

# Software Plan

Requested by: Ken Swarner  
System Administrator  
Siena College  
Computer Science Department

## TCP/IP Packet Descriptor

### Paradigm Solutions

Prepared by: Jim DeSario, Team Leader  
Ryan Fischer  
Mike Sebast  
Justin Waterman  
Jon Baker  
Mark Mossman

**September 27, 2004**

# TCP/IP Packet Descriptor

## Software Plan

### *Table of Contents*

#### **System Definition:**

Section 1: Problem Definition .....	1
Section 2: System Justification .....	1
Section 3: Goals for the System and the Project.....	1
Section 4: Constraints on the System and the Project.....	1
Section 5: Functions to be Provided (Hardware and Software/People).....	1
Section 6: User Characteristics .....	2
Section 7: Development/Operating/Maintenance Environments.....	2

#### **Project Plan:**

Section 1: Life-cycle Model.....	4
Section 2: Organizational Structure .....	5
Section 3: Preliminary Staffing and Resource Requirements.....	6
Section 4: Preliminary Development Schedule .....	6
Section 5: Project Monitoring and Control Mechanisms.....	6
Section 6: Tools and Techniques to be Used .....	6
Section 7: Programming Languages .....	6
Section 8: Testing Requirements .....	7
Section 9: Supporting Documents Required.....	7
Section 10: Manner of Demonstration and Delivery .....	7
Section 11: Sources of Information .....	8

#### **Appendix:**

Section 1: Gantt Chart (September – December) .....	9
Section 2: Glossary of Terms .....	17

## **1.1: Problem Definition**

With the Internet's growth and popularity continuously growing, knowing how computers communicate is becoming increasingly important. Last year, our client, Ken Swarner, had commissioned other groups to build a web-based educational tool that would parse, decipher, and show the user the contents of a TCP/IP (see appendix) packet in a meaningful and graphical way. This year Mr. Swarner, along with the help of Dr. Timothy Lederman, has requested that we improve upon these existing programs, both to enhance what's currently implemented as well as be able to dynamically load packets into the program.

## **1.2 System Justification**

The purpose of our software is to dynamically display the information contained in a variety of TCP/IP packets. These packets will be converted into a human readable form using a Graphical User Interface (GUI). This will make it easy to follow the process of network data transfer and will aid in the understanding of the information contained in the network packets being sent to the computer.

## **1.3 Goals for the System and Project**

The goal of the project is to alter any problems in the current interface and extend it to dynamically accept TCP/IP packets and display the desired packet to be interpreted.

Our team goal is to obtain a better understanding of TCP/IP, how computers communicate, and to see what is going on inside of the packets being transferred.

## **1.4 Constraints on the System and on the project**

The major constraint on this project is that it must be web based and run on an apache server. The deadline for this project is April 2005.

## **1.5 Functions to be Provided (Hardware and Software/People)**

A database that will store data from sequences of protocol packets captured and stored by Ethereal.

A GUI that will import packets from the database, and will convert them from binary into a user-friendly language. With our new enhancements to this product, the user will get an in-depth view of the IP packet and its related structures.

We will keep documentation on all meeting with Mr. Swarner and post them on our website for referral.

