

MASTER

Performance analyse van een hiërarchisch gestructureerd computernetwerk door middel van simulatie

Nunnink, C.J.

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

Technische Universiteit Eindhoven

Fakulteit der Technische Natuurkunde

Performance analyse van een hiërarchisch
gestructureerd computernetwerk door
middel van simulatie.

C.J.Nunnink

FI/FIV 88-14

Afstudeerverslag

augustus 1988

Afstudeerhoogleraar : prof. ir. M.P.J. Stevens
(fakulteit der Elektrotechniek)

Contactprofessor : prof. dr. A.J. van der Wal
(fakulteit der Technische Natuurkunde)

Begeleiders : ir. A.G.M. Geurts
(fakulteit der Elektrotechniek)

ir. M. Swennenhuis
(Philips)

Voorwoord

Voor het uitvoeren van mijn afstudeeronderzoek heb ik de voorkeur gehad voor een opdracht op het gebied van de computercommunicatie. Dit is één van de onderzoeksgebieden van de vakgroep Digitale Systemen van de fakulteit der Elektrotechniek van de Technische Universiteit Eindhoven. Deze vakgroep staat onder leiding staat van prof.ir. M.P.J. Stevens.

Vervolgens is mij de mogelijkheid geboden mijn afstudeeronderzoek in het bedrijfsleven te verrichten, namelijk bij het Projecten Centrum Geldrop van Philips. Van deze mogelijkheid heb ik gaarne gebruik gemaakt.

Op deze plaats wil ik een woord van dank uitspreken aan ir. M. Swennenhuis (Philips) en ir. A.G.M. Geurts (fakulteit der Elektrotechniek) die mij tijdens mijn afstudeeronderzoek hebben begeleid. Verder aan mijn afstudeerhoogleraar prof.ir. M.P.J. Stevens en aan mijn contactdocent bij de fakulteit der Technische Natuurkunde prof.dr. A.J. van der Wal (vakgroep Fysische Informatica).

Tenslotte wil ik al die medewerkers van het Projecten Centrum Geldrop bedanken die mij geïnformeerd hebben over de aanwezige systemen en het gebruik van verscheidene faciliteiten en mij geholpen hebben bij het verrichten van mijn werkzaamheden.

Samenvatting

Het ECHO-project richt zich op het automatiseren van complexe, arbeidsintensieve, administratieve taken binnen grote kantooromgevingen. Het project moet een systeem ontwikkelen voor het zo volledig en efficiënt mogelijk beheersen van de activiteiten van en de informatiestromen tussen de medewerkers in dergelijke kantoren. Dit systeem omvat naast een gecompliceerde software-applikatie een hiërarchisch gestructureerd computernetwerk. Over dit netwerk vindt een karakteristiek gegevensverkeer plaats in de vorm van grote aantallen omvangrijke imagebestanden. Teneinde inzicht te verkrijgen in de performanceaspecten van deze computernetwerken is daarvoor, in het kader van dit afstudeerwerk, een model opgesteld in de vorm van een wachtrijnetwerk op basis waarvan het simulatieprogramma ECHOSIM is geschreven.

ECHOSIM simuleert het gegevenstransport op een afzonderlijk segment van het hiërarchisch gestructureerde computernetwerk onder de typische belasting van het herhaaldelijk door werkstations inlezen van imagebestanden die op fileservers zijn opgeslagen. De tijdsduur van iedere leesactie wordt bepaald en na afloop van de simulatie worden statistische resultaten voor deze responsetijden uitgevoerd.

De tijdsduur van een leesactie omvat, indien meerdere werkstations tegelijkertijd een bestand inlezen, behalve de servicetijden bij de fileserver voor het aanleveren van het bestand, bij het netwerk voor transporteren ervan en bij het workstation voor het ontvangen ervan, tevens de wachttijden bij de fileserver en het netwerk daar dit systeemcomponenten zijn waarvan tijdens de leesacties door de werkstations gemeenschappelijk gebruik gemaakt wordt.

De wachttijd is afhankelijk van de omvang van de door de werkstations in het computernetwerk geïntroduceerde belasting. Deze belasting wordt bepaald door het aantal actieve werkstations, de frequentie waarmee deze werkstations leesacties uitvoeren en de grootte van de ingelezen bestanden.

Voor de parametrisering van het simulatieprogramma zijn met behulp van een network analyzer traces verzameld van het pakketverkeer op het netwerk van het Projecten Centrum Geldrop indien één of meerdere werkstations bestanden vanaf de fileserver inlezen.

Een trace geeft een overzicht van de gedurende een bepaalde periode over het ethernet getransporteerde pakketten waarin voor ieder pakket onder meer het tijdstip wordt gegeven waarop het op het ethernet aanwezig is geweest. De tijdsintervallen tussen opeenvolgende pakketten in een trace kunnen in bepaalde gevallen geïnterpreteerd worden als responsetijden van de fileserver of het workstation en aldus gebruikt worden voor de parametrisering van ECHOSIM. Dit programma simuleert het gegevenstransport over het computernetwerk namelijk op het niveau van de afzonderlijke ethernet-pakketten.

Voor de validering van ECHOSIM is gebruik gemaakt van een programma dat op maximaal twaalf werkstations een leesproces initieert. Ieder proces leest herhaaldelijk bestanden van de fileserver in en bepaalt de voor iedere leesactie vereiste tijdsduur.

Deze tijdsduur wordt samen met overige relevante gegevens omtrent de leesactie naar een logbestand weggeschreven op basis waarvan statistische resultaten voor de tijdsduur van de leesacties bepaald kunnen worden. Deze resultaten zijn vergeleken met de resultaten die het simulatieprogramma produceert indien hierbij een overeenkomstige configuratie en belasting wordt ingevoerd. De vergelijking van deze resultaten heeft een beoordeling opgeleverd van de nauwkeurigheid waarmee ECHOSIM het gedrag van werkelijke systemen simuleert.

Tenslotte zijn simulaties verricht voor meer uitgebreide configuraties dan die van het op het Projecten Centrum Geldrop aanwezige computernetwerk waarvoor statistische resultaten omtrent de responsetijden niet verkregen kunnen worden door het aan dit netwerk verrichten van metingen vanwege het beperkte aantal aanwezige werkstations. Op basis van deze resultaten alsmede de eisen die aan de responsetijden bij de leesacties worden gesteld, kan een oordeel geveld worden over het aantal werkstations dat bij deze typische belasting op een enkel segment van het netwerk kan worden aangesloten. Dit aantal bepaalt direkt het aantal voor een bepaalde organisatie vereiste segmenten van het computernetwerk alsmede het aantal koppelingen tussen deze segmenten.

In het simulatieprogramma zijn zeker nog niet alle aspecten van het gegevensverkeer geïmplementeerd die in ECHO-systemen aanwezig zijn. Er kunnen nog uitbreidingen op en verfijningen in worden aangebracht. Sommige daarvan worden bij de evaluatie van het simulatieprogramma besproken. Er kan geconcludeerd worden dat het programma ook in zijn huidige vorm reeds een redelijke functionaliteit heeft.

Inhoudsopgave

1. Inleiding	1
1.1. De afstudeeropdracht	3
2. ECHO : Electronic Case Handling in Offices	4
2.1. Het ECHO-project	4
2.2. ECHO-systemen	6
2.2.1. Performanceaspecten van ECHO-systemen	6
2.2.2. De logische structuur van ECHO-systemen	8
2.2.3. De hardwarecomponenten van ECHO-systemen	11
2.2.4. De gedistribueerde gegevensopslag	14
2.2.5. Het Network File System NFS	16
2.3. Het PCG-netwerk	20
3. Theorie	22
3.1. Modelling	22
3.2. Wachtrijnetwerken	22
3.2.1. Analytische aanpak van wachtrijsystemen	24
3.2.1.1. Analytische aanpak voor afzonderlijke wachtrijen	25
3.2.1.2. Analytische aanpak voor netwerken van wachtrijen	27
3.2.2. Simulatie van wachtrijnetwerken	28
3.2.3. Simulatietalen	31
4. ECHO-systeemmodellering en modelsimulaties	33
4.1. Modelling op casefolder-niveau	33
4.2. Modelling op ethernet-pakketniveau	36
4.3. Het gebruikte ECHO-systeemmodel	38
5. Metingen	41
5.1. Doelstellingen van de metingen	41
5.1.1. Metingen voor de modellering van ECHO-systemen	41
5.1.2. Metingen voor de parametrisering van het systeemmodel	42
5.1.3. Metingen voor de validering van de modelsimulaties	43
5.2. De meetinstrumenten	44
5.2.1. Het meetprogramma Simulation	45
5.2.1.1. De meetmethode en de meetomstandigheden	46
5.2.1.2. De cache-problematiek	48

5.2.2. De Excelan Ethernet Network Analyzer	50
5.2.3. De Sun-utilities traffic en etherfind.	52
5.3. Traces van het pakketverkeer op het ethernet	53
5.3.1. Traces van afzonderlijke leesacties	55
5.3.2. Traces van simultaan plaatsvindende leesacties	57
5.3.3. Meetresultaten van de traces	59
6. Het simulatieprogramma ECHOSIM	62
6.1. Algemene beschrijving van ECHOSIM	62
6.2. De parametrisering van ECHOSIM	63
6.3. De structuur van ECHOSIM	65
6.3.1. Het read-control-besturingsproces	66
6.3.2. Het token-proces	68
6.3.3. De servicetijden bij de fileserver	70
6.4. Een evaluatie van ECHOSIM	71
7. De resultaten van metingen en simulaties	75
7.1. De verrichte metingen	75
7.2. De verrichte simulaties	76
7.3. Resultaten van metingen en overeenkomstige simulaties	78
7.4. Resultaten van de uiteindelijke simulaties	83
8. Conclusies	86
Literatuurverwijzingen	88
Bijlage 1 ECHO-terminologie	90

1. Inleiding.

Deze afstudeeropdracht is verricht bij het Projecten Centrum Geldrop (PCG) van Philips. Het PCG is een dependance van het Philips Natuurkundig Laboratorium. Binnen dit centrum wordt aan verscheidene projecten uit verschillende vakgebieden gewerkt. Hierbij zijn zowel medewerkers van de researchafdelingen als uit de produkt-divisies van Philips betrokken met het doel resultaten van research sneller te integreren in de produkt(voor)ontwikkeling.

De looptijd van een projekt in het PCG bedraagt in het algemeen één tot enkele jaren. Na het bereiken van een prototype-fase wordt een projekt overgenomen door de betrokken produkt-divisie waar de verdere ontwikkeling tot een commercieel produkt plaatsvindt.

Tot de in voorgaande jaren op het PCG aanwezige projecten behoren onder meer :

- het DIVAC-glasvezelverbindingsprojekt,
- het PHILAN-glasvezel-LAN-projekt en
- het CARIN-verkeersgeleidingsprojekt.

Deze afstudeeropdracht is verricht binnen het ECHO-projekt. ECHO staat voor "Electronic Case Handling in Offices".

Het ECHO-projekt richt zich op het zo volledig en efficiënt mogelijk beheersen van de activiteiten en de informatiestromen binnen grote kantooromgevingen, in het bijzonder bij organisaties waar arbeidsintensieve, administratieve handelingen verricht worden met een sterk reproductierend karakter.

De daar gehanteerde werkwijzes zijn veelal nog sterk papiergeoriënteerd met als belangrijkste nadelen een relatief trage werkafhandeling en een relatief lage efficiency. Deze nadelen zijn te wijten aan het tijdsverlies dat optreedt bij het opzoeken, kopiëren, transporteren en archiveren van gegevens binnen de organisatie.

Het ECHO-projekt wil een alternatieve werkwijze mogelijk maken onder meer door het papier als gegevensdrager te vervangen door een elektronisch equivalent opdat zowel de opslag, het transport als de archivering van gegevens op efficiëntere wijze kunnen plaatsvinden. Hiertoe dienen de kantoormedewerkers in een organisatie voorzien te worden van professionele werkstations die samen met fileservers, waarop de gegevens zich bevinden, zijn opgenomen in een netwerk, waarover het gegevens-transport zich voltrekt. Alle acties in dit systeem vinden plaats binnen het raamwerk van de gedistribueerde ECHO-applikatie. Dit geheel wordt in het vervolg aangeduid als een ECHO-systeem.

Een belangrijk deel van de bedrijfsgegevens zal in ECHO-systemen in de vorm van images aanwezig zijn. Images worden verkregen door het scannen van documenten. Zo wordt de door de organisatie ontvangen post onmiddellijk bij binnenkomst gescand. Deze images vormen relatief grote bestanden waardoor zeker in grote organisaties, waar enige honderden medewerkers bij de administratieve werkzaamheden betrokken kunnen zijn, hoge eisen gesteld zullen worden aan de gegevensopslag- en

-transportcapaciteit van een ECHO-systeem.

Een ECHO-systeem moet zeker ook in dergelijke omvangrijke organisaties significante efficiencyverbeteringen opleveren. De mate waarin dit gerealiseerd wordt, is direkt afhankelijk van de performance van het totale ECHO-systeem. De performance is derhalve voor ECHO-systemen een faktor van cruciaal belang.

Over de performanceaspecten van deze grootschalige, gedistribueerde systemen voor de automatisering van complexe taken binnen kantooromgevingen zijn op applicatieniveau nog nauwelijks resultaten bekend. Evenmin is er sprake van een volledig inzicht in de aard en de omvang van de parameters die van invloed zijn op deze performanceaspecten. Binnen het ECHO-project wordt daarom onderzoek verricht op het gebied van de performance van ECHO-systemen teneinde in staat te zijn deze te optimaliseren.

De performance van een ECHO-systeem is zeker afhankelijk van de configuratie ervan. Deze configuratie omvat de netwerkstructuur, de types en de plaatsing van de daarin aanwezige systeemcomponenten alsmede de distributie van de bedrijfsgegevens over de verschillende daarvoor in aanmerking komende systeemcomponenten zoals bijvoorbeeld de fileservers.

De configuratie zal behalve door performanceaspecten ook bepaald worden door factoren als betrouwbaarheid, beveiliging van bedrijfsinformatie, handhaving van dataconsistentie en de complexiteit van het systeembeheer. Bij deze afstudeeropdracht heeft de relatie tussen de performance en de configuratie van een ECHO-systeem centraal gestaan.

De performance van een ECHO-systeem, in termen van responsetijden en verwerkingscapaciteit, zal aan bepaalde nader vast te stellen randvoorwaarden dienen te voldoen. Op basis van aannames omtrent de belasting die afzonderlijke medewerkers in een ECHO-systeem genereren moeten voor specifieke configuraties performanceresultaten kunnen worden afgeleid die een bijdrage leveren bij de bepaling van geschikte systeemconfiguraties.

Het afleiden van deze performanceresultaten is gedaan door middel van het opstellen van een model voor ECHO-systemen in de vorm van een wachtrijnetwerk en het vervolgens doorvoeren van simulaties op basis van dit systeemmodel.

Op het PCG is een netwerk aanwezig dat gebruikt wordt voor de ontwikkeling van de ECHO-applicatie. Aan dit netwerk zijn performancemetingen verricht onder belastingen als in een ECHO-systeem. De resultaten van deze metingen dienen voor de parametrisering en de validering van het systeemmodel en de modelsimulaties.

Bij de aanvang van het afstudeerwerk was weliswaar een softwarepakket voor simulatiedoeleinden aanwezig maar werd de geschiktheid daarvan betwijfeld. Op grond hiervan is de eerste tijd gebruikt voor een onderzoek van bestaande simulatiepakketten. De uiteindelijke simulaties zijn verricht met het pakket SIMENV afkomstig van PTDS te Hilversum.

