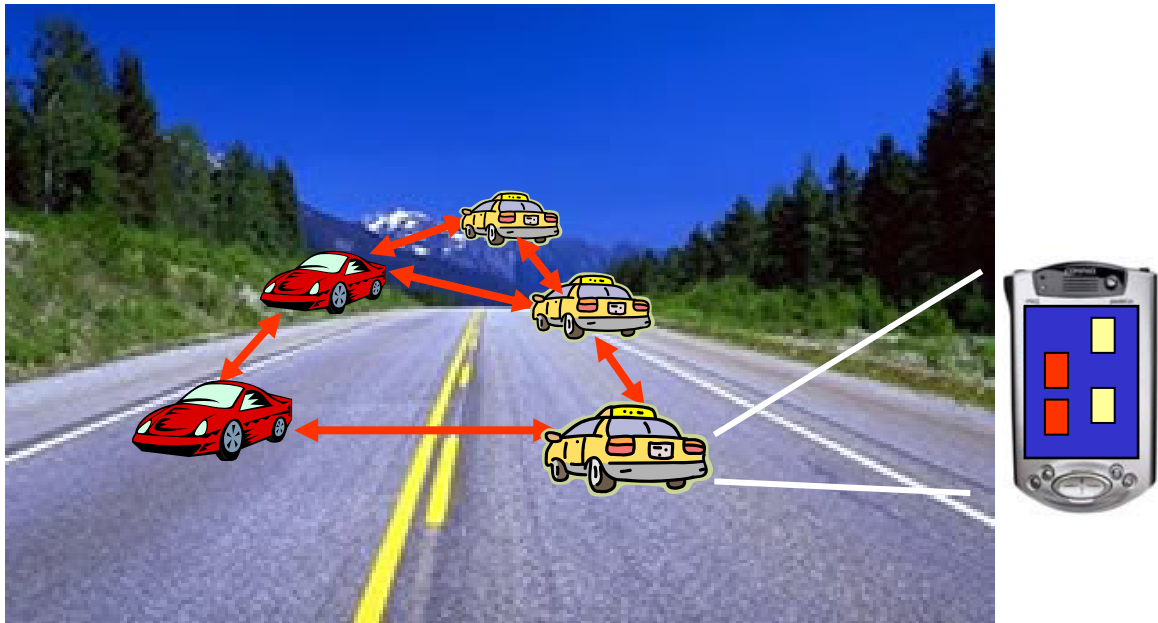


Ontwerprapport

Stelsiem om file-informatie te vergaren met behulp van wifi ad-hoc netwerken



Studenten

M. Vasic 1213849
A.M. Abikhaled 1198653

Begeleider

J.J. Reijmers

Voorwoord

Dit rapport is geschreven als onderdeel van het bachelor eindproject. Eén van de onderdelen was om een product/systeem te maken dat voortkwam uit het businessplan, dat ook onderdeel was van het bachelor eindproject.

In het rapport staat dan ook beschreven hoe een systeem is ontworpen om te worden gebruikt voor file-informatie vergaring met behulp van wifi ad-hoc netwerken.

Deze scriptie is niet alleen geschreven voor docenten, maar ook voor andere studenten en/of wetenschappers die iets willen onderzoeken op dit gebied. Dit schrijven zou in dat geval een mooie introductie kunnen zijn in het onderwerp.

Bij het project hebben wij veel hulp ontvangen de docenten: J.J. Reijmers, A.C.C. Lo, F.A. Kuipers, R. Hekmat en master studenten.

Wij hopen dat de lezers dit rapport interessant en vooral leerzaam zullen vinden.

Delft, 17.06.2007

M. Vasić

A.M. Abikhaled

Samenvatting

Algemeen

Er is een systeem ontworpen dat file-informatie kan vergaren. Dit systeem werkt met behulp van wifi ad-hoc netwerken. Het systeem is bedoeld om in elk toekomstig navigatiesysteem geïmplementeerd te worden. Een dergelijk navigatiesysteem heeft dan ook een GPS (globe positioning system) antenne aan boord.

Dit systeem is speciaal ontworpen voor de stad, wat betekend dat elke file wordt gedetecteerd in elke straat, mocht die aanwezig zijn.

Dit systeem kan echter pas goed functioneren, als er genoeg auto's rondrijden met het systeem aan boord. Daarom wordt het systeem pas in werking gesteld, als dat het geval is.

Systeemconcepten

Door verschillende concepten tegen elkaar af te wegen, is er uiteindelijk gekozen voor één concept.

Het *eerste concept* houdt in dat er een plaatsbepaling wordt uitgevoerd van verschillende auto's door middel van gsm-toestellen en gsm-masten. In elke auto wordt dan aangenomen dat er een gsm-toestel aanwezig is. Elk gsm-toestel op een wegstuk wat gedetecteerd wordt door gsm-masten, wordt dan automatisch als een auto gezien. Als er zich op een wegstuk meer dan X auto's bevinden, en deze langzamer rijden dan Y km/u, dan wordt er een besluit genomen dat daar een file is.

Het *tweede concept* houdt in dat een auto (de master) in zijn wifi-straal kijkt, hoeveel andere auto's zich daar bevinden. Van elke auto die zich in zijn wifi-straal bevindt, wordt de x- en y-coördinaat en snelheid opgevraagd. Aan de hand van deze gegevens bepaalt de master of zich daar werkelijk een file bevindt. Mocht dat het geval zijn, dan worden de begin- en eindcoördinaten en de gemiddelde snelheid van de file gebroadcast naar alle andere auto's via een wifi ad-hoc netwerk.

Het *definitieve concept* houdt in dat een auto (de master) in zijn wifi-straal kijkt, hoeveel andere auto's zich daar bevinden. Van elke auto die zich in zijn wifi-straal bevindt, wordt de x- en y-coördinaat en snelheid opgevraagd. Aan de hand van het aantal auto's, bepaalt de master of zich daar mogelijk een file bevindt. Mocht dat het geval zijn, dan worden de coördinaten, en snelheden van alle auto's gestuurd naar het basisstation (deze bevindt zich ongeveer in het midden van de stad) met behulp van wifi ad-hoc netwerken.

Het basisstation vergelijkt vervolgens deze gegevens met de coördinaten van de wegennetkaart. Van elke auto wordt er bepaald op welke weg en wegsoort deze zich bevindt. Aan de hand van het aantal auto's en de gemiddelde snelheid op een wegstuk, wordt er bepaald of zich daar een file bevindt. Mocht dat het geval zijn, dan worden de begin- en eindcoördinaten en de gemiddelde snelheid van de file gebroadcast naar alle andere auto's via een wifi ad-hoc netwerk.

Het eerste concept is ongeschikt bevonden, omdat de plaatsbepaling met behulp van gsm-masten veel te onnauwkeurig is, en daardoor het systeem niet naar behoren kan werken.

Het tweede concept is ongeschikt bevonden omdat de besluitneming of er een file is, door de master genomen wordt. Dit is heel lastig, aangezien de master geen gegevens heeft van de soort wegstukken waarop zich een mogelijke file bevindt.

Uiteindelijk is gekozen voor het derde concept. Hier heeft het basisstation alle gegevens over alle wegen, en kan zo heel nauwkeurig bepalen of er een file is.

Uitwerking systeem

Om het systeem te ontwerpen is er gebruik gemaakt van een simulatieprogramma QualNet, en een C-compiler.

QualNet is gebruikt om het verzenden van informatie te simuleren met behulp van wifi ad-hoc netwerken. Een C-compiler is gebruikt om een C-code te schrijven en vervolgens te compileren als een zelfstandig programma. Er zijn twee programma's gemaakt. Een voor de master, de andere voor het basisstation. In beide programma's staat hoe er een besluit wordt genomen of er (mogelijk) een file is.

Uitvoering project

In de eerste instantie was het de bedoeling om QualNet te laten communiceren met de C-programma's. Dit was echter moeilijk te realiseren binnen de 8 weken, dus is er besloten om deze onderdelen los te koppelen.

Vervolgens zijn er twee C-programma's ontworpen. Een voor de master, en een voor het basisstation. In plaats van dat de master de gegevens van de omringende auto's via QualNet binnenkrijgt, en deze vervolgens worden bewerkt door het programma, krijgt het programma deze gegevens nu binnen via een tekstbestand. In dit tekstbestand staan alle x- en y-coördinaten en snelheden van alle auto's.

Deze gegevens worden dan niet doorgestuurd van de master naar het basisstation via QualNet. Het programma van het basisstation krijgt nu een, door het programma van de master, nieuw aangemaakte tekstbestand binnen. Aan de hand hiervan bepaalt het programma van het basisstation of er daadwerkelijk een file is. Mocht dit het geval zijn, wordt er als output gewoon de begin- en eindcoördinaat en snelheid van de file gegeven.

Inhoudsopgave

	Voorwoord.....	2
	Samenvatting.....	3
1	Inleiding.....	6
2	Totstandkoming programma van eisen.....	7
	2.1 Programma van persoonlijke eisen.....	7
	2.2 Vertaling van persoonlijke eisen naar technische eisen.....	7
	2.3 Programma van technische eisen	8
3	Genereren van systeemconcepten.....	9
	3.2 Drie concepten om file-informatie te vergaren.....	9
	3.3 Keuze van concepten.....	10
	3.4 Functieblokschema van het systeem.....	11
4	Ontwerpen van het systeem.....	13
	4.1 Keuze simulatiemodellen.....	13
	4.2 Implementatie problemen in simulatiemodellen.....	14
	4.3 Resultaten van de simulatiemodellen.....	16
5	Testen van de simulaties.....	17
	5.1 Inleiding.....	17
	5.2 Resultaten van het testen.....	18
6	Discussie.....	20
7	Conclusie.....	21
	Geraadpleegde literatuur.....	22
	Bijlage A: C-code van de masternode.....	23
	Bijlage B: C-code van het basisstation.....	25
	Bijlage C: De scenario's.....	27
	Bijlage D: Wegennetkaart + coördinaten	31

1 Inleiding

Tegenwoordig is er steeds meer verkeer op de weg. Om snel van punt A naar punt B te komen wordt steeds lastiger door files die dan ontstaan. Voor veel zakenmensen vormt dit een groot probleem.

