

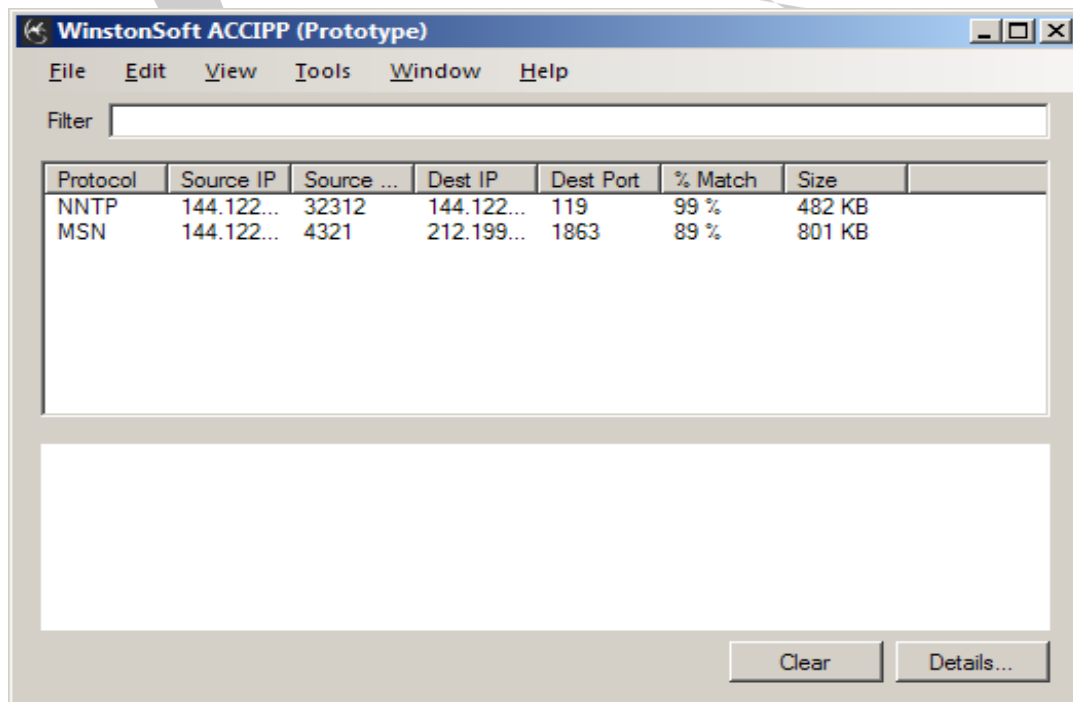
WEEKLY REPORT 3

This week, instead of doing individual work on the project, we worked as a team basically on the requirement analysis report. Regarding the analysis report, the primary step we took was to make a research and have a deep idea about what an analysis report is and how to write one. For doing that, we used the sample software requirements specification document of IEEE. After we examined the report carefully, we faced the fact we had to conduct a lot more research regarding our project. This research included the following topics:

1. Similar programs to Wireshark like SmartSniff which we can observe the network traffic by capturing PCAP files.
2. We studied TCP Headers and wrote some pseudo-code to parse the TCP packets into internal data structures. The code can be seen in the Appendix.
3. Functions and the possible requirements, user class and characteristics.

As a consequence of the research we conducted, some topics regarding the project, like design constraints and requirements, gained clarity and this was reflected in our analysis report.

In addition to the research we designed a prototype GUI for the user interface of our project. However, the whole design in our minds is not fully reflected in that GUI.



Appendix

```
struct packet {
    char dst_mac[6]; // mac address of receiver
    char src_mac[6]; // mac address of sender
    char type[2]; // type of packet (must be 0x08 0x00 for ip)
    struct {
        char version; // 0x45 for ipv4 (this also implicitly determines the length of
header (20 bytes in this case))
        struct differentiated_services_field { // 0x00 (didn't understand yet)
            unsigned codepoint: 6; // default 0
            unsigned ect: 1; // 0
            unsigned ecn_ce: 1; // 0
        }
        char total_len[2]; // total length of packet in bytes
        char identification[2]; // didn't understand yet
        struct flags {
            reserved : 1; // not set
            dontfragment: 1; // set (1 if packet is not fragmented, 0 if
fragmented)
            morefragments: 1; // not set; (packet is fragmented into more
packets, might be 0 if this is the last fragment)
        } ip_header;
        char fragmentoffset;
        char timetolive;
        char protocol; // 0x06 for tcp, 0x11 for udp;
        char headerchecksum[2]; // sum of all bytes in header
        char src_addr[4]; // ip address of sender
        char dst_addr[4]; // ip address of receiver
        char src_port[2]; // senders port
        char dst_port[2]; // receivers port
        char seq_number[4]; // relative ?
        char next_seq_number[4]; // relative ?
        char ack_number[4]; // relative ?
        char header_length;
        struct {
            congestion_window_reduced: 1;
            ecn_echo: 1;
            urgent: 1;
            acknowledgement: 1;
            push: 1;
            reset: 1;
            syn: 1;
            fin: 1;
        } flags;
        char window_size[2];
        char checksum[2];
        char options[12];
        char data[0]; // contents of the packet
    }
}
```

Sorular

- 1) Bu projenin sonunda ortaya cikacak programi kimler kullanacak? Biz network admins vs tarzından bişeyler düşündük ama Siemens'in hedeflediği bir kitle var mı ayrıca?
- 2) Siemens tarafından hazırlanmış, bu konuyla ilgili paylasabilecekleri bütün dokumantasyonu sağlanması mümkün müdür?
- 3) Programda ne gibi error message'lar verebiliriz? Bizden beklenen specific birşeyler var mı?
- 4) Database konusunda ne yapmamız lazım? Programa entegre bir database olacak mı? Olacak ise kullanılacak DBMS hakkında bir sınırlama mevcut mu?
- 5) Siemens'in belirlemiş olduğu "minimum hardware requirements" veya "recommended hardware requirements" var mı?