De indeling van dit verslag is als volgt :

Hoofdstuk 2 bevat een beschrijving van het ECHO-project, van de performanceaspecten van ECHO-systemen alsmede van de binnen deze systemen aanwezige hardware- en softwarecomponenten. Tevens worden enige van de voor het ECHO-project specifieke termen geïntroduceerd. Deze termen zijn nog eens samengevat in bijlage 1.

Hoofdstuk 3 behandelt theoretische aspecten. Dit betreft de modellering op basis van wachtrijnetwerken, de analytische aanpak daarvan alsmede simulaties op basis van deze netwerken.

Hoofdstuk 4 beschrijft het proces dat doorlopen is bij het opstellen van een systeemmodel voor ECHO-systemen alsmede de aannames die daarbij gedaan zijn.

In hoofdstuk 5 worden de aard en de doelstelling van de metingen besproken die verricht zijn voor de parametrisering van het systeemmodel en voor de validering van de modelsimulaties. Hierbij komen eveneens de gebruikte meetinstrumenten, de meetmethoden en de meetomstandigheden aan de orde. De meetresultaten, die dienen voor de parametrisering van de modelsimulaties, zijn in dit hoofdstuk opgenomen.

Hoofdstuk 6 behandelt de structuur en de werking van het op basis van het systeemmodel geschreven simulatieprogramma ECHOSIM.

Hoofdstuk 7 specificeert de metingen, die voor de validering van de modelsimulaties zijn verricht, en de verrichte simulaties en bevat een overzicht van de resultaten van deze metingen en simulaties.

Hoofdstuk 8 bevat de conclusies.

1.1. De afstudeeropdracht.

De letterlijke formulering van de afstudeeropdracht is als volgt :

Binnen een project, gericht op geïntegreerde kantoorautomatisering met nadruk op de coöperatieve afhandeling van complexe taken binnen kantooromgevingen, is onduidelijk op welke manier grootschalige, gedistribueerde systemen geconfigureerd dienen te worden. Hierbij bestaan met name vragen ten aanzien van de performance, security en betrouwbaarheid.

Tijdens de opdracht dient, gebruikmakend van een aantal aanwezige simulatiegereedschappen, het volgende gerealiseerd te worden :

1. Hoe kan een dergelijk systeem in een model beschreven worden, welke zijn de benodigde parameters en hoe groot zijn deze?
2. Simulaties dienen te worden uitgevoerd die, gebruikmakend van bovengenoemd model, op eenvoudige wijze antwoord kunnen geven op vragen betreffende de door een klant gewenste configuratie.

2. ECHO : Electronic Case Handling in Offices.

2.1. Het ECHO-project.

Het ECHO-project richt zich op het zo volledig en efficiënt mogelijk beheersen van de activiteiten en de informatiestromen binnen grote kantooromgevingen. In het bijzonder bij organisaties waar arbeidsintensieve, administratieve handelingen verricht worden met een sterk reproducerend karakter. Met name wordt hierbij gedacht aan de werkzaamheden bij grote verzekeringsinstellingen, zoals het afsluiten van polissen en het afhandelen van schadeclaims. In het vervolg zullen voorbeelden daarom steeds betrekking hebben op ECHO-systemen bij verzekeringsbedrijven en in het bijzonder op het afhandelen van schadeclaims op motorrijtuigverzekeringspolissen.

Het centrale ECHO-begrip "case" kan nu gedefinieerd worden als een door een bepaald voorval geïnitieerde en aan een specifieke polis gerelateerde reeks van routineactiviteiten bij verzekeringsbedrijven.

Het afhandelen van cases, "case handling", is op dit moment veelal nog sterk papiergeoriënteerd. Zo wordt bij het afhandelen een schadeclaim een case geïnitieerd door de binnenkomst van een schadeformulier bij het verzekeringsbedrijf. Dit formulier wordt toegekend aan één van de medewerkers die claims afhandelen, de "caseworkers". Het schadeformulier wordt in een map gedaan, de "casefolder", waarna de caseworker begint met het verzamelen van de informatie die nodig is voor het beoordelen van de claim. Hiertoe behoort het opvragen van gegevens uit een centrale database betreffende de cliënt en de betrokken polis, het opzoeken in het archief van eventueel aanwezige dossiers van aan de nieuwe case gerelateerde vroegere cases, het verzamelen van door de politie opgemaakte processen-verbaal, van getuigeverklaringen, van informatie van de verzekeringsmaatschappij van de tegenpartij, en dergelijke.

Alleen voorzover de benodigde informatie intern aanwezig is, kan de caseworker er min of meer direkt over beschikken. Voor de overige benodigde gegevens moet een verzoek gericht worden aan externe instanties. De casefolder wordt dan opgeborgen en pas weer geopend bij ontvangst van nieuwe informatie, die dan bij de reeds in de casefolder aanwezige documenten gevoegd wordt. De casefolder bevat op elk moment van het case handling proces alle tot op dat tijdstip verzamelde gegevens omtrent een case. Op een bepaald moment wordt op basis van de vergaarde informatie een beslissing over de claim genomen. De case wordt dan gesloten en de casefolder wordt toegevoegd aan de verzameling dossiers in het archief.

In het algemeen zijn bij één case meerdere medewerkers betrokken, zoals senior caseworkers, "case checkers", die beslissingen controleren en managers die de uiteindelijke uitkomst van een case beoordelen en al dan niet goedkeuren. Bij één caseworker behoren op elk moment een groot aantal cases in verschillende stadia van afhandeling.

Deze papiergeoriënteerde werkwijze heeft de volgende negatieve aspecten :

- door het trage fysieke transport van gegevens tussen de verschillende bij een case betrokken medewerkers kan er betrekkelijk veel tijd verlopen tussen de aanvang en de afronding van een case,
- het gemiddeld aantal cases, dat per dag door een caseworker behandeld wordt, is betrekkelijk laag vanwege de tijd die nodig is voor het bijeenzoeken van dossiers, het kopiëren van documenten en het archiveren van casefolders. Samenhangend hiermee is het aantal cases dat een caseworker simultaan uit kan hebben staan betrekkelijk laag,
- (telefonische) vragen van cliënten of andere betrokkenen omtrent een case kunnen pas worden beantwoord nadat de casefolder is opgehaald,
- het is niet of nauwelijks mogelijk om voor beleidsdoeleinden statistische informatie te verzamelen over de op een bepaald moment in de organisatie aanwezige openstaande cases,
- informatie over een case staat op ieder moment slechts ter beschikking van één persoon,
- er is een risico van verlies van documenten,
- er is veel ruimte vereist voor de fysieke opslag van gegevens.

Uit onderzoek is gebleken dat alle vormen van manipulatie van papieren gegevens samen ongeveer de helft van de werktijd van de medewerkers in beslag nemen.

De doelstellingen die het ECHO-project wil realiseren door het automatiseren van het proces van case handling zijn als volgt :

- stijging van de produktiviteit van de medewerkers door een optimale sturing van hun activiteiten en door het verminderen van de tijd die nodig is voor het verzamelen, versturen, en opslaan van gegevens,
- verbetering van de efficiency door het vermijden van het transport van papieren gegevens door de organisatie alsmede door beperking van de benodigde opslagruimte,
- betere serviceverlening aan de cliënten door het verkorten van de tijd voor het afhandelen van een case, het onmiddellijk kunnen beantwoorden van vragen omtrent een case en door vermindering van het aantal fouten.

Hiertoe dient een systeem ontwikkeld te worden met de volgende kenmerken :

- een optimale beheersing en sturing van de activiteiten binnen de organisatie,
- een snel en efficiënt transport van informatie door de gehele organisatie,
- een snelle toegang voor alle bij case handling betrokken medewerkers tot alle voor hen relevante bedrijfsgegevens,
- een optimaal mens-machine-interface,
- een compacte fysieke opslag van bedrijfsgegevens.

De ECHO-applicatie dient bij te houden welke cases bij welke caseworkers aanwezig zijn, in welke toestand iedere case zich bevindt en welke cases voor behandeling in

aanmerking komen, bijvoorbeeld omdat nieuwe informatie voor deze cases ter beschikking staat. De applicatie leidt de case naar een uiteindelijke toestand waarin een definitieve beoordeling plaats kan vinden.

Beslissingen zullen nog steeds door medewerkers genomen worden. Het ECHO-project wil het optreden van deze beslissingen echter door de ECHO-applicatie gestuurd laten verlopen. Alle administratieve routineactiviteiten zoals het distribueren, verzamelen, ordenen, transporteren en opslaan van gegevens dienen binnen een ECHO-systeem geautomatiseerd plaats te vinden.

De hoofddoelstellingen van het ECHO-project bestaan uit de manipulatie van de voor case handling relevante bedrijfsgegevens en de sturing van de bij case handling optredende activiteiten. Deze beide hoofddoelstellingen dienen gerealiseerd te worden binnen randvoorwaarden voor de performance van het systeem in termen van responsetijden en verwerkingscapaciteit. In het vervolg zullen deze beide hoofddoelstellingen zelf niet uitgebreid aan de orde komen maar staan de performanceaspecten van ECHO-systemen centraal.

2.2. ECHO-systemen.

2.2.1. Performanceaspecten van ECHO-systemen.

ECHO-systemen moeten significante efficiencyverbeteringen opleveren ten opzichte van de nu nog bij case handling gehanteerde, sterk papiergeoriënteerde werkwijzes. De mate waarin dit gerealiseerd wordt, is direkt afhankelijk van de performance van het totale ECHO-systeem. De performance is derhalve voor ECHO-systemen een faktor van cruciaal belang.

Over de performanceaspecten van deze grootschalige, gedistribueerde systemen voor de automatisering van complexe taken binnen kantooromgevingen zijn op applicatieniveau nog nauwelijks gegevens bekend. Evenmin is er sprake van een volledig inzicht in de aard en de omvang van de parameters die van invloed zijn op deze performanceaspecten. Binnen het ECHO-project wordt daarom onderzoek verricht naar de performance van de verschillende in ECHO-systemen aanwezige hardware- en softwarecomponenten alsmede de invloed die zij hebben op de performance van het totale ECHO-systeem. Er moet worden vastgesteld wat met betrekking tot de performance de kritieke componenten in ECHO-systemen zijn. Hierop dient in de eerste plaats de aandacht gericht te worden bij het optimaliseren van de performance van ECHO-systemen.

De volgende componenten en aspecten zijn van invloed op de performance :

- de ECHO-applikatiesoftware,
- de netwerksoftware,
- de windowingssoftware,

- de compressietechnieken die op de imagebestanden worden toegepast,
- het database management systeem,
- de software voor de gedistribueerde opslag van bestanden in het systeem,
- de snelheden van de hardwarecomponenten zoals
 - de optische en magnetische achtergrondgeheugens,
 - de processoren,
 - de communicatiecontrollers,
 - het transmissiesysteem,
- de configuratie van het ECHO-systeem.

De ECHO-applikatiesoftware prepareert, stuurt, controleert en/of effectueert alle binnen het systeem plaatsvindende acties die betrekking hebben op case handling.

De netwerksoftware realiseert het gegevenstransport tussen de op het netwerk aangesloten computersystemen voor de verschillende daarop aanwezige processen van de gedistribueerde ECHO-applikatie.

De windowingsoftware wordt door de ECHO-applikatie gebruikt voor het creëren van het gebruikersinterface. Deze softwarecomponent schermt applicaties af van het type en zelfs de lokatie van een beeldscherm waarop de ECHO-applikatie informatie presenteert aan een medewerker. De applicatie kan hierdoor onafhankelijk van het beeldscherm en met name onafhankelijk van de resolutie daarvan gerealiseerd worden. Voor de windowingsoftware wordt van de de facto standaard NeWS en/of X11 gebruik gemaakt.

Images worden verkregen door het scannen van documenten. Het scannen van een A4-document met een resolutie van 200 dots per inch (dpi) levert een imagebestand op met een omvang van ongeveer 500 kB. Het in deze vorm opslaan en transporteren van imagebestanden stelt buitensporige eisen aan de opslag- en transportcapaciteit van ECHO-systemen vanwege het zeer grote aantal images dat daarin aanwezig is en dagelijks getransporteerd wordt. Tevens treedt er in dat geval een buitensporige verspilling op van deze omvangrijke opslag- en transportcapaciteit vanwege de hoge redundantie van de beeldinformatie in de imagebestanden.

Door het toepassen van compressietechnieken op de imagebestanden wordt deze redundantie teruggebracht. Na compressie resteert een imagebestand met een grootte in de orde van 40 kB. De exacte omvang van het gecomprimeerde bestand is natuurlijk sterk afhankelijk van de specifieke beeldinformatie. Ook dan zal het in ECHO-systemen benodigde achtergrondgeheugen nog bijzonder omvangrijk zijn en reeds voor ECHO-systemen met een gemiddelde omvang ettelijke honderden Gigabytes tot zelfs enkele Terabytes kunnen bedragen. De opslag van de imagebestanden zal derhalve hoofdzakelijk plaatsvinden op optische systemen, zoals het Megadoc-systeem van Philips.

Naarmate de gebruikte compressietechniek een sterkere reductie van de imagebestanden oplevert, vermindert de voor de opslag ervan vereiste omvang van het achtergrondgeheugen. Ook is minder tijd vereist voor het transport van imagebestanden over

het netwerk. Bij de compressietechnieken is echter niet alleen de reductie van de omvang van imagebestanden van belang maar eveneens de snelheid waarmee gecomprimeerde images in een werkstation gedecomprimeerd en op de pixels van het beeldscherm geschaald kunnen worden. De tijd, die vereist is voor de decompressie, schaling en afbeelding van een image, ligt in de orde van 1 seconde en is daarmee in het algemeen groter dan de tijd die nodig is voor alleen het inlezen van een gecomprimeerd imagebestand door een werkstation vanaf een fileserver via het netwerk.

Het database management systeem beheert en regelt de toegang tot de voor case handling relevante bedrijfsgegevens. Bij het veranderen of toevoegen van informatie in de database is voor de handhaving van de dataconsistentie het uitvoeren van bepaalde controles op de ingevoerde gegevens vereist. Deze controles kunnen complex van aard zijn waardoor in uitgebreide ECHO-systemen hoge eisen gesteld worden aan de verwerkingscapaciteit van het database management systeem.

De voor case handling relevante informatie zal in ECHO-systemen over verscheidene in het netwerk aanwezige computersystemen gedistribueerd worden opgeslagen. De ECHO-applikatie dient echter onafhankelijk te zijn van de plaats waar een specifiek bestand is opgeslagen. De toegang tot de bestanden dient transparant te zijn voor de ECHO-applikatie. De softwarecomponent waarmee dit gerealiseerd wordt is de door Sun Microsystems ontwikkelde de facto standaard hiervoor, het Network File System NFS.

De configuratie van een ECHO-systeem omvat de netwerkstructuur, de types en de plaatsing van de daarin aanwezige systeemcomponenten alsmede de distributie van de bedrijfsgegevens over de daarvoor in aanmerking komende systeemcomponenten zoals bijvoorbeeld de fileservers. Verscheidene factoren zijn van invloed op de manier waarop ECHO-systemen geconfigureerd zullen worden waaronder performance, betrouwbaarheid, beveiliging van bedrijfsgegevens, handhaving van dataconsistentie en de complexiteit van het systeembeheer.

Tijdens deze afstudeeropdracht is de invloed van NFS, de netwerksoftware, de hardwarecomponenten en de configuratie op de performance van ECHO-systemen onderzocht. De overige genoemde componenten en aspecten, die eveneens hierop van invloed zijn, worden in dit onderzoek niet meegenomen. Wel zijn resultaten van dit onderzoek van invloed op de wijze waarop de ECHO-applikatie geïmplementeerd en gestructureerd wordt.