Sommige navigatiesystemen geven file-informatie weer voor de snelwegen, maar er bestaat nog geen systeem dat dit voor echt alle wegen doet, dus ook voor straten in steden.

Dit is vooral van belang als men in een grote stad genavigeerd wil worden. In een dergelijk geval moet een file-informatiesysteem voor alle wegen/straten werken.

Het doel is om een systeem te ontwerpen dat in de toekomst kan worden toegepast om file-informatie te vergaren. Dit systeem zou dan gebruik maken van wifi ad-hoc netwerken.

Wanneer het systeem ontworpen wordt, moet men eerst weten welk bestel het meest geschikt is om te worden gebruikt als file-informatiesysteem.

In dit verslag wordt duidelijk ingegaan op de belangrijkste aspecten, waarop gelet moet worden wanneer er een systeem wordt ontworpen.

Door middel van drie concepten wordt er naar een uiteindelijk systeem toegewerkt, dat door ons als beste wordt bevonden.

De opbouw van het verslag is als volgt:

In *hoofdstuk 2* wordt de totstandkoming van het programma van eisen besproken.

Hoofdstuk 3 zal ingaan op het genereren van oplossingen voor het probleem. Hierbij zal m.b.v. een functieblok-schema het systeem worden uitgelegd. In dit schema wordt dan de werking van het systeem duidelijk gemaakt. Uiteindelijk zal er een concept worden uitgekozen om uitgewerkt te worden.

In *hoofdstuk 4* wordt gekeken naar de simulatiemodellen die worden gebruikt om het systeem weer te geven. Hierin wordt de keuze voor bepaalde modellen gebruikt, de problemen die men tegenkomt worden besproken, en uiteraard worden de resultaten ook beoordeeld.

Hoofdstuk 5 borduurt hier verder op door. Hierin komt het testen aan de orde, en de resultaten hiervan, wat uiteindelijk zal leiden tot een goed beeld van het systeem.

2 Totstandkoming programma van eisen

2.1 Programma van persoonlijke eisen

Wanneer het systeem wordt ontworpen, moet het voldoen aan een aantal eisen van de gebruiker. Deze eisen kunnen verschillend zijn, maar ze laten wel zien wat de gebruiker verwacht van het systeem. Eisen zijn dus iets wat de gebruiker zelf wil van het systeem [Ten Haaf, 2002].

In dit geval gaat het om een file-informatie vergaringsysteem.

Eisen kun je onderverdelen in een aantal categorieën. Het beste is om 1 hoofdeis te gebruiken die zich relateert aan de uitgangspunten

Voor het file-informatie vergaringsysteem ziet dat er als volgt uit:

Hoofdeis

-Als ik in de auto zit, wil ik via mijn navigatiesysteem nauwkeurige up-to-date file-informatie binnenkrijgen.

Uitgangspunten

-Ik wil elke 5 minuten file-informatie binnenkrijgen. (1)

-Ik wil dat de positie wordt bepaald m.b.v. GPS. (2)

-Ik wil dat de file-informatie wordt verstuurd via wifi ad-hoc netwerken. (3)

-Ik wil het volgende in de informatie aantreffen; waar de file zich bevindt, hoe lang de file is, en hoe snel de file zich voortbeweegt. (4)

-Ik wil dat mijn anonimiteit gewaarborgd blijft. (5)

Als deze hoofdeis en uitgangspunten worden bekeken, wordt er ook een idee gevormd wat de gebruiker daadwerkelijk verwacht van het systeem.

Nu is het zaak om deze hoofdeis en uitgangspunten te vertalen naar de technische eisen die aan het systeem worden gesteld.

Hier wordt in de volgende paragraaf (2.2.)op ingegaan.

2.2 Vertaling van persoonlijke eisen naar technische eisen

Hierboven staat beschreven wat de persoonlijke eisen zijn van de klant. Nu wordt toegelicht hoe deze persoonlijke eisen vertaald kunnen worden naar de technische eisen van het systeem.

De uiteindelijke bedoeling is dat de technische eisen van het systeem, de persoonlijke eisen van de gebruiker representeren.

De beste manier om dit te doen is om goed te kijken naar de persoonlijke eisen van de gebruiker, en te zien hoe een persoonlijke eis vertaald kan worden in een technische.

Allereerst wordt er begonnen met het maken van een tabel die tracht globaal deze vertaling te visualiseren. Bij de persoonlijke eisen uit de tabel is d.m.v. een cijfer meteen aangegeven bij welke uitgangspunt ze horen.

Tabel 1 wordt hieronder weergegeven:

Tabel 1. Globale omzet van persoonlijke eisen naar technische eisen

Persoonlijke eisen	Technische eisen
Snelheid (1), (3)	Algoritme moet snel genoeg zijn met het verzamelen van gegevens
Nauwkeurigheid (2), (4)	Coördinaten van de file - nauwkeurig
Anonimiteit (5)	Het systeem mag onderscheid maken tussen auto's, maar geen identiteit eraan toekennen

In de tabel hierboven is deze vertaling globaal weergegeven. En de technische eisen zijn dan ook globaal geformuleerd. In de volgende paragraaf zal de uitwerking aan de orde komen van de eisen, wat vervolgens in een programma van eisen zal resulteren.

2.3 Programma van technische eisen

Het programma van eisen heeft als functie communicatie tussen iemand met een probleem en probleemoplosser mogelijk te maken. Het is bedoeld om een productdefinitie op te stellen aan welk degene met probleem zich kan committeren. Hierbij hoort een hoofdfunctie die moet voldoen aan een aantal randvoorwaarden. Deze komen voort uit de persoonlijke eisen die in paragraaf (2.2) zijn weergegeven.

De hoofdfunctie en randvoorwaarden worden hieronder weergegeven.

*Als **hoofdfunctie** hebben we verkregen:*

File-informatie verzamelen m.b.v. gps, en wifi adhoc-netwerken

*Als **randvoorwaarden** hebben we verkregen:*

Gebruik

- Het systeem moet om de 5 minuten nauwkeurige file-informatie kunnen geven: het basis station heeft maximaal vijf minuten om de data te verwerken en de file-informatie naar de andere auto's te versturen.
- De file-informatie moet uit 3 gegevens bestaan:
 - begin coördinaat van de file;
 - eind coördinaat van de file ;
 - snelheid van de file.
- Iedereen die met het systeem meedoet, moet de file-informatie kunnen krijgen, ook diegene die in de file zitten
- Het systeem moet in een bestaand navigatiesysteem kunnen werken.
- Het systeem moet ongeveer 5 jaar meekunnen.
- Systeem moet betrouwbaar zijn/ bedrijfszeker.
- Besluit of er een file plaatsvindt, wordt genomen door het basisstation.
- De maximale afstand dat de auto's gegevens naar elkaar kunnen sturen bedraagt 100 meter.

Veiligheid

- Iedereen mag de file-informatie binnenkrijgen, maar hoeft het niet per se te gebruiken.
- Iedereen mag zelf weten of hij aan het systeem meedoet, als ad-hoc node.

Ontwerpfase

- Moet in vier weken gemaakt kunnen worden.
- Er moet een gebruikersvoorschrift geschreven worden.

Hergebruik

- Recyclebaar in zoverre dat kan.

Met behulp van deze PvE zal het systeem dan ook ontworpen worden.

3 Genereren van systeemconcepten

Nu het programma van eisen bekend is, is het de bedoeling om deze om te zetten in systeemconcepten. Uiteindelijk zijn er drie systeemconcepten ontworpen. Deze concepten zijn tegen elkaar afgewogen, en vervolgens is de beste daarvan gekozen om compleet uitgewerkt te worden tot een werkend systeem.

3.1 Drie concepten om file-informatie te vergaren

Concept 1: gsm-masten

Bij dit concept is het de bedoeling dat de file-informatie word vergaard m.b.v. gsm-masten, en gsm toestellen. De posities van de gsm-toestellen worden in een zodanig geval bepaald door de gsm-masten . Vervolgens wordt er aan de hand van die posities bepaald of het om een auto gaat. Dit gebeurt door te kijken of de coördinaten zich op de weg bevinden of niet.

In een drukke stad lijdt dit hoogstwaarschijnlijk tot grote problemen. Ten eerste omdat de posities van de gsm toestellen heel onnauwkeurig worden bepaald, en ten tweede omdat er in een stad ook voetgangers op de straat kunnen lopen. In dat geval worden deze gezien als automobilisten. Je kunt er dan geen onderscheid meer tussen maken.

Concept 2: ad-hoc wifi (besluitvorming in de auto's)

Bij dit concept is het de bedoeling dat elke auto met het ingebouwde systeem, via wifi om zich heen kijkt, en kijkt hoeveel auto's zich in zijn wifi straal bevinden. (Er wordt ook aangenomen dat er heel veel auto's rondrijden). Als dit aantal groot is, dan wordt aan de hand van de gegevens (aantal auto's, coördinaten van de auto's, snelheid van de auto's) een besluit genomen of het om een file gaat. Mocht dit het geval zijn, dan wordt deze informatie verstuurd via een ad-hoc netwerk naar een wifi basisstation. Dit basisstation broadcast vervolgens via ad-hoc wijze deze informatie naar alle auto's.

