

## MASTER

### Real-time regeling van wegverkeer

Honig, E.

*Award date:*  
1988

[Link to publication](#)

#### **Disclaimer**

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

#### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

#### **Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**TECHNISCHE UNIVERSITEIT EINDHOVEN**  
Afdeling der Elektrotechniek

**Real-time regeling  
van wegverkeer**

Afstudeerverslag van  
E. Honig

Verslag van het afstudeerwerk uitgevoerd  
onder leiding van Prof. Ir. M.P.J. Stevens  
bij de vakgroep Digitale Systemen (EB)  
van januari tot augustus 1988

## Samenvatting

Dit verslag gaat over het regelen van verkeer met behulp van verkeerslichten. Hiermee is de afgelopen decennia al de nodige ervaring opgedaan. In de eerste eerste hoofdstukken van dit verslag wordt getracht een overzicht te geven van de diverse verkeersregelsystemen die zijn ontwikkeld.

In het kader van het afstudeeronderwerp is nagedacht over de opzet van een real-time verkeersregeling. Daar microprocessoren vandaag de dag goedkoop zijn, is er geen beletsel meer om deze in te zetten voor zo'n systeem. Voor real-time regeling moeten namelijk continu berekeningen uitgevoerd worden.

In de eerste instantie is een opzet gemaakt voor een systeem voor verkeersregeling op een enkel tracé. Vervolgens is dit concept uitgebreid tot een heel netwerk.

Voor een systeem ter regeling van het verkeer in een netwerk wordt een regelstrategie beschreven. Verder wordt van dit systeem een globale opzet gegeven.

## INHOUDSOPGAVE

Samenvatting . . . . .	1
Inhoudsopgave . . . . .	2
Hoofdstuk 1 : Inleiding . . . . .	4
Hoofdstuk 2 : Singuliere kruispunten . . . . .	6
2.1 Algemeen . . . . .	6
2.2 Criteria voor het plaatsen van verkeerslichten . . . . .	7
2.3 Het verzamelen van kruispuntgegevens . . . . .	8
2.4 Het berekenen van een optimale starre regeling . . . . .	8
2.4.1 Groen- en roodtijdstippen van de lichten . . . . .	9
2.4.2 Relaties tussen verkeerslichten . . . . .	10
2.4.3 Het totale probleem . . . . .	14
2.5 Verkeersafhankelijke regeling . . . . .	15
Hoofdstuk 3 : Netwerkregelingen . . . . .	17
3.1 Waarom, wanneer en hoe coördinatie ? . . . . .	17
3.2 Starre netwerkregelingen . . . . .	20
3.3 Verkeersafhankelijke programmaselektie . . . . .	22
3.4 Verkeersafhankelijk geregelde netwerken . . . . .	23
3.5 Real-time netwerkregelingen . . . . .	24
3.6 Huidige status . . . . .	25
Hoofdstuk 4 : Real-time verkeersregeling op een tracé . . . . .	26
4.1 Algemene beschrijving . . . . .	26
4.2 Opzet van de regeling . . . . .	27
4.3 De lokale regelaar . . . . .	29
4.4 Interactie tussen lokale regelaars . . . . .	30
4.5 Nadere beschouwing van de clusterafhandeling . . . . .	30
4.5.1 Gedrag van clusters . . . . .	30
4.5.2 Voorstartgroen . . . . .	32
4.5.3 Passeren van een cluster . . . . .	33
4.5.4 Communicatie . . . . .	34
4.6 Handelingen bij een passerend cluster . . . . .	36

4.7	Processtructuur van de lokale regelaar . . . . .	38
4.7.1	Het hoofdprogramma . . . . .	39
4.7.2	Detektormelding . . . . .	40
4.7.3	Bericht van andere lokale regelaar . . . . .	40
4.7.4	Bericht van / naar hoofdregelaar . . . . .	41
4.8	De hoofdregelaar . . . . .	46
4.8.1	Processen in de hoofdregelaar . . . . .	47
4.9	De communicatie . . . . .	49
4.9.1	Data formaat . . . . .	50
4.9.2	Bepaling van de baudrate . . . . .	51
Hoofdstuk 5 : real-time netwerkregeling . . . . .		54
5.1	Netwerkdefinitie . . . . .	54
5.2	Specificatie en eisen . . . . .	54
5.3	Opzet van het systeem . . . . .	56
5.3.1	Configuratie . . . . .	56
5.3.2	Regelstrategie . . . . .	57
5.4	De hoofdregelaar . . . . .	57
5.5	Knooppunt uit een netwerk . . . . .	58
5.6	Nadere beschouwing van de regelstrategie . . . . .	61
5.7	De lokale regelaar . . . . .	64
5.7.1	Algemeen . . . . .	64
5.7.2	Coördinatie in de tijd . . . . .	65
5.7.3	Het optimalisatie-proces . . . . .	66
5.7.4	Processen in de lokale regelaar . . . . .	69
Hoofdstuk 6 : Conclusies . . . . .		70
Literatuur . . . . .		71

## Hoofdstuk 1 : Inleiding

De afgelopen decennia is de verkeersdrukte explosief toegenomen. Dit heeft geleid tot het ontwerp van verkeerslichteninstallaties voor het regelen van het verkeer op drukker kruispunten en tracé's.

In de eerste instantie werden regelingen ontworpen voor de afwikkeling van het verkeer op singuliere kruispunten. Dit waren eerst zogenaamde *starre* regelingen. Een starre verkeersregeling draait een vast programma af, zonder te kijken naar het actuele verkeersaanbod op de toevoerwegen. Iedere cyclus krijgen de diverse fasen van de kruising in dezelfde volgorde groen en rood.

Werkten de regelaars in het eerste begin nog mechanisch (uitgevoerd met relais en schakelwalsen), later werden microprocessoren toegepast. Met microprocessoren werden starre verkeersregelaars gemaakt. Groot voordeel hierbij is uiteraard dat programma's achteraf gemakkelijk veranderd kunnen worden. Bij een regelaar opgebouwd met schakelwalsen is het beduidend moeilijker om een eenmaal gemaakt afwikkelings-programma later nog te veranderen.

Later werden de kruispunt-regelaars *verkeersafhankelijk*. In de toevoerwegen van de kruising zijn detectoren aangebracht, die de verkeersregelaar informatie geven over het actuele verkeersaanbod. Afhankelijk hiervan kunnen bepaalde fasen dan langer of korter of helemaal geen groen krijgen. Wanneer er op een zijweg geen verkeer aanwezig is, heeft het geen zin deze richting groen te geven. De regelaar zal dit dan ook niet doen. Een verkeersafhankelijke kruispuntregeling zal over het algemeen een efficiëntere afwikkeling van de verkeer op de kruising bewerkstelligen, en zo voor kortere wachttijden zorgen.

Wanneer een aantal kruisingen in een keten is opgenomen, is het wenselijk om voor het verkeer over deze keten een doorstroming (soms wel "groene golf" genoemd) te creëren.

Tussen de diverse kruispuntregelaars wordt dan een koppeling aangebracht, zodat de regelaars met elkaar kunnen communiceren. De groentijden van de kruisingen kunnen dan zodanig op elkaar worden afgestemd, dat verkeer op het tracé zo weinig mogelijk voor rood licht zal hoeven stoppen. In de loop der jaren zijn diverse methoden bedacht om coördinatie tussen de regelingen van kruispunten behorende bij een tracé aan te brengen.

In de eerste hoofdstukken van dit verslag wordt een overzicht gegeven van de diverse verkeersregelsystemen die tot nu toe ontwikkeld zijn.

Er is nagedacht over een systeem ter coördinatie van een verkeersstroom over een tracé. Dit tracé bestaat in de eerste instantie uit één keten van kruispunten. Van het systeem wordt in hoofdstuk 4 een beschrijving gegeven.

Microprocessoren zijn heden ten dage goedkoop. Wanneer een gecoördineerde verkeersregeling wordt opgebouwd, is er geen belemmering meer om de intelligentie te decentraliseren door iedere kruising door een microprocessor te laten regelen. Op deze manier is een regelsysteem voor een tracé opgezet. In hoofdstuk 5 wordt uiteindelijk toegewerkt naar een netwerk van wegen en kruisingen, waarbij iedere kruising zijn regelaar heeft, en waarbij een optimale regelstrategie voor het verkeer in het netwerk wordt toegepast.

## Hoofdstuk 2 : Singuliere kruispunten

In dit hoofdstuk beschouwen we de afwikkeling van het verkeer op een enkel kruispunt. Het verkeer op de kruising wordt geregeld met verkeerslichten. De regeling kan star of verkeersafhankelijk werken. Informatie over verkeersaanboden kan afkomstig zijn van detectoren, of, wanneer de kruising in een gecoördineerde netwerkregeling is opgenomen, van andere kruisingen.

### 2.1 Algemeen

Men kan criteria opstellen aan de hand waarvan kan worden vastgesteld of het nodig is om een kruising te voorzien van verkeerslichten. Stel dat voor een bepaalde kruising blijkt dat dit nodig is. Voor het ontwerp van de regeling kunnen we dan een reeks van stappen onderscheiden die in een vaste volgorde worden uitgevoerd. Ze zullen hieronder globaal opgenoemd worden:

- \* Formuleren van doelstellingen voor de verkeerssituatie op het kruispunt. De grootte van de diverse verkeersstromen moet hierbij in beschouwing worden genomen.
  
- \* Verzamelen van alle kruispuntgegevens zoals
  - kruispuntgeometrie : de exacte vorm en afmetingen van de kruising. Verkeer op een aanvoerweg kan op de kruising verschillende kanten op, er zijn verschillende fasen.
  - verkeersintensiteiten op toevoerwegen.
  - afrijcapaciteiten : capaciteiten van afvoerwegen.
  - strijdige fasen : fasen die nooit tegelijkertijd groen mogen voeren. Ook genoemd : konfliktfasen.
  - ontruimingstijden : minimale tijd die in acht moet worden genomen tussen overgang van strijdige fasen. Deze overgang wordt om veiligheidsredenen even uitgesteld, omdat verkeer van strijdige fasen nooit tegelijkertijd op het kruisingsvlak mag zijn. De minimale tijd



die het verkeer van de oude fase nodig heeft om de kruising te ontruimen heet de ontruimingstijd.

- \* Uit de kruispuntgegevens kunnen worden opgesteld
  - een konfliktmatrix : hierin zijn alle konfliktkerende fasen opgeslagen.
  - een ontruimingstijdenmatrix : hierin zijn de ontruimingstijden tussen alle fasen opgeslagen.
  - minimale groentijden voor de verschillende fasen. Deze kunnen worden afgeleid uit de gegevens over de drukte (intensiteit) van de diverse verkeersstromen.
  
- \* Uit bovenstaande gegevens kan een optimale starre kruispuntregeling ontworpen worden. Zie voor methoden voor het ontwerpen van een starre regeling paragraaf 2.4.
  
- \* Uitgaande van de verkregen starre regeling, kan eventueel een verkeersafhankelijke regeling ontworpen worden. Zie hiervoor paragraaf 2.5.
  
- \* Eventueel kan de regeling geschikt gemaakt worden om opgenomen te worden in een gecoördineerd netwerk.
  
- \* Een ontworpen regeling kan gesimuleerd worden, zodat het ontwerp aan de doelstellingen getoetst kan worden. Het kan vervolgens zonodig aangepast worden.

In de nu volgende paragrafen zullen de bovenstaande stappen nader uitgewerkt worden.

## **2.2 Criteria voor het plaatsen van verkeerslichten**

Voor het bepalen wanneer het noodzakelijk is om een kruising te voorzien van verkeerslichten spelen een aantal criteria een rol. Belangrijke gegevens van een tot dusver niet geregelde kruising zijn verkeersintensiteiten en wachttijden. Verkeersintensiteiten kunnen uiteraard gemeten worden. Voor het verwerken van deze metingen zijn computerprogramma's beschikbaar.

Ook voor wachttijdberekeningen zijn computerprogramma's voorhanden. Deze berekenen bijvoorbeeld capaciteit, wachttijd en filelengte voor verkeersstromen die op een kruising voorrang moeten verlenen.

De uitkomsten van bovengenoemde computerprogramma's geven meestal wel een goede indicatie of het op een bepaalde kruising al dan niet gewenst is om verkeerslichten te plaatsen.

### **2.3 Het verzamelen van kruispuntgegevens**

#### **\* intensiteiten van het te regelen verkeer:**

Deze worden bepaald door middel van tellingen. Op straat wordt registratie- en telapparatuur aangebracht. De gegevens kunnen vervolgens weer verwerkt worden door een computer.

#### **\* afrijkapaciteiten:**

De afrijkapaciteiten van rijstroken kunnen worden bepaald aan de hand van metingen of met behulp van berekeningen. Voor metingen is de hierboven beschreven apparatuur beschikbaar. Voor berekening van afrijkapaciteiten zijn computerprogramma's ontwikkeld. Deze programma's houden in hun berekeningen rekening met

- fasenindeling (konfliktfasen)
- type van de rijstrook (recht door, afbuigend, ..)
- aandeel van zwaar verkeer
- helling van de weg
- breedte van de rijstroken
- invloed van korte opstelvakken

### **2.4 Het berekenen van een optimale starre regeling**

Er zijn diverse methoden ontwikkeld om uit de kruispuntgegevens een optimale starre regeling te ontwerpen. In verreweg de meeste gevallen wordt een computer gebruikt voor de berekening. Aan het betreffende computerprogramma wordt een verzameling inputgegevens over de kruising aangeboden.

Aangezien er sprake is van een starre regeling, werkt het programma met een vaste cyclustijd. De output bestaat uit de cyclustijd en de set groen- en roodtijden van de diverse fasen van de kruising. Een voorbeeld van zo'n programma is het programma KRAAN, dat veel toegepast is.

Een kruispuntregeling moet het aangeboden verkeer redelijk kunnen verwerken, en moet vooral verhinderen dat strijdige verkeersstromen met elkaar in konflikt komen. Wanneer op een kruising een bepaalde verkeersstroom meerdere verkeerslichten passeert, is het wenselijk dat deze lichten een "groene golf" vertonen, om ongewenste vertragingen en stuwingen te vermijden. Men kan een regeling optimaal noemen als onder deze voorwaarden de som van de groentijden zo groot mogelijk is.

#### 2.4.1 Groen- en roodtijdstippen van de lichten

Beschouw een willekeurige kruising, welke voorzien is van verkeerslichten. Het aantal lichten is  $N$ . De regeling kent aan ieder verkeerslicht  $i$  van de kruising een *groentijdstip*  $g_i$  en een *roodtijdstip*  $r_i$  toe. Hieronder wordt verstaan het tijdstip dat het licht van rood op groen resp. van groen op rood springt. Het signaal oranje zullen we beschouwen als een deel van het signaal groen, en zo dus verder buiten beschouwing laten. Omdat hier sprake is van een starre regeling, liggen de groen- en roodtijdstippen vast ten opzichte van een zeker tijdstip  $t_0 + nC$ . Hierin is  $t_0$  een begintijdstip,  $n$  een geheel getal en  $C$  de cyclustijd. De *groentijd*, de tijd dat een licht op groen staat, is het verschil tussen het rood- en het groentijdstip.

- (1) Om het aangeboden verkeer redelijk te kunnen verwerken, mag de groentijd die aan een verkeerslicht wordt toegewezen, niet kleiner zijn dan de voor die fase geldende minimale groentijd. Dus:

$$r_i - g_i \geq M_i \quad i = 1..N$$

waarin  $M_i$  de minimale groentijd van verkeerslicht  $i$  voorstelt.

(2) Uiteraard mag mag de groentijd van een verkeerslicht niet groter zijn dan de cyclustijd. Dus:

$$r_i - g_i \leq C \quad i = 1..N$$

Dit levert dus voor iedere verkeerslicht van de kruising twee ongelijkheden (1) en (2) op.

#### 2.4.2 Relaties tussen verkeerslichten

Op de kruising kan tussen ieder paar verkeerslichten een zekere relatie bestaan, afhankelijk van de manier waarop deze lichten moeten samenwerken om het aangeboden verkeer te verwerken.

\* Wanneer, zoals hierboven al genoemd, een verkeersstroom over de kruising meerdere verkeerslichten passeert, is het gewenst tussen deze lichten een "groene golf" te creëren. De relatie tussen twee van deze lichten wordt een *golfrelatie* genoemd, de lichten vormen een *golfpaar*.

\* Verkeerslichten behorende bij konfliktierende fasen moeten verhinderen, dat de door hen bestuurde verkeersstromen met elkaar in konflikt komen. De relatie tussen twee van deze verkeerslichten noemen we een *konfliktrelatie*, de lichten vormen een *konfliktpaar*.

\* Tussen verkeerslichten die niet bij eenzelfde stroom horen, en die geen konfliktierende fasen besturen, bestaat geen relatie.

#### golfrelaties:

Een golfrelatie tussen twee lichten  $i$  en  $j$  levert ongelijkheden op met  $g_i$ ,  $r_i$ ,  $g_j$  en  $r_j$  als variabelen. De rijtijd van licht  $i$  naar licht  $j$  speelt hierin een rol. Omdat we willen dat de golfrelatie voor zowel de snelste als de langzaamste verkeersdeelnemer geldt, voeren we de twee rijtijden in:

$T_{ij}$  = toevoertijd van  $i$  naar  $j$  (waarbij men denkt aan de snelste verkeersdeelnemer).