2.2.2. De logische structuur van ECHO-systemen.

Een essentieel aspect van ECHO-systemen is dat alle voor case handling relevante bedrijfsinformatie in elektronische vorm wordt opgeslagen. Deze opslag geschiedt op entiteiten binnen het ECHO-systeem die op basis van de daarop aanwezige gegevens logisch worden onderscheiden in :

- de casefolderstore waar de casefolders aanwezig zijn,
- het dossier archief waar zich de dossiers van afgesloten cases bevinden,
- de ECHO-database waar de ECHO-systeeminformatie is opgeslagen en
- de corporate database met bedrijfsgegevens van algemene aard die eventueel ook voor case handling relevant kunnen zijn zoals het geval is met gegevens over cliënten en polissen.

De gegevens, die bij een organisatie binnenkomen, worden zoveel mogelijk al aan de periferie van de organisatie in elektronische vorm omgezet, voorzover dat nog niet reeds het geval is. Zo worden in de mailroom grote hoeveelheden ontvangen post gescand. Deze gescande documenten zijn dan in de vorm van images in het systeem aanwezig. Ook na datacompressie van de redundante beeldinformatie hebben deze images nog steeds een relatief grote omvang in de orde van 40 kB. Dit in vergelijking met ongeveer 2 kB voor een pagina met ASCII-gecodeerde karakters. De images maken vanwege hun omvang het grootste deel uit van de binnen ECHO-systemen te transporteren en te archiveren gegevens.

De logische entiteiten binnen een ECHO-systeem waartussen zich de meest omvangrijke gegevensstromen afspelen zijn de caseworker, de casechecker, de mailroom, de casefolderstore en het dossier archief, zie fig. 2.1. Deze gegevenstransporten bestaan in belangrijke mate uit images.

Daarentegen heeft de interactie met de ECHO-database en de corporate database grotendeels betrekking op ASCII-gecodeerde gegevens met een meer beperkte omvang.

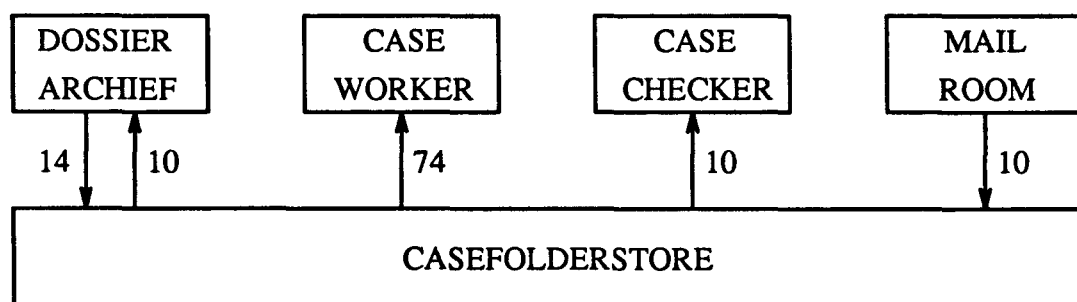


Fig. 2.1 De belangrijkste datastromen in een ECHO-systeem in MB per caseworker per dag.

Teneinde een indruk te verkrijgen van de omvang van de belastingen die in ECHO-systemen zullen optreden, is er bij verzekeringsbedrijven onderzoek verricht naar de werkzaamheden van caseworkers en de hoeveelheid gegevens die een caseworker per dag verzamelt, bekijkt, verstuurt en opslaat. Dit betreft onder meer :

- het aantal nieuwe cases per caseworker per dag,
- het aantal casefolders dat per caseworker per dag geopend wordt,

- het aantal dossiers van gesloten cases dat per caseworker per dag wordt opgezocht,
- de hoeveelheid post die per caseworker per dag bij de organisatie binnenkomt,
- het aantal en het type documenten in een casefolder.

Deze resultaten zijn natuurlijk sterk afhankelijk van de werkwijze bij een specifieke verzekeringsinstelling en van het type polis en claim.

Op grond van deze resultaten kunnen voor een ECHO-systeem gericht op het afhandelen van schadeclaims bij motorrijtuigverzekeringen de volgende aannames gedaan worden, zie [OUD87] :

- een casefolder bevat 10 tot 20 images en daarnaast nog een in omvang relatief beperkte hoeveelheid door de caseworker verzamelde of geproduceerde ASCII-gecodeerde gegevens. De grootte van een casefolder bedraagt gemiddeld 500 kB,
- een caseworker opent per dag 80 casefolders. Dit veroorzaakt een gegevensstroom van 40 MB tussen de casefolderstore en de caseworkers,
- een caseworker verzamelt per dag uit dossiers een hoeveelheid gegevens met een totale omvang van 14 MB. Deze gegevens worden eerst overgedragen van het dossier archief naar de casefolderstore en daarna tegelijk met casefolders door de caseworker ingelezen,
- een caseworker opent per dag ongeveer 20 nieuwe cases en sluit er eveneens 20. Dit laatste leidt tot een overdracht van 10 MB aan casefolders van de casefolderstore naar het dossier archief,
- per caseworker per dag worden er eveneens 20 casefolders geopend door casecheckers voor het beoordelen van de besluiten omtrent cases door de caseworkers,
- een caseworker ontvangt per dag 10 MB aan post in de vorm van gescande documenten. Deze documenten worden vanuit de mailroom overgedragen naar de betreffende casefolders in de casefolderstore,
- behalve het transport van gegevensbestanden leest ieder werkstation per dag ook nog een ruw geschatte 20 MB aan systeemsoftware in. Deze wordt eveneens verondersteld afkomstig te zijn van de casefolderstore.

De hier vermelde gegevensstromen zijn de naar omvang belangrijkste.

De belangrijkste geaggregeerde gegevensstromen tussen de verschillende entiteiten binnen een ECHO-systeem bedragen derhalve per caseworker per dag :

- | | |
|---|-------|
| - van de casefolderstore naar een caseworker | 74 MB |
| - van de casefolderstore naar een casechecker | 10 MB |
| - van de casefolderstore naar het dossier archief | 10 MB |
| - van het dossier archief naar de casefolderstore | 14 MB |
| - van de mailroom naar de casefolderstore | 10 MB |

De eerste twee gegevensstromen zijn van dezelfde aard en kunnen in wezen worden samengevoegd. De gegevensstromen zijn weergegeven in fig. 2.1. Het totale

gegevenstransport tussen de logische entiteiten van een ECHO-systeem bedraagt aldus 118 MB per caseworker per dag.

De gegevensopslag vereist bij een permanente, gelijktijdige aanwezigheid van 400 open cases per caseworker een capaciteit van 200 MB per caseworker in de casefolderstore. De groei van het dossier archief bedraagt 10 MB per caseworker per dag en derhalve 2 GB per caseworker per jaar.

Er dient op gewezen te worden dat bovenstaande resultaten slechts een globale indruk geven en dat binnen specifieke organisaties aanzienlijke afwijkingen daarvan op kunnen treden.

Indien de benodigde gegevens fysiek niet alle lokaal bij de werkstations van de caseworkers zijn opgeslagen, resulteert dit bij aanwezigheid van enige honderden caseworkers in grote belastingen van de fileservers en van het netwerk.

In dergelijke gevallen kan voor wat betreft het netwerk een verdeel-en-heers-strategie gevolgd worden door de gegevensstromen in het ECHO-systeem zoveel mogelijk gescheiden van elkaar te laten optreden. Hiervoor is vereist dat het transmissiesysteem in segmenten wordt opgedeeld. Tussen deze segmenten moeten filters aanwezig zijn die alleen verkeer door laten waarvan de bron en bestemming zich aan weerszijden van het filter bevinden. Aldus is het mogelijk de gegevenstromen zoveel mogelijk te beperken tot afzonderlijke segmenten van het transmissiesysteem en kunnen er binnen dit transmissiesysteem verscheidene, lokale gegevensstromen simultaan optreden.

Dit lokaliteitsprincipe vereist wel een optimale allocatie van de voor case handling relevante gegevens over de verschillende fileservers in het systeem teneinde te bewerkstelligen dat de bron en de bestemming van een gegevenstransport zoveel mogelijk op hetzelfde segment van het transmissiesysteem zijn aangesloten.

2.2.3. De hardwarecomponenten van ECHO-systemen.

Tot de hardwarecomponenten van een ECHO-systeem behoren het transmissiesysteem, de computersystemen en de randapparatuur voor de gegevensopslag.

Voor het transmissiesysteem van ECHO-systemen wordt uitgegaan van een Ethernet-Local Area Network. Ethernet is een gestandaardiseerd LAN met een relatief hoge transmissiecapaciteit en om deze redenen door PTDS als bedrijfsstandaard gekozen.

Het netwerk wordt opgebouwd uit afzonderlijke ethernet-segmenten die onderling gekoppeld worden door bridges. Ieder segment is een enkelvoudige ethernet-kabel die een bus vormt waarop computersystemen kunnen worden aangesloten.

In ECHO-systemen is de belasting, die afzonderlijke werkstations op het netwerk en de fileservers veroorzaken, relatief hoog. Een belangrijke faktor voor ECHO-systemen is derhalve het aantal werkstations dat per segment kan worden aangesloten.

Een bridge biedt een twee-weg koppeling tussen de beide erop aangesloten ethernet-segmenten. Alleen de ethernet-pakketten waarvan de bron en de bestemming zich aan weerszijden van een bepaalde bridge bevinden, worden door die bridge overgedragen

van het ene naar het andere segment. Op deze wijze kunnen de belastingen in het transmissiesysteem gescheiden gehouden worden en worden de verschillende segmenten, in tegenstelling tot bij het gebruik van repeaters, niet nodeloos belast.

Het is binnen een ethernet-netwerk niet toegestaan door het koppelen van ethernet-segmenten ringstructuren aan te brengen. Er zouden in dat geval meerdere transmissiewegen tussen bron en bestemming van een pakket aanwezig zijn. Vanwege het feit dat ethernet een broadcast-medium is, zou een pakket, dat verstuurd wordt, dan een botsing met zichzelf ondergaan waardoor een korrekte overdracht van informatie onmogelijk zou zijn.

Voor de structuur van het transmissiesysteem wordt bij ECHO-systemen uitgegaan van een beperkte boomstructuur : een backbone op het hoogste niveau met afzonderlijke ethernet-segmenten op het daaronderliggende departementale niveau, zie fig. 2.2. Voor zeer grote organisaties kan eventueel een derde subdepartementaal niveau ingevoerd worden.

Ook voor de backbone wordt in eerste instantie uitgegaan van een ethernet. In grote ECHO-systemen kan, afhankelijk van de gekozen configuratie, de backbone het zwaarst belaste deel van het transmissiesysteem zijn. Toch is voorlopig afgezien van alternatieven met een grotere transmissiecapaciteit daar hiervoor op dit moment nog geen gestandaardiseerde versies bestaan. De zich nu voltrekkende standaardisatie van het 100 c.q. 200 Mbit/s-FDDI-glasvezelnetwerk (Fiber Distributed Data Interchange), zie [SNI88], zal over enkele jaren een alternatieve keuze voor de backbone met een hogere capaciteit mogelijk maken.

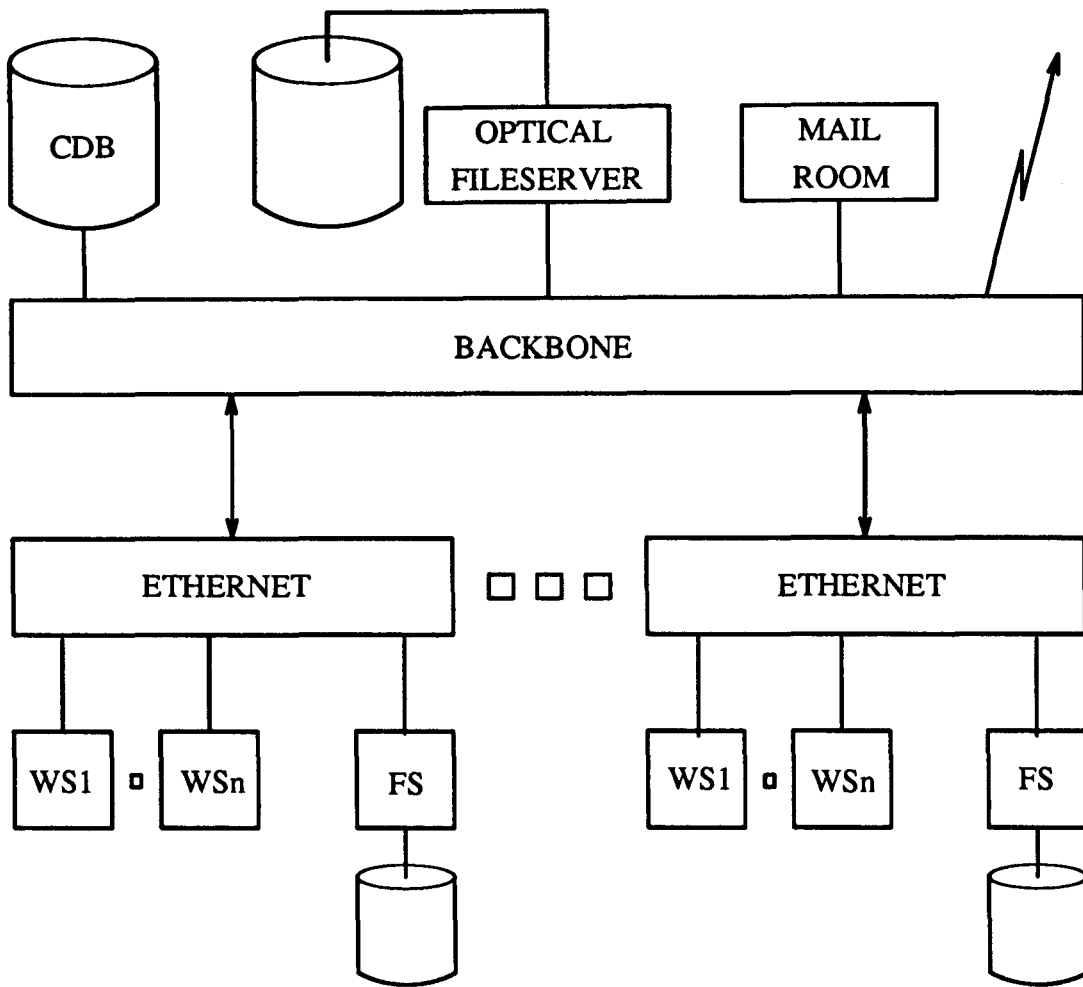
Op het transmissiesysteem worden de werkstations en de fileservers aangesloten. Fileservers beschikken over een in het algemeen groot achtergrondgeheugen waarop systeem- en applicatiesoftware alsmede databestanden zoals images worden opgeslagen. Deze gegevens zijn dan via het netwerk voor de overige daarop aangesloten computersystemen bereikbaar.

Voor de werkstations zal gebruik gemaakt worden van professionele systemen vergelijkbaar met Sun-3-werkstations van Sun Microsystems. Deze werkstations zullen voorzien zijn van een groot beeldscherm met een hoge resolutie teneinde de gescande documenten leesbaar te kunnen afbeelden. Tevens zal een relatief groot werkgeheugen aanwezig moeten zijn om de systeem- en applicatiesoftware alsmede de images van één of enkele casefolders te kunnen bevatten.

Als besturingssysteem van de computersystemen wordt Unix gebruikt.

In eerste instantie wordt uitgegaan van werkstations zonder achtergrondgeheugen, die alle benodigde systeem- en applicatiesoftware alsmede gegevensbestanden van fileservers betrekken. Gezien de per caseworker benodigde opslagruimte zullen reeds bij ECHO-systemen van enige omvang meerdere, snelle fileservers met een zeer grote opslagcapaciteit aanwezig moeten zijn.