Concept 3: ad-hoc wifi (besluitvorming in het basisstation)

Bij dit concept is het de bedoeling dat een auto die mogelijk een file waarneemt (als er zich meer dan X auto's in zijn straal bevinden), deze informatie doorzendt naar het basisstation via een ad-hoc netwerk. Het basisstation bepaalt aan de hand van die informatie op wat voor soort weg deze auto's zich bevinden, en of ze te langzaam rijden voor die weg. Mocht dit allemaal het geval zijn, dan neemt het basisstation het besluit dat het om een file gaat, en vervolgens wordt deze informatie naar alle auto's gebroadcast.

3.2 Keuze van concepten

Wat **systeemconcept 1** betreft zijn er twee van de dertien eisen (zie paragraaf 2.3) die zeker niet gehaald kunnen worden:

-Het basisstation heeft maximaal vijf minuten om de data te verwerken en naar andere auto's te versturen.

-Het besluit of er werkelijk een file is, wordt genomen door het basisstation.

Bij systeemconcept 1 is het onmogelijk om binnen 5 minuten de data te verwerken, aangezien de posities van al die gsm-toestellen bepaald moeten worden die ook nog eens heel onnauwkeurig zijn. Vervolgens moet er bepaald worden of het al dan niet om een automobilist gaat. De informatie die je binnenkrijgt is veel te onnauwkeurig en veel te onbetrouwbaar om deze binnen 5 minuten te kunnen verwerken, en aan te geven of er een file is.

En het besluit of er file is wordt in dit geval genomen door de auto zelf, wat onze eis tegenspreekt. Dit systeemconcept is dus verworpen.

Wat **systeemconcept 2** betreft is er zeker één van de dertien eisen (zie paragraaf 2.3) die niet gehaald kan worden:

-Het besluit of er werkelijk een file plaatsvindt, wordt genomen door het basis station.

Bij systeemconcept 2 wordt de uiteindelijke besluitvorming of er een file plaatsvindt gemaakt door de auto zelf. Dit spreekt onze eis tegen dat het in het basisstation moet gebeuren.

Dit is van belang omdat het niet de bedoeling is dat elke auto voor zich gaat bepalen of die in een file zit. Dit vereist teveel onnodige rekenkracht en energie. Daarom is er besloten dat dit bij het basisstation moet gebeuren. Het basisstation verzamelt alle gegevens, en aan de hand van die gegevens wordt er een besluit genomen of er ergens een file is.

Uiteindelijk is dus ook dit systeemconcept verworpen.

Wat **systeemconcept 3** betreft kunnen alle eisen gehaald worden.

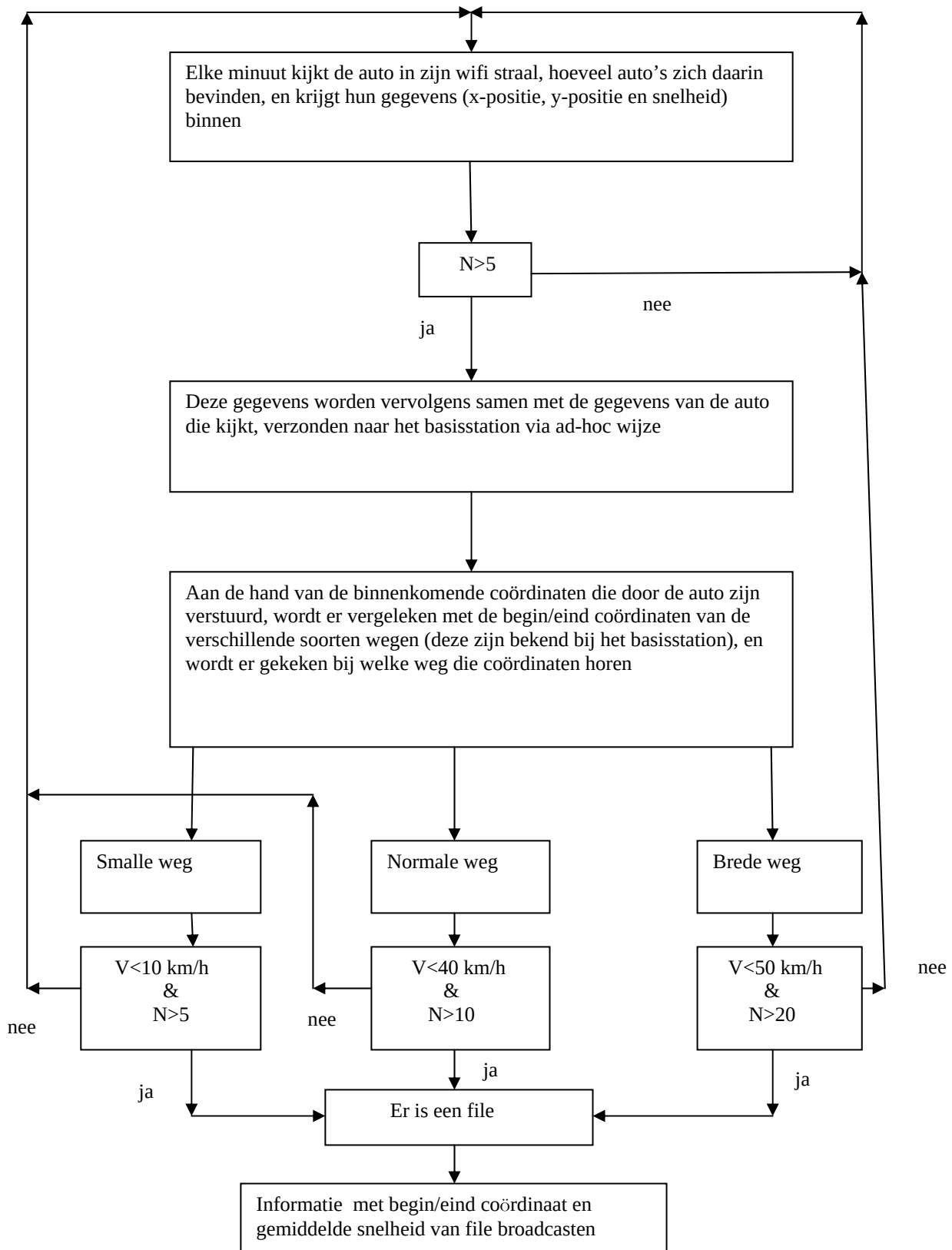
Omdat het systeem gebruik maakt van nauwkeurige positionering, en wifi ad-hoc netwerken, kunnen de gegevens binnen de 5 minuten bewerkt worden.

In dit concept wordt ook het besluit of er een file is, genomen door het basisstation.

Dit betekent dat systeemconcept 3 uiteindelijk is gekozen als systeem om daadwerkelijk uitgewerkt te worden.

3.3 Functieblokschema van het file-informatie vergaringsysteem

Een functieblokschema is een representatie van een systeem. In zo'n schema worden de ingangskkenmerken door middel van een enkele functie met de uitgangskkenmerken verbonden. De ingangskkenmerken geven een overzicht van eigenschappen in de ongewenste situatie, de uitgangskkenmerken bevatten de eigenschappen van de gewenste situatie of doelsituatie. Hieronder wordt het functieblokschema weergegeven van het file-informatie vergaringsysteem.



Figuur 1. Functieblokschema van het systeem

In het schema hierboven is in stappen laten zien hoe het file-informatie vergaringsysteem werkt. Als eerste worden de gegevens ingelezen van de omringende auto's. Als dat aantal groter is dan 5, dan worden die gegevens met de gegevens van de master gestuurd naar het basisstation. Het basisstation vergelijkt deze gegevens met zijn eigen gegevens, en kijkt op welke weg ze horen. Daarna wordt er bekeken of de gegevens aan de criteria voldoen, zodat er een file wordt gedetecteerd. Als dit het geval is, dan worden de benodigde gegevens vanuit het basisstation naar alle auto's gebroadcast.

4 Ontwerpen van het systeem

4.1 Keuze simulatiemodellen

Aangezien er niet genoeg tijd was om het product daadwerkelijk te maken, is er besloten om een simulatiemodel te gebruiken.

Bij de keuze van een simulatiemodel moet rekening gehouden worden met een aantal zaken:

- het simulatiemodel moet alle aspecten van je systeem kunnen laten zien;
- het simulatiemodel moet snel genoeg zijn;
- het simulatiemodel moet vrij makkelijk implementeerbaar zijn in een simulatieprogramma.

Bij de keuze van een simulatieprogramma moet rekening gehouden worden met het volgende:

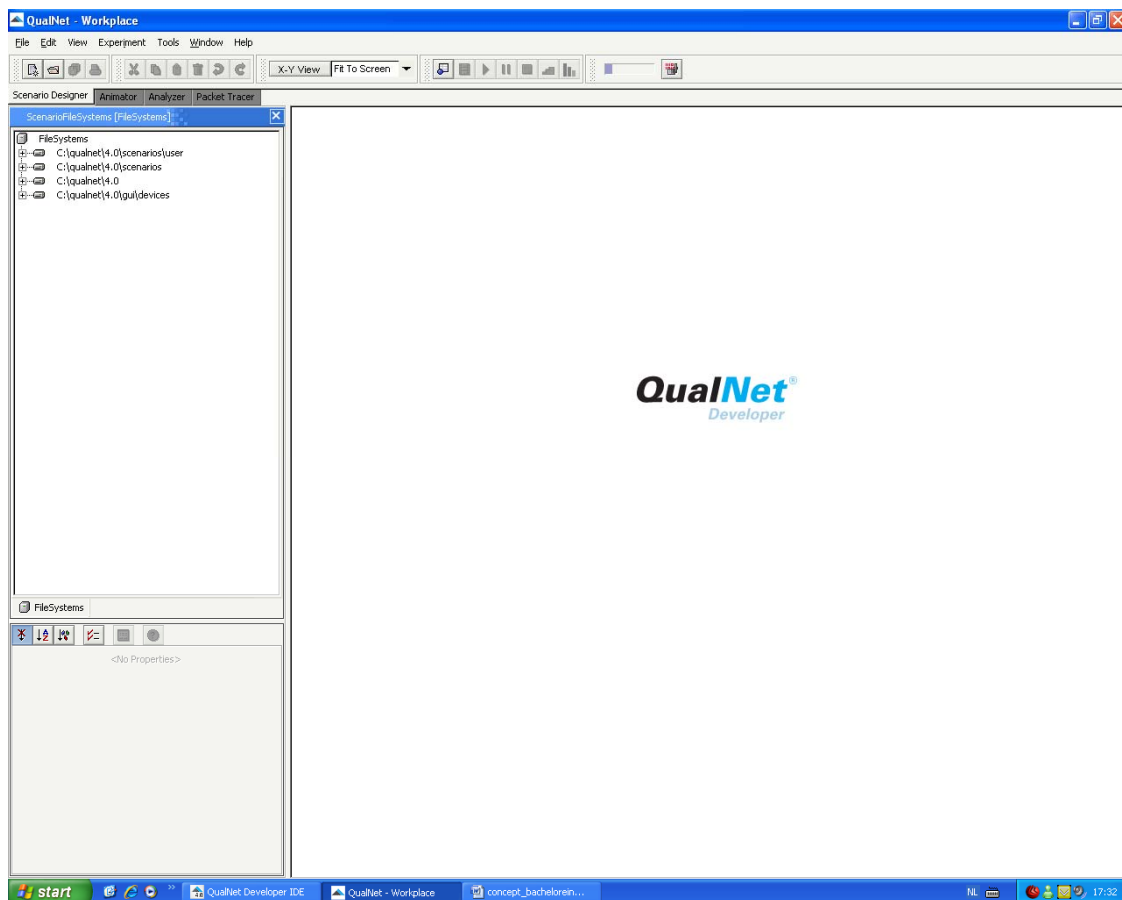
- het simulatieprogramma moet snel te begrijpen zijn;
- het complete simulatiemodel moet in het simulatieprogramma te maken zijn;
- het simulatieprogramma moet beschikbaar zijn.

Uiteindelijk is gekozen om het programma QualNet te gebruiken. QualNet is een netwerksimulatieprogramma dat vrij makkelijk in gebruik is, en waarin het gekozen simulatiemodel kan worden gemaakt (zie figuur 2).

Voor de besluitneming of er een file is, is een C-programma gemaakt (zie bijlage A en B).

Het simulatiemodel dat wordt ontworpen bootst het complete systeem na. Dus vanaf een auto die om zich heen kijkt, tot het broadcasten van de gegevens door het basisstation.

Echter is de koppeling tussen de C-programma en QualNet achterwege gelaten. Dit omdat er simpelweg geen tijd was om dat binnen de 8 weken te realiseren.



Figuur 2. De interface van QualNet.

4.2 Implementatieproblemen in simulatiemodellen

Als eerste moet de probleemoplossing geïmplementeerd worden in een simulatiemodel.

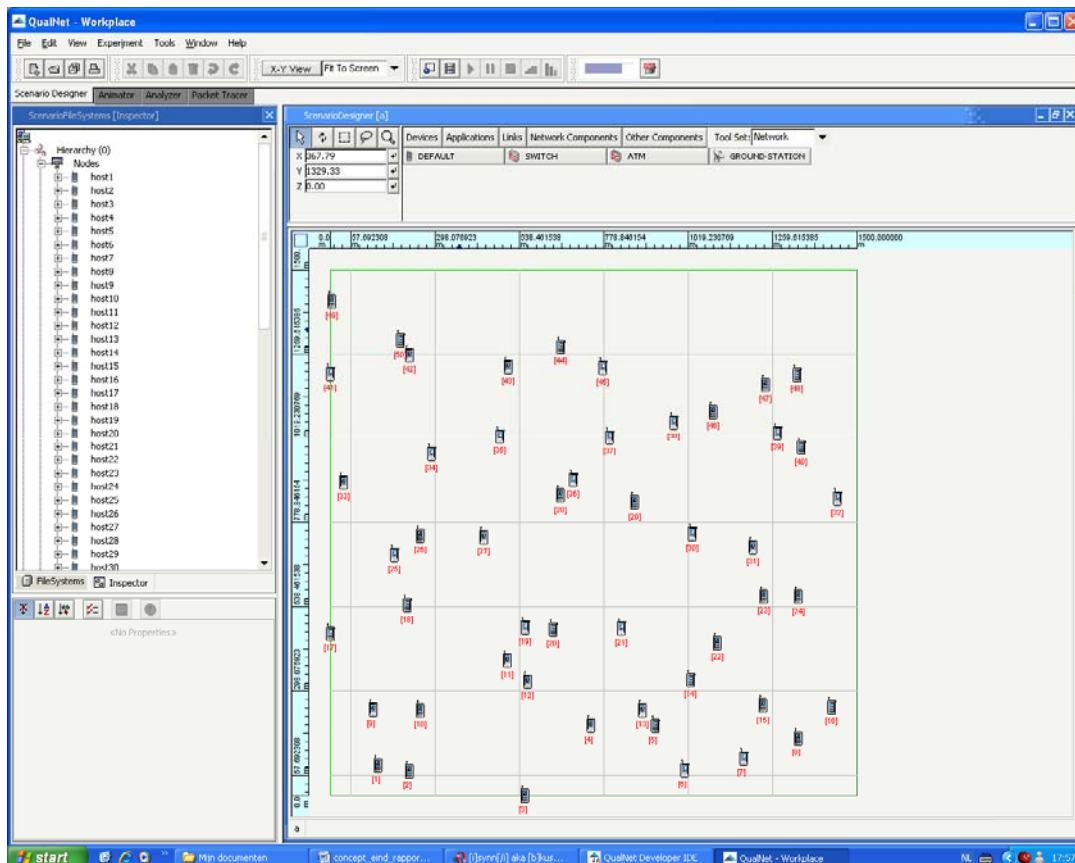
Het gene dat eerst geïmplementeerd wordt is een file. Om dat voor elkaar te krijgen zijn er in een gebied 50 wifi nodes geplaatst, die allemaal een auto vertegenwoordigen met het ingebouwde systeem (zie figuur 3).

Vervolgens moeten die nodes bewegen, zodat er op een gegeven moment een klein gebiedje ontstaat, met veel nodes erin (zie figuur 4). Dit zou een file kunnen zijn. Een van die nodes kijkt dan om zich heen, en neemt waar hoeveel nodes er zich bevinden in haar directe omgeving. Deze node wordt nu de *master* genoemd. Van elke node ontvangt de master een drietal gegevens. De x-positie, de y-positie, en de snelheid. Ook wordt het aantal nodes dat is gezien, vastgelegd. Deze procedure wordt beschreven in een C programma wat in de eerste instantie gekoppeld zou worden aan de master.

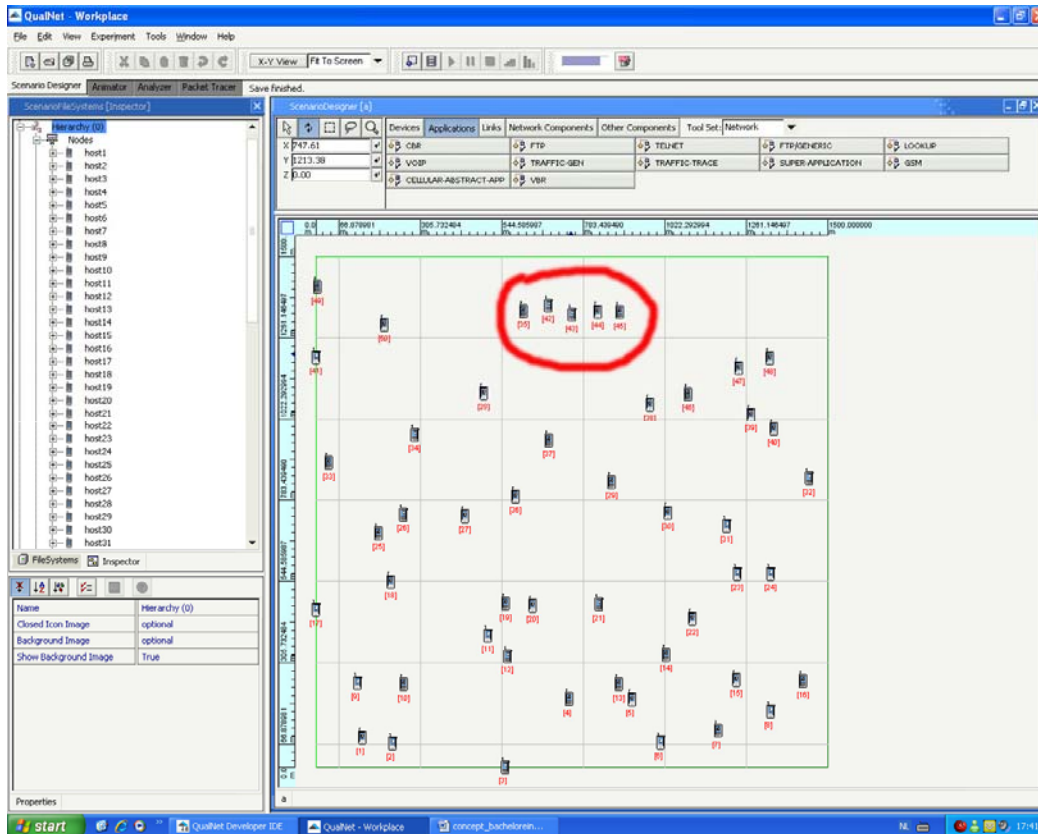
In geval dat de master meer dan een X aantal auto's om zich heen ziet, moet een lijst van de auto's met de posities en de snelheden naar een node in het midden van het gebied worden verstuurd (zie figuur 5). Deze node wordt nu het *basisstation* genoemd. Wanneer de gegevens binnenkomen, moet er besloten worden of het echt om een file gaat. Dit gebeurt door te kijken naar de coördinaten van de auto's en deze te vergelijken met de wegenkaart (zie bijlage D), die het basisstation kan gebruiken. Als er zich meer dan X auto's op een bepaald wegdek bevinden, en ze langzamer rijden dan Y km/u, dan moet het besluit worden genomen dat zich op dat wegdeel een file bevindt. Vervolgens wordt door het basisstation een bestand aangemaakt die het begin, het einde, en de gemiddelde snelheid van de file weergeeft.