$O_{ij}$  = ontruimingstijd van i naar j (waarbij men rekening houdt met de langzaamste verkeersdeelnemer).

Wanneer de snelste verkeersdeelnemer bij licht i vertrekt, moet bij aankomst bij licht j dit op groen staan. Dit levert:

$$g_j \leq g_i + T_{ij} \quad (3)$$

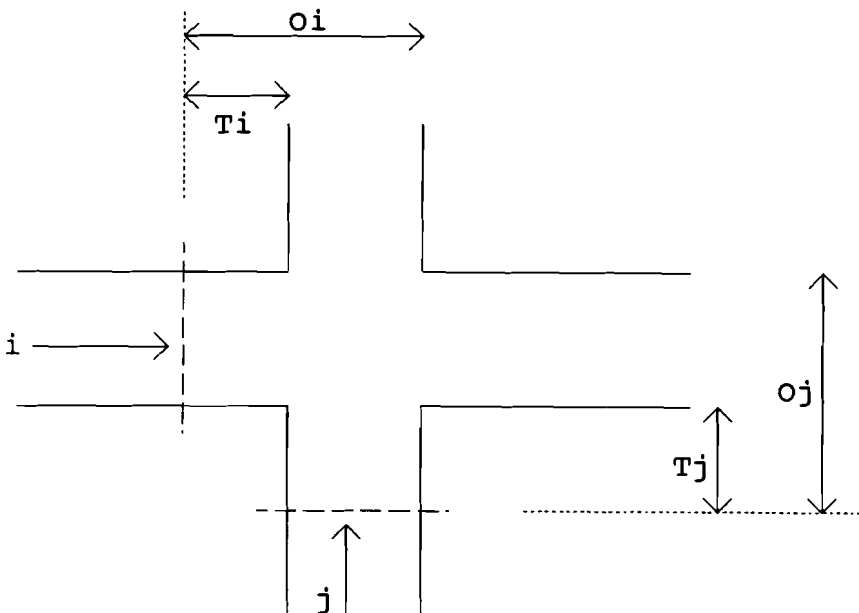
Wanneer de langzaamste verkeersdeelnemer bij licht i vertrekt, moet licht j bij aankomst nog steeds op groen staan. Dus:

$$r_i + O_{ij} \leq r_j \quad (4)$$

Iedere golfrelatie tussen twee verkeerslichten levert dus twee ongelijkheden (3) en (4) op.

#### konfliktrelaties:

We hebben hier te maken met relaties tussen verkeerslichten die konfliktgerende stromen besturen. Neem als voorbeeld de kruising van figuur 2.1.



Figuur 2.1 : Konfliktgerende stromen

We introduceren voor beide stromen:

- Toevoertijden  $T_i$  en  $T_j$ , de tijd die de snelste verkeersdeelnemer nodig heeft om vanaf de stopstreep naar de kruising te rijden.
- Ontruimingstijden  $O_i$  en  $O_j$ , de tijd die de langzaamste verkeersdeelnemer nodig heeft om het kruisingsvlak te ontruimen.

Omdat de stromen konfliktierend zijn, moet in ieder geval iedere groenperiode van  $i$  vallen in een roodperiode van  $j$  en omgekeerd. Verder moet men nog rekening houden met toevoeren ontruimingstijden.

Van groot belang is de volgorde waarin de groenperioden van  $i$  en  $j$  optreden. We beschouwen de situatie vanaf tijdstip  $t=0$  en onderscheiden nu twee gevallen:

- a) Na  $t=0$  gaat de eerste groenperiode  $(g_j, r_j)$  van  $j$  vooraf aan de eerste groenperiode  $(g_i, r_i)$  van  $i$ . De verkeersstroom die door  $j$  wordt doorgelaten in de groenperiode  $(g_j, r_j)$  zal het kruisingsvlak bezet houden gedurende het interval

$$(g_j + T_j, r_j + O_j).$$

Het kruisingsvlak is dus vrij voor het door  $i$  bestuurde verkeer in het interval

$$(r_j + O_j, g_j + T_j + C). \quad (I)$$

Het verkeer dat door  $i$  wordt vrijgegeven in de groenperiode  $(g_i, r_i)$  bezet het kruisingsvlak in het interval

$$(g_i + T_i, r_i + O_i). \quad (II)$$

Om conflicten te voorkomen moet het interval (II) geheel binnen het interval (I) liggen, dus:

$$r_j + O_j \leq g_i + T_i$$

$$r_i + O_i \leq g_j + T_j + C$$

Wanneer we invoeren:

$$V_{ij} = O_i - T_j$$

$$V_{ji} = O_j - T_i$$

gaat het stelsel over in:

$$r_j + V_{ji} \leq g_i$$

$$r_i + V_{ij} \leq g_j + C$$

$V_{ij}$  en  $V_{ji}$  worden ook wel de *veiligheidsmarges* genoemd.

- b) Na het tijdstip  $t=0$  volgt de eerste groenperiode ( $g_j, r_j$ ) van  $j$  op de eerste groenperiode ( $g_i, r_i$ ) van  $i$ . De groenperiode ( $g_j, r_j$ ) gaat dan echter vooraf aan de tweede groenperiode ( $g_i + C, r_i + C$ ) van  $j$ . Men vindt dan de relaties:

$$r_j + V_{ji} \leq g_i + C$$

$$r_i + V_{ij} \leq g_j$$

Afhankelijk van de fasenindeling van het verkeerslichten-systeem zullen ofwel de relaties onder a) ofwel die onder b) geldig zijn. Men kan de relaties onder a) en b) combineren door een grootte  $\delta_{ij}$  in te voeren. Deze grootte wordt de *konfliktdelta* behorende bij het konfliktpaar ( $i, j$ ). In geval a) is  $\delta_{ij}$  gelijk aan 0, in het geval b) is  $\delta_{ij}$  gelijk aan 1. Een paar konfliktkerende verkeerslichten levert zo uiteindelijk de volgende ongelijkheden op:

$$r_j + v_{ji} \leq g_i + \delta_{ij} \cdot c \quad (5)$$

$$r_i + v_{ij} \leq g_j + (1 - \delta_{ij}) \cdot c \quad (6)$$

### 2.4.3 Het totale probleem

We hebben nu het volgende probleem:

Voor een kruising met  $N$  verkeerslichten moeten de groentijdstippen  $g_1 \dots g_N$  en de roodtijdstippen  $r_1 \dots r_N$  berekend worden, zodanig dat de regeling optimaal is. In de voorgaande twee subparagrafen zijn een aantal ongelijkheden (1) t/m (6) afgeleid waaraan deze tijdstippen moeten voldoen. Samengevat:

- \* De ongelijkheden (1), die uitdrukken dat de groentijd van een licht niet kleiner mag zijn dan de minimale groentijd van dat licht. Er zijn dus  $N$  ongelijkheden (1).
- \* De ongelijkheden (2), die uitdrukken dat de groentijd van een licht de cyclustijd niet mag overschrijden. Er zijn dus ook  $N$  ongelijkheden (2).
- \* De golfrelaties (3) en (4). Er zijn evenveel relaties (3) en (4) als het aantal golfparen.
- \* De konfliktrelaties (5) en (6). Er zijn evenveel relaties (5) en (6) als het aantal konfliktparen.

We hebben nu een stelsel ongelijkheden die alle de gedaante van *lineaire* ongelijkheden hebben. Dit stelsel zal over het algemeen *onderbepaald* zijn, dat wil zeggen als het stelsel een oplossing heeft, heeft het ook meer dan een oplossing. De vraag is nu of een van deze oplossingen de beste is, en zo ja welk criterium men hiervoor kan hanteren.

Het ligt voor de hand dat men zoveel mogelijk groen in het verkeerslichtenprogramma zal willen zien onder de voorwaarde dat aan de eisen (1) t/m (6) is voldaan. De som van alle tijdseenheden groen in één cyclus is:

$$\sum_{i=1}^N (r_i - g_i) \quad (7)$$



Het probleem is dus nu : zoek die oplossing van het stelsel lineaire ongelijkheden waarvoor (7) maximaal is.

Het ontwerpen van een optimale starre regeling verloopt nu in twee stappen. Eerst wordt voor de kruising een fasenindeling gemaakt. Dit zal in de regel "met de hand" geschieden. Vervolgens volgt het oplossen van een stelsel lineaire ongelijkheden. We hebben hier te maken met een zogenaamd 'lineair programmerings probleem'. Immers de ongelijkheden zijn lineair, en ook de functie (7) die gemaximaliseerd moet worden. Voor het oplossen van lineaire programmerings problemen zijn computerprogramma's beschikbaar, welke in staat zijn om grote stelsels ongelijkheden binnen redelijke tijd door te rekenen. Wanneer voor een kruising een fasenindeling is gemaakt, kan vervolgens de computer ingeschakeld worden voor het berekenen van alle groen- en roodtijden.

## **2.5 Verkeersafhankelijke regeling**

Een starre regeling kent aan elke fase iedere cyclus dezelfde groentijd toe, onafhankelijk van het aktuele verkeersaanbod. Uiteraard kan dit tot een situatie leiden dat een regeling aan zijrichtingen groen geeft terwijl er daar helemaal geen verkeer aanwezig is. Verkeer op de andere richtingen staat dan voor niets te wachten. Het is dus wenselijk om een regeling te laten reageren op het verkeersaanbod. Wanneer op bovengenoemde zijrichtingen detectoren geplaatst worden, weet de regeling of er daar al of niet verkeer aanwezig is. Zo niet, dan hoeven de betreffende fasen geen groen te krijgen.

Uit onderzoek is gebleken dat een voertuigafhankelijke regeling zeer goed uit een starre regeling kan voortvloeien. Vaak wordt voor een bepaalde kruising eerst een starre regeling ontworpen, die dan vervolgens verkeersafhankelijk wordt gemaakt.

Wanneer de verkeersintensiteit fluktuaties vertoont, moet de regeling zich zodanig aanpassen dat toch een optimale doorstroming verkregen wordt. De stromen die op een bepaald

moment een groot verkeersaanbod hebben, kunnen dan een langere groentijd dan normaal toegekend krijgen.

Met behulp van detectoren worden metingen verricht. Bij ieder voertuig dat passeert, ontvangt de regelaar een puls van de detector. Door nu de tijd tussen twee pulsen te meten, krijgt de regeling een indruk van de verkeersdrukke op de toevoerwegen. Deze tijd wordt de intervaltijd of hiaattijd genoemd. Een korte hiaattijd betekent grote drukke. Bij een korte hiaattijd kan de regeling beslissen om aan de betreffende stroom verlenging van de groentijd toe te kennen. We spreken van het verlengen van een stroom.

Uiteraard kan niet onbeperkt verlengd worden. Er zijn immers nog andere stromen die op groen wachten. Er zijn twee factoren die bepalen tot wanneer er verlengd kan worden. Ten eerste kan aan de hiaattijd een maximum waarde toegekend worden (grenshiaattijd). Wanneer de tijd tussen twee pulsen deze maximumtijd overschrijdt, is kennelijk de drukke zover afgenomen dat niet verder verlengd hoeft te worden. Echter : wanneer de grenshiaattijd het enige criterium voor het beëindigen van het verlengen zou zijn, zou er bij een aanhoudend drukke verkeersstroom onbeperkt verlengd worden. Daarom wordt er voor iedere fase een maximum groentijd ingesteld. Is deze overschreden, dan wordt een verlenging afgebroken, ook al is de grenshiaattijd nog niet overschreden. Het verkeerslicht zal op rood springen, uiteraard via een fase oranje.

Een andere mogelijkheid is progressie-verlenging. Hierbij wordt de grenshiaattijd kleiner naarmate de verlenging langer duurt. Dus : hoe langer de verlenging duurt, hoe kleiner de maximaal toegestane hiaattijd en dus hoe strenger de eis voor verlenging.

Wanneer een bepaalde stroom verlenging krijgt, is het mogelijk om niet-strijdige stromen (bijvoorbeeld de tegengestelde richting) "mee te verlengen". Immers, de strijdige stromen voeren vanwege het verlengen toch rood. De meeverlengde stroom kan hiervan profiteren, en eventueel hier wachtende voertuigen kunnen doorgelaten worden.

## Hoofdstuk 3 : Netwerkregelingen

Tot dusver was er sprake van een regeling voor het besturen van één kruispunt. Vaak is een kruising in de praktijk opgenomen in een netwerk. Een netwerk is een verzameling van twee of meer geregelde kruisingen. Een kruising behoort tot het netwerk wanneer de verdeling van het aankomende verkeer op minstens één richting beïnvloed wordt door de verkeersregeling van een andere kruising van het netwerk.

### 3.1 Waarom, wanneer en hoe coördinatie ?

Om diverse redenen kan men besluiten om tussen de diverse kruispuntregelingen een coördinatie aan te brengen (gecoördineerde regeling - "groene golf"). Een aantal redenen zijn:

- \* Wanneer kruisingen op korte afstand van elkaar liggen, kunnen bij afwezigheid van coördinatie de wegen tussen de kruisingen "dichtslibben".
- \* Een gecooördineerde regeling zorgt ervoor dat het aantal keren dat moet worden gestopt, wordt beperkt of voor bepaalde richtingen zelfs nul wordt gemaakt. Dit verkleint de kans op aanrijdingen, levert tijdwinst op, en is gunstiger voor het milieu.
- \* De door een kruispunt door te laten hoeveelheid verkeer kan worden aangepast aan de capaciteit van de erachter liggende kruispunten. Er kan dus doserend worden opgetreden.

In het algemeen kan men stellen dat, om bovenstaande redenen, coördinatie tussen kruisingen zo mogelijk altijd moet worden toegepast. De kruispunten moeten echter zo dicht bij elkaar gelegen zijn, en zodanig zijn gesitueerd, dat de verkeersafwikkeling op een kruispunt die op de andere kruispunten werkelijk beïnvloedt, anders is coördinatie zinloos. Verder is het belangrijk dat de verkeersstroom de kruispunten in praktisch ongewijzigde vorm passeert, zodat er sprake is van een constante doorgaande verkeersstroom. Wanneer er veel verkeer het tracé op en af rijdt is coördinatie moeilijker.

Aangezien via coördinatie de verkeersafwikkeling op een kruispunt wordt beïnvloed door de verkeersregeling op één of meer andere kruispunten, zullen er in de verkeersafwikkeling van een gecoördineerd kruispunt elementen komen die een voor die kruising optimale afwikkeling onmogelijk kunnen maken. Bij coördinatie verkiest men de optimale doorstroming van een bepaalde rijrichting of richtingen. De regeling was tot nu toe voor de kruising zelf optimaal, nu moet met minder dan de optimale regeling genoegen worden genomen.

Indien geen informatie wordt verzameld over de gedragingen van een verkeersstroom tussen twee gecoördineerde kruispunten, is een gecoördineerde regeling zeer kwetsbaar. Er wordt er dan immers van uit gegaan, dat een voertuig dat op een kruispunt vertrekt, na een tijd gelijk aan de rijtijd ook op het volgende kruispunt aankomt. Wanneer een aanrijding plaatsvindt, een voertuig langzaam rijdt, er halterende bussen zijn, er een bezinepomp aanwezig is, is hiervan geen sprake. Een gecoördineerde regeling kan zo volledig ontregeld worden. Verzamelen van informatie over gedragingen van voertuigen tussen twee kruisingen, bijvoorbeeld door het plaatsen van meer detektoren, is dus zinvol.

Bij het creëren van een groene golf rijst de vraag of deze in twee richtingen mogelijk is. Dit wordt bepaald door de afstand tussen de kruispunten, de cyclustijd op de kruispunten, en de rijsnelheid van het verkeer. In de paragraaf over starre regelingen zal hierop nader worden ingegaan.

De regelingen van kruisingen die tot een netwerk behoren, zijn dus zodanig op elkaar afgestemd, dat voertuigen die een bepaalde route door het netwerk volgen een zo gering mogelijke verliestijd ondervinden ten gevolge van de verkeerslichten.

Bij starre regelingen betekent dit dat de cyclustijden en de groentijden van de kruisingen zodanig zijn ingesteld dat er "groene golven" ontstaan. Een onderlinge koppeling van de kruispuntregelaars is hierbij nodig, omdat de regelingen anders op den duur uit fase zouden gaan lopen.

Ook verkeersafhankelijke regelingen kunnen worden opgenomen in een netwerk. Aangrenzende kruisingen worden dan zodanig geregeld dat clusters voertuigen zo weinig mogelijk verlies-tijd ondervinden.

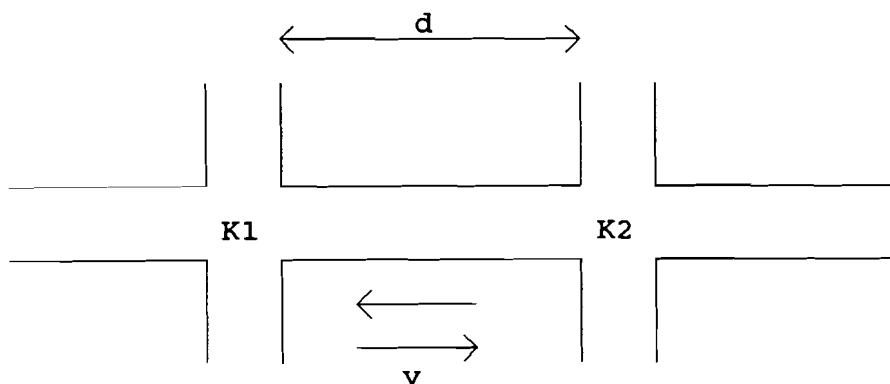
Met behulp van "verkeersafhankelijke programmaselektie" kan uit een aantal regelingen (zowel star als voertuigafhankelijk), automatisch de meest optimale regeling worden gekozen. Welke regeling de meest optimale is, hangt af van het tijdstip van de dag. Het verkeerspatroon is immers op geen enkel moment van de dag hetzelfde.