De opslag van gegevens op het achtergrondgeheugen kan op magnetische en op



- CDB Corporate DataBase
- WS Werkstation
- FS Fileserver

Fig. 2.2 Een mogelijke hardwareconfiguratie van een ECHO-systeem.

optische wijze geschieden. Bij magnetische opslag is er een snellere toegang tot de opgeslagen gegevens. Bij het optische equivalent is de opslagcapaciteit groter en zijn de opslagkosten per bit lager.

Optische opslag betreft op dit moment alleen nog zogenaamde WORM-schijven (Write Once Read Many times) waarbij eenmaal beschreven gedeeltes van een schijf niet hergebruikt kunnen worden voor de opslag van andere gegevens. Indien er geen behoefte meer is aan bepaalde op de WORM-schijf aanwezige gegevens, is het gebied waar deze op de schijf zijn opgeslagen verder onbruikbaar. Op een WORM-schijf dienen derhalve bij voorkeur gegevens opgeslagen te worden die niet meer aan veranderingen onderhevig zijn. Dit is het geval bij imagebestanden die ook vanwege hun relatief grote omvang voor optische opslag in aanmerking komen.

De opslagcapaciteit van een WORM-schijf bedraagt typisch 1 GB per zijde. Een

Megadoc-systeem bestaat uit een jukebox met een twintigtal WORM-schijven.

De uitwisseling van informatie tussen het ECHO-systeem en de papieren buitenwereld geschiedt door middel van printers en scanners. Scanners produceren een image van een document door dit punt voor punt af te tasten. Dit gebeurt met een bepaalde resolutie. Veelal wordt hiervoor 200 dpi gebruikt.

2.2.4. De gedistribueerde gegevensopslag.

De fysieke opslag van de gegevens in de casefolderstore, het dossier archief en de ECHO-database, die alleen voor case handling relevant zijn, kan in theorie variëren van opslag op één of meerdere fileservers centraal in het systeem tot volledig gedistribueerd, lokaal bij de werkstations.

De corporate database zal geïmplementeerd zijn op een reeds in de organisatie aanwezige mainframe die aan het ECHO-netwerk gekoppeld wordt.

De verschillende factoren, die van invloed zijn op de configuratie van ECHO-systemen, wijzen in tegengestelde richtingen voor wat betreft de mate waarin de voor case handling relevante bedrijfsgegevens gedistribueerd moeten worden opgeslagen.

Zo kan de performance gebaat zijn bij de aanwezigheid van lokale achtergrondgeheugens bij de werkstations zodat de benodigde gegevens niet alle meer betrokken hoeven te worden van en via systeemcomponenten die met anderen gedeeld worden. In dat laatste geval treden er bij deze gedeelde systeemcomponenten wachttijden op die langer zijn naarmate de belasting ervan hoger is. Deze lokale harde schijven zullen dan echter niet alleen kleiner zijn maar in het algemeen ook minder bedrijfszeker en langzamer dan de grote schijven bij fileservers. Door het laatste wordt de performancewinst voor een groot deel weer ongedaan gemaakt.

Handhaving van dataconsistentie is gebaat bij het op één fysieke plaats in het systeem opslaan van bestanden. Indien dit gegevens betreft die voor meerdere of alle medewerkers van belang zijn, ligt de opslag op een centrale fileserver het meest voor de hand. Ook het systeembeheer wordt eenvoudiger indien bestanden door de werkstations worden ingelezen van een beperkt aantal fileservers.

Op deze gronden lijkt het het meest waarschijnlijk dat in ECHO-systemen werkstations zonder achtergrondgeheugen gebruikt zullen worden en dat de gegevensopslag op fileservers plaatsvindt.

(De performanceaspecten hiervan zouden nader onderzocht kunnen worden.)

In dat geval dient de allocatie van de voor case handling relevante gegevens zodanig te geschieden dat de gegevens die door op departementaal niveau aanwezige werkstations worden opgevraagd, zoveel mogelijk gelezen worden vanaf op datzelfde departementale niveau aangesloten fileservers. In dat geval wordt het transmissiesysteem en in het bijzonder de backbone zo laag mogelijk belast.

Dit lokaliteitsprincipe zal voor wat betreft de casefolders vrijwel volledig gerealiseerd kunnen worden. De casefolders zijn namelijk hoofdzakelijk relevant voor de bij de case betrokken caseworker en kunnen derhalve op de fileserver worden opgeslagen die op hetzelfde segment is aangesloten als het werkstation van de caseworker.

Ook bij een eventueel gebruik om performanceredenen van lokale achtergrondgegevens bij de werkstations zullen de daarop opgeslagen gegevens voornamelijk casefolders betreffen. Het is in dit geval wel van belang dat caseworkers steeds aan hetzelfde werkstation werkzaam zijn.

Voor dossiers ligt opslag op één of meerdere optische fileservers, die op de backbone zijn aangesloten, meer voor de hand. Deze dossiers van gesloten cases bevatten stabiele, permanente gegevens en zijn in principe voor iedere medewerker in gelijke mate van belang.

Een organisatorische indeling van het werk waarbij er een relatie ontstaat tussen de departementen en de cliënten van de organisatie, bijvoorbeeld aan de hand van polisnummers of de beginletters van de namen van de cliënt, zou de mogelijkheid bieden ook het dossierarchief gedistribueerd op te slaan op optische fileservers op departementaal niveau. Alle dossiers blijven dan voor alle medewerkers opvraagbaar, maar medewerkers zullen hoofdzakelijk dossiers inlezen vanaf de op hetzelfde ethernet-segment aangesloten optische fileserver.

Het is voor de modellering van ECHO-systemen van belang te weten hoe de interactie tussen de werkstations en de fileservers zich voltrekt. Via het gebruikersinterface kan de caseworker een casefolder identificeren en vervolgens daaruit één of meerdere images selecteren om op het scherm te laten afbeelden.

Dit wordt dan door de applicatie vertaald in het inlezen van deze images of van de casefolder in zijn geheel. De precieze implementatie hiervan ligt nog open en zal mede op basis van performanceaspecten bepaald worden.

Indien de gehele casefolder wordt ingelezen zal een relatief grote wachttijd optreden voordat het eerste image getoond wordt, maar is voor de daaropvolgende images slechts kleine wachttijd vereist daar deze images dan reeds lokaal aanwezig zijn.

Indien echter images pas worden ingelezen op het moment dat ze geselecteerd worden, zal het eerste image in vergelijking met de vorige mogelijkheid relatief snel aanwezig zijn maar is voor de daaropvolgende images een relatief lange wachttijd vereist omdat deze images nu ook steeds van de fileserver moeten worden ingelezen. Nu wordt echter voorkomen dat images worden ingelezen waaraan de caseworker geen behoefte blijkt te hebben. Anderzijds kunnen bij het inlezen van gehele casefolders in zekere mate schaalvoordelen optreden.

Naar welke van de beide mogelijkheden de voorkeur van de caseworkers uitgaat, is een vraag die op basis van perceptieonderzoek beantwoord zou moeten worden. Het is echter te verwachten dat het behandelen van een case als een eenheid van werk gezien wordt en dat een langere wachttijd aan het begin geaccepteerd wordt indien dat verbeteringen gedurende het werken aan een case met zich meebrengt.

Verder kan nog vermeld worden dat binnen het ECHO-project de mogelijkheid onderzocht zal worden om in ECHO-applicaties de intelligentie in te bouwen om te voorspellen welke case een caseworker tijdens zijn werkzaamheden als eerstvolgende voor afhandeling zal uitkiezen. De betreffende gegevens kunnen dan al door de ECHO-applicatie naar het werkstation getransporteerd worden en aldus daar reeds lokaal aanwezig zijn op het moment dat de caseworker de case selecteert. Dit kan dan wederom een gehele casefolder betreffen dan wel één of enkele images. De images die

een caseworker als eerste zal willen zien, zullen in het algemeen de images van nieuw binnengekomen documenten zijn. Op deze wijze kan de performance, zoals de caseworker deze ervaart, verhoogd worden.

Bij de simulaties op basis van het systeemmodel van ECHO-systemen is in de meeste gevallen aangenomen dat een casefolder in zijn geheel getransporteerd wordt waardoor bestanden met een grootte in de orde van 500 kB door de fileserver aangeleverd en door het netwerk overgedragen worden. Er zijn enkele simulaties uitgevoerd waarbij 40 kB-grote bestanden ingelezen worden.

2.2.5. Het Network File System NFS.

Bij de over verschillende systemen gedistribueerde gegevensopslag is het gewenst dat ieder op het netwerk aangesloten systeem in principe toegang kan hebben tot alle in het netwerk aanwezige bestanden. Eveneens is het gewenst dat deze gedistribueerde opslag transparant is voor gebruikers en applicatieprogramma's oftewel dat de toegang tot bestanden kan plaatsvinden zonder dat de exacte opslaglocatie ervan in het netwerk bekend hoeft te zijn. Beide aspecten worden gerealiseerd door het door Sun Microsystems ontwikkelde Network File System NFS.

NFS vormt het mechanisme dat in ECHO-systemen de toegang realiseert tot de op fileservers gedistribueerd opgeslagen gegevensbestanden voor de verschillende op het netwerk aangesloten werkstations. NFS regelt de gegevensoverdracht tussen deze componenten.

NFS is gebaseerd op het client-server-concept waarbij dit concept wordt toegepast op het bestandssysteem. Binnen NFS kan een client-systeem een verzoek met betrekking tot een bepaald bestand richten aan een server-systeem waarop dit bestand aanwezig is. Een specifiek systeem kan hierbij tegelijkertijd zowel zelf een server zijn voor verscheidene clients als een client van een andere server.

Een server biedt aan bepaalde of alle systemen in het netwerk één of meerdere op de server aanwezige bestandssystemen aan. Ieder van deze client-systemen kan dan door middel van een "mount"-operatie deze aangeboden bestandssystemen in logische zin onderbrengen in oftewel toevoegen aan zijn eigen hiërarchische bestandssysteem. De client bouwt aldus een eigen blikveld op over de in het netwerk aanwezige bestanden. De toegevoegde bestanden kunnen vervolgens door op de client aanwezige applicaties benaderd worden alsof ze bij iedere client lokaal aanwezig zijn. De mount-operatie vindt normaliter plaats bij het opstarten van een client.

Op deze wijze is het mogelijk voor alle op het netwerk aangesloten systemen een gemeenschappelijke, transparante toegang tot alle in het netwerk aanwezige bestanden te realiseren.

In applicatieprogramma's kunnen de bestaande procedureaanroepen van het besturingssysteem gebruikt worden voor het benaderen van de bestanden, zie fig. 2.3. Op basis van de file-identificatie, die als parameter aan de procedure wordt

meegegeven, wordt binnen de systeemprocedure vastgesteld of dit bestand tot een lokaal bestandssysteem behoort dan wel op een bepaalde server in het netwerk is opgeslagen.

Voor de benadering van niet lokaal aanwezige bestanden wordt het door de applicatie verrichte verzoek vervolgens via het netwerk naar het betreffende NFS-serversysteem verstuurd. Deze voert dan de voor de honorering van het verzoek vereiste acties met betrekking tot het bestand in kwestie uit en stuurt de resultaten daarvan terug naar het proces op het client-systeem.

Het Network File System biedt applicaties aldus virtuele bestandssystemen (VFS) aan waarbij het voor de applicatie geen verschil maakt of deze wel of niet lokaal aanwezig zijn.

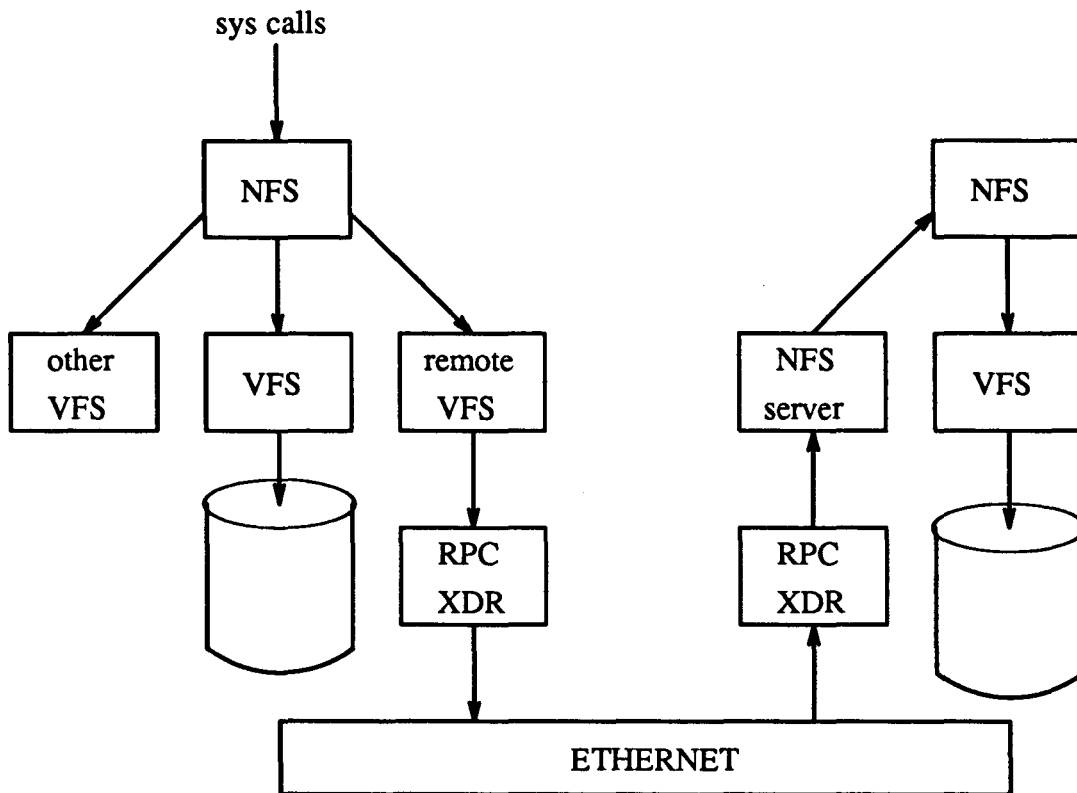


Fig. 2.3 De verwerking van bestand-georiënteerde systeemaanroepen door NFS.

NFS bestaat in wezen uit een verzameling Remote Procedure Calls (RPC).

RPC is een Sun-implementatie van inter-proces-communicatie. RPC maakt communicatie mogelijk tussen verschillende processen, al dan niet op verschillende systemen, in een vorm waarbij procedures door een bepaald proces aangeroepen en door een ander proces uitgevoerd worden.

RPC is eveneens gebaseerd op het client-server-concept, zie fig. 2.4.

Een client-proces, dat de service behoeft van een bepaald server-proces ergens in het netwerk, verricht een RPC-procedureaanroep waarin de service en het server-systeem gespecificeerd worden en blokkeert vervolgens zelf.

De RPC-procedure bouwt een bericht met het serviceverzoek op en verstuurt dit vervolgens al dan niet via het netwerk naar het specifieke server-proces. Dit proces analyseert het verzoek, voert de acties van de in het verzoek gespecificeerde serviceprocedure uit, construeert een resultaatbericht en stuurt dit tenslotte terug naar het client-proces. Na de ontvangst daarvan komt de procedureaanroep van het client-proces terug met het resultaat, zie fig. 2.4.