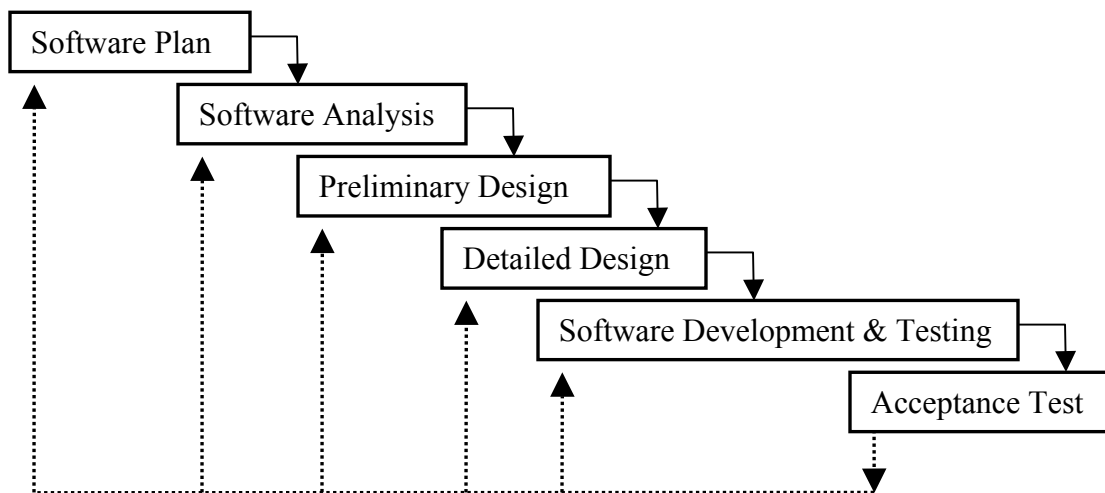
### 1.6: User Characteristics

This program is designed for the Computer Science professors to teach the students how the inner system of the TCP/IP packet works. The main user of the program is Ken Swarner, the system administrator of the Computer Science Department.

### 1.7: Development/Operating/Maintenance Environments

The project will be designed and created on the Siena College Software Engineering workstations. The project will require the user to select from a list of packets, captured from a program, Ethereal, which the program should display. The software will then display and describe the packet in full color detail. The project will require little maintenance. The system will be developed in such a way that any future work on the system can be easily picked up by other programmers.

### 2.1 Life-cycle model: Linear Sequential (Classic Waterfall) Model



#### Software Plan

The purpose of the software plan is to define the problem that needs to be solved and a preliminary project plan is formulated.

### Software Analysis

During the software analysis phase the software engineers work understand the nature of the program they are producing and all information regarding the required function, behavior, performance, and interface of the software.

### Preliminary Design

In the preliminary design, requirements are translated into a representation that can be appraised for quality before coding begins. The design for the code is formulated and descriptions of all functions and data structs are written.

### Detailed Design

Framework for the entire code is developed. Individual functions and sub-routines are illustrated.

### Software Development & Testing

Code design is implemented, compiled, and tested. Any bugs or errors will be worked out in this phase. Software will be fully functional after this step of the software development process.

### Acceptance Test

Before the software is delivered to the customer, the software engineers make sure that the software provides every interface and function desired by the customer. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested. In this way, errors may be corrected and it can be established that defined input will produce actual results that agree with the required results.

## **2.2: Organizational Structure**

Paradigm Solutions is comprised of the following members:

<u>Name</u>	<u>E-Mail Address</u>	<u>Phone Number</u>
Jim DeSario	james.desario@students.siena.edu	(518) 782-6365
Ryan Fischer	ryan.fischer@students.siena.edu	(518) 782-5655
Mike Sebast	michael.sebast@students.siena.edu	(518) 782-5650

Justin Waterman	justin.waterman@students.siena.edu	(518) 782-5930
Jon Baker	jonathan.baker@students.siena.edu	(518)782-6363
Mark Mossman	mark.mossman@students.siena.edu	(518) 782-5900

Paradigm Solutions is organized as follows for the TCP/IP Packet Descriptor Project:

Jim DeSario – Team Leader  
Ryan Fischer – Web Master  
Justin Waterman – Librarian  
Mike Sebast – System Administrator  
Jon Baker – Public Relations  
Mark Mossman – Consultant

The team structure of Paradigm Solutions is Democratic. All of the decisions will be decided by majority vote. The team leader will be the tiebreaker in the case of split decisions. There will be no hierarchy as all positions are considered to be of equal importance to our company's success.

The Work assignment for each member is as follows:

Team Leader – Organize meetings and interviews, assign individual project work to other members, assemble gathered information, guide the team throughout the semester.

Web Master – Designs and develops the project web page. Updates project information as completed.

Librarian – Keeps track of all documentation needed for project.

System Administrator – Sets up and maintains team member accounts on project workstations.

Public Relations – Organizes meetings with clients, serves as the public focal point, passing on all communications to clients and the rest of the group.

Consultant – Performs various tasks required by the project and other team members.

### **2.3: Preliminary Staffing and Resource Requirements**

The software resources we will require include an Apache web server with PHP support, Microsoft Office software package, as well as a web page editor (Dreamweaver and Fireworks). The only hardware resources required will be the computer systems and printers used in developing the software. Our primary resource will be our client, Ken Swarner, as well as Dr. Timoth Lederman, our Software Engineering professor.