De meest recente vorm van netwerkregeling is real-time regeling. Bij deze vorm van regelen worden de verkeerslichtenregelingen voortdurend aan de hand van actuele verkeersinformatie geoptimaliseerd.

Gecoördineerde regeling van een netwerk zou uitgevoerd kunnen worden door een centrale computer. De afzonderlijke regelaars op de diverse kruispunten zouden dan vervallen, terwijl in plaats hiervan één centrale computer het hele netwerk regelt. Groot nadeel hiervan is, dat wanneer een storing in de centrale computer zou optreden, de hele keten van kruispunten buiten werking is. Daarom worden in de praktijk bijna altijd de kruisingen door eigen regelaars geregeld. Tussen deze regelaars is dan nog een koppelnetwerk aangebracht, zodat de regelaars kunnen communiceren. Zou er in het koppelnetwerk een storing ontstaan, dan kunnen de kruispuntregelaars nog "standalone" werken. Alleen de coördinatie van de keten is dan verdwenen, het verkeer op de kruisingen wordt echter nog wel afgehandeld.

Voor het ontwerpen van netwerkregelingen is veel gebruik gemaakt van speciaal daarvoor ontworpen computerprogramma's. Daarnaast vervulde de computer soms een belangrijke plaats bij het realiseren van de netwerkregelingen.

### 3.2 Starre netwerkregelingen



Figuur 3.1 : Twee opeenvolgende kruisingen in een netwerk

Neem de situatie van figuur 3.1. We hebben hier twee opeenvolgende kruispunten K1 en K2 op een onderlinge afstand  $d$ . Deze kruispunten zijn opgenomen in een gecoördineerde regeling. Beide kruisingen zijn star geregeld.

Coördinatie wordt hier bereikt door te zorgen dat het verschil tussen de groentijdstippen van beide kruisingen gelijk is aan de benodigde rijtijd. Een voertuig dat bij de ene kruising groen krijgt en wegrijdt, zal bij de volgende kruising op het juiste moment groen krijgen. Deze continue offset tussen de groentijden moet voor iedere cyclus hetzelfde blijven, er is immers sprake van starre regelingen. Dit houdt in, dat de cyclustijden van beide kruisingen even groot moeten zijn, omdat de regelingen anders uit fase gaan lopen. Algemeen geldt dat voor een gecoördineerd netwerk van starre regelingen geldt dat alle kruisingen hiervan dezelfde cyclustijd  $C$  hebben, of eventueel de helft hiervan.

Het liefst willen we een groene golf in twee richtingen hebben. De offset van de groentijden tussen de kruisingen moet dus voor beide richtingen gelijk zijn aan de rijtijd tussen de kruisingen. Wanneer we er vanuit gaan dat de rij snelheid  $v$  op beide richtingen gelijk is, is de rijtijd voor beide richtingen dus ook gelijk. Deze rijtijd is gelijk aan  $d / v$ , en is uit symmetrie-overwegingen ook gelijk aan de halve cyclustijd. We komen zo uiteindelijk op de formule

$$d = \frac{1}{2} c.v$$

Wanneer een tweezijdige groene golf gewenst is, bestaat er dus bovenstaand verband tussen de afstand tussen de kruisingen, de cyclustijd op de kruisingen en de rijsnelheid op de wegen tussen de kruisingen. De rijsnelheid ligt meestal vast. Ook de cyclustijd kan maar binnen zeer beperkte grenzen gevarieerd worden, omdat het verkeer op alle richtingen van een kruispunt goed verwerkt moet kunnen worden. Stel dat de rijsnelheid 50 km/u is (binnen de bebouwde kom). Een acceptabele waarde voor de cyclustijd is 60 sekonden. Dit betekent dan dat de afstand tussen de kruisingen 400 meter moet bedragen voor het realiseren van een tweezijdige groene golf.

Uiteraard zal het in de praktijk zelden voorkomen dat de afstand tussen kruisingen 400 meter bedraagt. Meestal zullen de afstanden tussen de kruispunten ook zeker niet gelijk zijn, maar een zeer grote variatie vertonen. Wanneer toch een tweezijdige groene golf gewenst is, is het dus nodig om tussen de diverse kruispunten verschillende rijsnelheden aan te geven. De cyclustijden moeten immers allemaal gelijk zijn. In de praktijk blijkt dit echter geen onverdeeld succes te zijn. In het buitenland, waar snelheidsaanduidingen werden toegepast, is gebleken dat een groot gedeelte van de weggebruikers zich niet houdt aan de aangegeven snelheden, zodat toch weer voor kruisingen moet worden gestopt en weer moet worden opgetrokken, zodat de doorstroming nadelig beïnvloed wordt en de groene golf teniet gedaan wordt. Vooral wanneer een lagere snelheid wordt aangegeven dan de bij het wegtracé passende snelheid, zal men zeker geneigd zijn de snelheidsaanduiding te negeren. In zo'n geval is het vaak beter om voor één richting (en wel de zwaarst belaste) een groene golf te creëren, terwijl men in de tegenovergestelde richting eventueel voor een verkeerslicht zal moeten stoppen. Welke richting de zwaarst belast is, hangt vaak af van het tijdstip van de dag, en moet dus variabel blijven.

Er zijn computerprogramma's geschreven waarmee starre netwerkregeringen geoptimaliseerd kunnen worden. Het programma TRANSYT is hier een voorbeeld van. Aan het programma worden gegevens van een star te regelen netwerk aangeboden. Voorwaarde is dat alle kruisingen dezelfde cyclustijd hebben, of een cyclustijd die de helft is van de cyclustijd van de overige kruisingen. Het programma berekent dan een optimale starre regeling. Als criterium voor optimaal zijn wordt vaak de zogeheten "Performance Index" gebruikt. Deze ziet er als volgt uit:

$$P.I. = \sum_{i=1}^n (W \cdot A_i \cdot V_i + 0.01 \cdot K \cdot B_i \cdot S_i)$$

Hierin is :  $V_i$  : gemiddelde verliestijd op richting  $i$   
 $S_i$  : aantal stops op richting  $i$   
 $W$  : totale kosten per gemiddeld verliesuur  
 $K$  : totale kosten per 100 voertuigstops  
 $A_i, B_i$  : gewichtsfactoren

Optimaal is een regeling waarvan de Performance Index zo klein mogelijk is.

### 3.3 Verkeersafhankelijke programmaselektie

Een stap in de richting van een verkeersafhankelijke netwerkregeling is de verkeersafhankelijke programmaselektie. Hierbij wordt uit een aantal starre regelingen (of regelingen waarvan de kruispuntregelingen beperkt verkeersafhankelijk werken) steeds de meest geschikte geselecteerd. De verkeerssituatie verandert voortdurend met de tijd (ochtend-, avondspits, dag, nacht), zodat soms een andere regeling zal worden ingeschakeld.

Dit wisselen van regeling kan gebeuren op vaste tijdstippen van het etmaal. De begin- en eindtijdstippen van bepaalde verkeerssituaties kunnen echter ten gevolge van bijvoorbeeld evenementen of weersinvloeden veranderen. Een betere methode is daarom het gebruik van verkeersafhankelijke programma-selektie. Er wordt hierbij getracht bepaalde verkeerssitua-



ties te herkennen, en de daarbij passende regeling in te schakelen. Dit herkennen gebeurt aan de hand van tellingen en bezettingsgraadmetingen met behulp van detectoren.

Het zal duidelijk zijn dat op deze manier niet "echt" verkeersafhankelijk geregeld wordt. Er wordt gekozen uit een beperkt aantal programma's, terwijl verder een nieuwe regeling eigenlijk steeds net iets te laat wordt ingeschakeld.

### **3.4 Verkeersafhankelijk geregelde netwerken**

Wanneer een netwerk niet star, maar verkeersafhankelijk geregeld wordt, zal het vaak toch gewenst zijn om coördinaties tussen de geregelde kruisingen tot stand te brengen.

Bij kruisingen die vlakbij elkaar liggen, kunnen er golfrelaties opgesteld worden als beschreven in hoofdstuk 2. Wanneer de afstand tussen de kruisingen wat groter is, zal er tussen de twee verkeersafhankelijke regelingen informatie uitgewisseld worden, zodanig dat zoveel mogelijk een doorstroming bewerkstelligd wordt.

Voor de koppeling tussen verkeersafhankelijke regelaars zijn verschillende methoden ontwikkeld. Deze methoden realiseren allemaal een zekere vorm van coördinatie, echter geen van alle realiseren een absolute coördinatie. Ofwel : de doorstroming van het verkeer over de diverse kruisingen zal verbeterd worden, maar de ideale situatie dat een voertuig dat bij de eerste kruising vertrekt, op alle volgende kruisingen kan doorrijden, wordt niet bereikt.

Een koppelingsmethode die toegepast wordt is bijvoorbeeld het aanpassen van de groentijden van een verkeersafhankelijke kruispuntregelaar aan de hand van informatie komend van de vorige kruising. Ook worden soms "wachtende installaties" toegepast, waarbij een kruispuntregelaar op bepaalde momenten van zijn cyclus gaat staan wachten op synchronisatiesignalen van andere regelaars. Soms wordt door een kruispuntregelaar

ook gebruik gemaakt van detektorinformatie van het vorige kruispunt. De regelaar kan hier zijn groentijden alvast aan aanpassen.

Zoals gezegd zijn bovenstaande methoden niet optimaal. Zij creëren een beperkte vorm van coördinatie, maar garanderen geen volledige doorstroming van het verkeer over het tracé. Wanneer we dit willen realiseren, komen we terecht bij de "real-time" netwerkregelingen.

### **3.5 Real-time netwerkregelingen**

Real-time netwerkregelingen zijn regelingen die voortdurend aan de hand van actuele verkeersinformatie aangepast worden aan de toestand in het netwerk. In het buitenland is hieraan enig onderzoek verricht, en zijn enkele systemen ontwikkeld.

Bij een real-time regeling zijn in het algemeen veel detectoren aanwezig. Er is bijvoorbeeld een systeem, genaamd SCOOT, ontwikkeld waarbij alle geregelde wegvlakken voorzien zijn van detectoren. Deze liggen op een zo groot mogelijke afstand van de stopstreep. Aan de hand van de verzamelde verkeersgegevens wordt voorspeld hoe het verkeer bij de stopstreep zal aankomen. Op ieder moment is er voor de kruising een bepaalde regeling gepland. Enkele sekonden voordat een verkeerslicht volgens die regeling groen zou worden, gaat een optimalisatieprogramma na of een wijziging van de geplande groentijd tot een verlaging leidt van de grootste verzadigingsgraad van de geregelde richtingen op de betreffende kruising. Indien een wijziging tot een verbetering van de regeling leidt, wordt die wijziging aangebracht. De offset tussen de kruisingen wordt iedere cyclus aan de hand van de voorspelde verkeerspatronen m.b.v. een Performance Index geoptimaliseerd.

In het bovenstaande voorbeeld is sprake van een systeem waarbij min of meer vaste regelingen steeds, aan de hand van actuele verkeersinformatie, "on line" aangepast worden. De in het volgende hoofdstuk besproken netwerkregeling is ontworpen

om "clusters" voertuigen over een tracé van het begin naar het einde te geleiden. Er is hierbij geen sprake meer van vaste regelingen of cyclustijden.

### **3.6 Huidige status**

Tot nu toe bestond het ontwerpen van netwerkregelingen meestal uit het aanbieden van de specifieke netwerkgegevens aan een computerprogramma. Deze programma's berekenden dan een zo optimaal mogelijke netwerkregeling. Het betrof dan meestal starre regelingen. Zoals al in de inleiding werd opgemerkt, zijn microprocessoren heden ten dage zo goedkoop, dat hiermee zonder meer real-time netwerkregelingen kunnen worden ontworpen. In de volgende hoofdstukken volgen beschrijvingen van zulke regelingen.

## Hoofdstuk 4 : Real-time verkeersregeling op een tracé

Beschouwd wordt een verkeersstroom over een tracé met hierop een aantal door verkeerslichten geregelde kruispunten. We willen voor deze verkeersstroom een zo goed mogelijke doorstroming creëren. Hiervoor is een verkeersregeling opgezet, die in dit hoofdstuk beschreven wordt.

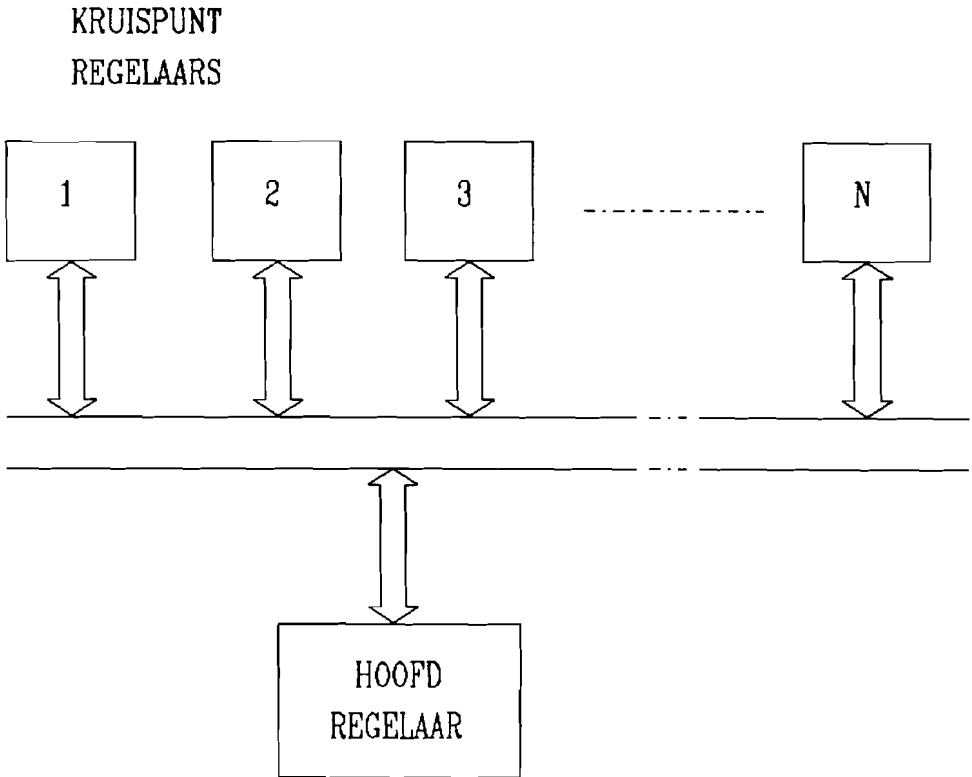
### 4.1 Algemene beschrijving

Er is sprake van een *event-driven* systeem. Hiermee wordt bedoeld, dat een willekeurige regelaar reageert op gebeurtenissen. Zo'n gebeurtenis is in dit geval een naderend voertuig, of een cluster naderende voertuigen, die hun komst melden, en die verzoeken om doorgelaten te worden. Treden er op een bepaald moment geen gebeurtenissen op, dan is het systeem in rust : het wacht op een melding. Zo'n event-driven systeem staat tegenover de systemen beschreven in de vorige hoofdstukken, waarbij steeds met (meer of minder vaste) cyclustijden werd gewerkt. De regelingen werken hierbij een cyclus af, ook als er geen gebeurtenis optreedt.

Het systeem bestaat uit één centrale intelligente regelaar, en voor ieder kruispunt een lokale regelaar. Deze zijn door middel van een datanet onderling doorverbonden. Iedere lokale regelaar heeft een adres, zodat via dit datanet iedere regelaar geadresseerd kan worden. Zie figuur 4.1.

De regeling kan in één richting een volledige doorstroming op de hoofdrichting realiseren. Deze richting wordt de voorkeursrichting genoemd. Voor deze richting is gegarandeerd dat iedere auto die het tracé oprijdt, in één keer door kan rijden naar het einde van het tracé. Voor de tegenovergestelde richting wordt de meest optimale doorstroming gerealiseerd. Soms, bij een bepaalde grootte van het verkeersaanbod, kan in twee richtingen een volledige doorstroming gerealiseerd worden. Wanneer de ene richting duidelijk veel zwaarder belast is dan de andere, bijvoorbeeld in de spits, wordt deze zwaarst belaste richting uiteraard als voorkeursrichting

gekozen. Is er geen duidelijk zwaarst belaste richting, dan kan een willekeurige richting voorkeursrichting worden gemaakt.



Figuur 4.1 : Regelsysteem voor een tracé

#### 4.2 Opzet van de regeling

De regeling werkt op een wegtracé met door verkeerslichten geregelde kruispunten op onregelmatige afstanden. Deze afstanden, de rijnsnelheden, de verkeersdichtheden, kortom alle verkeersgegevens van het tracé zijn parameters voor het systeem.