Dit bestand wordt vervolgens gebroadcast naar alle auto's die meedoen aan het systeem, zodat zij weten waar de file zich bevindt, en het navigatiesysteem dat samenwerkt met ons file-informatie vergaarsysteem een snellere route kan berekenen.

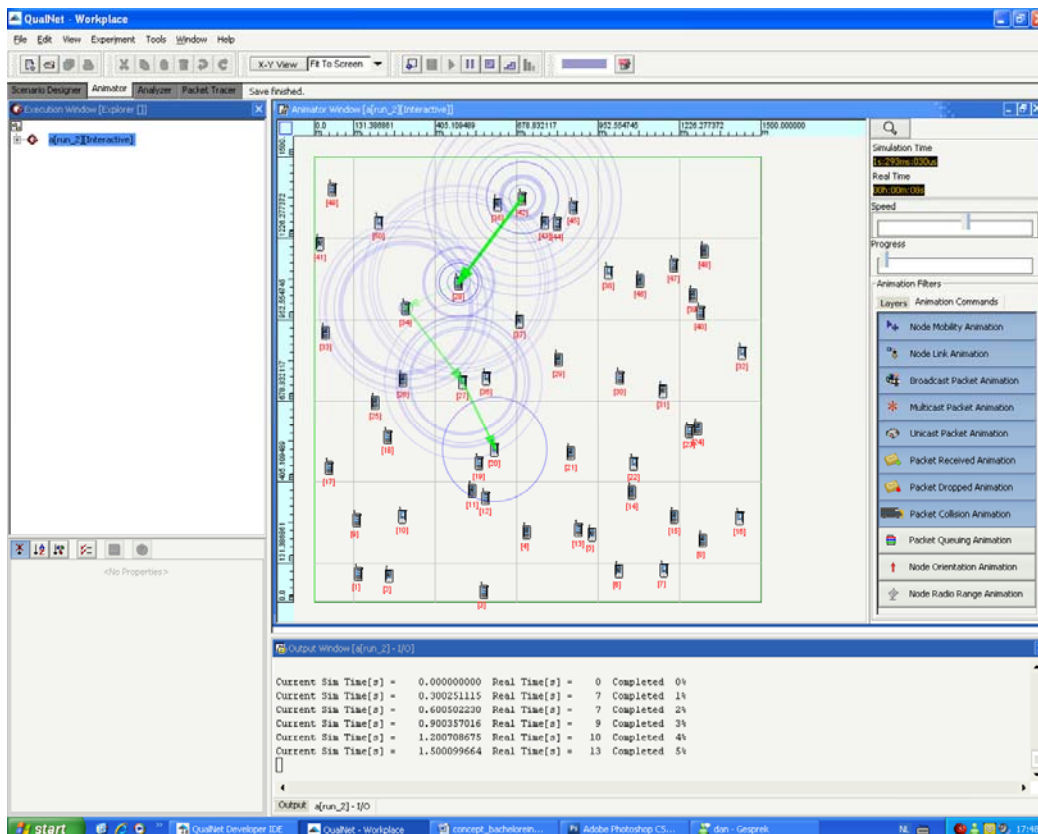
Het hele systeem staat beschreven in het functie blokschema (zie figuur 1).



Figuur 3. Een gebied met 50 nodes uniform verdeeld.



Figuur 4. De nodes zijn een beetje verschoven, en in het omringde gedeelte is er mogelijk een file ontstaan.



Figuur 5. Hier worden de gegevens van de master naar een node in het midden verstuurd via ad-hoc wifi.

4.3 Resultaten van de simulatiemodellen

Zoals hierboven vermeldt, zijn de onderdelen van het C-programma en QualNet losgekoppeld. Dit omdat het te veel tijd in beslag neemt om ze te laten communiceren. QualNet is een goed programma om netwerken mee te simuleren, maar niet echt geschikt om er een heel fileinformatie vergaringsysteem mee te simuleren.

Omdat het C-programma en QualNet losgekoppeld zijn, zullen de resultaten dus ook los besproken worden.

QualNet

Het simulatieprogramma QualNet wordt dus vooral gebruikt om te laten zien hoe informatie verzonden kan worden door middel van wifi ad-hoc netwerken.

Als men informatie wil verzenden van punt A naar punt B, dan ziet men dat er een routerings-algoritme wordt gebruikt, om te bepalen waar alle nodes zich bevinden. Daarna wordt de informatie pas verstuurd. Dit gebeurt vrij snel, want een route is al binnen 1 seconde gevonden.

C-programma (master)

Waar de gegevens normaal via QualNet binnen zouden komen, komen ze nu uit een tekstbestand. Dit zijn verschillende scenario's (zie bijlage C) van verschillende verkeerssituaties. In het tekstbestand zijn de begincoördinaten, eindcoördinaten, en snelheden opgenomen van alle auto's die zich in zijn wifi straal bevinden. Aan de hand van dit tekstbestand wordt er bepaald of er genoeg auto's zijn, zodat er mogelijk een file is ontstaan. Als dit het geval is, dan wordt er een nieuw tekstbestand aangemaakt dat identiek is aan het eerste, alleen zitten er nu op de eerste regel, gegevens van de master. Deze horen er immers ook bij, omdat de master zelf ook mogelijk in een file zit.

C-programma (basisstation)

Het tekstbestand wordt dan door het basisstation gebruikt om het te vergelijken met de gegevens van de wegnetaart (zie bijlage D) om te bepalen op welke soort weg de auto's zich bevinden, in welke richting de auto's rijden (op welke rijstrook?), en of ze een file vormen.

Op de wegnetaart zijn alle wegen opgedeeld in wegstukken. Elk wegstuk kan klein, normaal, of breed zijn, en heeft een begin- en eindcoördinaat. In het tekstbestand van de master zitten de coördinaten van de auto's. Bij al deze coördinaten wordt er gekeken of ze tussen die begin- en eindcoördinaten van zo'n wegstuk zitten.

Als dit het geval is, dan wordt er een nieuwe file aangemaakt met de begincoördinaat, eindcoördinaat en gemiddelde snelheid van de file.

5 Testen van de simulaties

5.1 Inleiding

Bij het testen van het systeem in QualNet moet eerst gekeken worden naar wat voor soort data er verstuurd kunnen worden. Er kan een CBR (constant bit rate), VBR (variable bit rate), FTP (file transfer protocol) gebruikt worden, en nog andere types.

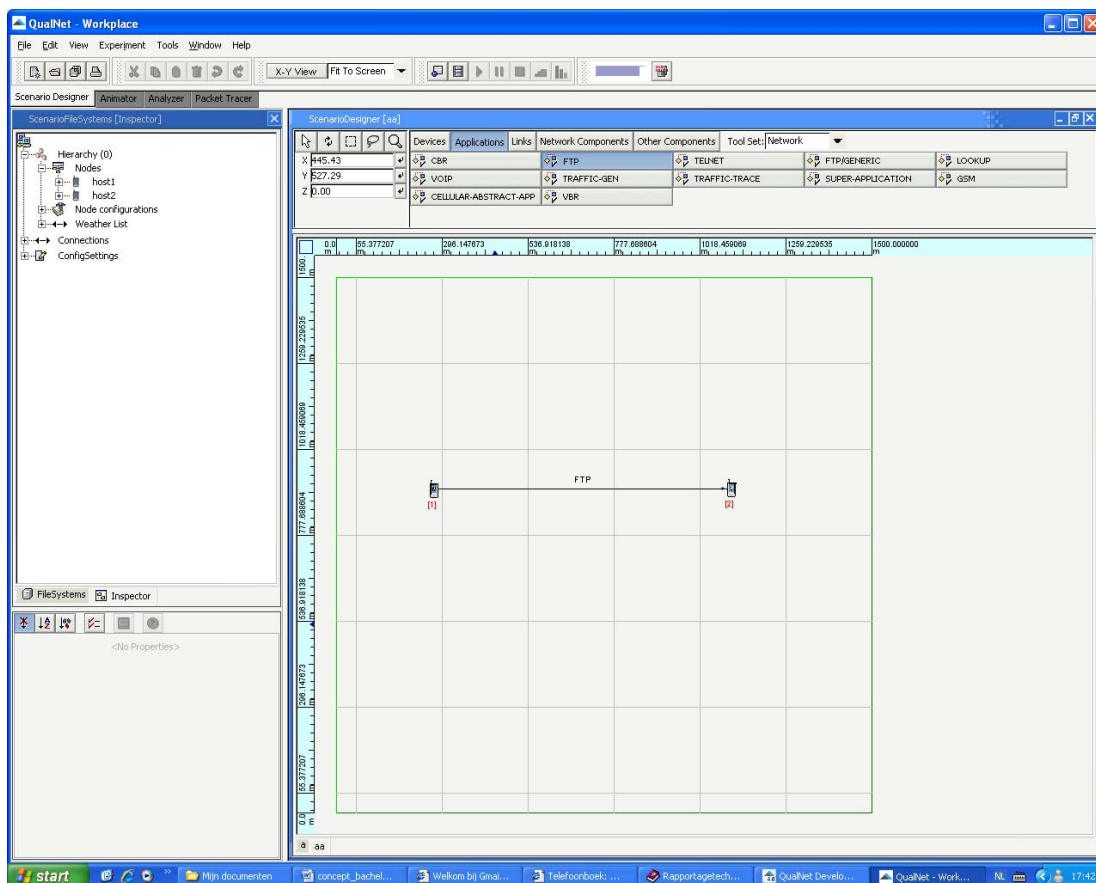
De hierboven genoemde type bestanden hebben allemaal hun voor- en nadelen. CBR en VBR kunnen heel goed gebruikt worden bij applicaties waarbij een bepaalde kwaliteit gegarandeerd moet worden, zoals bij het verzenden van videobestanden. Omdat bij dit systeem een simpel bestand wordt overgezonden, en daar geen rekening mee gehouden hoeft te worden, is er besloten om met FTP te werken.

