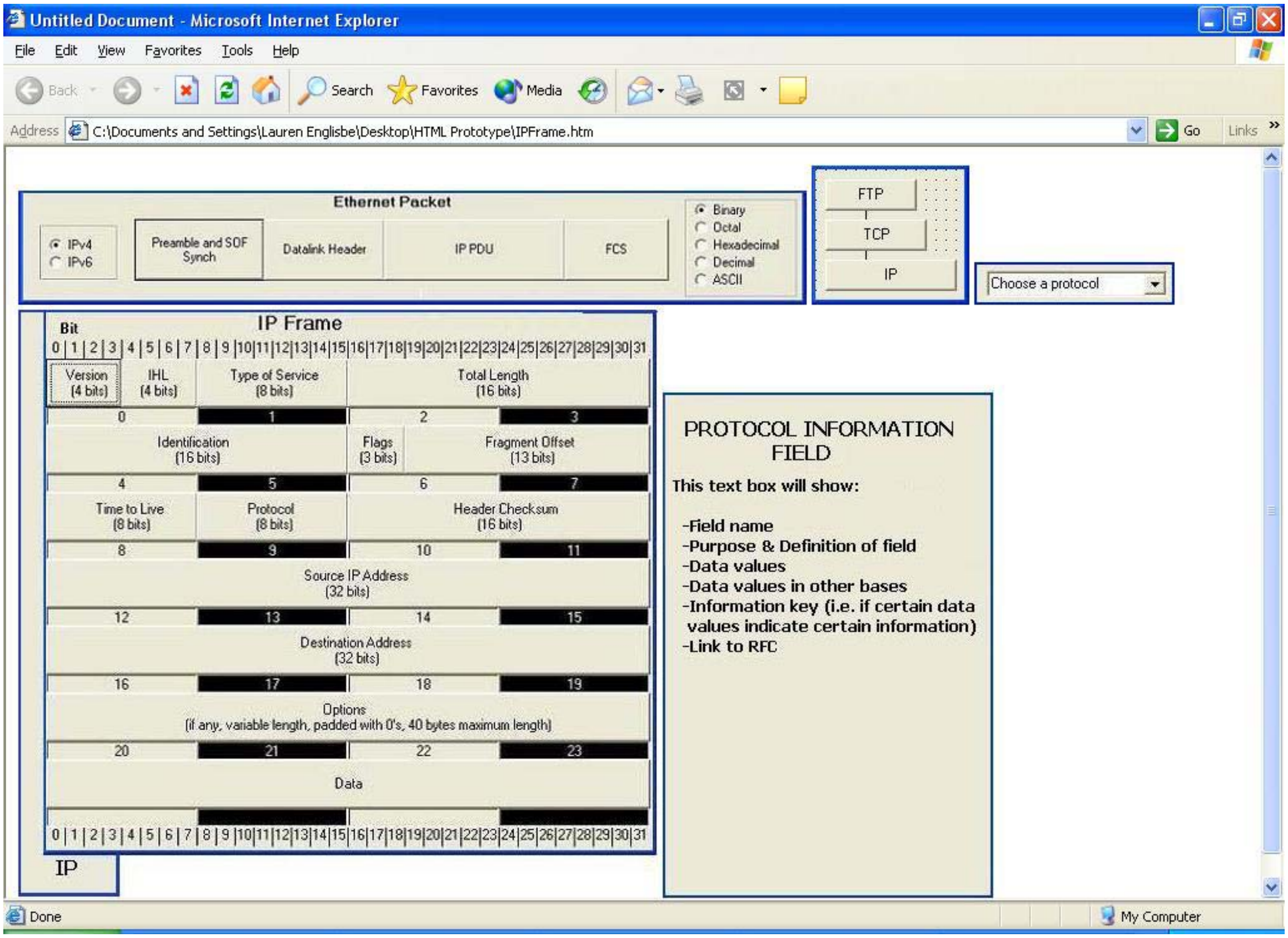
A general schema for this process is illustrated by the following prototype screenshots.

1) The user chooses a protocol to view.



2) Once the FTP protocol is chosen, the IP frame is displayed. This IP frame is the basis for all the protocols we will display in our software.

(The Protocol Information Field at the lower right will display the definition and purpose of the field, the given data values and what those data values indicate, the data values in other radixes, and a link to the RFC, if it exists.)



Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links

Address: C:\Documents and Settings\Lauren Englisbe\Desktop\HTML Prototype\IPFrame.htm

Ethernet Packet

● IPv4
○ IPv6

Preamble and SOF Synch	Datalink Header	IP PDU	FCS

● Binary
○ Octal
○ Hexadecimal
○ Decimal
○ ASCII

FTP
TCP
IP

Choose a protocol

IP Frame

Bit
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31

Version (4 bits) | IHL (4 bits) | Type of Service (8 bits) | Total Length (16 bits)
0 | 1 | 2 | 3
Identification (16 bits) | Flags (3 bits) | Fragment Offset (13 bits)
4 | 5 | 6 | 7
Time to Live (8 bits) | Protocol (8 bits) | Header Checksum (16 bits)
8 | 9 | 10 | 11
Source IP Address (32 bits)
12 | 13 | 14 | 15
Destination Address (32 bits)
16 | 17 | 18 | 19
Options (if any, variable length, padded with 0's, 40 bytes maximum length)
20 | 21 | 22 | 23
Data

0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31

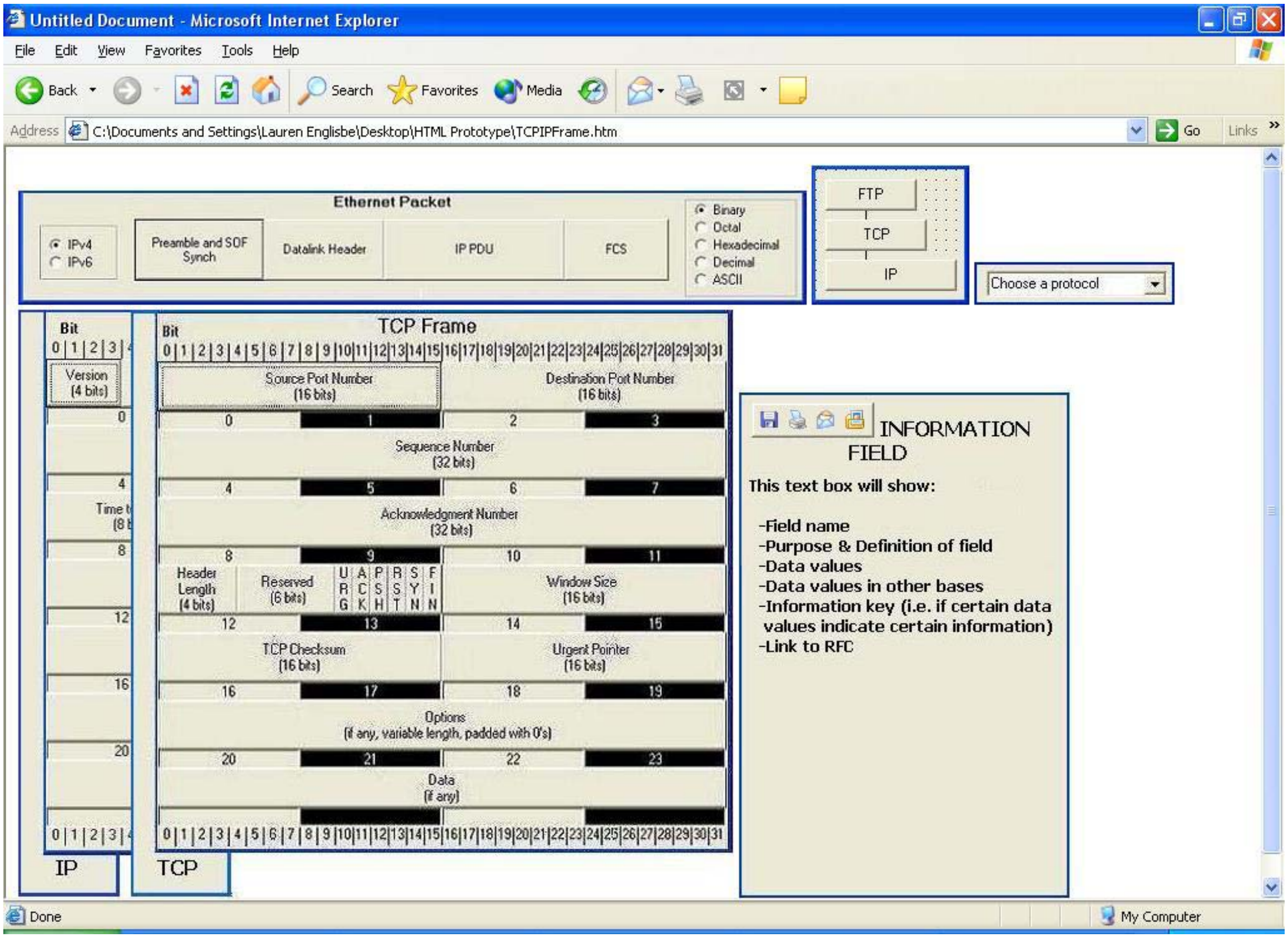
IP

PROTOCOL INFORMATION FIELD

This text box will show:

-Field name
-Purpose & Definition of field
-Data values
-Data values in other bases
-Information key (i.e. if certain data values indicate certain information)
-Link to RFC

3) The user is then able to click on the data portion of the IP frame, which will display the TCP frame, the next level before we get to FTP.



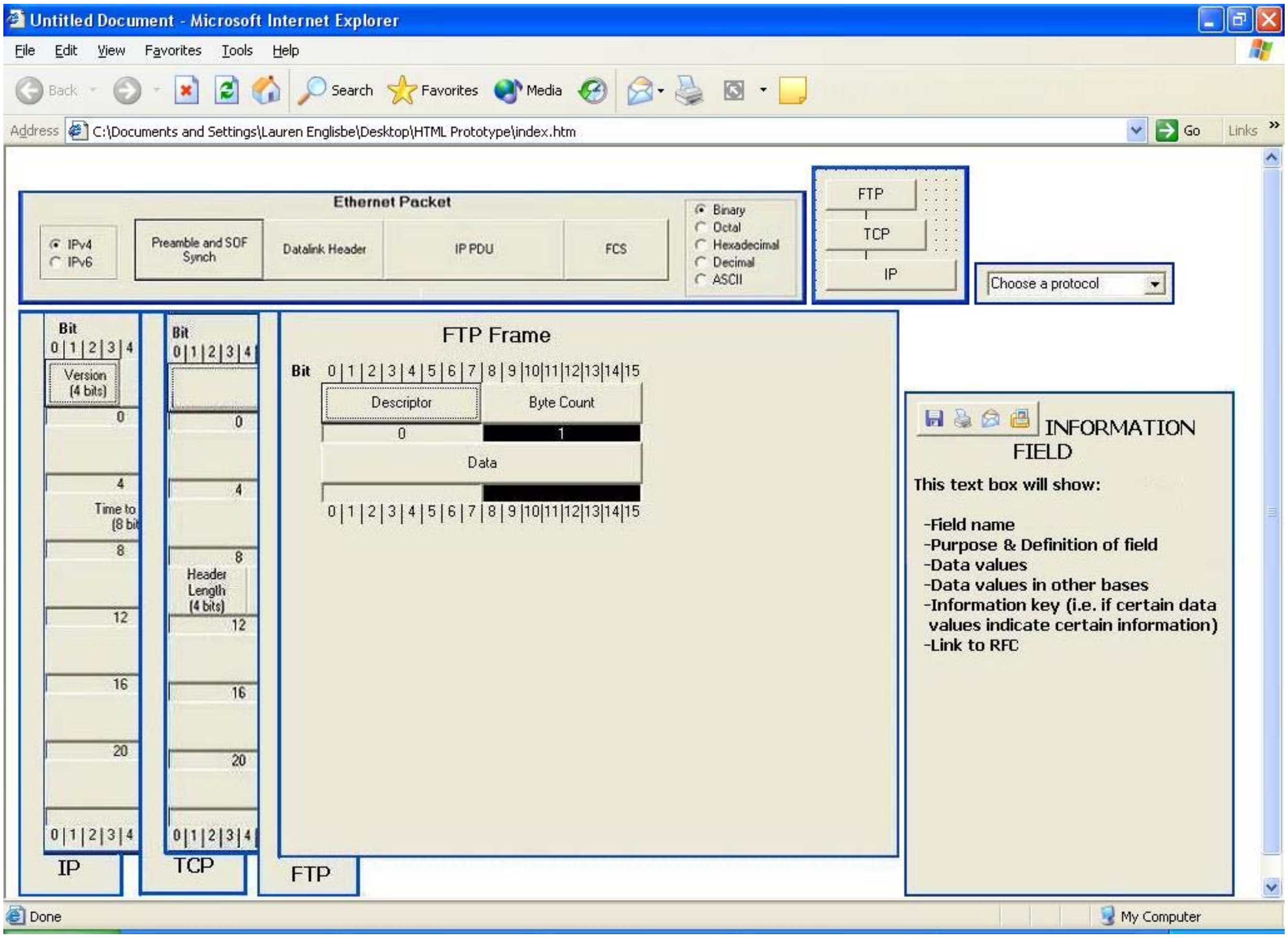
C:\Documents and Settings\Lauren Englisbe\Desktop\HTML Prototype\TCPIPFrame.htm