Het eerste kruispunt van elke rijrichting wordt de poort van de betreffende richting genoemd. De beide poorten gaan fungeren als *sturende* kruispunten. De poort van een rijrichting vervult een belangrijke taak, omdat deze het verkeer dat het tracé oprijdt doseert. De regelaar van het als poort

fungerende kruispunt krijgt steeds gegevens binnen omtrent het aantal door te laten voertuigen. Dit cluster voertuigen, dat nu het tracé oprijdt, moet goed verwerkt kunnen worden. Dat wil zeggen : in de voorkeursrichting moet het cluster in één keer het tracé over, terwijl in de tegengestelde richting geen opstoppingen mogen ontstaan.

De kruispunten met het grootste verkeersaanbod op de dwarsfasen worden de *bottle-necks* genoemd. Zij vormen de flessehalzen van het tracé, die in feite de capaciteit van het hele tracé bepalen. De doorlaatcapaciteit van de bottlenecks bepaalt hoeveel verkeer er per keer door het sturende kruispunt (de poort) kan worden doorgelaten.

N.B. In dit hoofdstuk zullen we voortaan elke "keer" dat een poort een cluster voertuigen doorlaat een cyclus noemen. Deze cyclus moet niet verward worden met de cyclus uit de vorige hoofdstukken. Hierbij was sprake van cyclustijden op de diverse kruispunten. Bij het hier beschreven systeem hoort er bij iedere cyclus een cluster voertuigen, dat alle kruispunten passeert, of ergens van het tracé afbuigt. De "cyclustijden" op de diverse kruisingen zijn dus iedere keer verschillend, en worden bepaald door het sturende kruispunt.

Voor iedere cyclus bekijkt de hoofdregelaar hoeveel voertuigen het tracé op mogen rijden. Hiervoor wordt het verkeersaanbod op de diverse dwarsfasen, vooral die van de *bottle-necks*, in ogenschouw genomen. Afhankelijk van deze verkeersaanboden, worden meer of minder voertuigen toegelaten. De hoofdregelaar beschikt voor het nemen van deze beslissingen over de nodige informatie van zuiver verkeerstechnische aard, alsmede over detektorinformatie betreffende het aktuele verkeersaanbod op de diverse zijwegen. Deze detektorinformatie kan de hoofdregelaar inwinnen bij de diverse lokale regelaars.

Wanneer de hoofdregelaar voor een cyclus bepaald heeft, hoeveel voertuigen er doorgelaten worden, verleent het sturende kruispunt (de poort) het cluster voertuigen toegang tot het tracé. Het idee van het systeem is nu, dat dit

cluster op de rijrichting die gekozen is als voorkeursrichting, in één keer het tracé over gaat, zonder ergens voor een verkeerslicht te hoeven stoppen (uiteraard zolang de hoofdrichting gevolgd wordt).

Dit gaat als volgt in zijn werk. Het sturende kruispunt geeft, om het cluster voertuigen door te laten, een bepaalde groentijd aan de hoofdrichting. Vervolgens wordt deze groentijd doorgeseind naar het volgende kruispunt, en wanneer het cluster voertuigen daar arriveert, aan het verkeer op de hoofdrichting gegeven. Vervolgens wordt op dit kruispunt door detectie vastgesteld of er verkeer het tracé verlaat, of hoeveel verkeer zich van de dwarsrichting bij de hoofdstroom voegt. Op grond van deze gegevens wordt weer een groentijd doorgeseind naar het volgende kruispunt, waarna op dit kruispunt en het daarop volgende weer hetzelfde wordt gedaan.

Zoals verderop zal blijken, worden, om zo adequaat mogelijk op situaties te kunnen reageren, uiteindelijk geen groentijden doorgeseind, maar aantallen voertuigen.

De cyclustijden op alle kruisingen worden, zoals reeds opgemerkt, door de grootte van het begincluster bepaald en zijn steeds wisselend. Een groene golf in twee richtingen is daarom bij dit systeem niet mogelijk. Bovenstaande methodiek wordt toegepast op de voorkeursrichting, en wordt op de tegengestelde richting zo goed als mogelijk is gevolgd.

#### **4.3 De lokale regelaar**

Ieder kruispunt heeft zijn eigen regelaar. Deze handelt de volledige gang van zaken op het kruispunt af. Van andere lokale regelaars, van detectoren op de kruising, of van de hoofdregelaar, komt informatie binnen over verkeersdichtheden, naderende clusters, enzovoort. De regelaar gaat deze informatie verwerken, zodanig dat het naderende verkeer zo efficiënt mogelijk wordt afgehandeld. Naderende clusters voertuigen op de voorkeursrichting moeten geheel worden doorgelaten, voor andere richtingen moeten de wachttijden zo klein mogelijk zijn.

#### **4.4 Interactie tussen lokale regelaars**

Wanneer een andere lokale regelaar informatie over een naderend cluster doorseint, wordt ook het adres van deze regelaar meegegeven, zodat bekend is van welke kant de voertuigen komen. De informatie bestaat uit het aantal auto's waaruit het cluster bestaat, en een tijdsoffset in seconden. Wanneer een tijdsoffset van  $T$  seconden binnenkomt, betekent dit dat op het moment van binnenkomst van deze informatie, over  $T$  seconden het cluster de stopstreep van de kruising genaderd zal zijn.

Zoals verderop nog zal worden besproken, moet de regelaar de informatie een bepaalde tijd  $T_d$  eerder krijgen dan het tijdstip waarop de op de informatie volgende actie moet worden uitgevoerd. Wanneer bijvoorbeeld informatie binnenkomt over naderende voertuigen op de voorkeursrichting, moet aan deze richting over  $T$  seconden groen gegeven worden. De regelaar moet hiervoor echter eventuele groen voerende dwarsfasen op rood zetten, een ontruimingstijd in acht nemen, en eventueel voorstartgroen aan de hoofdrichting geven in het geval hier voertuigen staan te wachten. De tijd die de regelaar hiervoor nodig heeft is hier  $T_d$ . Dit betekent dat de tijdsoffset  $T$  in ieder geval groter moet zijn dan  $T_d$ .

#### **4.5 Nadere beschouwing van de clusterafhandeling**

In deze paragraaf zal wat gedetailleerder worden ingegaan op enkele aspecten van het systeem.

##### **4.5.1 Gedrag van clusters**

We hebben gezien dat, wanneer een cluster auto's een kruispunt gepasseerd is, de betreffende regelaar naar het volgende kruispunt doorseint hoeveel auto's er aan komen, en wanneer. Bij het volgende kruispunt wordt de doorgeseinde groentijd aan het cluster gegeven zodra het daar arriveert. In het ideale geval vertrekt er dus op tijdstip  $t$  een cluster voer-



tuigen bij een kruising, en komt op tijdstip  $t + T$  als één blok bij de volgende kruising aan. Van deze ideale situatie zal in de praktijk geen sprake zijn om de volgende redenen:

- (1) De voertuigen van het cluster zullen verschillende optrektijden hebben, en zullen ook niet allemaal even snel rijden, vooral niet als er per rijrichting meerdere rijstroken zijn. Er zal dan ook een zekere spreiding zijn in de tijdstippen waarop de verschillende voertuigen van een cluster bij het volgende kruispunt aankomen. Men noemt dit verschijnsel *pelotondiffusie* : het cluster voertuigen wordt "verstrooid". Sommige voertuigen zullen dus op een tijdstip bij het volgende kruispunt aankomen dat iets afwijkt van  $t + T$ . Wanneer als doorgeseinde tijdsoffset  $T$  gehanteerd wordt de tijd die het snelste voertuig nodig heeft om de afstand tussen de kruisingen af te leggen, dan zal de aankomsttijd van sommige voertuigen alleen later liggen dan  $t + T$ , niet eerder.
- (2) Het kan ook voorkomen dat sommige voertuigen van een cluster niet een klein beetje later aankomen, zoals onder (1), maar een relatief veel grotere tijd later. Dit kan komen door zaken als halterende bussen op het tracé, of een bezinestation dat tussen twee kruisingen gelegen is, en waardoor voertuigen oponthoud kunnen oplopen.
- (3) Een heel cluster kan veel later dan het tijdstip  $t + T$  bij het volgende kruispunt arriveren. Dit kan komen door bijvoorbeeld een ongeluk op het tracé.

In verband met bovenstaande punten, wordt er bij dit systeem niet zonder meer vanuit gegaan dat een cluster inderdaad een kruispunt bereikt op het door het vorige kruispunt doorgeseinde tijdstip. Het arriveren van de voertuigen op het betreffende kruispunt wordt met behulp van detectoren vastgesteld.

#### 4.5.2 Voorstartgroen

Het kan voorkomen dat er op de hoofdrichting van een kruising voertuigen voor rood licht staan te wachten. Komt er een cluster aan, dan zou de kruispuntregelaar in principe groen geven op het moment dat dit cluster de kruising bijna bereikt heeft. De voor de kruising wachtende voertuigen gooien dan roet in het eten, omdat die nu moeten gaan optrekken, zodat de aankomende voertuigen toch moeten afremmen. Om deze reden geeft een lokale regelaar, wanneer er voertuigen staan te wachten op de hoofdrichting, bij een naderend cluster een hoeveelheid voorstartgroen om de wachtende voertuigen te laten optrekken. De naderende voertuigen kunnen de kruising dan ongehinderd passeren. De situatie dat er op de voorkeursrichting voertuigen voor rood staan te wachten lijkt in tegenspraak met het feit dat op deze richting ieder voertuig ongehinderd het tracé over kan. Het komt echter voor in de volgende twee gevallen:

- \* Een voertuig is afkomstig van de dwarsfase van de vorige kruising, en is dus op die kruising doorgelaten op een tijdstip liggend tussen twee groenperioden van de hoofdrichting. Het wacht dus bij de volgende kruising voor rood.
- \* Een voertuig van een door de vorige kruising doorgelaten cluster heeft oponthoud gehad, of te langzaam gereden. Het komt te laat aan bij de volgende kruising, zodat de hoofdrichting hier inmiddels alweer rood is.

In het geval een voertuig bij een kruising op de hoofdrichting voor rood staat te wachten, is dit de kruispuntregelaar bekend (zie verderop). Bij een volgend naderend cluster wordt voorstartgroen gegeven, en worden de wachtende voertuigen aldus bij het naderende cluster gevoegd. Dit cluster wordt dus groter, en hiermee dient bij het doorseinen naar de volgende kruising rekening gehouden te worden.

### 4.5.3 Passeren van een cluster

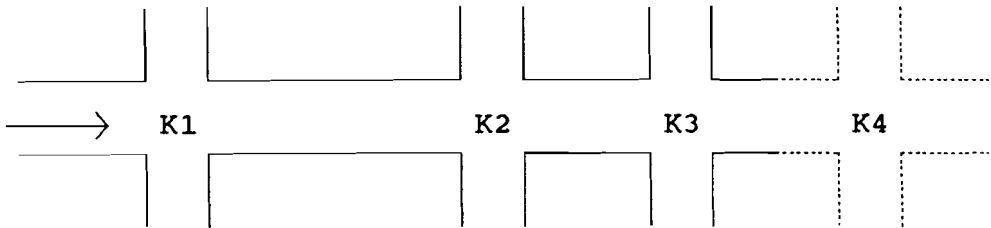
Algemeen geldt dat het aantal voertuigen van een cluster bij ieder kruispunt dat het passeert zal veranderen. Immers, er kunnen op de hoofdrichting wachtende voertuigen bij gevoegd worden, er kunnen voertuigen van het cluster te langzaam zijn zodat ze de groenperiode niet meer halen, en er kunnen voertuigen van het tracé afbuigen. Op dit moment zal men misschien geneigd zijn om ook de van de dwarsfase van het kruispunt afkomstige voertuigen erbij te tellen. Deze voertuigen worden echter niet bij het huidige cluster gevoegd, omdat de dwarsfasen pas groen krijgen als het cluster gepasseerd is en de hoofdrichting dus weer rood heeft. Ze zullen dus bij de volgende kruising voor rood moeten wachten en aldus bij het volgende passerende cluster gevoegd worden.

Wanneer een cluster een kruispunt gepasseerd is, kan de kruispuntregelaar dus de nieuwe grootte ervan bepalen. Deze grootte wordt bepaald door het aantal voertuigen te tellen dat de detektor op de hoofdrichting, die net iets na de kruising ligt, passeert. Deze nieuwe grootte kan vervolgens doorgeseind worden naar de volgende kruising. De regelaar is vervolgens in staat om het aantal voertuigen te berekenen dat nu voor de kruising staat te wachten. Dit aantal is gelijk aan het door de vorige kruising doorgeseinde aantal, minus het aantal voertuigen dat op deze kruising afgebogen is (detektoren op de afbuigende voorsorteervakken), minus het aantal getelde op de hoofdrichting passerende voertuigen. Dit aantal wachtende voertuigen kan in het tijdsinterval tussen twee passerende clusters nog groter worden, omdat de vorige kruising dan zijn dwarsfasen naar de hoofdrichting doorlaat. Dit zal deze kruising dan doorgeven (naderend cluster afkomstig van dwarsfase), en dit kan eventueel nog met een voor de kruising gelegen detektor gecontroleerd worden. De regelaar weet nu hoeveel voertuigen er op de hoofdrichting staan te wachten wanneer het volgende cluster nadert, zodat een evenredige hoeveelheid voorstartgroen gegeven kan worden.

#### 4.5.4 Communicatie

Tot nu toe is er steeds van uit gegaan dat de informatie over naderende clusters steeds op tijd binnenkomt. Wanneer kruisingen dicht bij elkaar gelegen zijn, hoeft dit echter niet het geval te zijn. Immers, wanneer de afstand tussen twee kruispunten klein is, kan de door te seinen tijdsoffset  $T$  kleiner zijn dan de  $T_d$  van de volgende regelaar.  $T_d$  is de minimale tijd die de regelaar de informatie van te voren moet hebben (zie paragraaf 4.4).

Neem de situatie van figuur 4.2.



Figuur 4.2 : Een stukje van een tracé

De afstand tussen de kruisingen  $K_2$  en  $K_3$  is zo klein, dat er problemen optreden. De tijdsoffset  $T$  die  $K_2$  naar  $K_3$  seint, en die gelijk is aan de tijd die een voertuig nodig heeft om van  $K_2$  naar  $K_3$  te rijden, is hier kleiner dan de minimale tijd  $T_d$  die de regelaar van  $K_3$  nodig heeft om, na het ontvangen van de melding dat er een cluster nadert, de hoofdrichting groen te geven.

Conclusie : wanneer  $K_2$  aan  $K_3$  meldt dat er een cluster nadert, is dit te laat. De regelaar van  $K_3$  heeft niet voldoende tijd meer om op een veilige manier de dwarsfasen op rood te zetten. Daarom moet deze melding eerder gebeuren, voordat het cluster bij  $K_2$  is. Op dit moment is echter nog niet bekend hoeveel voertuigen er bij  $K_2$  bij het cluster gevoegd

worden of er vanaf gehaald worden. Wanneer aan K3 gemeld wordt dat er een cluster nadert, zal dus een schatting van de grootte hiervan doorgeseind moeten worden. K3 weet dan alvast dat het er aan komt, alleen de grootte is niet exact bekend. Deze kan na het passeren van K2 als correctie doorgeseind worden. Om te weten welke correctie bij welk cluster hoort, wordt een clusternummer meegezonden. Dit nummer wordt bij ieder volgend cluster cyclisch opgehoogd, met een cyclus van bijvoorbeeld 10.

Uit het voorgaande blijkt dat er maatregelen moeten worden genomen om lokale regelaars steeds tijdig van informatie te voorzien. Dat wil zeggen : in sommige gevallen deze regelaars alvast doorseinen dat er een cluster nadert, waarbij de grootte van dit cluster nog kan veranderen omdat er nog kruisingen tussen liggen.

In het systeem is dit als volgt geregeld. Iedere lokale regelaar weet van zijn beide burens of deze al dan niet te dicht bij liggen. In bovenstaand voorbeeld weet de regelaar van K2 dat K3 te dicht bij ligt. Wanneer K2 nu een melding binnenkrijgt dat er vanuit K1 een cluster nadert, zal deze dit ook meteen naar K2 doorseinen, met een geschatte grootte en tijdsoffset van dit cluster. Nadat het cluster K2 gepasseerd is, zendt K2 een correctie op de grootte en offset naar K3.

In het algemeen geldt dus : iedere regelaar geeft een clustermelding door naar de volgende regelaar indien de afstand tot deze kruising te klein is. "Te klein" wil zeggen : de tijdsoffset  $T$  tussen de kruisingen is kleiner dan de  $T_d$  van de volgende kruising. Zou er na K3 nog een kruising K4 volgen met ook een kleine afstand tot K3, dan wordt de melding als een "emmertje" doorgegeven. K1 seint naar K2, deze seint meteen door naar K3, en deze seint weer meteen door naar K4. Wanneer een cluster K1 gepasseerd is, zijn K2, K3 en K4 zijn dus meteen op de hoogte van het feit dat er een cluster nadert.

Bij de bovenstaande "recursieve" methode kan opgemerkt worden, dat communicatie alleen plaats vindt tussen naast elkaar gelegen kruispunten. Verder hoeft iedere regelaar alleen van zijn beide burens te weten of zij op een te kleine afstand liggen.