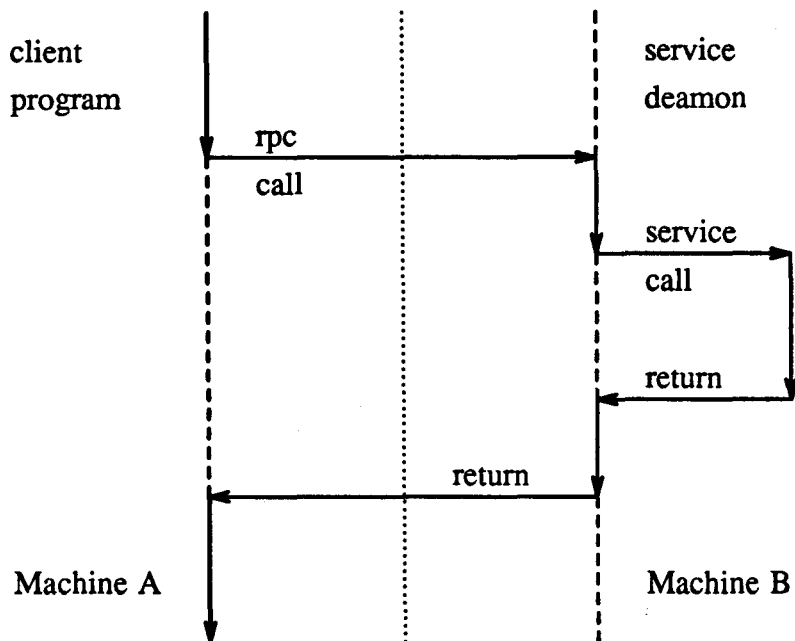


Fig. 2.4 Client-serverinteractie bij Remote Procedure Calls.

De middels RPC tussen client- en serverprocessen over te dragen gegevens worden gecodeerd op basis van de eXtended Data Representation (XDR).

XDR is een standaard voor de representatie van elementaire datatypes en daaruit opgebouwde, complexere datastructuren en maakt een machine- en besturingssysteemafhankelijke gegevensoverdracht mogelijk. XDR verzorgt bijvoorbeeld woord- en byte-verwisselingen bij op verschillende machines verschillend gerepresenteerde integer-types.

RPC is een high-level protocol voor communicatie tussen processen. Applicaties kunnen voor wat betreft de communicatieaspecten op basis van RPC gerealiseerd worden zonder geconfronteerd te worden met de aspecten van low-level transmissieprotocollen. Deze worden intern binnen RPC afgehandeld.

Standaard maakt RPC, en daarmee ook NFS, gebruik van het User Datagram Protocol (UDP) uit de TCP/IP- oftewel Internet-protocolfamilie.

Het UDP-protocol is een datagramprotocol en garandeert derhalve niet de korrekte overdracht van de informatie die middels dit protocol verstuurd wordt. Deze garantie dient dus op RPC-niveau gerealiseerd te worden. Hieronder valt onder meer het herhalen van een verzoek indien na een bepaalde tijd nog steeds geen (korrekt) antwoord ontvangen is.

In fig. 2.5 worden de bij de gegevensoverdracht tussen de verschillende computersystemen gebruikte protocollen weergegeven in de vorm van de gelaagde structuur van het hiervoor opgestelde OSI-model.

ECHO	applikatielaag
XDR	presentatielaag
RPC	sessielaag
UDP	transportlaag
IP	netwerklaag
CSMA/CD	datalinklaag
ETHERNET	fysieke laag

Fig. 2.5 OSI-protocol-stack voor ECHO-systemen.

Het UDP-protocol staat slechts gegevenstransporten toe in blokken van maximaal 8 kB. Hierdoor wordt bijvoorbeeld een door een applicatie aangeroepen read-systeemprocedure voor het inlezen uit een bestand van een blok data van 512 kB binnen NFS vertaald in een reeks van 64 NFS-remote procedure calls waarmee na elkaar opeenvolgende blokken van 8 kB uit het bestand ingelezen worden.

Om performanceredenen wordt hierbij binnen NFS zowel bij het werkstation als de fileserver read-ahead en write-behind toegepast.

Read-ahead kan optreden indien een werkstation een bestand inleest met een omvang van meer dan 8 kB. Het werkstation zal in dat geval alvast een NFS-verzoek om het volgende 8 KB-datablok naar de fileserver versturen nog voordat het vorige verzoek beantwoord is. Evenzo zal de fileserver, na de ontvangst van een NFS-verzoek om een bepaald blok uit een bestand, zo mogelijk alvast het volgende blok uit dit bestand van disk lezen. Dit blok kan dan indien er bij een volgend NFS-verzoek om gevraagd wordt, direkt vanuit de cache van de fileserver aangeleverd worden.

Bij write-behind verstuurt een werkstation tijdens het wegschrijven van een bestand naar de fileserver alvast een volgend blok data nog voordat het resultaatbericht van de schrijffactie van het vorige blok ontvangen is. De blokken blijven echter in de lokale cache van het werkstation aanwezig totdat de fileserver het werkstation mededeelt dat de blokken naar disk geschreven zijn.

Binnen NFS wordt voor de hertransmissie van verzoeken, waarop geen (korrekt) antwoord ontvangen is, een time-out van 700 ms gehanteerd. Deze time-out is vrij groot in vergelijking met de responsetijd bij de fileserver van een NFS-verzoek om een 8 kB-blok data die bij een overigens onbelaste fileserver in de orde van 20 tot 50 ms ligt. Deze time-out is echter toch noodzakelijk om te voorkomen dat bij een zwaar belaste fileserver prematuur herhalingen van verzoeken optreden. Bij hoge belasting kunnen zeker 10 tot 20 van dergelijke verzoeken in de wachtrij van de fileserver aanwezig zijn en gaan de responsetijden in de richting van deze time-out. De waarde van deze time-out kan door de systeembeheerder naar wens worden aangepast.

Duplicaten van verzoeken worden bij de fileserver niet weggegooid maar eveneens uitgevoerd. Er treden hierbij echter toch geen fouten op indien een applicatie bijvoorbeeld in append-mode gegevens naar een bestand schrijft. Er wordt in dit geval voorkomen dat een blok data tweemaal achtereen aan een bestand wordt toegevoegd. Alle akties op bestanden worden namelijk binnen NFS vertaald naar akties die betrekking hebben op een bepaald blok data op een specifieke lokatie in een bestand. In dit geval betekent het herhaald uitvoeren van een NFS-verzoek om een bepaald blok data op een specifieke lokatie in een bestand te schrijven het overschrijven van de reeds aanwezige data met zichzelf. Het NFS-protocol is idempotent hetgeen betekent dat het herhaald uitvoeren van een aktie gelijk is aan het eenmalig uitvoeren van een aktie.

Deze eigenschap van NFS is een consequentie van de toestandloosheid van de implementatie van NFS hetgeen wil zeggen dat een server bij het verwerken van opeenvolgende verzoeken niets hoeft te onthouden over zijn clients, de verwerkte verzoeken of de bestanden waarop de verzoeken betrekking hebben.

Een voordeel van de toestandloosheid van NFS-servers is dat een server uit kan vallen en weer operationeel kan worden zonder na te hoeven gaan in welke toestand de verzoeken, die op het moment van uitvallen van de server reeds ontvangen waren, zijn achtergelaten. De clienten dienen verzoeken, die niet of niet volledig afgehandeld zijn, eenvoudig te herhalen. Voor hen is er in dit geval slechts sprake van een trage server.

Een tekortkoming van de huidige implementatie van NFS is dat het locken van bestanden niet binnen NFS geïntegreerd is. Wel zijn er aparte lock-serverprocessen aanwezig.

Er worden voor de toegang tot bestanden twee typen locks gehanteerd.

Locks verkregen door een proces op een bepaald werkstation met behulp van de flock-systeemprocedure verhinderen de toegang tot een bestand op een fileserver door andere processen op hetzelfde werkstation maar voorkomen niet de toegang door processen die op andere systemen in het netwerk aanwezig zijn.

Een lock op een bestand op een fileserver, waarmee de toegang door alle overige processen in het netwerk verhinderd wordt, kan wel verkregen worden met behulp van de lockf-systeemprocedure. Hierbij treedt echter een tijdrovende communicatie op tussen verscheidene lock-serverprocessen in het netwerk waardoor met het verkrijgen van de lock in het bijzonder bij zware belastingen ettelijke seconden gepaard kan gaan.

2.3. Het PCG-netwerk.

Op het PCG is een netwerk aanwezig dat gebruikt wordt voor de ontwikkeling van de ECHO-applicatie. Het netwerk bestaat uit een ethernet-segment waarop verscheidene computersystemen zijn aangesloten.

De reden dat dit netwerk hier besproken wordt is dat aan dit netwerk de metingen zijn verricht waarvan de resultaten gebruikt worden bij het parametriseren van de simulatiemodellen alsmede voor het valideren van het systeemmodel en de modelsimulaties.

Bij aanvang van het afstudeerwerk waren op het netwerk behalve twee Sun fileservers en enkele PC's ongeveer 15 Sun werkstations aangesloten. Dit aantal is geleidelijk

gegroeid tot ruim 20. Alle Sun-systemen zijn voorzien van een naam in de vorm van een vierletterig geluid. Zo heten de fileservers respectievelijk "echo" en "galm".

De aanwezige Sun-systemen vormen een heterogene verzameling van de verschillende door Sun geleverde types. Dit zijn onder meer de Sun 3/xx , Sun 3/1xx en Sun 3/2xx-types met een 68020-microprocessor en een klokfrequentie van respectievelijk 15, 20 en 25 MHz. Bij metingen komen de verschillen in snelheid ten gevolge van de verschillende klokfrequenties duidelijk naar voren. Voor de fileservers worden SUN 3/280-types gebruikt.

Alle Sun-systemen zijn voorzien van een ethernet-controller-kaart waarmee ze aan het netwerk zijn gekoppeld.

De werkstations worden over het netwerk geboot. Ieder werkstation is daarvoor aan een specifieke fileserver toegekend. Tijdens het opstarten van het werkstation wordt door deze fileserver aan het werkstation een versie van het bootprogramma opgestuurd. Tot aan het moment van ontvangst daarvan staat het werkstation onder besturing van code in een PROM. Het bootprogramma neemt daarna het werkstation over laadt de relevante delen van het Unix-besturingssysteem vanaf de fileserver.

Unix verricht vervolgens de nodige acties voor systeemconfiguratie en -administratie waaronder het opstarten van standaard server-processen en het mounten van NFS-bestandssystemen. Hierna is het werkstation klaar voor multi-user gebruik.

Ieder werkstation heeft twee eigen gebieden op de disk van de fileserver waaraan het is toegekend. Het eerste bevat een rudimentair bestandssysteem, dat de "root" van het eigen bestandssysteem van het werkstation en enige directories met systeemsoftware omvat. Het tweede gebied is de swap area van het werkstation en vormt een extensie van het fysieke werkgeheugen. Daarnaast is er nog een publieke read-only sectie waar onder meer het boot-programma aanwezig is.

Deze beide eigen gebieden en de publieke read-only sectie vormen voor het werkstation een zogenaamde netwerk disk. Het gegevenstransport van en naar deze gebieden vindt plaats volgens het ND-(Network Disk)-protocol.

De voor ieder werkstation aparte gebieden op de disk van de fileserver zijn voor het opstarten vereist omdat op dat moment de NFS-bestandssystemen nog niet bereikbaar zijn daar de mount-operatie nog niet kan hebben plaatsgevonden.

De verschillende werkstations in het PCG-netwerk zijn hiervoor vrijwel gelijk over de beide aanwezige fileservers, "echo" en "galm", verdeeld.

Het ND-protocol transporteert data in blokken van 1 kB over het netwerk. De defaultwaarde voor het aantal pakketten dat verstuurd mag worden voordat de verzender een acknowledgement-bericht van de ontvanger verkregen moet hebben, is zes.

De fileserver waaraan een werkstation is toegekend en waar zich dus de netwerk disk van dit werkstation bevindt, zal in het vervolg worden aangeduid als de ND-fileserver. Een fileserver waarop zich bestandssystemen bevinden, die door werkstations middels NFS benaderd worden, zal ter onderscheiding als NFS-fileserver worden aangeduid. Beide fileservers kunnen op dezelfde fysieke fileserver betrekking hebben.

3. Theorie.

3.1. Modelling.

De performance van een bestaand computersysteem of -netwerk kan vastgesteld worden door het verrichten van metingen bij een kunstmatig aangebrachte belasting of, bij een operationeel systeem, onder belasting van de gebruikers.

ECHO-systemen zijn echter nog niet in de praktijk gerealiseerd. De performance analyse van ECHO-systemen kan derhalve alleen geschieden door het opstellen van een systeemmodel op basis waarvan resultaten voor de performance van ECHO-systemen kunnen worden afgeleid.

Een model is een abstractie van een systeem waarin de essentiële eigenschappen van het systeem zijn opgenomen en de minder belangrijke zijn weggelaten. Wat wel of niet essentieel is hangt af van de specifieke doelstelling van de systeemanalyse.

Het model dient een voldoende nauwkeurige beschrijving van de werking van het systeem te geven. Enerzijds mag het model niet te ruw zijn om onaanvaardbaar grote modelfouten in de op basis van het model verkregen resultaten te voorkomen, anderzijds moet gewaakt worden voor overmodellering waardoor het model onhanteerbaar wordt en de analyse bijzonder gecompliceerd of onmogelijk. De graad van modellering is derhalve tevens afhankelijk van de ter beschikking staande oplossingsmethodieken.

De wijze waarop een systeem gemodelleerd wordt, hangt af van de aard van het systeem. Een ECHO-systeem kenmerkt zich door de aanwezigheid van gemeenschappelijk gebruikte systeemcomponenten, zoals de fileservers en het netwerk, die op verzoek een bepaalde service verlenen, zoals het aanleveren of transporteren van een bestand. Deze verzoeken zijn afkomstig van een bepaald aantal werkstations.

De door de werkstations verrichte leesacties van casefolders vinden op willekeurige momenten onafhankelijk van elkaar plaats. Dit impliceert dat een serviceverzoek aan een gedeelde systeemcomponent kan optreden op een moment dat deze niet vrij is maar service verleent aan een eerder ontvangen verzoek. In dergelijke gevallen treedt voor de server een wachtrij van verzoeken op en worden wachttijden in het systeem geïntroduceerd.

De hier beschreven aspecten van ECHO-systemen zijn kenmerkend voor computernetwerken in het algemeen en geven aanleiding tot een modellering op basis van wachtrijnetwerken.

3.2. Wachtrijnetwerken.

Wachtrijnetwerken modelleren systemen waarin bepaalde componenten diensten verlangen van bepaalde andere componenten. De tweede groep componenten wordt met de term servers aangeduid. De snelheid waarmee deze servers hun diensten verlenen is eindig. De tijdstippen waarop de verzoeken om dienstverlening optreden alsmede de omvang van de verlangde diensten zijn in het algemeen beide stochastische grootheden. Bijgevolg kunnen er voor de servers wachtrijen van verzoeken optreden.

De verdeling van de tijdstippen waarop verzoeken bij een server aankomen kan ook beschreven worden in termen van de grootte van de tijdsintervallen tussen de opeenvolgende aankomsten, de zogenaamde "interarrival-tijden". Voor zowel de servicetijden als de interarrival-tijden gelden bepaalde verdelingsfuncties.

Wachtrijnetwerken bestaan uit een verzameling knooppunten met onderlinge verbindingen. Ieder knooppunt representeert een servicesysteem, waarin zich één of meerdere servers bevinden, alsmede een wachtruimte, waar serviceverzoeken tijdelijk verblijven indien alle servers in het servicesysteem belast zijn met de afhandeling van andere verzoeken en er derhalve geen enkele server onmiddellijk beschikbaar is. De servers binnen één servicesysteem worden identiek verondersteld. De wachtruimte heeft een capaciteit die eindig, eventueel nul, of oneindig is. Het gehele servicesysteem wordt ook wel aangeduid met de term wachtrij of queue.