INFORMATION FIELD

This text box will show:

-Field name
-Purpose & Definition of field
-Data values
-Data values in other bases
-Information key (i.e. if certain data values indicate certain information)
-Link to RFC

3) Now that TCP is displayed, the user can click on its data portion to display the protocol for FTP.



C:\Documents and Settings\Lauren Englisbe\Desktop\HTML Prototype\index.htm

This text box will show:

-Field name
-Purpose & Definition of field
-Data values
-Data values in other bases
-Information key (i.e. if certain data values indicate certain information)
-Link to RFC

*Note that any information pertinent to FTP or to any of the protocols will be displayed in the Protocol Information Field to the right of the protocol frames.

4.3: Glossary of Terms

ASCII:

American Standard Code for Information Interchange: a code for representing English characters as numbers, with each letter assigned a number from 0 to 127.

Attribute:

A named value or relationship that exists for some or all instances of some entity and is directly associated with that instance.

Binary:

Pertaining to a number system that has just two unique digits, 0 and 1. Computers operate on a binary number system.

Code:

The symbolic arrangement of data or instructions in a computer program or the set of such instructions.

Data Flow Diagram:

A graphical notation used to describe how data flows between processes in a system. They are a representation of the functional decomposition of a system.

Decimal:

Refers to numbers in base 10—the numbers we use in everyday life.

Dynamic Combo Menu:

Menu showing all actions possible at the current moment.

Frame:

A feature that divides a browser's window into separate segments that can be scrolled independently of each other; a single step in a sequence of programmed instructions

GUI:

Graphical User Interface: A user interface based on graphics (icons, pictures, and menus) instead of text; uses a mouse as well as a keyboard as an input device.

Gantt Chart:

A chart that depicts progress in relation to time, often used in planning and tracking a project

HTML:

Hypertext Transfer Markup Language: A markup language used to structure text and multimedia documents and to set up hypertext links between documents, used extensively on the World Wide Web.

Hexadecimal:

Refers to the base-16 number system which consists of 16 unique symbols: the numbers 0 to 9 and the letters A to F.

Hypertext:

A computer-based text retrieval system that enables a user to access particular locations in web pages or other electronic documents by clicking on links within specific web pages or documents.

Internet:

An interconnected system of networks that connects computers around the world via the TCP/IP protocol.

Linear Sequential Model:

Sometimes called the *classic life cycle* or the *waterfall model*, this model suggests a systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing, and support.

Linux:

A trademark for an open-source version of the UNIX operating system.

Network:

A group of two or more computer systems linked together.

Open-Source:

A method and philosophy for software licensing and distribution designed to encourage use and improvement of software written by volunteers by ensuring that anyone can copy the source code.

PHP:

PHP Hypertext Preprocessor (server-side scripting language)

Packet:

A short block of data transmitted in a packet switching network.

PDU:

Protocol Data Unit: A packet of data passed across a network.

Protocol:

A set of formal rules describing how to transmit data, especially across a network.

RFC:

Request for Comments: One of a long-establish series of numbered Internet informational documents and standards widely followed by commercial software and freeware in the Internet and Unix communities.

Software:

The code executed by a computer, as opposed to the physical device which they run on.

TCP/IP:

Transmission Control Protocol/Internet Protocol: A suite of protocols for communication between computers, used as a standard for transmitting data over networks and as the basis for standard Internet protocols.

UNIX:

A powerful operating system developed at the ATT Bell Laboratories.

Use Case:

The specification of sequences of actions that a system, subsystem, or class can perform by interacting with outside actors.

Visible Analyst:

Project management software used in Computer-Aided Software Engineering (CASE) to create such illustrations as the data flow diagrams.