#### 4.6 Handelingen bij een passerend cluster

Laten we nu de gang van zaken eens op een rijtje zetten wanneer een cluster voertuigen een kruising passeert op de voorkeursrichting.

Stel : een cluster nadert kruising  $K_i$ . Dit cluster is op tijdstip  $t_1$  bij kruising  $K_{i-1}$  vertrokken. Kruising  $K_{i-1}$  heeft toen aan kruising  $K_i$  doorgeseind dat er op tijdstip  $t_2 = t_1 + T$  een cluster voertuigen ter grootte  $G$  aankomt. Wanneer kruising  $K_{i+1}$  dicht bij ligt, seint  $K_i$  meteen aan deze kruising door dat er voertuigen naderen, met een geschat aantal. De regelaar van kruising  $K_i$  doorloopt nu de hierna volgende sequence.

N.B. In het onderstaande wordt gebruik gemaakt van een aantal gegevens, zoals het te geven voorstartgroen, gegevens over verkeersaanbod op dwarsfasen en tegengestelde richting, enz. Deze gegevens worden uit het geheugen van de regelaar gehaald. Ze worden door andere routines in het geheugen geschreven. Hoe dit gebeurt wordt verderop beschreven.

[1] Tref maatregelen om het cluster op het tijdstip  $t_2$  door te laten. Ofwel : zorg ervoor dat de hoofdrichting op tijdstip  $t_2$  groen kan worden. Indien de dwarsfasen nog verkeersaanbod hebben kunnen deze nog zo lang groen blijven voeren als verantwoord is in verband met de ontruimingstijden. Tevens moet het te geven voorstartgroen worden meegenomen.

[2] Wacht tot het tijdstip  $t_2$  minus het te geven voorstartgroen. Op dit tijdstip voeren alle met de hoofdrichting strijdige fasen rood. Indien er voorstartgroen gegeven

moet worden : geef groen aan de hoofdrichting, zodat de wachtende voertuigen kunnen wegrijden.

[3] Wacht op de detektormelding van het naderende cluster (het gaat hier om de detector die vlak voor het kruispunt gelegen is). In het geval van voorstartgroen was de hoofdrichting al groen, geef anders nu groen. Indien op het wegvak tussen de kruisingen detectoren geplaatst zijn, kan het doorgeseinde aantal voertuigen gecorrigeerd worden, wanneer bijvoorbeeld een aantal voertuigen oponthoud heeft. Tel vervolgens het aantal afbuigende voertuigen (detectoren) en verminder het aantal hiermee. Laat het cluster over de hoofdrichting passeren, en tel de voertuigen.

[4] Bekijk de situatie op de dwarsfasen en op de tegengestelde richting. Wanneer alle voertuigen gepasseerd zijn, kan bij veel verkeer op de dwarsfasen de hoofdrichting op rood gezet worden, of bij veel verkeer op de tegengestelde richting kan deze nog even groen blijven. Wanneer nog niet alle voertuigen van het cluster gepasseerd zijn, en de dwarsfasen hebben een hoog verkeersaanbod, kan besloten worden om de hoofd-fase naar rood te laten gaan. De resterende voertuigen van het cluster waren dan te langzaam en blijven wachten, zij worden zo bij het volgende cluster gevoegd.

[5] Sein het aantal naderende voertuigen door naar de volgende kruising.

[6] Bereken het aantal nu op de hoofdrichting wachtende voertuigen. Dit is gelijk aan:

- # door vorige kruising doorgeseind -
- # afgebogen -
- # gepasseerde (geteld)

Haal uit een tabel het hierbij behorende voorstartgroen, schrijf die op de betreffende plaats in het

geheugen. Dit aantal wachtende voertuigen kan nog groter worden indien de voorliggende kruising voertuigen van de dwarsfase naar de hoofdrichting doorlaat. Wanneer dit gebeurt, zal die kruising dat ook nog doorseinen (zie het volgende punt).

[7] Handel de dwarsfasen af. Sein naar de volgende kruising door hoeveel voertuigen afkomstig van de dwarsfasen via de hoofdrichting de volgende kruising naderen.

[8] Zend een correctie op een eerder gezonden clustergrrootte wanneer de volgende kruising op een kleine afstand gelegen is.

#### **4.7 Processtructuur van de lokale regelaar**

In deze paragraaf wordt beschreven hoe de lokale regelaar opgebouwd is. Dit wordt voornamelijk gedaan in termen van interrupts en processen.

De regelaar werkt op interrupt-basis. Deze interrupts treden allemaal volkomen onafhankelijk van elkaar op, en kunnen in principe zelfs tegelijkertijd optreden. Er is daarom een interruptcontroller aanwezig die al deze interrupts afhandelt.

Iedere regelaar heeft een interne (lokale) lopende tijdvariabele. Deze is dus onafhankelijk van de andere regelaars.

Er is een hoofdprogramma dat continu wordt afgelopen, en een aantal Interrupt Service Routines, die aangeroepen worden door binnenkomende interrupts.

De volgende interrupts kunnen optreden:

- (1) error (van lampbewaking of strijdigheidsmatrix)
- (2) detektormelding
- (3) binnenkomende data van andere regelaar
- (4) binnenkomende data van hoofdregelaar



Interrupt (1) geeft altijd meteen een sprong naar een noodprogramma. De door de interrupts (2), (3) en (4) gestarte processen zullen verderop nader besproken worden. Sommige van deze processen schrijven data in bepaalde geheugenlocaties, die door andere processen weer gebruikt wordt.

#### **4.7.1 Het hoofdprogramma**

Het hoofdprogramma wordt continu door de regelaar doorlopen. Het kan steeds worden onderbroken door optredende interrupts.

Een van die interrupts is de errormelding die wordt gegenereerd door de controle-hardware. Deze controleert continu of geen strijdige fasen tegelijk groen voeren, en of er geen lampen kapot zijn. Wanneer deze interrupt optreedt, wordt meteen naar een noodprogramma gesprongen, dat de taak van het hoofdprogramma volledig overneemt. Op wat dit noodprogramma verder doet wordt hierop niet verder ingegaan. Meestal zal een star noodprogramma worden afgedraaid, of, in geval dit niet meer mogelijk is, oranje knipperlicht worden ingeschakeld.

De regelaar is zodanig opgebouwd, dat het hoofdprogramma de sturing van de lichten verzorgt. Het is te vergelijken met de schakelwals uit vroegere tijden. Het hoofdprogramma leest continu geheugenplaatsen uit waarin de standen van de lichten zijn opgeslagen. Deze geheugenplaatsen worden gevuld door de diverse Interrupt Service Routines. In plaats van rechtsreeks de lichtstanden in het geheugen te schrijven, kunnen sommige routines ook toekomstige lichtstanden in het geheugen plaatsen. Er wordt dan een tijdstip meegegeven waarop deze lichtstand ingaat. Dit tijdstip ligt ten opzichte van de lopende lokale tijvariabele.

Wanneer geen verkeer op de dwarsfasen aanwezig is, staat het systeem in de "wachtstand groen". De hoofdrichting voert dan continu groen. Er treden in dit geval ook geen gebeurtenissen, dus ook geen interrupts op. De geheugenplaatsen vertonen

de "wachtstand groen", het hoofdprogramma geeft zodoende continu groen aan de hoofdrichting.

#### **4.7.2 Detektormelding**

Detektoren zijn geplaatst op de dwarsfasen, en op de hoofdrichting. Detektormeldingen hiervan veroorzaken sprongen naar verschillende Interrupt Service Routines (dit wordt afgehandeld door de interruptcontroller).

Wanneer een detektormelding van een dwarsfase optreedt, schrijft de bijbehorende routine de nodige relevante waarden in het geheugen. Deze geheugenplaatsen worden door andere routines (zie verderop) weer gebruikt.

Detektoren op de hoofdrichting kunnen geplaatst zijn:

- D1 : (eventueel) op het weggedeelte tussen twee kruisingen
- D2 : op de voorsorteervakken voor afbuigend verkeer
- D3 : op de hoofdrichting vlak voor de kruising
- D4 : op de hoofdrichting vlak na de kruising

Melding D1 geeft informatie over het gedrag van de voertuigen van clusters tussen twee kruisingen. Deze informatie wordt in het geheugen geschreven.

Melding D2 schrijft aantallen afbuigende voertuigen in het geheugen.

Melding D3 start routine [3] van paragraaf 4.6 op : de afhandeling van het passerende cluster wordt gestart.

Melding D4 schrijft aantallen passerende voertuigen op de hoofdrichting in het geheugen.

Een en ander is te zien in de flowchart van figuur 4.3.

#### **4.7.3 Bericht van andere lokale regelaar**

De volgende berichten kunnen door een andere lokale regelaar gestuurd worden:

- (B1) aantal voertuigen van naderend cluster + T
- (B2) correctie op eerdere 1) transmissie + T
- (B3) aantal naderende voertuigen komend van dwarsfase

De T bij sommige berichten is de reeds eerder beschreven tijdsoffset.

Wanneer bericht B1 binnenkomt, wordt routine [1] van paragraaf 4.6 gestart : de voorbereidingen voor het laten passeren van het cluster worden genomen. Afhankelijk van het feit of het cluster op de voorkeursrichting nadert of niet, wordt het cluster op het juiste moment doorgelaten, of kan nog even gewacht worden.

De door bericht B2 opgestarte routine corrigeert enige gegevens in het geheugen.

Wanneer B3 binnenkomt, moet het aantal door [6] berekende, voor de kruising wachtende voertuigen met het doorgeseinde aantal worden opgehoogd. Deze waarde in het geheugen wordt vernieuwd.

We krijgen zo de flowchart van figuur 4.4.

#### **4.7.4 Bericht van / naar hoofdregelaar**

De hoofdregelaar kan de volgende berichten sturen / ontvangen:

- (B4) vraag naar aantal voertuigen op hoofd- of dwarsfase
- (B5) antwoord : aantal voertuigen op hoofd- of dwarsfase
- (B6) te verlenen groentijd aan fase + T
- (B7) grootte van door te laten cluster + T
- (B8) voorkeursrichting

Wanneer bericht B4 binnenkomt, worden de geheugenlokaties uitgelezen de gevuld worden bij optredende detektormeldingen. Vervolgens worden de gevraagde aantallen door middel van bericht B5 naar de hoofdregelaar gezonden.

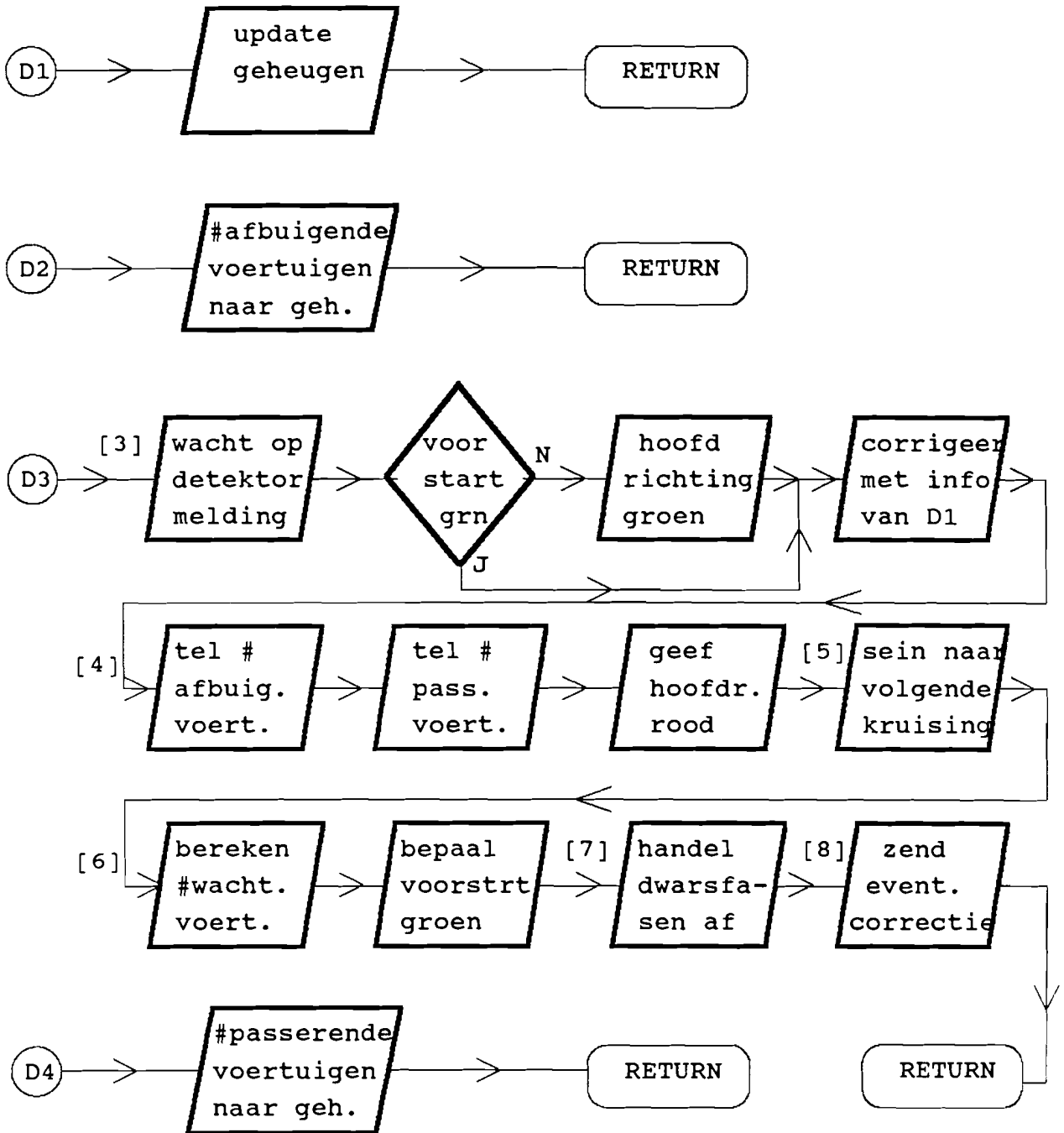
Bericht B6 heeft tot gevolg dat de nodige geheugenlokaties met groentijdstippen overschreven worden. De oude waarde wordt weggegooid, de van de hoofdregelaar afkomstige waarden komen hiervoor in de plaats.

Bericht B7 komt binnen bij de beide poortregelaars. Dit heeft voor de rest hetzelfde effect als bericht B1.

Bericht B8 wijzigt de in de regelaar opgeslagen voorkeursrichting. De nieuwe voorkeursrichting komt in het geheugen te staan. Deze geheugenplaats wordt gebruikt door routine [4] van paragraaf 4.6.

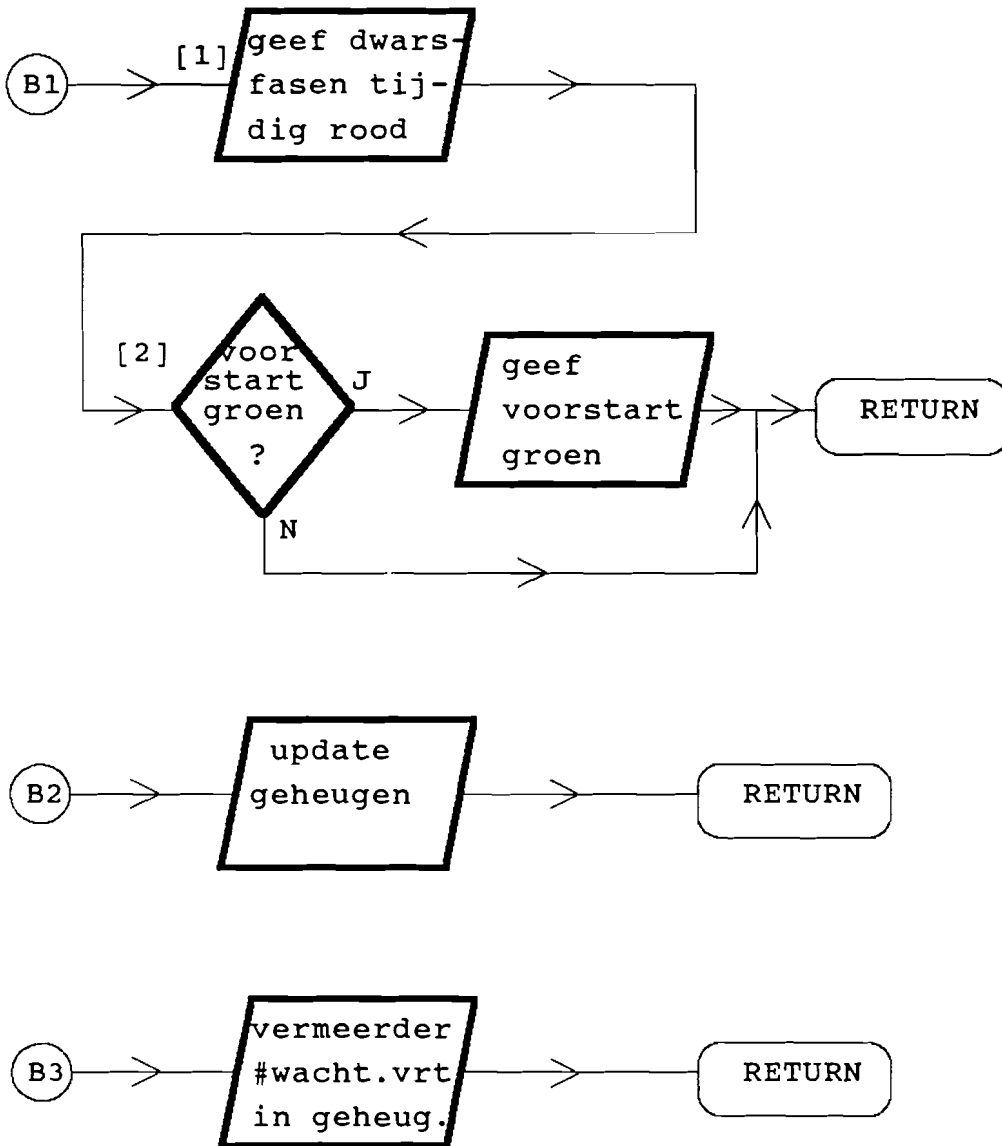
Zie ook de flowchart van figuur 4.5.

