

---

**Team 12**

---

Frédéric Blondeel	h
Martijn Debeuf	h
Toon Sauvillers	h
Dirk Vanbeveren	15h
Bert Van den Bosch	h
Seppe Van Steenberghe	h

---

---

## Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>2</b>
<b>2</b>	<b>Vertraging berekenen</b>	<b>2</b>
<b>3</b>	<b>Kloksynchronisatie</b>	<b>2</b>
<b>4</b>	<b>Aftelklok</b>	<b>2</b>

## 1 Introductie

Bij een live applicatie is het belangrijker dat de aangesloten apparaten de mogelijkheid hebben synchroon een beeld weer te geven. Dit is niet altijd vanzelfsprekend dankzij verschillende factoren, zoals vertraging op en snelheid van een apparaat. Hiervoor zijn er verschillende technieken om toch synchronisatie te bekomen.

## 2 Vertraging berekenen

Er is 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 en gedeelt door twee waarmee we de ping verkrijgen.

## 3 Kloksynchronisatie

Het is niet gegarandeerd dat de klok van de clients allemaal gesynchroniseerd zijn met de server. Daarom is het ook nodig om te weten wat het verschil is tussen de tijd aan de kant van de client en de server. Bij het terug verzenden van de client naar de server wordt de client tijd er bij het bericht gezet. Met deze clienttijd en de berekende *ping*, is het mogelijk het tijdsverschil tussen de client en de server te bepalen. Door dit verschil toe te voegen aan de servertijd is het mogelijk de correcte clienttijd te vinden. Dit wordt gebruikt om een starttijd te bepalen voor elke client dat op exact hetzelfde moment zal beginnen.

## 4 Aftelklok

Bij de naïve implementatie van de aftelklok is er gebruikt gemaakt van de *setTimer()* functie die recursief een getal aftelt en tekent op een canvas. Een probleem hierbij is dat apparaten niet even snel het getal kunnen tekenen op het scherm waardoor er apparaten kunnen zijn die sneller zijn dan anderen. Daarom is er gebruik gemaakt van *setInterval()* dat periodisch het getal berekend, relatief ten opzichte van de meegegeven starttijd. Als er een client trager is en niet optijd op het scherm kan tekenen, dan zal het getal worden overgeslagen omdat het apparaat het sowieso niet zou aankunnen. Hierdoor blijven de getallen op het scherm synchroon en zal het aftellen ook op het zelfde moment stoppen op elk scherm.