De mogelijkheid van het moeten wachten op serviceverlening brengt met zich mee dat voor een wachtrij een queueing discipline gedefinieerd moet zijn op basis waarvan op het moment dat een server vrijkomt het eerstvolgende verzoek uit de wachtrij voor serviceverlening geselecteerd wordt. De meest voorkomende queueing discipline is FCFS (First Come First Served), waarbij het verzoek dat het langst in de wachtruimte aanwezig is, als eerstvolgende gekozen wordt.

De entiteiten, die bij een servicesysteem aankomen en om service verzoeken, worden "tokens" of "customers" genoemd. Deze tokens kunnen tot één of meerdere klassen behoren. Tokens uit verschillende klassen kunnen verschillen met betrekking tot de verdelingen van de interarrival-tijden en de servicetijden alsmede met betrekking tot de prioriteiten bij de queueing discipline en de routeringsregels.

Voor een wachtrijnetwerk moeten routeringsregels gedefinieerd zijn indien er bij het verlaten van een bepaalde server in het netwerk een keuzemogelijkheid bestaat ten aanzien van de door het token te volgen weg. De overgang van een token tussen verschillende servicesystemen geschiedt zelf tijdloos.

Wachtrijnetwerken kunnen open of gesloten zijn.

In een open wachtrijnetwerk treden aankomsten bij servicesystemen van buiten het netwerk op en verlaten tokens op een bepaald moment het netwerk weer. In dit geval is er altijd sprake van een oneindig groot aantal externe bronnen van serviceverzoeken.

In een gesloten netwerk zijn geen externe bronnen en bestemmingen van tokens aanwezig. In het systeem circuleert een vast aantal tokens. Meestal zijn er in dit geval één of meerdere nodes die als bronnen en bestemmingen beschouwd kunnen worden en vormen de overige nodes in het netwerk het eigenlijke model oftewel het gedeelte dat centraal staat bij de bestudering van het wachtrijnetwerk. De tijdsduur gedurende welke een token in dit laatste gedeelte van het wachtrijnetwerk aanwezig is, wordt (netwerks-)verblijfstijd of responsetijd genoemd. De tijd tussen twee opeenvolgende verblijfstijden, gedurende welke een token aanwezig is bij een node die als bron en eindbestemming van tokens wordt beschouwd, wordt met de term "thinktime" aangeduid.

Bij een gesloten model is het aantal bronnen van verzoeken eindig waardoor de momentane verdeling van de interarrival-tijden afhankelijk is van het aantal tokens dat zich reeds in het systeem bevindt.

Ook mengvormen van open en gesloten wachtrijnetwerken kunnen voorkomen.

Voor het bestuderen van (netwerken van) wachtrijen dienen alle basiselementen bekend te zijn. Dit betreft voor ieder servicesysteem het aantal servers, de capaciteit van de wachtruimte, de queueing discipline, de verdelingen van de servicetijden en de interarrival-tijden en eventuele routeringsregels.

Bij de bestudering van wachtrijsystemen kunnen onder bepaalde randvoorwaarden langs analytische weg resultaten worden afgeleid. Deze resultaten zijn statistisch van aard en hebben betrekking op systeemaspecten als queuelengtes, verblijfstijden bij servers en/of in (delen van) het netwerk alsmede bezettingsgraden van de servers.

Voor de aanpak van wachtrijsystemen die niet voldoen aan de randvoorwaarden waarbinnen een exacte analytische oplossing kan worden afgeleid, staan in sommige gevallen benaderende analytische oplossingsmethodieken ter beschikking. Zijn ook deze niet toepasbaar of te gecompliceerd dan resulteert de mogelijkheid om simulaties op basis van het wachtrijsysteem door te voeren.

3.2.1. Analytische aanpak van wachtrijsystemen.

Het gedrag van wachtrijsystemen wordt beschreven door de wachttijdtheorie. Deze theorie wordt uitgebreid behandeld in [KLE75]. De term wachtrijsysteem kan zowel op een enkele wachtrij als op een netwerk van wachtrijen betrekking hebben.

De analytische aanpak van wachtrijsystemen betreft de analyse van stochastische modellen en met name de analyse van Markov-processen.

Deze bespreking beperkt zich tot de klasse van Markov-processen waarbij discrete toestanden optreden waartussen op discrete tijdstippen overgangen kunnen plaatsvinden, de zogenaamde Markov-ketens.

Bij een wachtrijsysteem wordt de toestand gedefinieerd door het aantal tokens dat op een bepaald moment in iedere wachtrij van het wachtrijsysteem aanwezig is.

De Markov-eigenschap houdt in dat de kans q_{ij} op de overgang naar een volgende toestand $X_{t+1} = j$ alleen afhangt van de huidige toestand $X_t = i$ maar onafhankelijk is van alle vroegere toestanden van het Markov-proces X :

$$q_{ij} = P [X_{t+1}=j \mid X_0=a, X_1=b, \dots, X_t=i] = P [X_{t+1}=j \mid X_t=i] \quad (3.1)$$

In dit geval wordt ook wel gesproken van een eerste-orde Markov-keten.

De overgangswaarschijnlijkheden q_{ij} tussen de verschillende toestanden kunnen dan in een matrix Q worden weergegeven. De Markov-keten is stationair indien deze overgangswaarschijnlijkheden niet tijdsafhankelijk zijn.

De kans $p_k(t) = P [X_t = k]$ dat het proces zich op het tijdstip t in een bepaalde toestand k bevindt, oftewel de toestandsverdeling, kan dan nog wel tijdsafhankelijk zijn. Hoewel het theoretisch mogelijk is om het tijdsafhankelijke gedrag van het Markov-

proces c.q. het wachtrijsysteem te analyseren, ligt de nadruk in het algemeen op de bestudering van het limietgedrag waarbij de toestandsverdeling convergeert naar een niet-tijdsafhankelijke verdeling en het gedrag van het wachtrijsysteem stationair is.

De overgangswaarschijnlijkheden q_{ij} zijn afhankelijk van de verdelingen van de interarrival-tijden en de servicetijden van de tokens. De verdeling, die hiervoor in de wachttijdtheorie het meest gebruikt wordt, is de exponentiële verdeling. Deze verdeling is namelijk als enige continue verdeling geheugenloos. Dit houdt in dat bij een verschijnsel, waarvan de totale tijdsduur negatief-exponentieel verdeeld is, de op een bepaald moment gedurende de aanwezigheid van het verschijnsel nog resterende tijdsduur daarvan dezelfde negatief-exponentiële verdeling heeft en onafhankelijk is van de tijd gedurende welke het verschijnsel reeds is opgetreden.

Dit verschijnsel kan in de wachttijdtheorie zowel betrekking hebben op de (resterende) servicetijd van een verzoek bij een server als op de (resterende) tijd tot aan de aankomst van een volgend verzoek bij een servicesysteem.

Zonder de geheugenloosheid van de exponentiële verdeling wordt de analytische aanpak van de stochastische processen, die bij wachtrijen optreden, aanzienlijk gecompliceerder.

Bij de bepaling van exacte analytische oplossingen voor systeemaspecten van een wachtrijsysteem wordt op basis van de overgangswaarschijnlijkheden q_{ij} de toestandverdeling p_k van het wachtrijsysteem afgeleid. Op basis daarvan worden dan de statistische resultaten afgeleid voor belangrijke systeemaspecten zoals de verblijfstijden en de wachttijden van de tokens bij de verschillende servicesystemen, de verblijfstijd in (delen van) het wachtrijnetwerk, de bezettingsgraden van servers en het per tijdseenheid door de servicesystemen verwerkte aantal tokens oftewel de throughput ervan.

Met name voor wachtrijnetwerken kan het afleiden van exacte oplossingen zeer gecompliceerd of onmogelijk zijn. In sommige gevallen kunnen dan nog analytische methodieken toegepast worden die een benadering van de oplossing bepalen, zie [DOR86] en [LAV83]. In de overige gevallen is simulatie de enig begaanbare weg.

3.2.1.1. Analytische oplossingen voor afzonderlijke wachtrijen.

Exacte analytische oplossingen zijn in de eerste plaats vastgesteld voor afzonderlijke wachtrijen. Deze wachtrijen worden geklassificeerd als A/B/C/D/E, met :

- A : de distributie van de interarrivaltijden van de serviceverzoeken.
- B : de distributie van de grootte van de serviceverzoeken
- C : het aantal servers in het servicesysteem.
- D : het aantal plaatsen in de wachtrij voor de server(s).
- E : het aantal bronnen van serviceverzoeken.

Indien C, D en/of E niet gespecificeerd worden, betreft het een oneindig groot aantal.

Voor de distributies A en B worden de volgende afkortingen gebruikt :

- M : de exponentiële verdeling
- Hr : de hyperexponentiële verdeling van graad r.
- Er : de Erlang-r-verdeling
- D : de uniforme verdeling
- G : de algemene verdeling

Bij de analyse van wachtrijen wordt ervan uitgegaan dat het aankomen bij en het verlaten van een servicesysteem steeds op afzonderlijke tokens en nooit op groepen van tokens betrekking heeft.

In dat geval kan een wachtrij beschreven worden als een geboorte-sterfte-proces. Dit is een speciale klasse van Markov-processen waarbij alleen toestandsvergangen optreden tussen naburige toestanden. De toestand is hierbij het aantal tokens dat op een bepaald moment in een wachtrij aanwezig is. Dit betekent voor een wachtrij dat vanuit een toestand met N tokens in de wachtrij alleen kan worden overgegaan naar een toestand met 1 token meer, door een aankomst, dan wel naar een toestand met 1 token minder, door het beëindigen van de serviceverlening aan een token en het vertrek daarvan uit de server. Deze overgangen kunnen in een toestandsdiagram worden weergegeven, zie fig. 3.1.

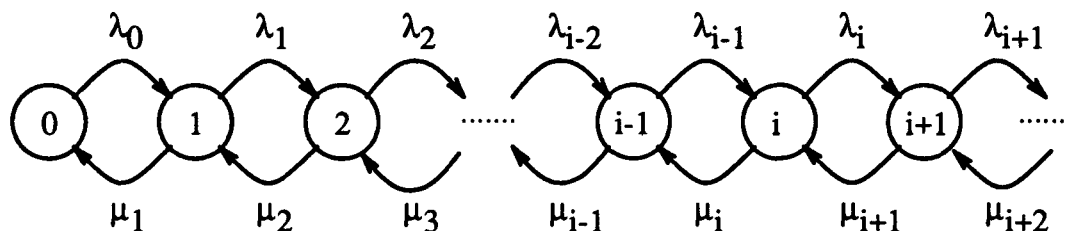


Fig. 3.1 Toestandsdiagram van de M/M/1-queue.

Voor dergelijke geboorte-sterfte-processen kan een uitdrukking voor de stationaire kansverdeling van de toestand van de wachtrij worden afgeleid :

$$p_k = \frac{\prod_{i=0}^{k-1} \lambda_i}{\mu_{i+1}} \bigg/ \left(1 + \sum_{k=0}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right), \quad k \geq 0 \tag{3.2}$$

met λ_i het gemiddeld aantal aankomsten bij de wachtrij per tijdseenheid in toestand $X_t = i$, en μ_i het gemiddeld aantal beëindigingen van serviceverlening per tijdseenheid in toestand $X_t = i$ oftewel het omgekeerde van de gemiddelde servicetijd in toestand $X_t = i$.

Voor specifieke wachtrijen moet deze uitdrukking opgelost worden opdat statistische resultaten voor relevante systeemaspecten zoals queuelengte, verblijfstijden en bezettingsgraad afgeleid kunnen worden.

De wachtrij, waarvoor het afleiden van deze statistische resultaten het eenvoudigst verloopt, is de M/M/1-queue vanwege de exponentiële verdelingen.

Het aantal bronnen bij een M/M/1-queue is oneindig. Derhalve is de verdeling van de aankomsten van de tokens c.q. de verdeling van de interarrival-tijden tussen de tokens onafhankelijk van het aantal tokens dat zich reeds in queue bevindt.

Voor de M/M/1-queue resulteert de uitdrukking voor de stationaire toestandsverdeling in een geometrische verdeling in de bezettingsgraad van de server. De bezettingsgraad is hierbij het quotiënt van het gemiddeld aantal aankomsten per tijdseenheid en het gemiddeld aantal verzoeken dat per tijdseenheid door de server afgehandeld kan worden.

$$p_k = (1-\rho) \cdot \rho^k \quad , \quad k \geq 0 \quad (3.3)$$

met $\rho = \lambda / \mu$ de bezettingsgraad van de server en λ het gemiddelde aantal aankomsten van tokens per tijdseenheid en μ het gemiddeld aantal tokens dat een server kan bedienen indien deze volledig belast wordt.

Voor de M/M/1-queue kunnen op basis van de stationaire toestandsverdeling alle verdelingen en daarmee alle statistische resultaten voor de relevante systeemaspecten afgeleid worden. Naarmate meer van de M/M/1-queue wordt afgeweken worden de afleidingen gecompliceerder en het aantal resultaten, dat bepaald kan worden, beperkter.

Bij de M/G/1-queue zijn de interarrival-tijden negatief-exponentieel verdeeld maar worden geen nadere veronderstellingen gedaan omtrent de servicetijdverdeling. Het aantal bronnen is ook hier oneindig groot verondersteld.

Vanwege de algemene servicetijdverdeling zijn de stochastische processen nu niet meer geheugenloos. Toch kunnen ook nu formules worden afgeleid voor de verwachtingswaarde en de hogere momenten van stochastische grootheden als de queuelengte, de wachttijd en de verblijfstijd van de tokens bij de queue, zie [TUE79]. De grootheden worden dan uitgedrukt in bepaalde aspecten van de algemene verdeling zoals de verwachtingswaarde en de variantie.

3.2.1.2. Analytische oplossingen voor netwerken van wachtrijen.

Het grootste probleem bij de analytische aanpak van wachtrijnetwerken is dat alleen voor een betrekkelijk kleine klasse van deze wachtrijssystemen exacte resultaten bekend zijn.

Onder bepaalde voorwaarden is bij wachtrijnetwerken de toestandsverdeling p_k de oplossing van een stelsel lineaire vergelijkingen, de evenwichtsvergelijkingen.

Een klasse van wachtrijnetwerken waarvoor dit geldt, wordt gevormd door de separeerbare of produktvorm netwerken. Deze netwerken worden gekenmerkt door het feit dat de oplossing van het stelsel evenwichtsvergelijkingen een produktvorm aanneemt. Bij deze netwerken wordt zowel voor de servicetijden als voor de interarrival-tijden uit-

gegaan van exponentiële verdelingen.

Bij separeerbare open wachtrijnetwerken gaat de evaluatie van het evenwichtsgedrag rechtstreeks. Iedere wachtrij kan afzonderlijk gevalueerd worden. Globaal gezien is hierbij de oplossing van het netwerk als geheel gelijk aan het produkt van de oplossingen van de afzonderlijke wachtrijen :

$$p(\underline{k}) = p(k_1, k_2, k_3, \dots, k_n) = p_1(k_1) p_2(k_2) p_3(k_3) \cdots p_n(k_n) \quad (3.4)$$

Hierin is $p_i(k_i)$ voor $1 \leq i \leq n$ de geometrische verdeling die betrekking heeft op wachtrij i en n het aantal wachtrijen in het netwerk.