Detektor-  
melding:



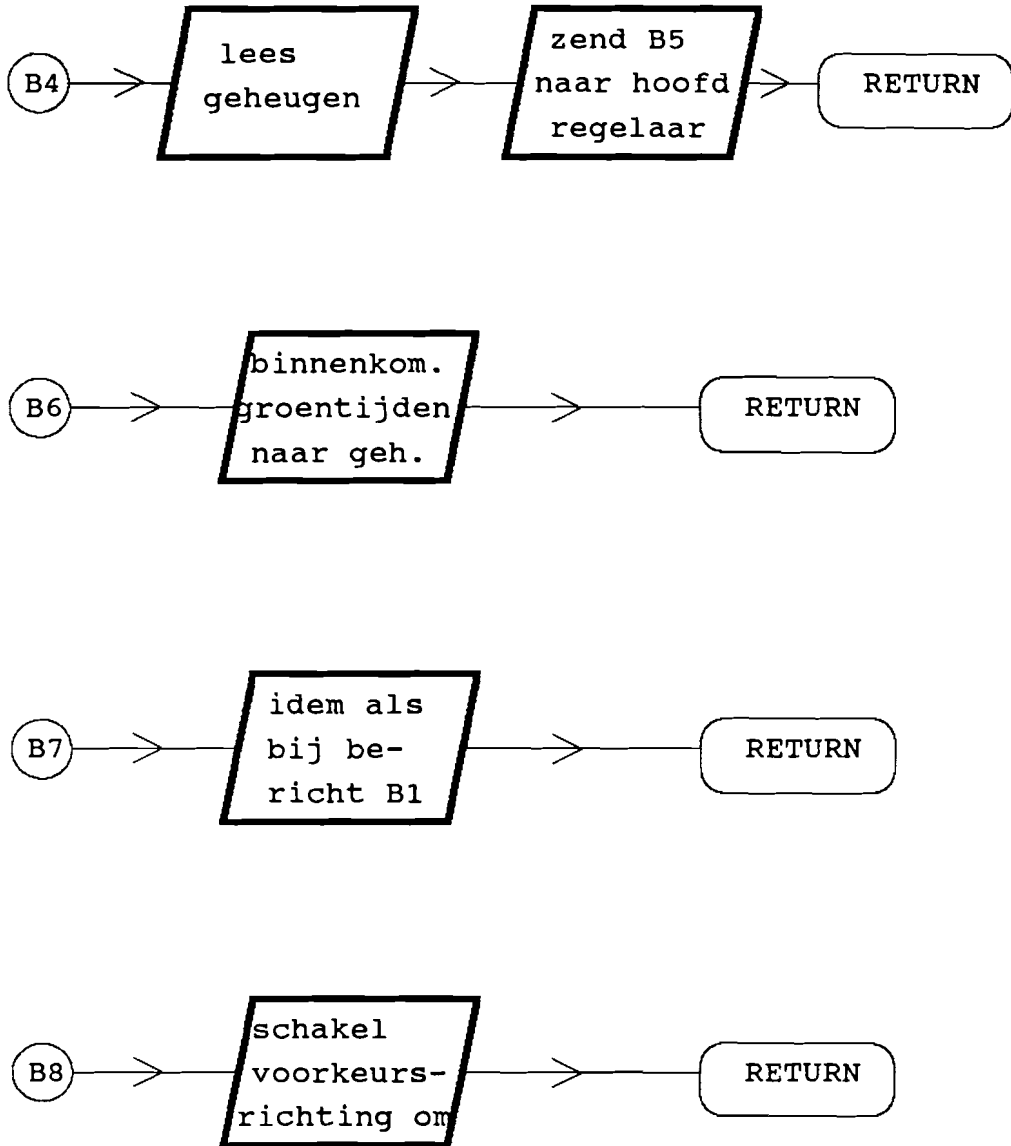
Figuur 4.3 : Processen die worden opgestart door detektor-  
meldingen

Binnenkomend  
bericht:



Figuur 4.4 : Processen die worden opgestart door binnenkomende data van andere lokale regelaars

Binnenkomend bericht:



Figuur 4.5 : Processen die worden opgestart door binnenkomende data van de hoofdregelaar

#### 4.8 De hoofdregelaar

De hoofdregelaar (centrale regelaar) heeft een overzicht over het actuele verkeersverloop op het tracé. Bij iedere lokale regelaar kan de hoofdregelaar informatie opvragen over de situatie op de betreffende kruising. Hiertoe kunnen alle lokale regelaars geadresseerd worden. De lokale regelaar krijgt deze informatie weer binnen door middel van de op de diverse wegvlakken geplaatste detektoren.

In principe regelen de lokale regelaars in onderlinge samenwerking het verkeer over het tracé. De hoofdregelaar heeft tot taak om als een overall-controller het hele tracé in de gaten te houden, en om eventueel in te grijpen als er ergens iets fout dreigt te gaan. Hiertoe heeft de hoofdregelaar de mogelijkheid om iedere lokale regelaar bepaalde groentijden dwingend op te leggen. Een commando van de hoofdregelaar aan een bepaalde lokale regelaar om een bepaalde groentijd te verlenen heeft voorrang boven de door de lokale regelaar zelf vastgestelde groentijd.

Een andere belangrijke taak van de hoofdregelaar is het doseren van de hoeveelheid voertuigen die steeds het tracé op rijdt. Zoals reeds beschreven, worden deze voertuigen door de beide poorten van het tracé doorgelaten. De hoofdregelaar berekent uit alle opgehaalde detektorinformatie hoeveel voertuigen het tracé op mogen, en seint dit door naar de poort.

De hoofdregelaar moet het verkeersaanbod op de dwarsfasen van alle kruisingen beschouwen. Voor het bepalen van de grootte van het begincluster heeft de hoofdregelaar als criterium dat op alle kruisingen de afhandeling van het dwarsverkeer voldoende gewaarborgd is. De bottlenecks zijn hierdoor het meest bepalend voor de door te laten hoeveelheid verkeer.

Omdat de aldus vastgestelde cyclustijden op de kruisingen ook bepalen hoe de tegengestelde richting kan worden afgehandeld,



kijkt de hoofdregelaar bij het bepalen van de grootte van het begincluster op de voorkeursrichting ook of de tegengestelde richting voldoende kan doorstromen. Op deze richting mogen in ieder geval geen opstoppingen ontstaan. Deze kunnen ontstaan ofwel doordat de poort van de voorkeursrichting door plotse-ling sterk veranderende verkeersaanboden toch te weinig voertuigen doorgelaten blijkt te hebben dat de tegengestelde richting te weinig groen aan de hoofdrichting kan geven, ofwel doordat de poort van deze tegengestelde richting teveel voertuigen heeft doorgelaten. In dit geval grijpt de hoofdregelaar in.

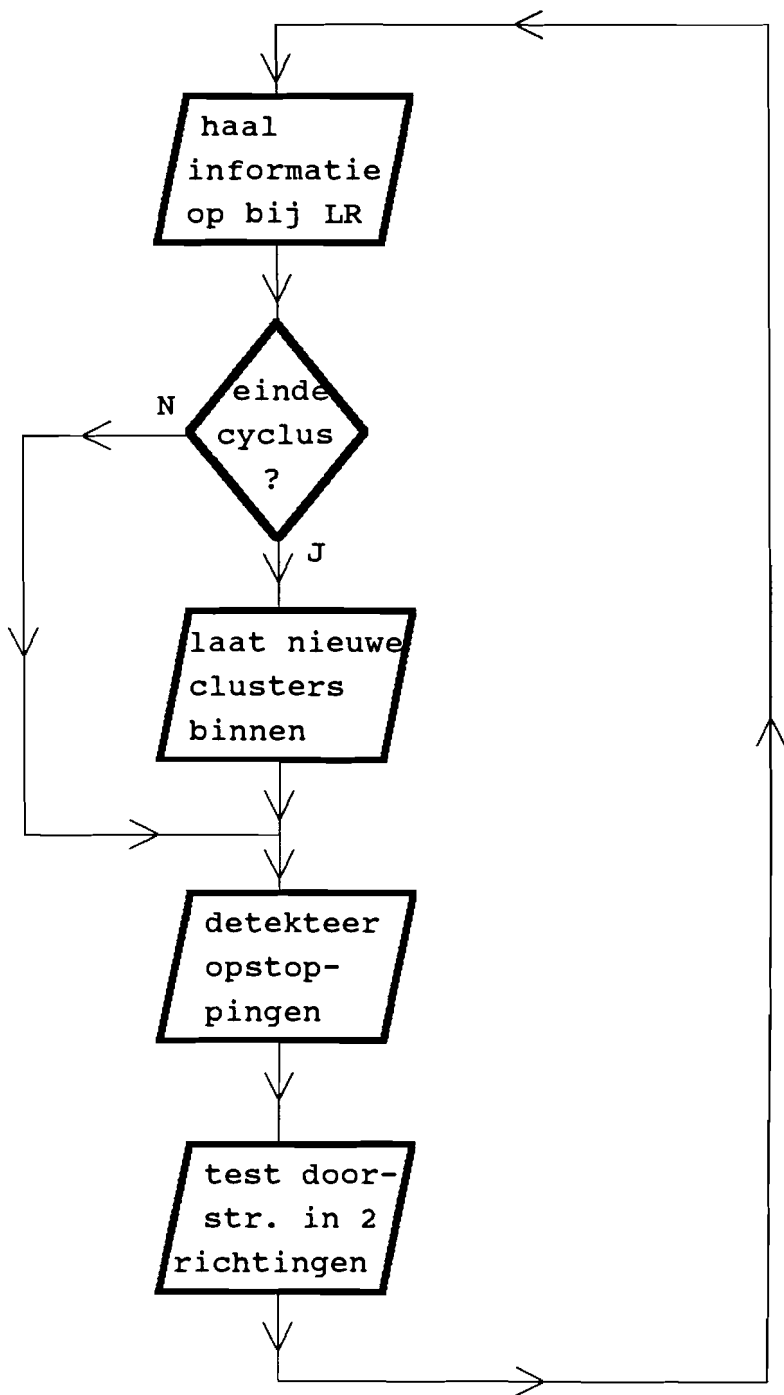
Aan de andere kant kan in sommige gevallen de verkeerssitua-tie dermate gunstig zijn dat in beide richtingen volledige doorstroming bereikt kan worden. Ook in dit geval kan de hoofdregelaar extra commando's naar lokale regelaars zenden.

#### **4.8.1 Processen in de hoofdregelaar**

De hoofdregelaar handelt de volgende zaken af:

- (1) Ophalen van informatie over verkeersdichtheden bij alle lokale regelaars.
- (2) Uit de gegevens van (1) bepalen hoeveel voertuigen in beide richtingen het tracé op mogen, dit aantal door-seinen naar de beide poorten.
- (3) In de gaten houden of op de niet-voorkeursrichting geen grote wachttijden of opstoppingen ontstaan. Zonodig ingrijpen door lokale regelaars extra groentijden door te seinen.
- (4) In de gaten houden of het verkeerspatroon dermate gunstig is (weinig verkeersaanbod) dat een volledige doorstroming in twee richtingen mogelijk is. In dat geval kan de hoofdregelaar de groentijden van sommige kruisingen nog iets aanpassen.

In de hoofdregelaar loopt maar een proces, dat bovenstaande punten sequentieel doorloopt. Zie de flowchart van figuur 4.6.



Figuur 4.6 : Proces van de hoofdregelaar

Eerst worden bij alle lokale regelaars de aktuele verkeersdichtheden opgehaald. Hieruit kan worden vastgesteld of er weer een nieuwe cyclus kan beginnen, dat wil zeggen of de poorten weer een volgend aantal voertuigen tot het tracé mogen toelaten. Indien dit het geval is, worden de aantallen naar de poorten doorgeseind. Vervolgens worden de slagen (3) en (4) afgelopen, waarna de informatie over de verkeersaanboden weer geaktualiseerd kan worden.

Het zal in de praktijk zo zijn dat het toelaten van nieuwe clusters ongeveer om de 60 sekonden gebeurt. In deze tijd kan de loop een aantal malen doorlopen worden. Bepalend voor dit aantal is vooral slag (1), omdat hierbij alle regelaars moeten worden afgelopen. De snelheid van het communicatienet is hier dus van groot belang. In de paragraaf over de communicatie (4.9) zal de te gebruiken baudrate hierop berekend worden.

#### **4.9 De communicatie**

Voor de communicatieverbinding tussen de kruispuntregelaars onderling en de hoofdregelaar kunnen twee mogelijkheden in beschouwing worden genomen.

- (1) Een datanet waarbij iedere regelaar data kan sturen naar iedere willekeurige andere regelaar. Iedere regelaar heeft dan een adres aan de hand waarvan het ontvangende station geselecteerd kan worden. Uiteraard is een protocol nodig om dit alles in goede banen te leiden.
- (2) Iedere twee opeenvolgende kruispuntregelaars zijn met elkaar verbonden. Bij kruispuntregelaars onderling kan dus alleen regelaar  $n$  informatie zenden naar regelaar  $n-1$  en  $n+1$ . Verder loopt er nog een verbinding van de hoofdregelaar langs alle kruispuntregelaars, zodat de hoofdregelaar alle kruispuntregelaars kan adresseren en naar iedere regelaar data kan sturen of data kan opvragen.

Mogelijkheid (1) is de elegantste. Het systeem is hierbij het meest flexibel, terwijl een minimale lengte aan kabel nodig is. Het vergt echter wel extra hard- en software voor de implementatie van het dataprotocol, terwijl transmissietijden langer kunnen zijn omdat het datanet maar één transmissie tegelijk kan verwerken. Er kunnen dus wachttijden ontstaan.

Omdat het in het hier beschreven systeem alleen voorkomt dat twee opeenvolgende regelaars met elkaar communiceren, kan voor mogelijkheid (2) gekozen worden. De protocollen kunnen hierbij eenvoudig gehouden worden. Transmissies tussen opeenvolgende kruispuntregelaars kunnen onafhankelijk van elkaar plaatsvinden, terwijl de hoofdregelaar met iedere kruispuntregelaar kan communiceren. Het nadeel is uiteraard dat meer kabel nodig is : ongeveer 2 maal zoveel als bij (1). Wanneer een relatief goedkoop netwerk volgens (1) beschikbaar is, verdient het vanwege de grotere flexibiliteit van het systeem aanbeveling om deze mogelijkheid te kiezen.

In beide gevallen wordt een seriële verbinding gebruikt.

#### **4.9.1 Data formaat**

Voor de lengte van zowel data als adressen is gekozen voor 8 bit.

De berichten die kunnen worden overgezonden zijn de berichten B1 t/m B8. Ieder bericht bestaat uit een aantal bytes. Het eerst byte van en bericht bestaat uit het adres van de regelaar waar het naar toe moet. Hierna volgt een byte met een code die aangeeft welk van de berichten B1..8 het betreft. Hierna volgen eventueel nog data bytes, afhankelijk van het type bericht.

De data die wordt overgezonden bij de hierboven genoemde transmissies:

- B1 : code, aantal, T, clusternr : 4 bytes
- B2 : code, aantal, T, clusternr : 4 bytes
- B3 : code, aantal : 2 bytes

- B4 : adres, code, fassenr : 3 bytes
- B5 : code, fassenr, aantal : 3 bytes
- B6 : adres, code, groentijd, T : 4 bytes
- B7 : adres, code, clustergrootte, clusternr, T : 5 bytes
- B8 : adres, code, voorkeursrichting : 3 bytes

Wanneer voor de communicatieverbinding mogelijkheid (1) gekozen zou worden, moet bij het bepalen van de tijdsoffset T rekening gehouden worden met een vertraging die de verbinding veroorzaakt. Stel dat regelaar A een tijdsoffset van 10 seconden verzendt naar regelaar B. Dit betekent dat vanaf het moment van verzenden de offset van 10 seconden geldt, ofwel : de offset is relatief ten opzichte van het moment dat A wil verzenden. Wanneer de vertraging van het datanet ervoor zorgt dat regelaar A bijvoorbeeld een seconde moet wachten voordat de offset kan worden overgezonden, moet nu een offset van 9 seconden worden overgezonden. Uiteraard zal een dermate grote vertraging alleen optreden wanneer de baudrates laag liggen.

#### **4.9.2 Bepaling van de baudrate**

Wanneer voor de communicatie mogelijkheid (1) gekozen wordt (zie paragraaf 4.9), moet beschouwd worden hoe groot de dichtheid van de transmissies over het datanet is, omdat alles over dezelfde lijn gaat. Bij mogelijkheid (2) verlopen de transmissies tussen lokale regelaars parallel.

