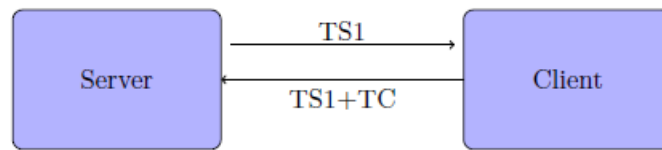

Team 12

Frédéric Blondeel
Dirk Vanbeveren
Bert Van den Bosch

Inhoudsopgave

1	Introductie	2
2	Testen	2
2.1	Algemeen concept	2
2.2	UDP vs TCP	2
2.3	Analyse van de data	2
3	Besluit	2



Figuur 1: Ping

1 Introductie

Het synchroniseren van de slave devices onder elkaar is essentieel om video of animatie op een correcte manier de tonen. In dit eerste deel van de opgave wordt gekeken naar het absolute verschil tussen client tijd en wereld tijd. Hiervoor wordt gebruik gemaakt van een server die in sync staat met een atomische wereld klok en zo meet men de verschillen in tijd. Er worden verscheidene besturing systemen en browsers gebruikt. Het verschil van UDP vs TCP zal ook verder worden toegelicht.

2 Testen

2.1 Algemeen concept

Voor de testen wordt gebruik gemaakt van een simpele server waar clients op kunnen verbinden. De client kan vervolgens het absolute tijd verschil meten met een atomische wereld klok.

Eerst en vooral is er een vertraging tussen een aangesloten client en de server, de *ping*. Dit is gemeten in miliseconden. Deze wordt gemeten door een bericht met de actuele tijd te verzenden van de server naar de client, en terug. De verzonden tijd wordt afgetrokken van de actuele tijd waarmee de ping verkregen is. In figuur 1 is de informatieoverdracht zichtbaar. In de server wordt de servertijd ($TS1$) berekent door middel van een worldtime API die de exacte wereld tijd teruggeeft, verzonden naar de client en teruggekregen. Nu wordt de actuele tijd berekent in de server $TS2$. Dus de uiteindelijke ping is:

$$ping = TS2 - TS1$$

Het is niet gegarandeerd dat de klokken van de clients allemaal gesynchroniseerd zijn met de server. Bij het terug verzenden van de client naar de server wordt de clienttijd (TC) bij het bericht gezet. Met deze TC en de berekende *ping*, is het mogelijk het tijdsverschil tussen de client en de server te bepalen (*DeltaTime*).

$$DeltaTime = (TC + ping/2) - TS2$$

Dit is de data die zal worden bekenken en vergeleken tussen browsers en besturings systemen.

2.2 UDP vs TCP

UDP heeft in tegenstelling tot TCP geen verbinding nodig tussen bijvoorbeeld server en client. TCP zal garanderen dat data correct aankomt door middel van error checking en zal ook in de goede volgorde binnen stromen. Zodat geen pakketen worden verloren zal TCP deze in een receive buffer steken en zal de applicatie de data pas lezen als ze er klaar

voor is. Dit in tegenstelling tot UDP waar de data continue zal binnen stromen, ontvangen of niet. Hier wordt ook geen error checking uitgevoerd en de volgorde is niet gegarandeerd. Het is duidelijk dat UDP veel sneller is doordat er veel minder moet worden gedaan. Dit is ook de reden dat het NTP protocol UDP zal gebruiken in plaats van TCP. Het is logisch dat voor een simpele synchronisatie tussen client en server geen complex protocol nodig is. Socket.io gebruikt het TCP protocol.

2.3 Analyse van de data

3 Besluit

Referenties