# WinstonSoft Weekly Report

## *14.10.2007 – 21.10.2007*

In the first meeting, we were asked to design the website for the group. This website was to contain the project description and information about group members and their roles. We had already began designing the website the week before, so this week we just made some minor changes (like adding the group logo and updating the roles of each member) to our design, and pretty much completed it.

On Tuesday, we had a meeting where everyone proposed their ideas on the project topic. We discussed about an initial design, and our project assistant provided feedback on our ideas. This feedback was essentially helpful since our project demanded knowledge about several subjects and we needed some advice on where to start.

We decided to start by doing research about four widely used mail protocols, namely IMAP, POP3, SMTP and NNTP. The reason for choosing these as a starting point is that these protocols share some common properties. To begin with, they are all used for a similar purpose (mail messaging), thus they have semantically similar properties. Second, they are all simple ASCII based protocols that that adopt the command-response model in a line oriented fashion, which makes them also syntactically similar.

We divided these four protocols among team members so that each member should conduct research and have a brief understanding about the protocol assigned to him/her. In the following section, we will provide a brief summary of what each member has done during the week.

1. Can Hoşgör: SMTP (Simple Mail Transfer Protocol)
   SMTP is a text based protocol that is used for sending e-mail messages. Normally, the SMTP protocol operates over TCP port 119, but since this setting can be overridden by configuration, we need to perform further inspection of packets sent and received. When the connection is established between the client and the server, the client sends commands, and the server sends back replies. SMTP is a line oriented protocol, and uses CRLF pair as end of line markers. Commands and replies are not case sensitive, but some of their arguments, for example, mailbox user names may be so. Commands and replies are composed of characters form the ASCII character set. Non-ASCII characters in email messages are encoded using MIME. Most commands and replies are single-line, with the exception of DATA command. This command spans multiple lines, and uses a dot (.) on a line by itself to indicate end of mail. Replies sent by the server is prefixed with a 3 digit status code that indicates the success or failure of the command. Most SMTP sessions are very short and straightforward: The client connects to server, (optionally) authenticates himself with a user name and password, sets the sender and receiver(s) of the message, sends the message body and finally quits. For this protocol, I think it will not be hard to implement a parser and extract useful information from network packets.

2. Elvan Gülen: My research topic is (RFC 1939) Post Office Protocol - Version 3. So apart from helping to form our website for the project, I've read some documents about POP3. First of all, I obtained a general idea about how the server host starts the POP3 service and also how the server and the client respond to the commands until the connection is lost. Besides, I looked through nearly all commands (like QUIT, LIST, UIDL, RETR, etc..) in the POP3. They have some specifications that will be useful for our auto-sensing mechanism. One of these specifications is that there are two status indicators: "+OK" and "-ERR". The given reply by the

POP3 server to nearly all commands (there are some exceptions but we know what are those commands so this doesn't become a problem for us) is significant to these two status. Also there are three way for the client to identify itself to POP3 server such as USER/PASS , APOP and AUTH. This information is also useful for us. And about commands, we roughly know what can be the restrictions and possible responses and the arguments that can be given after the commands. In addition, I connected to my account in mail.metu.edu.tr through POP3 service, and tested how POP3 works directly and this try-outs intensified my knowledge about this protocol.

3. N.Ilker ERCIN – 1395003: This week the RFC document for IMAP4 ( RFC 3501 ) is studied for useful information to identify the packets of IMAP protocol. IMAP *allows a client to access and manipulate e-mail messages on a server, it supports a single server.* All interactions transmitted by client and server are in the form of line, that is, strings that end with CRLF. Client commands are prefixed with an identifier (a short alphanumeric string) called a tag. Data transmitted by the server to the client and status responses that do not indicate command completion are prefixed with the token "*", and are called untagged responses.Messages have several attributes, which may be useful in identifying the packets, such as message numbers (Unique *Identifier, Message Sequence Number*), flags (*\Seen, \Answered, \Flagged, \Deleted, \Draft, \Recent*) which is probably the most useful attribute for identifying operation, internal date, size message, envelope structure, body structure. In addition to these, the commands (*LOGIN, CAPABILITY, AUTHENTICATE etc..*) seems useful for identification. Using experimental commands ( commands of the form “X<atom>” ) for identification can be a practical solution.
4. Çağla Çığ - As for my part of RFC research regarding mail/news protocols, I did research on NNTP (network news transfer protocol). This protocol, being used over the internet, is a facility for reading and posting Usenet articles, as well as stream-based transferring of news among news servers. The well-known TCP port 119 is reserved for NNTP. However, clients may connect to a news server using another port, so the aim of my research is to find out a way of generalizing NNTP packet contents by finding common patterns among them so that it will be possible to distinguish these packets without the need for port information. NNTP is defined by RFC 977, so the following part of my writing consists of the results of my research on the RFC 977 document. The design of NNTP allows the subscribers to select only the items they wish to read by storing news articles in a central database. Also provided are indexing, cross-referencing and expiration of aged messages. The news server being mentioned in the document uses a stream connection like TCP and SMTP-like commands and responses. Since my aim is to distinguish NNTP packets, I am planning to do so by stating common patterns among commands used in NNTP packets. Some properties of these commands are as follows:
    - Commands consist of a command word, which in some cases may be followed by a parameter.
    - Commands and command parameters are not case sensitive.
    - Each command line must be terminated by a Carriage Return – Line Feed pair.
    - Command lines shall not exceed 512 characters in length, counting all characters including spaces, separators, punctuations, and the trailing CR-LF pair.

    On the other hand, the responses to these commands are of two kinds: textual and status. The intention is that text messages will usually be displayed on the user’s terminal whereas command/status responses will be interpreted by the client before any possible display is done.