Het eerste wat je doet is een FTP verbinding opzetten tussen twee nodes, en kijken wat er gebeurt (zie figuur 6).

Ten tweede kun je de nodes laten bewegen, en kijken hoe zij zich gedragen.

Bij het testen van het systeem is het van belang dat je alle mogelijkheden langsloopt.

De hierboven genoemde onderdelen zijn allemaal uitvoerig getest.



Figuur 6. Een FTP verbinding tussen twee nodes.

5.2 Resultaten van het testen

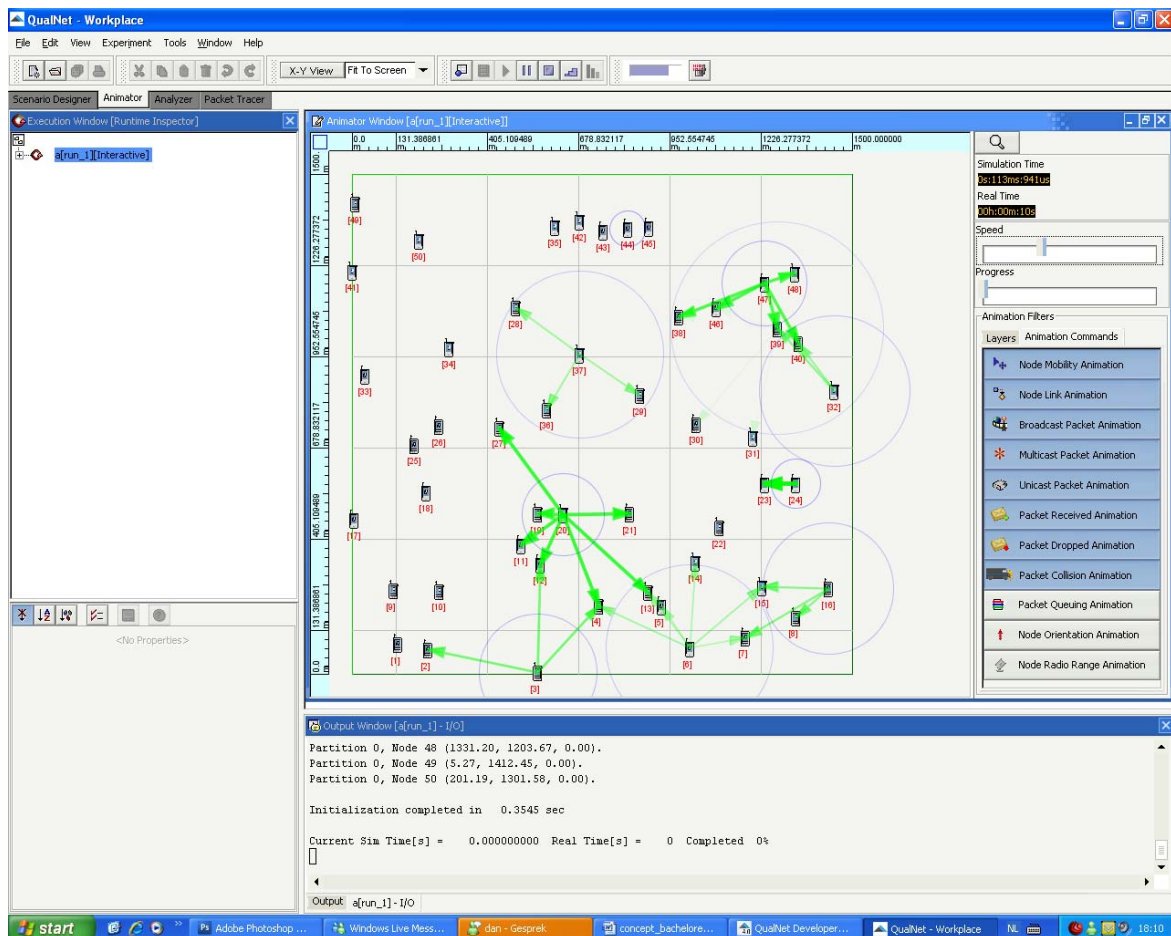
Bij het testen van een verbinding tussen twee nodes zijn er een aantal zaken naar voren gekomen. Om van node A naar node B te komen via andere nodes in een ad-hoc netwerk, heb je een routeringsprotocol nodig om de weg te vinden. Hierbij is gebruikgemaakt van de AODV (ad-hoc on demand vector routing) protocol. Dit protocol is van de beschikbare protocollen in QualNet het meest geschikt bevonden voor de routing in het systeem [Aggleou, 2005].

Wat je meteen opmerkt bij het simuleren van de verbinding, is dat eerst alle nodes om zich heen gaan kijken, en met elkaar gaan communiceren (zie figuur 7). Op deze manier wordt er gekeken hoe alle nodes ten opzichte van elkaar zijn gepositioneerd. Zodra dit is gevonden, wordt de verbinding gelegd tussen node A en node B.

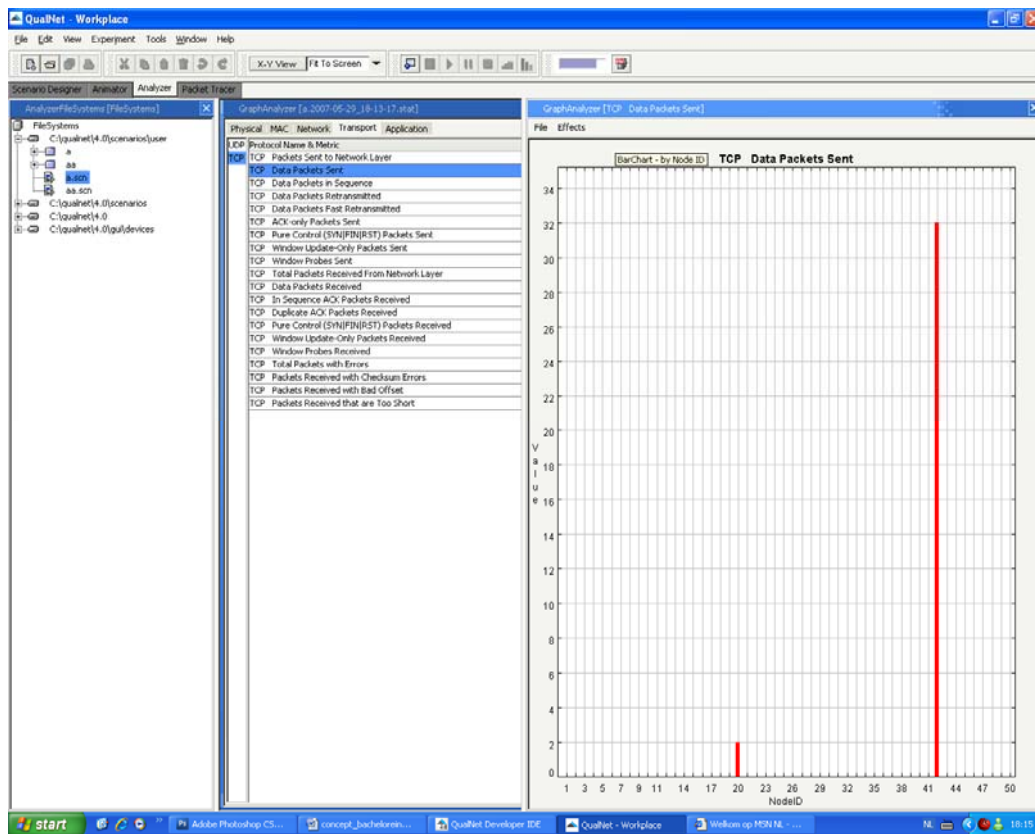
Bij routeringsprotocollen, wordt er gebruikt gemaakt van UDP (User Datagram Protocol), en bij het versturen van echte data wordt er gebruik gemaakt van TCP (Transmission Control Protocol).

Bij het analyseren van de verzonden bestanden kun je zien dat er alleen bij de nodes A en B een TCP overdracht is geweest. Dit houdt in dat er echte data overdracht is geweest. Bij de verzonden data zien we dat het grotendeels bij node 42 is gebeurt, en bij de ontvangen data grotendeels bij node 20. En dit klopt dan ook met onze verwachtingen. (Zie figuur 7. en figuur 8.)