Bij een systeem van N lokale regelaars kan ruwweg voor elke van de hierboven genoemde transmissies het aantal malen dat deze per cyclus voorkomt afgeschat worden:

- (B1) : 2N
- (B2) : N
- (B3) : N
- (B4) : 2N
- (B5) : 2N
- (B6) : komt weinig voor
- (B7) : 2
- (B8) : komt weinig voor
- \_\_\_\_\_ +

8N+2 transmissies gemiddeld per cyclus.

(Het aantal malen B1 is afgeschat op 2N door het meestal aanwezig zijn van dicht op elkaar liggende kruispunten - dus doorgeven van naderende clusters aan volgende kruisingen treedt op)

Hieruit kan men ruwweg berekenen hoe groot de gemiddelde vertragingstijd van het netwerk is. Dit levert dan weer eisen op met betrekking tot de te gebruiken baudrate.

Wanneer communicatieverbinding (2) gekozen wordt, wordt de minimaal te gebruiken baudrate bepaald door de te eisen maximale tijd die een transmissie nodig heeft.

Transmissie tussen lokale regelaars:

Stel dat de kleinst mogelijke afstand tussen twee kruisingen 10 meter bedraagt. Bij een rijsnelheid van 100 km/h (uiteraard zeer ruim genomen) bedraagt de rijtijd tussen deze kruisingen 0.4 seconde. Er kan dan geëist worden dat transmissies (B1), (B2) en (B3) niet langer dan 0.05 seconde mogen duren. Lengte van deze transmissies : 4 bytes = 32 bits. Met eventuele foutcorrigerende bits en start/stop bits bestaat de transmissie uit 60 bits. Een baudrate van 1200 baud is hier dus bruikbaar : de transmissie duurt dan maximaal 0.05 seconde.

Transmissies tussen hoofdregelaar en lokale regelaar:

Hierbij zijn transmissies (B4) en (B5) van belang. Stel : er zijn 255 lokale regelaars (maximum) en de gemiddelde cyclustijd is 60 seconden. In deze tijd moet de hoofdregelaar alle regelaars langsgaan. Lengte 3 bytes. Met start/stop bits en foutcorrigerende informatie : 50 bits. Bij een baudrate van 1200 baud duurt zo'n transmissie dus 0.04 seconde. Bij 255 regelaars en steeds transmissie (B4) en (B5) neemt dit een tijd van  $2 \times 10 = 20$  seconden. Dit is acceptabel bij een cyclustijd van 60 seconden.

De voor een goede werking van het systeem minimaal benodigde baudrate is dus : 1200 baud.

Uiteraard kan ook een hogere baudrate gekozen worden.

## **Hoofdstuk 5 : real-time netwerkregeling**

In het voorgaande hoofdstuk is gekeken naar een real-time verkeersregeling op een enkel tracé. Dit tracé bestond uit een enkele keten van kruispunten, zodat ieder kruispunt op zijn hoogst twee burens heeft. Dit concept kan worden uitgebreid naar een netwerk van wegen en met verkeerslichten geregelde kruispunten.

### **5.1 Netwerkdefinitie**

We hebben dus een netwerk van wegen en kruisingen. Ieder kruispunt wordt gemodelleerd als een knooppunt in het netwerk. Zo'n knoop heeft een aantal ingangen en een aantal uitgangen. De wegen tussen de diverse kruispunten zijn uiteraard de verbindingen tussen in- en uitgangen van knopen. Het netwerk heeft in- en uitgaande wegen. Deze sluiten aan op kruispunten die aan de rand van het netwerk gelegen zijn.

Voor dit netwerk willen we nu een real-time verkeersregeling ontwerpen. In de nu volgende paragraaf zal eerst worden gespecificeerd wat deze regeling allemaal moet doen, en zullen tevens enige aan de regeling te stellen eisen worden opgesteld.

### **5.2 Specificatie en eisen**

We gaan uit van het zojuist gespecificeerde netwerk. We willen hiervoor een regelsysteem ontwerpen waarmee het verkeer in het netwerk optimaal geregeld wordt.

We moeten nu eerst definiëren wat we onder "optimaal" verstaan.

Beschouw eerst een enkele verkeersstroom. We zullen een verkeersstroom definiëren als zijnde optimaal geregeld wanneer de gemiddelde wachttijd per voertuig minimaal is. Het gaat hier om de tijd die een voertuig gemiddeld voor rood licht staat te wachten wanneer het aan de verkeersstroom deelneemt.



In een netwerk lopen vele verkeersstromen. Sommige hiervan zullen een hogere bezettingsgraad hebben dan andere. Neem bijvoorbeeld het geval van een drukke doorgaande weg door een stad met rondom de nodige kleinere straten. Het zal meestal zo zijn dat men er meer belang aan hecht dat de gemiddelde wachttijden op deze doorgaande weg klein zijn dan dat dit het geval is op de omliggende wegen. Men zal meestal wensen dat er op de doorgaande weg een volledige "groene golf" aanwezig is, terwijl op de omliggende wegen wachttijden voor lief worden genomen. In het algemeen is het dus zo dat bepaalde verkeersstromen een hogere prioriteit hebben dan andere.

Met het voorgaande kunnen we een optimaal geregeld netwerk definiëren als een netwerk waarvan de *gewogen* som van de gemiddelde wachttijden van de diverse verkeersstromen minimaal is. De gewichtsfactoren in dit gemiddelde zijn dan evenredig met de prioriteit die de verkeersstroom heeft. En deze prioriteit zal in het algemeen groter zijn naarmate de stroom drukker is.

We hebben nu gedefinieerd wat we onder een optimaal geregeld netwerk verstaan. De te ontwerpen regeling moet het gegeven netwerk optimaal regelen.

Het regelsysteem moet het mogelijk maken om een willekeurige route door het netwerk als "groene golf" route te definiëren. Voor voertuigen die deze route volgen is dan gegarandeerd dat ze deze route kunnen afleggen zonder een keer voor rood licht te hoeven stoppen. We zullen deze route de *voorkeursroute* noemen. Voor de resterende routes kunnen gewichtsfactoren aan het systeem worden aangeboden.

De eigenschappen van het netwerk (geometrie, toegestane rijnsnelheden, verkeersdichtheden, prioriteiten van stromen) moeten als parameters aan het systeem kunnen worden aangeboden. Verder moet de intelligentie liefst zoveel mogelijk gedecentraliseerd worden, om de kans op uitval van grote stukken van de regeling zo klein mogelijk te maken.

### 5.3 Opzet van het systeem

In deze paragraaf wordt besproken, hoe het systeem globaal is opgezet.

#### 5.3.1 Configuratie

Evenals bij het systeem beschreven in het vorige hoofdstuk, wordt iedere kruising, ofwel iedere knoop in het netwerk voorzien van een lokale regelaar. De lokale regelaars seinen informatie over naderende voertuigen aan elkaar door. Ze zijn daartoe onderling gekoppeld door middel van een communicatienet, waarbij een regelaar iedere willekeurige andere regelaar kan adresseren.

De intelligentie is op deze manier gedecentraliseerd. Alle informatie over het netwerk, geometrisch en verkeerstechnisch, is opgeslagen in de diverse kruispuntregelaars. We kunnen hier onderscheid maken tussen vaste informatie, welke tijdens het functioneren van de regeling niet of zelden zal veranderen, en *veranderlijke* informatie.

Vaste informatie betreft bijvoorbeeld de geometrie van het netwerk. Denk aan gegevens over de wegen tussen kruisingen, zoals lengte en toegestane snelheid. Veranderingen in de eerstgenoemde zullen uiteraard zeer zelden optreden. Wanneer op een wegvak een andere snelheid wordt toegestaan, zal de nieuwe waarde hiervan in de regelaar moeten worden opgenomen.

Er zijn ook grootheden die op bepaalde momenten, bijvoorbeeld op bepaalde tijdstippen van de dag, kunnen veranderen. Hieronder valt bijvoorbeeld de voorkeursroute die door het netwerk gedefinieerd is. Vanuit bijvoorbeeld een centrale commandopost kan op ieder moment een bepaalde voorkeursroute vastgelegd worden. Dit wordt dan naar de lokale regelaars doorgeseind. Hierin is immers opgeslagen hoe de voorkeursroute loopt. Voor het doseren van de hoeveelheid verkeer die op de voorkeursrichting wordt toegelaten, is, net als in het

systeem van het vorige hoofdstuk, een hoofdregelaar aanwezig. Deze valt samen met de zojuist genoemde commandopost.

### 5.3.2 Regelstrategie

Continu veranderend is uiteraard het verkeersaanbod. Zoals reeds opgemerkt, hebben de diverse regelaars hier continu onderling contact over, zoals mag worden verwacht van een real-time verkeersregeling.

Iedere kruispuntregelaar handelt het verkeer op de betreffende kruising af. Hierbij heeft op iedere kruising een bepaalde verkeersstroom, de *voorkeursstroom*, de hoogste prioriteit. De afhandeling van alle andere verkeersstromen op de kruising wordt *ten opzichte van* deze voorkeursstroom gedaan. Hoe dit in zijn werk gaat wordt in de volgende paragrafen beschreven.

Een belangrijk punt is, dat de lokale regelaar zijn hele afwikkeling van de diverse fasen ophangt aan de melding van naderend verkeer op de voorkeursstroom van de betreffende kruising. Voor deze stroom *moeten* dus meldingen binnenkomen. Voor niet aan de rand van het netwerk gelegen kruisingen kan gesteld worden dat deze meldingen steeds door de voorliggende kruising(en) gestuurd worden. Van de aan de rand van het netwerk gelegen kruisingen is er één de kruising waar de voorkeursroute begint. De hoofdregelaar bepaalt voor deze route de grootte van het begincluster. Voor de andere kruisingen worden de meldingen gegenereerd door detektoren. Hoe verder deze detektoren van de kruising gelegen zijn, des te verder van te voren heeft de regelaar informatie over naderend verkeer en des te optimaler kan er dan geregeld worden.

### 5.4 De hoofdregelaar

De hoofdregelaar heeft een soortgelijke taak als die in het systeem dat in het vorige hoofdstuk beschreven werd. Ook hier wordt de gang van zaken op de voorkeursroute geregeld. Het verschil is hier echter, dat in het voorgaande systeem

slechts twee voorkeursrichtingen mogelijk waren (de beide hoofdrichtingen), terwijl dat er hier veel meer zijn.

De hoofdregelaar weet, hoe de voorkeursroute door het netwerk loopt. Dit is door een hiertoe bevoegd persoon vastgesteld. Deze persoon kan de hoofdregelaar op ieder moment een andere voorkeursroute opgeven. De hoofdregelaar zorgt er dan voor dat deze informatie naar de juiste lokale regelaars wordt doorgeseind.

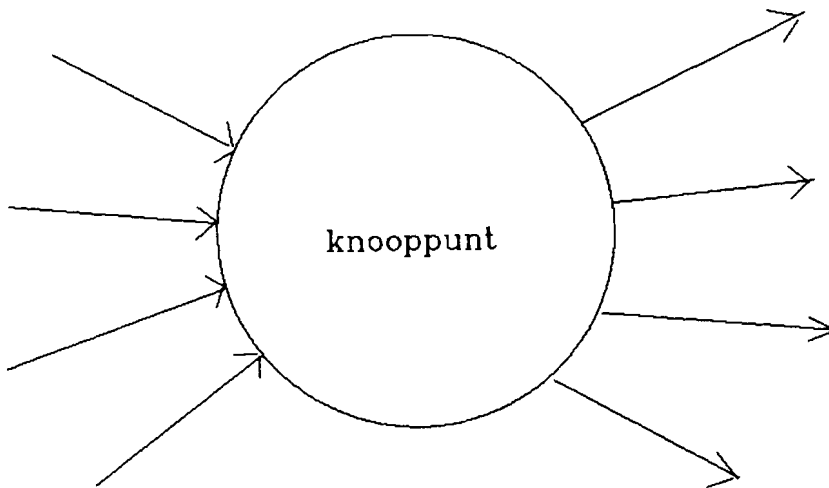
Op dezelfde manier als in het vorige hoofdstuk beschreven, wordt bij alle lokale regelaars waar de voorkeursroute langskomt, informatie opgevraagd over de belasting van de fasen die strijdig zijn met de hoofdrichting. Aan de hand van deze informatie wordt bepaald hoe groot het cluster voertuigen wordt dat op de voorkeursroute wordt toegelaten. Voertuigen van dit cluster kunnen deze route afleggen zonder voor rood licht te hoeven stoppen, zolang de aangegeven snelheid aangehouden wordt. Op deze manier wordt de afhandeling van het verkeer op de kruisingen waar de voorkeursroute langsloopt, sterk beïnvloed door de grootte van het cluster voertuigen dat op deze route wordt toegelaten.

### **5.5 Knooppunt uit een netwerk**

Beschouw een willekeurige knoop van het netwerk. Zoals uit het voorgaande blijkt, komt deze overeen met een kruispunt. Dit kruispunt wordt geregeld door een lokale regelaar.

ingangen

uitgangen



Figuur 5.1 : Willekeurig knooppunt in een netwerk.

Figuur 5.1 laat een knoop zien. Een knoop heeft een aantal ingangen, en een aantal uitgangen. De kruising is de "doorlaat" tussen de in- en uitgangen : tussen bepaalde ingangen en uitgangen zijn verkeersstromen mogelijk.

- \* Deze verkeersstromen zijn in het algemeen niet tussen alle in- en uitgangen mogelijk. Alle *mogelijke* overgangen van ingang naar uitgang zijn vastgelegd in de *stromen* 1..N.
- \* Een aanvoerweg naar een kruising splitst zich meestal in een aantal voorsorteevakken. Ieder van deze voorsorteevakken wordt beschouwd als een *ingang* van de knoop. Het aankomende verkeer verdeelt zich over deze voorsorteevakken met een voor de kruising min of meer vaste verdelingskans voor ieder vak.
- \* De uitgaande verkeersstromen (de afvoerwegen) van de kruising zijn de *uitgangen* van de knoop.

Uiteraard heeft de regelaar de beschikking over de nodige vaste gegevens.

- \* De ingangen corresponderen met de voorsorteervakken op de aanvoerwegen. Van iedere ingang is de kans opgeslagen dat het aankomende verkeer dit vak kiest.
  - \* De uitgangen corresponderen met de afvoerwegen. Van het stuk weg tot de volgende kruising zijn opgeslagen:
    - lengte
    - toegestane rijsnelheid
    - capaciteit (het maximale aantal voertuigen dat per tijdseenheid op een bepaalde plaats kan passeren)
- Wanneer de betreffende afvoerweg het netwerk uit gaat, wordt hier in plaats van bovenstaande gegevens een code opgeslagen die dit aangeeft.
- \* Van de N stromen zijn opgeslagen:
    - van welke ingang naar welke uitgang
    - prioriteit

Met dit laatste wordt aangegeven welke prioriteit de betreffende verkeersstroom heeft. De N stromen hebben een oplopende prioriteit van 1 tot N. Hiermee wordt de in paragraaf 5.2 genoemde prioriteit van stromen in het netwerk in de diverse regelaars gerepresenteerd. Hoe hoger het prioriteit-nummer, hoe minder "belangrijk" de stroom is. Dit zal vervolgens neerkomen op : hoe langer de gemiddelde wachttijd voor rood voor voertuigen van de betreffende stroom is. De stroom met prioriteit 1 is de *voorkeursstroom*. Verkeer van deze stroom zal altijd ongehinderd worden doorgelaten. Wanneer de kruising in de in paragraaf 5.2 genoemde voorkeursroute is opgenomen, zal de stroom die de voorkeursroute over de kruising representeert de voorkeursstroom zijn. Stromen over de kruising kunnen ook een gelijke prioriteit hebben. Ze hebben dan hetzelfde prioriteits-nummer.

De kruispuntregelaar ontvangt van de omliggende regelaars informatie over naderende clusters voertuigen. Deze informatie bestaat uit de grootte van deze clusters, en het tijdstip dat de voertuigen de kruising bereiken zullen hebben. Samen

met de verdelingskansen van de diverse ingangen weet de regelaar dan hoe zwaar de diverse stromen belast gaan worden. Verder kan ook nog informatie binnen komen afkomstig van detectoren, gelegen op wegvlakken in de buurt van de kruising.

### 5.6 Nadere beschouwing van de regelstrategie

Tussen lokale regelaars wordt informatie doorgeseind over clusters voertuigen die de kruising naderen. Een regelaar weet zo een bepaalde tijd van tevoren dat er een op tijdstip  $t$  een cluster voertuigen ter grootte  $G$  op een bepaalde aanvoerweg de kruising bereikt. Hoe dit wordt doorgeseind wordt verderop gedetailleerder behandeld. Met behulp van de verdelingskansen van de aanvoerweg naar de diverse bij die weg behorende ingangen (voorsorteervakken), is dus bekend hoeveel voertuigen er op iedere ingang aankomen op tijdstip  $t$ .