### **2.4: Preliminary Development Schedule**

Refer to the preliminary development schedules or Gantt charts on pages [insert page numbers here] of this document.

### **2.5: Project Monitoring and Control Mechanisms**

The project team will meet with the client regularly twice a week to discuss the progress on the project and to receive feedback. The team itself will meet frequently to assess the client's responses to the team's presentations of the Problem Definition/Project Plan, Software Requirements Specifications and Preliminary Design and assign tasks for each member to reach the client's goals. If at any time the team or client is not satisfied with the progress of the project then the necessary steps will be taken to correct the problem.

### **2.6: Tools and Techniques to be Used**

Paradigm Solutions uses two computers running Microsoft Windows XP Professional and Microsoft Windows 2000. Microsoft Project and Microsoft Office will be used to organize and collect details for the project. Macromedia Dreamweaver and PHP will be used to design and develop the web application. To obtain packets for the application, the team will be using Ethereal. The project will require the following web browsers: Internet Explorer, Netscape Navigator, and Mozilla.

### **2.7: Programming and Scripting Languages**

This project will require the integration of several programming and scripting languages, listed below:

HTML  
PHP.

## **2.8: Testing Requirements**

The testing will take place as each part of the program of the system is completed. The clients will be notified as each testing session occurs. The clients will also be encouraged to participate in the testing process. The final testing of the system will configure to all specifications of Mr. Swarner. The final product will be able to run on all major platform configurations.

## **2.9: Supporting Documents Required**

The following supporting documentation will be provided to our clients on the following dates:

- |  |                    |
|--|--------------------|
| 1) Project Definition/Project Plan:      | September 27, 2004 |
| 2) Software Requirements Specifications: | November 1, 2004   |
| 3) Preliminary Design:                   | December 6, 2004   |

Additional documentation on the Detailed Design will be provided to the client in the Spring of 2004.

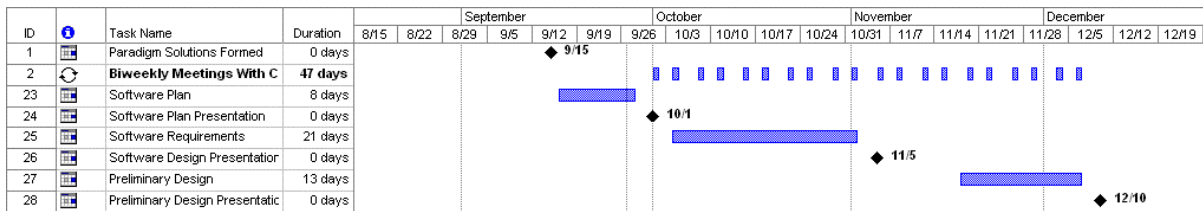
## **2.10: Manner of Demonstration and Delivery**

In addition to delivery of supplemental documents, there will be several presentations given throughout the year consisting of PowerPoint presentations, speeches, demonstrations, and handouts. These presentations serve to not only demonstrate that our planned goals are being met, but also to demonstrate what has been completed as of the date of the presentation. The dates of delivery and presentations are as follows:

- |   |                    |
|---|--------------------|
| 1) Problem Definition/Project Plan:                   | September 27, 2004 |
| 2) Problem Definition/Project Plan Presentation:      | October 1, 2004    |
| 3) Software Requirements Specifications:              | November 1, 2004   |
| 4) Software Requirements Specifications Presentation: | November 5, 2004   |
| 5) Preliminary Design:                                | December 6, 2004   |
| 6) Preliminary Design Presentation:                   | December 10, 2004  |



### 3.1: Gantt Chart (Current Semester)



### 3.2: Glossary of Terms

#### 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

#### HMTL:

*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.

#### 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.

**Network:**

An openwork fabric or structure in which cords, threads, or wires cross at regular intervals.

**PHP:**

Hypertext Preprocessor (server-side scripting language)

**Packet:**

The unit of data sent across a network. "Packet" is a generic term used to describe a unit of data at any layer of the OSI protocol stack, but it is most correctly used to describe application layer data units ("application protocol data unit", APDU).

**Protocol:**

A standard procedure for regulating data transmission between computers.

**Software:**

Written programs, procedures, or rules and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory.

**TCP/IP:**

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