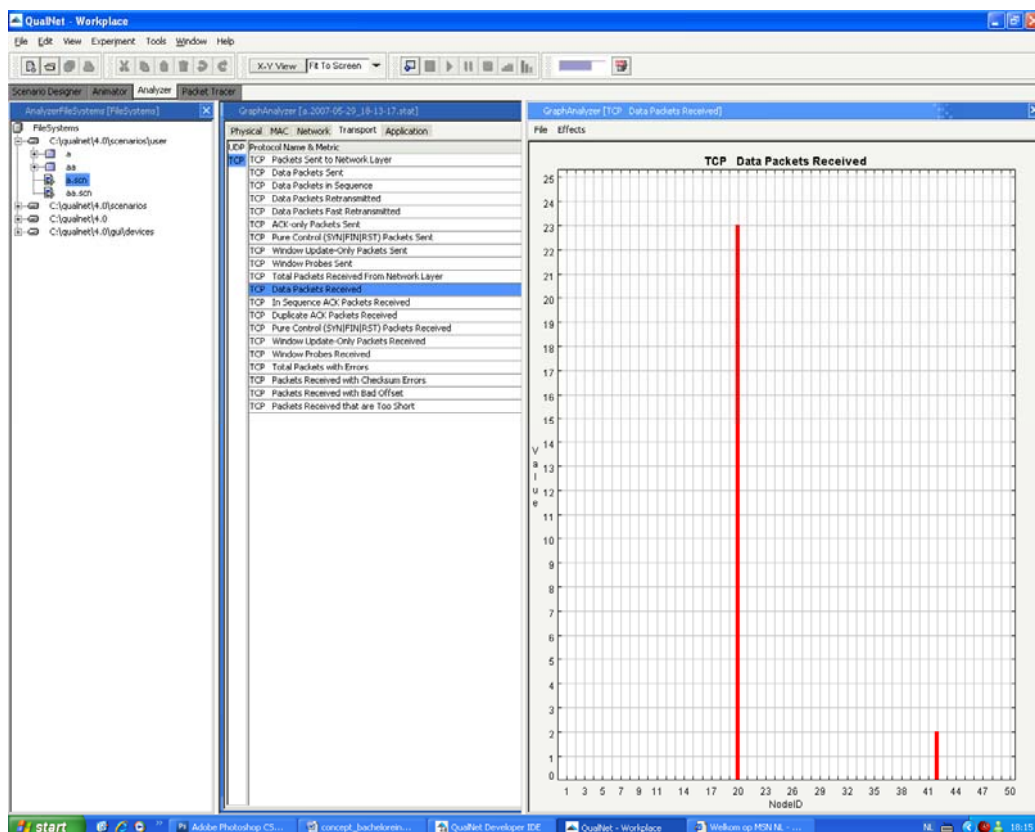
Het laten bewegen van de nodes is ook vrij makkelijk gegaan. Het is gewoon mogelijk om in QualNet een route uit te stippelen hoe de nodes zich moeten bewegen, en met welke snelheid. Bij het simuleren zijn er geen verrassingen naar boven gekomen.



Figuur 7. Hier wordt weergegeven hoe meerdere nodes data verzenden naar andere nodes. Dit wordt weergegeven met groene pijltjes.



Figuur 8. Hier zijn de TCP data pakketten weergegeven die zijn verzonden vanaf node 42.



Figuur 9. Hier zijn de TCP data pakketten weergegeven die zijn ontvangen bij node 20.

6 Discussie

De opdracht die we ons zelf gegeven hebben aan het begin van dit project, was om een systeem te ontwerpen dat slim, file-informatie kon vergaren. Om dit te laten zien wilden wij een simulatieprogramma maken. Alles zou in principe via dat simulatieprogramma moet worden laten zien.

Uiteindelijk kwamen erachter dat er niet genoeg tijd was om een koppeling te maken tussen het C-programma en QualNet. Daarom is er uiteindelijk besloten om deze twee onderdelen apart te vervolmaken. Het systeem is dus in principe hetzelfde gebleven, alleen de wijze waarop het gedemonstreerd wordt verschilt.

7 Conclusie

Het doel was om een systeem te ontwerpen dat file-informatie kan vergaren met behulp van wifi ad-hoc netwerken.

Na drie systeemconcepten te hebben overwogen, is er uiteindelijk besloten het concept uit te werken dat gebruik maakt van wifi ad-hoc netwerken, en waar het besluit wordt gemaakt of er een file is, in het basisstation plaatsvindt.

Dat het middels wifi ad-hoc tot stand is gekomen, heeft te maken met het feit dat zo'n systeem heel snel werkt en heel snel informatie kan versturen van het ene naar het andere punt. En dit aspect is essentieel in ons systeem. Met behulp van GPS kan de positie in elk navigatiesysteem worden bepaald. Als men van alle auto's de positie en snelheid kan weten, dan kan er ook een besluit worden genomen of er een file is.

Na elk tijdsinterval kijkt een auto (de master), hoeveel auto's zich in zijn wifi straal bevinden. Van al deze auto's vraagt de master de positie en snelheid op. Aan de hand van het aantal auto's neemt de master een besluit of zich mogelijk een file voordoet. Als dit het geval is, dan worden die gegevens samen met zijn eigen gegevens verstuurd naar het basisstation. Deze bevindt zich hoogstwaarschijnlijk in het midden van de stad. Deze vergelijkt deze informatie met zijn eigen gegevens, en neemt vervolgens een besluit of er een file is, en waar deze te vinden is.

De output wordt de begincoördinaat, eindcoördinaat en gemiddelde snelheid van de file.

De werking van de master en van het basisstation is gesimuleerd met behulp van twee C-programma's. Het transport van de gegevens is gesimuleerd met behulp van het netwerksimulatieprogramma QualNet.

Wij kunnen hierdoor stellen dat het ons is gelukt om een dergelijk systeem te ontwerpen.

Geraadpleegde literatuur

Boeken

- Haaf, W ten e.a., *Fundamentals of business engineering and management*, Delft University press, 2002
- Aggleou, George e.a., *Mobile Ad Hoc Networks, From Wireless LANs to 4G Networks*, McGraw-Hill, 2005
- Hekmat, Ramin, *Fundamental Properties of Wireless Mobile Ad-hoc Networks*, proefschrift
- Perkins, Charles, *Ad Hoc Networking*
- Toh, C-K, *Ad Hoc Mobile Wireless Networks, Protocol and Systems*

Handleidingen

- QualNet 4.0 Product Tour
- QualNet 4.0 Users Guide
- QualNet 4.0 Programmers Guide

Bijlage A: C-code van de masternode

```
/* Author :Andre Michel Abikhalel
* Study# :1198653
* Goal : making a primary decision whether there is a traffic or not
*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define FILENAME "data.txt"

void print_message(void);
void analyze_traffic_jam(int number);

main()
{
    int k;
    FILE *file_in, *file_in1, *file_in2, *file_out;
    int ch;
    int x[5], y[5], v[5];
    int x1[5], y1[5], v1[5];
    int x2[5], y2[5], v2[5];
    file_in = fopen("data.txt", "r");
    file_in1 = fopen("data1.txt", "r");
    file_in2 = fopen("data2.txt", "r");
    file_out = fopen("output.txt", "w");
    /* printf("print a character and hit enter%c\n", ch); */
    ch = getchar();
    /* printf("%d", ch); */
    if(ch == 97)
    {
        for(k=0; k<5; k++)
        {
            fscanf(file_in, "%d %d %d", &x[k], &y[k], &v[k]);
            /* printf("Scan is working\n"); */
            /* printf("%d %d %d \n", x[k], y[k], v[k]); */

        }
        fclose(file_in);
        analyze_traffic_jam(k);
        for(k=0; k<5; k++)
        {
            fscanf(file_out, "%d %d %d", &x[k], &y[k], &v[k]);
            /* printf("These are the vaules : %d %d %d\n", x[k], y[k], v[k]); */
            fprintf(file_out, "%d %d %d\n", x[k], y[k], v[k]);
        }
        fclose(file_out);
    }
    if(ch == 98)
    {
        for(k=0; k<5; k++)
        {
            fscanf(file_in1, "%d %d %d", &x1[k], &y1[k], &v1[k]);
        }
        fclose(file_in1);
        analyze_traffic_jam(k);
        for(k=0; k<5; k++)
        {
            fscanf(file_out, "%d %d %d", &x1[k], &y1[k], &v1[k]);
            fprintf(file_out, "%d %d %d\n", x1[k], y1[k], v1[k]);
        }
    }
}
```

```

        }
        fclose(file_out);
    }
    if(ch == 99)
    {
        for(k=0; k<5; k++)
        {
            fscanf(file_in2, "%d %d %d", &x2[k], &y2[k], &v2[k]);
        }
        fclose(file_in2);
        analyze_traffic_jam(k);
        for(k=0; k<5; k++)
        {
            fscanf(file_out, "%d %d %d", &x2[k], &y2[k], &v2[k]);
            fprintf(file_out, "%d %d %d\n", x2[k], y2[k], v2[k]);
        }
        fclose(file_out);
        if(ch == 99)
    {
        for(k=0; k<5; k++)
        {
            fscanf(file_in2, "%d %d %d", &x2[k], &y2[k], &v2[k]);
        }
        fclose(file_in2);
        analyze_traffic_jam(k);
        for(k=0; k<5; k++)
        {
            fscanf(file_out, "%d %d %d", &x2[k], &y2[k], &v2[k]);
            fprintf(file_out, "%d %d %d\n", x2[k], y2[k], v2[k]);
        }
        fclose(file_out);
    }
}

/*
 * Action : prints a message
 */

void print_message(void)
{
    printf("There is a chance for traffic jam \n");
}

/*
 * Action : The master node analyzes how
 *          many nodes are there around
 *          its small area and then decides
 *          whether there is a chance of
 *          having a traffic jam or not.
 */
void analyze_traffic_jam(int number)
{
    if (number >= 5)
        print_message();
}