Bij separeerbare gesloten wachtrijnetwerken is de evaluatie gecompliceerder maar ook hier treedt een produktvormoplossing voor de evenwichtsverdeling op waarin echter nog een nader te bepalen normalisatieconstante aanwezig is.

$$p(\underline{k}) = p(k_1, k_2, k_3, \dots, k_n) = \frac{1}{G(K)} x_1(k_1) x_2(k_2) x_3(k_3) \cdots x_n(k_n) \quad (3.5)$$

Hierin is $x_i(k_i)$ voor $1 \leq i \leq n$ een term die is opgebouwd uit factoren die alleen betrekking hebben op wachtrij i . Het berekenen van de normalisatieconstante $G(K)$ veroorzaakt hier numerieke problemen.

Belangrijker is echter dat de meeste realistische wachtrijnetwerken niet separeerbaar zijn. Zo ook in het onderhavige geval van het model voor ECHO-systemen. Dit systeemmodel heeft verscheidene aspecten die een analytische aanpak verhinderen.

Het systeemmodel is een gesloten wachtrijnetwerk waarin de afzonderlijke servicetijden bij de fileservers niet onafhankelijk en identiek verdeeld zijn. Dit is een standaardaanname bij de analytische aanpak. De servicetijd van een token is bij de fileserver afhankelijk van het type van het voorafgaande token. De servicetijden in het simulatieprogramma zijn ook niet exponentieel verdeeld. Dit is evenmin het geval met de interarrival-tijden tussen opeenvolgende verzoeken om diskblokken bij een leesactie van een casefolder. Verder treedt in het model "multiple resource possession" op, wat inhoudt dat een token tegelijkertijd de serviceverlening van meerdere servers uit verschillende servicesystemen ondergaat.

Een analytische aanpak is in dit geval niet mogelijk.

3.2.2. Simulatie van wachtrijnetwerken.

Simulatie van wachtrijnetwerken wordt toegepast indien het langs analytische weg afleiden van exacte dan wel benaderde resultaten te gecompliceerd of onmogelijk is.

Het op basis van het wachtrijmodel opgestelde simulatieprogramma bevat representaties van de verschillende systeemcomponenten en een beschrijving van de acties die deze

componenten uitvoeren, waaronder de onderlinge interacties tussen deze componenten.

Simulatie van het model bootst het dynamische gedrag van het stochastische systeem na op het niveau van afzonderlijke gebeurtenissen, "events", bij de verschillende systeemcomponenten. De events treden op discrete tijdstippen op en veroorzaken toestandsveranderingen in het systeem. Deze vorm van simulatie wordt derhalve met discrete simulatie aangeduid.

De simulatie wordt gestuurd door met behulp van "pseudo-random-number-generatoren" verkregen waarden voor de stochastische grootheden in het model.

Het oplossen van een probleem door middel van modelsimulaties geschiedt in verschillende fases :

- het ontwerp van het simulatiemodel.
- de implementatie van het simulatiemodel in een simulatietaal.
- het verzamelen van de voor de simulatie vereiste invoergegevens.
- het bepalen van de te verrichten simulatieexperimenten.
- de executie van de simulaties.
- de analyse van de uitvoergegevens en de validering van de simulatie.

Het ontwerp van het simulatiemodel vereist een specificatie van de verschillende componenten van het systeem, van de acties en interacties die in het systeem plaatsvinden, van de invoergegevens, die verzameld moeten worden, en van de uitvoergegevens, waarin men geïnteresseerd is en die men door middel van de simulatie wil bepalen.

Bij het ontwerp van het systeemmodel moet allereerst worden vastgesteld met welke mate van detail de simulator moet werken. Dit is afhankelijk van de doelstelling van de simulatie en van de mate van begrip en kennis van het te simuleren systeem. Een juiste keuze voor deze mate van detaillering is niet triviaal en kan niet altijd vooraf reeds gemaakt worden. Interacties van systeemcomponenten (hard- en software) kunnen zo complex zijn dat niet vooraf is in te schatten wat de invloed van keuzes bij het ontwerp van het simulatiemodel is op de performanceaspecten ervan. Veelal is de bouw van een adequaat model en de simulaties daarvan een iteratief proces.

Voor de implementatie van het simulatiemodel moet een keuze gedaan worden voor een bepaalde simulatietaal. Deze zullen in de volgende paragraaf kort besproken worden.

De invoergegevens van de simulatie kunnen verdeeld worden in gegevens die de gebruiker niet kan dan wel niet wenst te variëren en die als een vaste waarde in het programma worden ingebouwd en gegevens die de gebruiker juist wel in verschillende simulaties wil variëren om de invloed ervan op de resultaten te onderzoeken.

Tot de eerste groep behoort bijvoorbeeld de transmissiesnelheid van een ethernet, die een vast waarde heeft, tot de tweede groep behoort in het algemeen in het systeem geïntroduceerde belasting teneinde de gevolgen daarvan voor de performance van het systeem vast te kunnen stellen. De snelheid van bijvoorbeeld een cpu of een disk kan een vaste of een variabele invoergrootheid zijn afhankelijk van het al dan niet aanwezig

zijn van keuzemogelijkheden voor de gebruiker.

De systeemafhankelijke invoergegevens zullen gemeten of, indien dat niet mogelijk is, zo goed mogelijk geschat moeten worden.

Het model moet de aankomsten van tokens bij de servicesystemen en de serviceverlening aan deze tokens beschrijven en synchroniseren. De simulatie van een systeemmodel strekt zich uit over een bepaald gesimuleerd tijdsinterval waarin het gedrag van het werkelijke systeem wordt nagebootst. Dit is uitdrukkelijk niet de voor de executie van de simulatie op een computersysteem vereiste rekentijd. Het gesimuleerde interval kan zowel in de orde van microseconden als in de orde van jaren liggen afhankelijk van de aard van het te simuleren systeem.

Tijdens de simulatie moet de gesimuleerde tijd vanaf het begin van tijdsinterval van event tot event herhaaldelijk worden opgehoogd totdat het einde van dit tijdsinterval bereikt wordt. De simulatie wordt dan beëindigd en de resultaten kunnen worden uitgevoerd.

De simulator houdt een lijst bij van events, die op bepaalde tijdstippen in de toekomst moeten plaatsvinden. Deze eventlijst bevat tevens deze eventtijden.

Een cyclus binnen de simulatie begint met het verwijderen van de eerstvolgende gebeurtenis uit de eventlijst. Deze gebeurtenis wordt tot "current event" benoemd en de simulatietijd wordt opgehoogd naar de aan dit event gerelateerde event-tijd.

Het optreden van een event gaat gepaard met de executie van één of meerdere in het simulatieprogramma gespecificeerde acties. Deze acties kunnen aanleiding geven tot nieuwe toekomstige events die op de juiste plaats in de eventlijst ingevoegd worden. De simulator baant zich gedurende de simulatie een weg door de eventlijst, die echter steeds met nieuwe events wordt aangevuld en normaliter nooit leeg raakt. De simulatie eindigt bij het bereiken van een door de gebruiker gespecificeerd simulatie-tijdstip.

Tijdens de simulatie treden realisaties op van grootheden waarvan het statistische gedrag juist door middel van de simulatie vastgesteld moet worden. Deze afzonderlijke waarden worden door de simulator verzameld, in specifieke datastructuren opgeslagen en statistisch verwerkt.

Resultaten worden na afloop van de simulatie en eventueel op bepaalde tijdstippen tijdens de simulatie uitgevoerd.

In het algemeen is men geïnteresseerd in het stationaire gedrag van een systeem. Bij de simulaties van het systeemmodel treden aan het begin overgangsverschuiven op voordat ook de simulatie een stationaire fase bereikt. De statistische resultaten van de simulatie dienen in het algemeen alleen betrekking te hebben op gegevens die tijdens de stationaire fase zijn verzameld. De datastructuren, waarin deze gegevens worden opgeslagen, dienen derhalve op een tijdstip aan het begin van de stationaire fase gereset te worden. De duur van de overgangsfase is afhankelijk van de initiële toestand. De stationaire toestand is veelal een toestand die pas in de limiet bereikt wordt. De reset van de datastructuren wordt derhalve gegeven op een tijdstip waarop zeker aangenomen mag worden dat de overgangsverschuiven voor het overgrote deel verdwenen zijn en de stationaire toestand voldoende dicht benaderd wordt.

De duur van de simulatie bepaalt de nauwkeurigheid van de resultaten. De voor een bepaalde nauwkeurigheid benodigde duur dient bepaald te worden op basis van een statistische analyse van de resultaten.

3.2.3. Simulatietalen.

De implementatie van een simulatiemodel wordt een bepaalde simulatietaal of simulatietool gebruikt. Deze hebben de volgende algemene eigenschappen :

- ze bieden een geschikte representatiemogelijkheid voor de verschillende elementen die in simulatiemodellen voorkomen.
- ze beschikken over interne timing- en besturingsmechanismen die vereist zijn voor de executie van een simulatie.
- ze bieden faciliteiten voor het verzamelen van gegevens en statistieken over het gedrag van het gesimuleerde systeem alsmede voor de productie van overzichten van deze statistieken.

De bestaande simulatietalen en -tools kunnen in drie groepen verdeeld worden :

- de visueel-interactieve tools, zoals PAW en RESQ.
- expliciet voor simulatie bedoelde talen, zoals GPSS en SLAM II.
- algemene programmeertalen, eventueel met extensies die simulatietoepassingen vereenvoudigen, zoals SIMULA/DEMOS, SIMENV.

Visueel-interactieve simulatietools beschikken over een grafisch gebruikersinterface waarmee wachtrijmodellen snel, in het algemeen muisgestuurd, ingevoerd en geparametriseerd kunnen worden. Niet in de tool geïmplementeerde aspecten van een wachtrijmodel kunnen echter soms op een gekunstelde wijze maar meestal in het geheel niet gerealiseerd worden.

Expliciet voor simulatie bedoelde talen bieden de gebruiker een verzameling kompakte, krachtige opdrachten. Simulatieapplicaties bestaan uit een sequentie van deze opdrachten, kenmerken zich door een eenvoudige syntax en zijn in het algemeen beperkt van omvang.

Het gebruik van algemene programmeertalen, zoals Simula, voor het schrijven van simulatieapplicaties betekent een aanzienlijke programmeerinspanning.

Om de omvang daarvan te beperken zijn er in deze algemene programmeertalen geïmplementeerde simulatiepakketten opgesteld die de gebruiker een raamwerk bieden waarbinnen deze een simulatieapplicatie kan realiseren. Bij het gebruik van een dergelijk simulatiepakket staan tevens alle faciliteiten en de flexibiliteit van de onderliggende programmeertaal nog ter beschikking van de gebruiker.

DEMOS is een dergelijk pakket, dat in Simula geïmplementeerd is. Het biedt de gebruiker een verzameling datastructuren en procedures en regelt elementaire aspecten als het bijhouden van de event-lijst, het voortstuwen van de simulatietijd en het verzamelen en uitvoeren van statistische gegevens.

SIMENV is een met DEMOS vergelijkbaar pakket, echter geïmplementeerd in Vax-Pascal en Turbo-Pascal, zie [WIT87]. SIMENV is gebruikt voor het realiseren van het simulatieprogramma ECHOSIM.

4. Modelling van ECHO-systemen.

Modelling van computersystemen en -netwerken voor performance analyse doeleinden geschiedt in de vorm van wachtrijnetwerken oftewel queueing networks. Voor ECHO-systemen zijn dergelijke modellen opgesteld op basis waarvan simulaties zijn doorgevoerd.

De modelling van ECHO-systemen is een iteratief proces geweest waarbij eerst is uitgegaan van modellen op een hoog abstractieniveau en later is overgegaan tot meer gedetailleerde modellen.

Modelling van een systeem vereist een abstractie van detailaspecten en het benadrukken van de karakteristieke systeemcomponenten en systeemeigenschappen. In dit geval wordt voor simulatiedoeleinden een model opgesteld van een hiërarchisch gestructureerd computernetwerk. Dit vereist de specificatie van de statische systeemcomponenten, zoals de werkstations, de fileservers en het netwerk, van de dynamische componenten, zoals de verzoeken om serviceverlening en de daaropvolgende resultaatberichten alsmede van de wijze waarop de interactie tussen de verschillende systeemcomponenten plaatsvindt.

De specificatie van deze componenten geeft impliciet het abstractieniveau van het model weer. Beide soorten componenten kunnen namelijk met verschillende mate van detaillering in het model worden ondergebracht.

Zo kan een fileserver in een wachtrijnetwerk in principe worden voorgesteld door één afzonderlijke node maar evenzeer door een groep onderling verbonden nodes waarmee bijvoorbeeld de processor, de disk, de communicatie-controller en eventueel ook de i/o-kanalen en het werkgeheugen ieder afzonderlijk gerepresenteerd worden.

Een dynamische component kan een bestand in zijn geheel voorstellen maar evenzeer betrekking hebben op de afzonderlijke blokken waarin een bestand van disk wordt gelezen of waarin het bestand door de communicatiesoftware voor transport wordt opgedeeld, zoals de ethernet-pakketten.

Binnen een ECHO-systeem bestaat het verkeer tussen de verschillende systeemcomponenten in kwantitatieve zin hoofdzakelijk uit het transport van casefolders en dossiers ofwel images. Dit transport betreft dan voornamelijk leesacties door werkstations vanaf fileservers.

In het vervolg wordt er daarom vanuitgegaan dat het verkeer op het netwerk bestaat uit door werkstations verrichte verzoeken om serviceverlening, die betrekking hebben op het inlezen van een casefolder of dossier, en dat het resultaat het door de fileserver via het netwerk aanleveren van de benodigde casefolder of dossier is.

4.1. Modelling op casefolder-niveau.

De doelstelling bij de modelling van ECHO-systemen is het bepalen van prestatieresultaten als responsetijden en verwerkingscapaciteit met betrekking tot het transport van casefolders, dossiers en images door het systeem.

De aspecten van het transport van bijvoorbeeld afzonderlijke diskblokken of ethernet-

pakketten zijn voor deze doelstelling in wezen minder interessant. Initieel is daarom uitgegaan van modellen op een hoog abstractieniveau waarbij een door een werkstation verricht verzoek om serviceverlening betrekking heeft op een gehele casefolder en waarbij de casefolder als één entiteit, één "token", door het model getransporteerd wordt.

Een dergelijk model is weergegeven in fig. 4.1a. Dit is een gesloten wachtrijnetwerk dat een eenvoudig systeem weergeeft met één ethernet, in principe één fileserver en een willekeurig aantal werkstations.

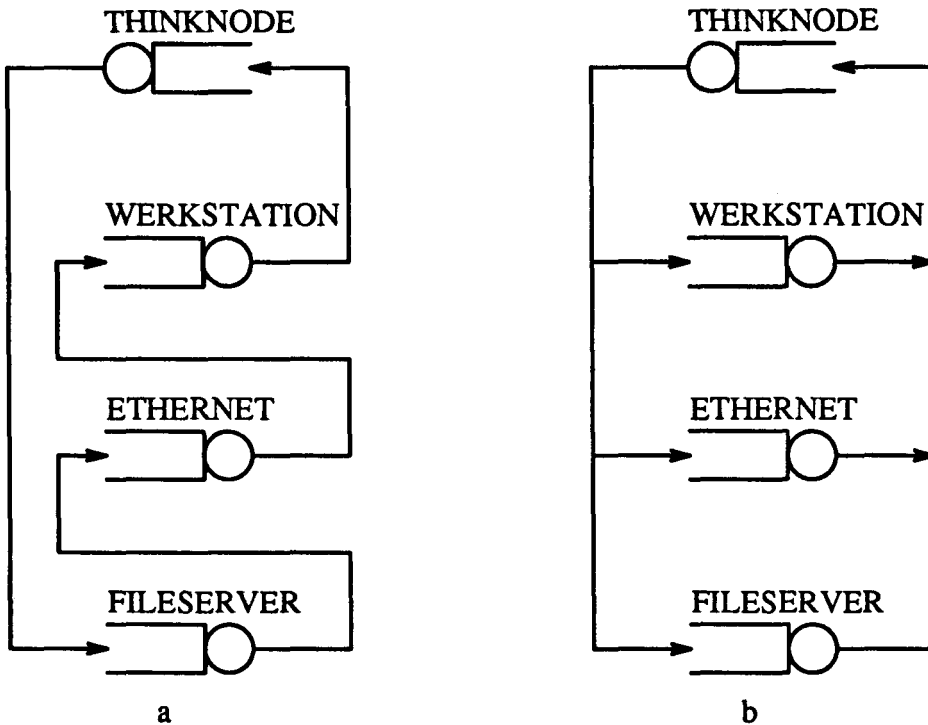


Fig. 4.1 Sequentieel en parallel wachtrijmodel voor ECHO-systemen.