Op het moment dat deze informatie voor alle fasen binnen is, wordt een optimalisatie-proces gestart. Er wordt nu berekend hoe de regelaar de naderende voertuigen het meest efficiënt kan gaan afhandelen. Hiermee wordt bedoeld : de afhandeling waarbij de wachttijden minimaal zijn. Er wordt op de volgende punten gelet:

- [1] Prioriteiten van de stromen. De eis voor minimale wachttijden wordt voor een stroom sterker naarmate de prioriteit hiervan hoger is.
- [2] Verkeersintensiteiten. Hoe groter een naderend cluster op een bepaalde stroom is, hoe groter uiteraard de hiervoor benodigde groentijd.
- [3] Nadertijden van clusters. Een cluster dat pas over  $T$  seconden de kruising bereikt, hoeft uiteraard ook niet eerder groen te krijgen.

Op het moment dat het optimalisatie-proces gestart wordt, ziet het probleem er dus als volgt uit. Er zijn  $N$  verkeersstromen over de kruising, die allemaal op een bepaald moment verkeersaanbod krijgen. Noem deze tijdstippen, waarop de clusters de kruising bereiken de aankomsttijden  $a_1$  tot  $a_N$ . Deze clusters hebben grootten  $G_1 \dots G_N$ . De kruising moet deze verkeersstromen nu allemaal laten passeren, waarbij het echter een onvermijdelijk gegeven is dat bepaalde stromen strijdig zijn, dat wil zeggen dat voertuigen hiervan niet gelijktijdig op het kruisingsvlak mogen verblijven.

Het optimalisatieproces moet nu voor de  $N$  stromen een set groen- en roodtijdstippen  $g_1 \dots g_N$  resp.  $r_1 \dots r_N$  gaan berekenen zodanig dat met de strijdigheid en ontruimingstijden tussen de stromen rekening wordt gehouden, en dat tevens de wachttijden minimaal zijn. Dit probleem hadden we ook al bij de berekening van een optimale starre regeling in paragraaf 2.4.

Uit het voorgaande blijkt dat het probleem nu neerkomt op het in de tijd plaatsen van een aantal groenintervallen. Met de intervallen  $[g_i, r_i]$  ( $i$  van  $1..N$ ) moet zodanig geschoven worden dat aan bovengenoemde eisen voldaan is. Het interval  $i$  begint op een tijdstip  $g_i$  dat samenvalt met of later ligt dan de aankomsttijd  $a_i$ .

Neem nu een bepaalde stroom over de kruising. Het optimalisatie-proces plaatst voor deze stroom een groeninterval  $[g_i, r_i]$  in de tijd. Afhankelijk van deze plaats komen de doorgelaten voertuigen eerder of later bij de volgende kruising aan. Het tijdstip  $g_i$  van een stroom heeft dus invloed op het aankomsttijdstip  $a_j$  van de doorgelaten voertuigen bij de volgende kruising.

Het blijkt dus dat de te berekenen optimale groentijdstippen van een kruising mede worden bepaald door de groentijdstippen van de omliggende kruisingen. Het ligt dus voor de hand om in het optimalisatie-proces ook de omliggende kruisingen te betrekken. Het wordt dan mogelijk om de aankomsttijdstippen



$a_i$  niet meer als een vast gegeven te beschouwen. Een en ander gaat dan als volgt in zijn werk.

De lokale regelaar berekent eerst de set groen- en roodtijdstippen zoals hierboven beschreven werd. Hierbij zijn de aankomsttijdstippen  $a_i$  nog niet in beschouwing genomen. Vervolgens wordt voor iedere  $g_i$  gekeken of deze groter is dan  $a_i$ . De groentijd mag immers nog niet gegeven worden wanneer de betreffende voertuigen de kruising nog niet bereikt hebben. Is een bepaalde  $g_i$  kleiner dan de  $a_i$ , dan moet het betreffende interval in de tijd vooruit verschoven worden. Daar alle andere intervallen van dit interval afhankelijk zijn, moet nu opnieuw geoptimaliseerd worden. In deze tweede optimalisatie-slag worden de tijdstippen  $a_i$  wel meegenomen, zodat de nu berekende set groen- en roodtijden wel bruikbaar is.

Wat is nu het nut van de eerste optimalisatie-slag? Wel, de hier berekende groentijdstippen  $g_i$  zijn in feite de ideale aankomst-tijdstippen. In het ideale geval zouden de aankomst-tijdstippen namelijk precies samenvallen met de berekende groentijdstippen. De tweede optimalisatie-slag is in dat geval niet nodig.

Wat er nu gebeurt is dat de regelaar na de eerste optimalisatie-slag een voorkeurstijdstip  $v_i$  naar de voorliggende regelaar doorzendt. Dit is het ideale aankomsttijdstip, ofwel de bij de eerste optimalisatie-slag berekende  $g_i$ . De voorliggende regelaar kan de binnenkomende tijdstippen  $v_i$  dan gebruiken bij de optimalisatie van de volgende cluster-afhandeling. Deze tijdstippen leveren dus een extra set ongelijkheden waaraan de  $g_i$ 's moeten voldoen om de aankomst-tijdstippen voor de volgende kruising zo gunstig mogelijk te leggen. Uiteraard wordt er dan wel vanuit gegaan dat de verkeerssituatie de volgende cyclus precies hetzelfde is, met andere woorden dat de aankomende clusters even groot zijn als in de vorige cyclus. Dit is in het algemeen uiteraard niet het geval, daar de verkeerssituatie steeds verandert. Deze veranderingen zullen echter zelden abrupt zijn, maar relatief langzaam in de tijd verlopen. De voorkeurstijdstippen afkomstig van omliggende kruisingen zullen daarom in het

algemeen meehelpen aan een efficiëntere afwikkeling van het verkeer op de kruisingen.

## **5.7 De lokale regelaar**

We kunnen nu definiëren wat een kruispuntregelaar tijdens normaal bedrijf allemaal moet doen.

### **5.7.1 Algemeen**

In het proces van het verwerken van clusters voertuigen kunnen twee duidelijke hoofdfuncties onderscheiden worden:

- (1) Op het moment dat van alle fasen de informatie over naderend verkeer binnen is, moet het optimalisatieproces gestart worden. Daar de afstanden tot de omliggende kruisingen verschillend zullen zijn, zullen de tijdstippen waarop deze informatie binnen komt random in de tijd verspreid liggen. Er moet dus een mechanisme gemaakt worden dat er voor zorgt dat de regelaar op het juiste moment over de informatie van alle fasen beschikt.
- (2) Het eigenlijke optimalisatieproces. Dankzij het zojuist genoemde mechanisme heeft dit proces op het moment dat het wordt opgestart de beschikking over grootte en moment van aankomst van clusters voertuigen op alle fasen van de kruising. Het gaat nu de set groen- en roodtijdstippen (voor definitie hiervan zie paragraaf 2.4.1) voor alle fasen berekenen. Dit programma wordt afgedraaid op het moment dat de eerste voertuigen van de voorkeursstroom de kruising bereiken.

De functies (1) en (2) zullen in de paragrafen 5.7.2 resp. 5.7.3 nader uitgewerkt worden.

Het zal blijken dat bepaalde details hetzelfde zijn als in het systeem beschreven in het vorige hoofdstuk. Er zal dan ook af en toe naar dit hoofdstuk verwezen worden.

### 5.7.2 Coördinatie in de tijd

In paragraaf 5.3.2 is reeds vermeld dat het hele verloop van de verkeersafwikkeling op een kruispunt geshiedt ten opzichte van de clustermelding op de voorkeursstroom. Wanneer een melding binnenkomt dat er hier een cluster nadert, wordt in principe het optimalisatie-proces gestart. Over alle andere stromen moet dan ook informatie binnen zijn.

Om dit laatste te bereiken, is in iedere regelaar opgeslagen welke omliggende kruisingen moeten worden voorzien van informatie. Er ontstaan hierbij dezelfde problemen als die beschreven in paragraaf 4.5.4. Wanneer de afstanden tot de volgende kruising groot genoeg zijn, wordt per stroom steeds informatie doorgeseind naar de volgende kruising. Zijn deze afstanden echter klein, dan worden ook de verder liggende kruisingen van informatie voorzien. Dit omdat het optimalisatie-proces op tijd de nodige informatie binnen moet hebben. Er is in de lokale regelaar dus een proces dat we P1 zullen noemen, dat steeds relevante data naar omliggende regelaars verstuurt. Deze data betreft clusters voertuigen. P1 weet hoe groot een cluster is, wanneer het de kruising gepasseerd is. Deze grootte wordt op dezelfde manier berekend als beschreven in paragraaf 4.5.3. Het tijdstip waarop het cluster bij de volgende kruising zal aankomen, berekent P1 uit de lengte van en de te voeren snelheid op de wegen tussen de kruisingen.

De bij een regelaar binnenkomende data wordt opgeslagen in het geheugen van de regelaar. Er is een gebied in het geheugen waarin voor elke stroom het aankomsttijdstip van het eerstvolgende cluster, en de grootte hiervan wordt opgeslagen. Er is hiervoor een proces P2 dat alle binnenkomende data opvangt, en in het geheugen opslaat. Deze data betreft dan naderende clusters, en door de voorliggende regelaar doorge-seinde voorkeurstijdstippen. Net zoals in paragraaf 4.5.4, kan het bij dicht op elkaar liggende kruispunten voorkomen dat, nadat over een cluster de eerste keer informatie is binnengekomen, hierop later nog een correctie volgt. De data in het geheugen wordt dan gewoon overschreven. Wanneer op een

gegeven moment het optimalisatie-proces gestart wordt, wordt de data gebruikt die dan in het geheugen staat. Het geheugen is dan geblokkeerd voor eventueel nieuw binnenkomende data.

Voor het verzenden in binnenhalen van data van andere lokale regelaars zijn er dus de processen P1 en P2. Deze processen lopen onafhankelijk van elkaar.

### 5.7.3 Het optimalisatie-proces

Op een gegeven moment wordt het optimalisatie-proces gestart. Dit proces gebruikt de door P2 in het geheugen geschreven data, en een aantal vaste gegevens. Belangrijke vaste gegevens zijn

- \* de ontruimingstijdematrix  $O$ . Hierin zijn de minimaal benodigde ontruimingstijden tussen alle stromen opgeslagen. Er geldt : Het matrixelement  $O_{ij}$  is de ontruimingstijd tussen stroom  $i$  en stroom  $j$  wanneer  $i$  rood wordt en  $j$  groen. In het algemeen is  $O_{ij}$  ongelijk aan  $O_{ji}$ . Wanneer stromen niet strijdig zijn kan dit bijvoorbeeld gecodeerd worden door  $O_{ij}$  negatief te maken.

- \* De prioriteitstabel  $P$ .  $P_i$  geeft de prioriteit aan van stroom  $i$ . Voor de voorkeursstroom  $k$  geldt dus :  $P_k = 1$ .

Wanneer het optimalisatie-proces start, zijn de aankomsttijdstippen  $a_1..a_N$  en de clustergrootten  $G_1..G_N$  bekend. Hieruit kunnen via een tabel de benodigde doorlaattijden voor de clusters worden bepaald. Deze tijdsintervallen  $T_1..T_N$  zijn de minimaal benodigde tijden om het cluster geheel door te laten.

Het optimalisatie-proces moet nu de set groen- en roodtijdstippen  $g_1..g_N$  resp.  $r_1..r_N$  gaan berekenen. Het proces bestaat, zoals in paragraaf 5.6 werd beschreven, uit twee delen. Het eerste proces berekent een set groen- en roodtijden, waarbij de aankomsttijdstippen van de diverse clusters nog buiten beschouwing worden gelaten.

Om te beginnen moeten, omdat de clusters allemaal worden doorgelaten, moeten alle roodtijdstippen later liggen dan de bijbehorende groentijdstippen:

$$g_i \leq r_i \quad (i = 1..N) \quad (1)$$

De hele cyclus kan niet onbeperkt duren. Hiervoor is een bepaald maximumtijd  $t_m$  in de regelaar opgeslagen. De roodtijdstippen liggen dus eerder dan  $t_m$ :

$$r_i \leq t_m \quad (i = 1..N) \quad (2)$$

De groenintervallen hoeven de benodigde doorlaattijden  $T_i$  niet te overschrijden:

$$r_i - g_i \leq T_i \quad (i = 1..N) \quad (3)$$

Afhankelijk van de prioriteit van de betreffende stroom wordt het cluster geheel doorgelaten, of kan de groentijd korter zijn:

$$r_i - g_i \geq \frac{N - P_i + 1}{N} \cdot T_i \quad (i = 1..N) \quad (4)$$

De conflictrelaties kunnen we als volgt representeren (zie ook paragraaf 2.4) :

$$g_j - r_i \geq O_{ij} \cdot \delta_{ij} \quad (i = 1..N) \quad (5)$$

$$g_i - r_j \geq O_{ji} \cdot \delta_{ij} \quad (1 = 1..N) \quad (6)$$

De set  $g_i$ 's en  $r_i$ 's moet dus in ieder geval aan de bovenstaande ongelijkheden voldoen. Net als in paragraaf 2.4 zijn er ook hier weer zeer veel oplossingen mogelijk. Hieruit moet dan nog een meest optimale gekozen worden.

Uiteraard willen we gegeven de maximum tijd  $T_m$  zoveel mogelijk groen hebben:

$$\text{maximaliseer } \sum_i \frac{(r_i - g_i)}{P_i} \quad (7)$$

De afwikkelingstijd moet zo klein mogelijk zijn:

$$\text{minimaliseer } (r_i)_{\max} \quad (8)$$

Ook moet nog rekening gehouden worden met de door omliggende regelaars doorgeseinde voorkeurstijdstippen:

$$\text{minimaliseer } \sum_i |g_i - v_i| \quad (9)$$

Het tweede optimalisatie-proces doet hetzelfde als het eerste, alleen worden hierbij ook nog de aankomsttijdstippen  $a_1..a_N$  in beschouwing genomen. We hebben hier dus de hierboven genoemde ongelijkheden, plus nog een extra ongelijkheid. Deze ongelijkheid drukt uit dat alle groentijdstippen  $g_i$  later moeten liggen dan de bijbehorende aankomsttijdstippen  $a_i$  :

$$g_i \geq a_i \quad (i = 1..N) \quad (10)$$

Verder moeten de groentijdstippen zo dicht mogelijk bij de aankomsttijdstippen liggen:

$$\text{minimaliseer } \sum_i \frac{(g_i - a_i)}{P_i} \quad (11)$$

We zien, net als in paragraaf 2.4, dat de beide optimalisatie-processen een stelsel lineaire ongelijkheden moeten oplossen, en bepaalde grootheden moeten maximaliseren of minimaliseren. Ook hier is weer sprake van een Lineair Programmerings Probleem. We hebben hiervoor het proces P3, dat in een bepaalde tijd het LP-probleem oplost. Hiervoor maakt P3 gebruik van de in het geheugen opgeslagen data, en van de vaste gegevens. Wanneer de eerste voertuigen van de voorkeursstroom de kruising bereiken, worden de door P3 berekende groentijden aan de diverse stromen gegeven. Na deze afwikkeling kan voor de volgende clusters weer een nieuw optimalisatie-proces gestart worden.

#### **5.7.4 Processen in de lokale regelaar**

In de lokale regelaar lopen diverse processen. Drie hiervan, de processen P1, P2 en P3 zijn hiervoor beschreven. Verder zijn er nog de processen die de nodige "administratieve" handelingen verrichten. Hieronder valt bijvoorbeeld het op aanvraag van de hoofdregelaar doorsturen van detektorinformatie, en het op commando van de hoofdregelaar veranderen van voorkeursstroom.

De regelaar is qua structuur opgebouwd als die beschreven in hoofdstuk 4. Ook hier vinden we een hoofdprogramma dat de "schakelwals" funktie vervult, en een aantal Interrupt Service Routines.

De exacte invulling van de in de regelaar lopende processen zal nog moeten plaatsvinden. In het voorafgaande is slechts globaal de opzet van het systeem beschreven.

## Hoofdstuk 6 : Conclusies

Gedurende de afstudeerperiode is een zeer globale opzet voor een systeem voor real-time netwerkregeling gemaakt. Hierin zijn nog vele zaken opengelaten. Deze zijn deels van verkeerstechnische aard, en deels van computertechnische aard. Voordat een prototype van het systeem ontwikkeld kan worden, zal nog het nodige onderzoek verricht moeten worden. Belangrijke punten die nog zijn opengelaten zijn bijvoorbeeld het regelgedrag van de hoofdregelaar, waar een verkeersdeskundige aan te pas zal moeten komen, en een implementatie van het algoritme ter oplossing van het Lineaire Programmerings Probleem.

Het systeem voor real-time verkeersregeling op een tracé is iets gedetailleerder beschreven. Ook hier echter wordt nergens het niveau van hardware of microprogrammering bereikt.

Aan het begin is een gedeelte van de tijd besteed aan het maken van een overzicht van de tot nu toe ontwikkelde verkeersregelsystemen. Een klein deel van deze kennis is gebruikt bij het opzetten van de real-time regelingen. In beginsel echter werken deze regelingen volgens geheel andere uitgangspunten.



## Literatuur

- [1] " Berekening van verkeerslichtenprogramma's met behulp van lineaire programmering "  
De Ingenieur, 24 februari 1967 Pagina V11-18  
A.J. Dekkers, A. van Duuren, F.A. Lootsma, J. Vlietstra
- [2] " Verkeersafhankelijke verkeerslichten 8 -  
De verkeersafhankelijke groene golf "  
Verkeerstechniek 1966 nr.12 Pagina 562  
J. van der Meer