```

Bijlage B: C-code van het basisstation

```
/* Author : Andre Michel Abikhaled
 * Study# : 1198653
 * Goal : comparing the data coming
 * the master node via ad-hoc
 * network to the basis Station
 */

# include <stdio.h>
# include <string.h>
# include <stdlib.h>
#define FILENAME "data_base.txt"
void traffic_check(void);
int coordinate_matching(float a , float b ,float c , float d, float e,
float f);
main()
{
    traffic_check();
}

/*
 * Action : Comparing the coordinates coming
 * from the master node via ad-hoc network
 * and the coordinates that are already stored
 * in the database of the basis station.
 */

void traffic_check(void)
{
    int k,j, result;
    int av_v;
    float x[22],y[22],v[22];
    float x1[22],x3[22],x4[22],x5[22],x6[22];
    char x2[22];
    int velocity =0;
    FILE *file_in,*file_database,*file_out;
    file_in = fopen("output.txt", "r");
    file_database = fopen("data_base.txt", "r");
    file_out = fopen("out.txt","w");
    for (k = 0; k<22; k++)
    {
        fscanf(file_in,"%f %f %f", &x[k], &y[k], &v[k]);
        fscanf(file_database," %f %c %f %f %f %f ",&x1[k], &x2[k],
&x3[k], &x4[k],&x5[k],&x6[k]);
        /* printf(" %f %c %f %f %f %f\n
",x1[k],x2[k],x3[k],x4[k],x5[k],x6[k]);*/
        result = coordinate_matching(x[k] , y[k] , x3[k] , x4[k] ,
x5[k], x6[k]);
        fprintf(file_out," %f %c %f %f %f \n ", x1[k] , x2[k] ,
x[k],y[k],v[k]);
    }

    for(j = 0; j<22; j++)
    {
        velocity = velocity + v[j];
        av_v = velocity/22;
    }

    /* printf("The average velocity is : %d \n",av_v);*/

    if(av_v<10 && k>5)
```

```

    {
        fprintf(file_out,"file file op de smalle  weg\n");
        fprintf(file_out,"Begin eind gemiddeld_snelheid %f %f %f %f %d
\n",x[1],y[1],x[19],y[19],av_v);
    }
    if(av_v<40 && k>10)
    {
        fprintf(file_out,"file file op de normale weg\n");
        fprintf(file_out,"Begin eind gemiddeld_snelheid %f %f %f %f %d
\n",x[1],y[1],x[19],y[19],av_v);
    }
    if(av_v<50 && k>20)
    {
        fprintf(file_out,"file file op de brede  weg\n");
        fprintf(file_out,"Begin eind gemiddeld_snelheid %f %f %f %f %d
\n",x[1],y[1],x[19],y[19],av_v);
    }
    fclose(file_in);
    fclose(file_database);
    fclose(file_out);
}

```

Bijlage C: De scenario's

Scenario 1 (file op S22)

X	Y	V
87	32	7
78	32	8
81	32	10
84	32	9
90	32	9
93	32	10
96	32	8

Scenario 2 (mogelijk een file op S22 / geen file op S22)

X	Y	V
87	32	10
78	32	12
81	32	11
84	32	11
90	32	10
93	32	12
96	32	9

Scenario 3 (file op B42)

X	Y	V
41	164	46
12	164	45
15	164	46
18	164	45
21	164	46
24	164	47
27	164	50
30	164	51
33	164	48
37	164	48
40	164	48
43	164	46
46	164	45
49	164	47
52	164	48
55	164	46
58	164	45
61	164	44
64	164	45
67	164	48
70	164	49
73	164	48

Scenario 4 (geen file op B42)

X	Y	V
35	164	47
12	164	45
15	164	46
18	164	45
21	164	46
24	164	47
27	164	50
30	164	51
33	164	48
37	164	48
40	164	48
43	164	46
46	164	45
49	164	47
52	164	48
55	164	46
58	164	45

Scenario 5 (mogelijk een file op N12 / geen file op N12)

X	Y	V
74	28	42
74	13	41
74	16	40
74	19	42
74	22	42
74	25	42
74	31	39
74	34	40
74	37	41
74	40	42
74	43	41
74	46	40

Scenario 6

X	Y	V
74	28	38
74	13	39
74	16	38
74	19	40
74	22	40
74	25	41
74	31	37
74	34	36
74	37	38
74	40	39
74	43	38
74	46	40

Scenario 7

X	Y	V
66	50	8
67	50	7
68	50	6
69	50	9
70	50	8
71	50	7
72	50	6
73	50	8
71	50	35
71	51	33
71	52	32
71	53	30
71	54	36
66	80	41
67	80	42
68	80	43
69	80	44
70	80	43
71	80	45

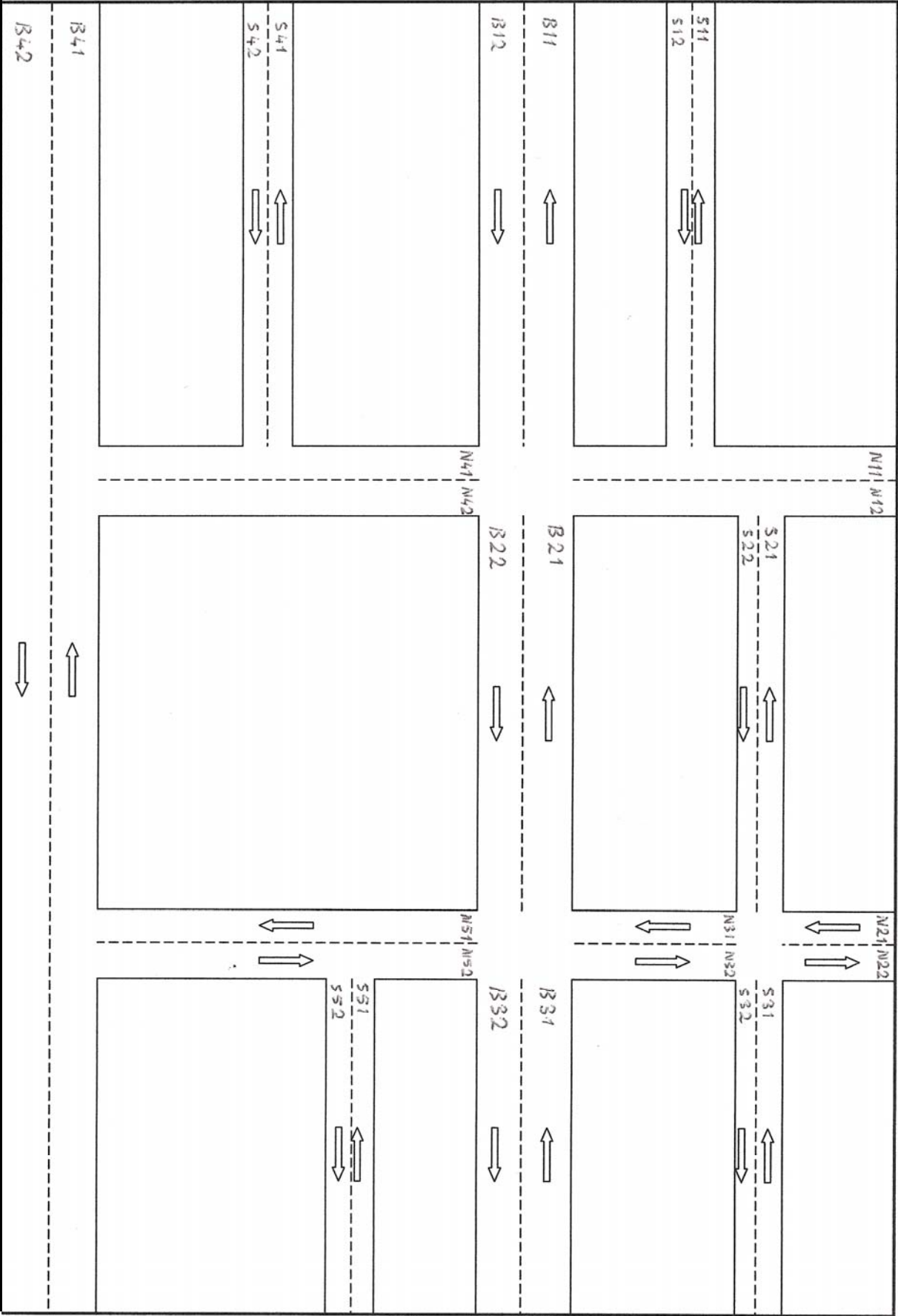
Scenario 8

X	Y	V
0	164	40
0	165	40
0	166	40
0	167	41
0	168	41
0	169	42
0	170	43
0	171	43
0	172	45
0	173	43
0	174	42
0	175	44
0	176	45
0	177	49
0	178	49
0	179	48
0	180	45
0	181	46
0	182	47

Scenario 9

X	Y	V
71	0	26
71	1	26
71	2	27
71	3	27
71	4	28
71	5	29
71	6	29
71	7	23
71	8	24
71	9	39
71	10	35
74	199	36
74	198	37
74	197	35
74	196	35
74	195	34
74	194	39
74	193	34
74	191	32

Bijlage D: Wegennetkaart + coördinaten



Wegstuk	begin (x,y)	eind (x,y)
S11	(0,50)	(70,50)
S12	(0,52)	(70,52)
S21	(75,30)	(135,30)
S22	(75,32)	(135,32)
S31	(140,30)	(190,30)
S32	(140,32)	(190,32)
S41	(0,130)	(70,130)
S42	(0,132)	(70,132)
S51	(140,115)	(190,115)
S52	(140,117)	(190,117)
N11	(71,0)	(71,79)
N12	(74,0)	(74,79)
N21	(136,0)	(136,29)
N22	(139,0)	(139,29)
N31	(136,33)	(136,79)
N32	(139,33)	(139,79)
N41	(71,85)	(71,159)
N42	(74,85)	(74,159)
N51	(136,85)	(136,159)
N52	(139,85)	(139,159)
B11	(0,80)	(70,80)
B12	(0,84)	(70,84)
B21	(75,80)	(135,80)
B22	(75,84)	(135,84)
B31	(140,80)	(190,80)
B32	(140,84)	(190,84)
B41	(0,160)	(190,160)
B42	(0,164)	(190,164)