Integration Project

Kimberly Hengst *1305093*
Tim Sonderen *1465252*
Kevin Hetterscheid *1490443*
Martijn de Bijl *1470108*

14 april 2014

Inhoudsopgave

1 Inleiding

Dit is ons verslag van het project van de derde module. Voor het project hebben wij een chat applicatie gemaakt. In deze applicatie kan een gebruiker in een ad-hoc netwerk met andere gebruikers berichten uitwisselen. De applicatie ondersteunt maximaal vier gebruikers. Een gebruiker kan ervoor kiezen om met n persoon, of drie personen te chatten. De berichten zijn beveiligd met behulp van encryptie. Hierdoor kunnen alleen de andere gebruikers de berichten uitlezen. Als twee gebruikers alleen willen chatten, is dit voor de andere gebruikers gencrypt. Het netwerk waarop de applicatie draait is een ad-hoc netwerk. Dit betekent dat het netwerk geen router of server nodig heeft. De computers maken een eigen netwerk via Wi-Fi.
In het verslag zullen we het proces van het ontwerpen van de applicatie, en de keuzes die we daarbij gemaakt hebben uitleggen. Als eerste bekijken we het systeem ontwerp, hier zal de basis uitgelegd worden. Daarna zal de implementatie van de beveiliging uitgelegd worden. Als laatste zal het test proces beschreven worden.

2 Systeem ontwerp

De gebruiker kan in een user interface berichten typen en versturen. Deze berichten worden dan in een pakket opgeslagen, en het pakket wordt verstuurd. Het versturen gebeurt via een multicast socket. Dit betekent dat het bericht naar iedereen verstuurd wordt. Als een socket een pakket ontvangt, wordt deze in de GUI weergeven.

2.1 Pakketjes

Voor het versturen van de pakketjes gebruikt multicast socket een Datagram packet. Deze gebruiken wij ook voor het versturen van data. Een datagram packet heeft een byte array met data, een IP adres van de ontvanger, en een poort van de ontvanger. In mulitcast wordt een pakketje dus naar iedereen verstuurd, het IP adres is dan ook het adres van een multicast network, een IP van de IP-klasse D. De poort is een standaard poort waar wij voor hebben gekozen. In de data van de datagram packet worden nog enkele andere velden aangewezen. De eerste byte bevat de sequencenummer, de tweede byte de teller voor het aantal hops, de derde tot zesde byte bestaan uit het IP-adres van de verzender, de zevende tot tiende byte bestaan uit het IP-adres van de ontvanger. We slaan de IP-adressen ook in de header op, omdat het IP-adres in een datagram packet veranderd volgens de context. Bij een ontvangen bericht is het IP-adres die van de verzender, bij een pakket wat verzonden wordt, is het IP-adres die van de ontvanger.

Het pakket ziet er dan als volgt uit:

Byte array data	lengte van data	IP-adres verzender of ontvanger	Poort nummer

Het byte array van de data ziet er dan als volgt uit:

Byte array data				
Header				Bericht
Sequence nummer	Hop teller	IP-adres verzender	IP-adres ontvanger	

We gebruiken pakketten niet alleen voor het verzenden van berichten uit de GUI, maar ook voor een aantal andere functionaliteiten. De functionaliteiten worden dan in het bericht van de data van het packet toegevoegd. Als het bericht dan met een bepaalde string begint, wordt hiermee een functionaliteit uitgevoerd. De volgende berichten gebruiken wij:

Bericht	Functionaliteit
$[FILE]$	Geeft aan dat het bericht een byte array van een bestand is
$[BROADCAST]$	Geeft aan dat het bericht een broadcast is, een bericht dat elke seconde verstuurd wordt om aan te geven dat een persoon er nog is
$[NAME\_IN\_USE]$	Geeft aan dat een naam al in gebruik is, en niet nog een keer gebruikt mag worden
$[PRIV\_MESSAGE]$	Geeft een privé bericht aan. Het bericht zal dan alleen aan de ontvanger worden weergeven
$[NACK]$	Geeft aan dat het bericht een NACK is
$[TOO\_LATE]$	Geeft aan dat een bericht waarvoor een NACK is gestuurd niet meer in de buffer voorkomt
$[EOF]$	Geeft het einde van een bestand aan

2.2 Bestanden verzenden

2.3 Pakket verlies

2.4 User interface

De gebruiker kan via de user interface met het programma communiceren. De interface bestaat uit een aantal elementen:

- Tekstveld
- Typveld
- Een knop voor het verzenden van tekst en bestanden
- Een knop voor privé chat
- Een knop voor het afsluiten van het applicatie

3 Beveiling

Om te zorgen dat niet iedereen de inhoudt van de pakketten kan lezen, gebruiken we encryptie.

3.1 Encryptie

Voor het encryptie gebruiken we symetrische sleutel encryptie. Dit betekent dat de clients een sleutel delen, waarmee ze hun berichten kunnen coderen en decoderen.

4 Test plan

4.1 Test cases