Unreliable data transfering

Michael Koopman s1401335 en Sven Konings s1534130

# Inleiding

Tijdens de challenge van de tweede week hebben wij een bestand moeten sturen over een onbetrouwbare verbinding. In dit document beschrijven wij de veranderingen die wij hebben gemaakt, ten opzichte van de versie die op Blackboard stond.

# Sender

Na wat experimenteren met de packet size, hebben we besloten dat 128 bytes een gulden middenweg was. Als we deze lager maakten, dan moesten er heel veel pakketjes geacknowledged worden. Als we het verhoogden, werden er veel pakketten gedropt, die opnieuw verstuurd moesten worden.

Wij sturen de pakketten met een interval van 10 keer het maximaal aantal pakketten (30) naar de server. Er wordt dus ongeveer 300ms tussen elk pakket gewacht en er worden maximaal 30 onbeantwoorde pakketten verstuurd, daarna moet er eerst een pakket beantwoord worden voordat er door wordt gegaan. De time-out is het maximaal aantal pakketten vermenigvuldigd met de tussentijd, dus bij deze waardes is het 9 seconden. Iedere keer begon de server bij ongeveer 34 pakketten tegelijk de pakketten te droppen vanwege congestion. Bij het versturen van ieder pakket wordt dit naar de console geprint, en worden de pakketten toegevoegd aan een Map. Ieder pakket krijgt aan het begin een nummer, deze zijn oplopend volgens de volgorde van de pakketten. Na het laatste pakketje wordt een end-of-file pakketje gestuurd, als deze niet beantwoord wordt, dan wordt deze nog een keer gestuurd.

Ieder pakket krijgt een time-out toegewezen. Op het moment dat een time-out triggert, checken we of er pakketten geacknowledged zijn. Zo ja, dan halen we het pakket uit de map. Daarna wordt er gekeken of het pakket waarvan de tag is meegestuurd aan de time-out nog in de lijst staan. Zo ja, dan versturen we de packet opnieuw, en wijzen we er opnieuw een time-out aan toe.

# Receiver

Op het moment dat wij een niet-leeg pakket ontvangen, checken we of dit een end-of-file pakketje is. Zo ja, dan sturen we dit naar de sender, verwerken alle pakketjes, en stoppen we het programma. Zo niet, dan checken we of we het pakket al ontvangen hebben. Als dit niet zo is dan voegen we deze toe aan onze map. Vervolgens wordt er een acknowledgement verstuurd.

# Resultaten

Met onze oplossing hebben wij een server tick time van 2903 bereikt. Dit kregen we redelijk in het begin van de middag. Aan het eind van de middag was de server continu overbelast, waardoor de tijd tussen de pakketjes erg varieerden. We kregen bijvoorbeeld een halve minuut helemaal niks en vervolgens alles in 1 keer, terwijl ze wel op verschillende tijden verstuurd waren en de time-out vaak niet klopte. Hierdoor werden er veel pakketjes te vaak gestuurd en ging de tijd omlaag.