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 30 pakketten per keer naar de server. Iedere keer begon de server rond 34 pakketten een deel te congesten. Bij het versturen van ieder pakket wordt dit naar de console geprint, en worden de pakketten toegevoegd aan een HashMap. Ieder pakket krijgt aan het begin een sequence number. Na het laatste pakketje wordt een end-of-file pakketje gestuurd.

Ieder pakket krijgt een timeout toegewezen. Op het moment dat een timeout triggert, checken we of het pakket al geacknowledged is. Zo ja, dan halen we het pakket uit de HashMap, en zo nee, dan versturen we de packet opnieuw, en wijzen we er opnieuw een timeout aan toe. Het acknowledgen van een pakket en het resenden van een pakket worden ook naar de console geprint.

# 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 de packet al ontvangen hebben. Zo ja, dan wordt dit naar de console geprint, zo nee, dan voegen we deze toe aan onze HashMap en printen we dat naar de console.

# 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, en kregen we lagere tick times.