In dit model worden de fileserver en het ethernet beide door een enkele node voorgesteld. Voor beide nodes is de processor sharing queueing discipline aangenomen. Deze aanname is gedaan op grond van het feit dat in werkelijkheid het inlezen van een bestand door NFS vertaald wordt in een reeks verzoeken om blokken uit dat bestand. Indien nu meerdere werkstations gelijktijdig bestanden inlezen, zullen deze van verschillende werkstations afkomstige NFS-verzoeken om datablokken tussen elkaar door bij de fileserver aankomen en in min of meer dezelfde volgorde door de fileserver worden beantwoord. Aldus wordt de servicecapaciteit van het ethernet en de fileserver min of meer gelijkmatig over de gelijktijdig plaatsvindende leesacties verdeeld. Derhalve ligt in dit model voor deze nodes de processor sharing queueing discipline het meest voor de hand.

Het werkstation is logisch opgedeeld in twee nodes : de THINKNODE en de WERKSTATION-node.

Een token, dat in het wachtrijnetwerk circuleert, verblijft gedurende een (in het

algemeen negatief-exponentieel veronderstelde) thinktime in de THINKNODE. De thinktime is de tijd die verloopt tussen twee door een specifiek werkstation verrichte, opeenvolgende leesacties van een casefolder. Gedurende deze thinktime vinden er bij een werkstation alleen lokaal activiteiten plaats en geen (noemenswaardige) acties over het netwerk.

Bij het verlaten van de THINKNODE stelt een token een verzoek om een casefolder voor, bij aankomst bij deze node de casefolder zelf. In fig. 4.1a wordt het verzoek om een casefolder niet via het ethernet maar direct van het werkstation naar de fileserver geleid. Dit is gedaan vanwege het feit dat de tijd die vereist is voor het transport over het ethernet van een enkel verzoekbericht, dat het aanleveren van een casefolder door de fileserver initieert, zeer klein is ten opzichte van de tijd gedurende welke het ethernet belast is met het bulktransport van de casefolder. Indien verzoekberichten wel via het ethernet naar de fileserver geleid zouden worden, zou dat geen noemenswaardige veranderingen in de resultaten tot gevolg hebben.

Een token verblijft in de WERKSTATION-node gedurende de tijd die gepaard gaat met de verwerking van communicatiesoftware door het werkstation tijdens het inlezen van een casefolder. Deze tijd is mede bepalend voor de totale tijd die tijdens het inlezen van een casefolder verloopt voordat deze lokaal in het werkstation ter beschikking van de applicatie staat. De WS-node is ingevoerd om de communicatie-georiënteerde en de overige activiteiten binnen het werkstation duidelijk van elkaar te scheiden.

De THINK- en de WERKSTATION-node zijn ieder een N-voudig server-complex en geven aldus N werkstations weer. De queueing disciplines bij deze nodes zijn irrelevant indien wordt uitgegaan van de veronderstellingen dat ieder werkstation hooguit één casefolder tegelijk inleest en dat er tijdens het inlezen van een casefolder geen overige processen op het werkstation actief zijn c.q. dat het leesproces volstrekte prioriteit geniet. In dit geval treden er voor wat betreft het gegevenstransport bij de werkstations wel servicetijden maar geen wachttijden op en is de queueing discipline bij deze nodes derhalve irrelevant.

Voor de modellering en de simulaties is steeds van deze veronderstellingen uitgegaan.

Tijdens het onderzoek is op grond van de resultaten van simulaties, die op basis van dergelijke eenvoudige, abstracte modellen zijn uitgevoerd, geconcludeerd dat het abstractieniveau bij deze modellering te hoog is. Het is bij deze abstracte modellen niet mogelijk gebleken zowel de totale verblijfstijd van het token in het gehele wachtrijnetwerk als de verblijfstijden, servicetijden en bezettingsgraden bij de afzonderlijke nodes korrekt te realiseren. Met de verblijfstijd van het token in het gehele wachtrijnetwerk, de "netwerkverblijfstijd", wordt hierbij het tijdsinterval bedoeld dat ligt tussen het moment waarop een verzoek om een casefolder de thinknode verlaat en het tijdstip waarop de volledige casefolder bij dezelfde thinknode is aangekomen.

Dit probleem wordt veroorzaakt doordat gedurende de overdracht van een casefolder er in werkelijkheid gelijktijdig in verschillende componenten van het systeem, de fileserver, het netwerk en het werkstation, op deze overdracht betrekking hebbende activiteiten plaatsvinden.

Zo treedt bijvoorbeeld de belasting van het ethernet altijd simultaan op met die van op zijn minst twee communicatiecontrollers, die van de bron en de bestemming van een

pakket. Indien een blok data in de vorm van een reeks opeenvolgende ethernet-pakketten wordt overgezonden, zal op het moment dat pakket nummer i op het ethernet aanwezig is (de processor van) het ontvangende werkstation belast zijn met het afhandelen van communicatiesoftwaremodules voor pakket $i-1$ en de zendende fileserver met de overeenkomstige software voor pakket $i+1$. Dergelijke overlappingsen treden ook op voor de afzonderlijke NFS-blokken die samen een casefolder vormen. De netwerkverblijfstijd van een token, dat een gehele casefolder voorstelt, is aldus in werkelijkheid kleiner dan de som van de verblijfstijden van het token bij de afzonderlijke nodes.

Er is vervolgens geprobeerd ook in het model belastingen van verschillende nodes simultaan te laten plaatsvinden. Een alternatief, zij het niet erg gebruikelijk model, waarin de belasting van de verschillende nodes volledig parallel plaatsvindt, is weergegeven in fig. 4.1b. In dit model zal een token door zich in drieën te splitsen simultaan de verschillende nodes belasten. Voor de netwerkverblijfstijd zal de langste van de bij de verschillende nodes gerealiseerde verblijfstijden genomen worden. Dat zal in dit model in het algemeen de verblijfstijd bij de fileserver zijn. Dit model heeft een analoge tekortkoming namelijk dat in werkelijkheid de belasting van de verschillende nodes evenmin volkomen parallel plaatsvindt.

Bij modellen waarin de serviceverlening aan een specifiek token bij de verschillende nodes gedeeltelijk sequentieel en gedeeltelijk parallel verloopt, dient de aanvang en de beëindiging van de serviceverlening bij deze afzonderlijke nodes op een bepaalde manier gesynchroniseerd te worden. Het implementeren van dergelijke synchronisatiemechanismen in deze abstracte modellen levert constructies op die de complexiteit van de gebeurtenissen, zoals deze in werkelijkheid plaatsvinden, op een alternatieve, weinig realistische wijze proberen weer te geven en tenderen in de richting van het toevoegen van detailaspecten aan het model.

Op grond van het bovenstaande is geconcludeerd dat de modellering op dit hoge abstractieniveau niet volstaat en dat gekozen dient te worden voor een meer gedetailleerde modellering die dichter staat bij de gebeurtenissen die in werkelijkheid in het systeem optreden.

4.2. Modellering op ethernet-pakketniveau.

De modellering op het hoge abstractieniveau, waarbij afzonderlijke tokens gehele casefolders voorstellen, is opgevolgd door een modellering op ethernet-pakketniveau, waarop de gebeurtenissen zich ook in werkelijkheid afspelen.

Het via het netwerk inlezen van een casefolder met een omvang van 500 kB leidt tot de overdracht van ongeveer 500 pakketten over het ethernet. Een consequentie van de modellering op pakket-niveau is dat veel extra events gecreëerd worden in de modelsimulaties. Aangezien de afzonderlijke gebeurtenissen zich nu op een veel fijnere tijdschaal afspelen zijn er ook meer en nauwkeuriger gegevens nodig voor het parametriseren van de modelsimulaties.

ECHO-systemen zijn hiërarchisch gestructureerde computernetwerken. Hiervoor kan een generiek model in de vorm van een wachtrijnetwerk worden opgesteld, zie

[SCH87]. Eén segment uit dit hiërarchisch gestructureerde wachtrijnetwerk met eventuele koppelingen naar hogere en lagere segmenten is afgebeeld in fig. 4.2. De koppelingen tussen de verschillende ethernet-segmenten worden gerealiseerd door bridges.

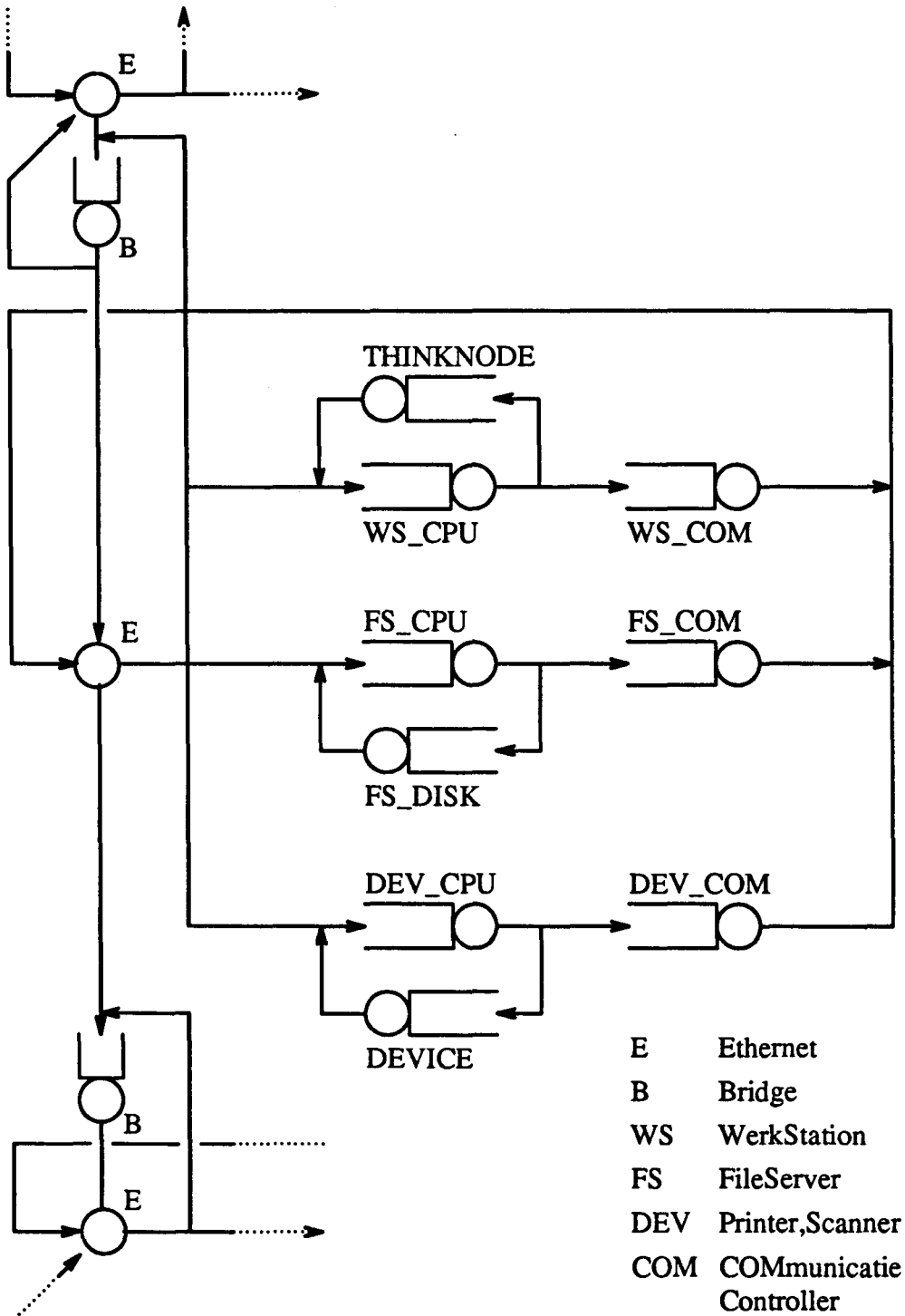


Fig. 4.2 Generiek wachtrijmodel voor ECHO-systemen.

In dit generieke model worden de op een ethernet aangesloten systeemcomponenten

ieder weergegeven door meerdere nodes, namelijk een processor, een communicatie-controller en daarnaast voor een file-, scanner- of printerserver een disk, scanner of printer.

De communicatie-controller moet in het model een voldoende realistische toegang tot het ethernet mogelijk maken. Voor het ethernet wordt een eenvoudige FCFS-queueing discipline gebruikt.

Een gedetailleerde modellering van het CSMA/CD-toegangsprotocol zou tijdens de simulatieruns nog meer events veroorzaken en weinig bijdragen aan de resultaten op casefolderniveau.

Op ieder niveau in de netwerkhierarchie worden verscheidene systeemcomponenten aangesloten. De samenstelling van de aangesloten componenten hangt af van het betreffende niveau. Zo zullen op het hoogste niveau, de backbone, relatief weinig werkstations aanwezig zijn en relatief veel fileservers, waarop dan gegevens worden opgeslagen die voor alle medewerkers van belang kunnen zijn, terwijl op de lagere departementale niveaus relatief veel werkstations aanwezig zullen zijn.

Het is niet noodzakelijk het generieke model, dat een willekeurig grote uitgebreidheid kan hebben, in zijn geheel te simuleren. Dit zou een eveneens willekeurig uitgebreid simulatieprogramma en alsmede willekeurig veel rekentijd vergen. Het is mogelijk de simulatie te beperken tot afzonderlijke segmenten waarbij de koppelingen naar andere segmenten worden weergegeven in de vorm van externe bronnen en bestemmingen van verkeer.

Het vaststellen van korrekte parameters op pakket-niveau kan voor deze externe bronnen en bestemmingen gecompliceerder zijn dan voor lokale bronnen en bestemmingen. Bij verschillende keuzes voor de configuratie en aannames omtrent het verkeer tussen de verschillende logische systeemcomponenten kunnen dan simulaties worden doorgevoerd en performanceresultaten worden afgeleid.

4.3. Het gebruikte ECHO-systeemmodel.

Het opgestelde simulatieprogramma is geen volledige implementatie van het generieke model. Enerzijds zijn er in dit generieke model beperkingen aangebracht in gevallen waar details niet voldoende interessant lijken om ze te implementeren, anderzijds zijn bepaalde aspecten van het generieke model nog niet gerealiseerd.

Het simulatieprogramma bevat één type proces dat leesacties genereert. Er kan een willekeurig groot aantal van deze processen gedeclareerd worden die alle identiek geparametriseerd worden. De processen representeren (de leesacties van) de werkstations die direkt op het betreffende segment zijn aangesloten. De binnen een bepaald segment optredende leesacties zullen voor het grootste deel afkomstig zijn van deze werkstations.

Externe bronnen kunnen in het simulatieprogramma weergegeven worden door ze op te vatten als een aantal additionele werkstations, met dan wel dezelfde parametrisering als de direkt op het segment aangesloten werkstations.

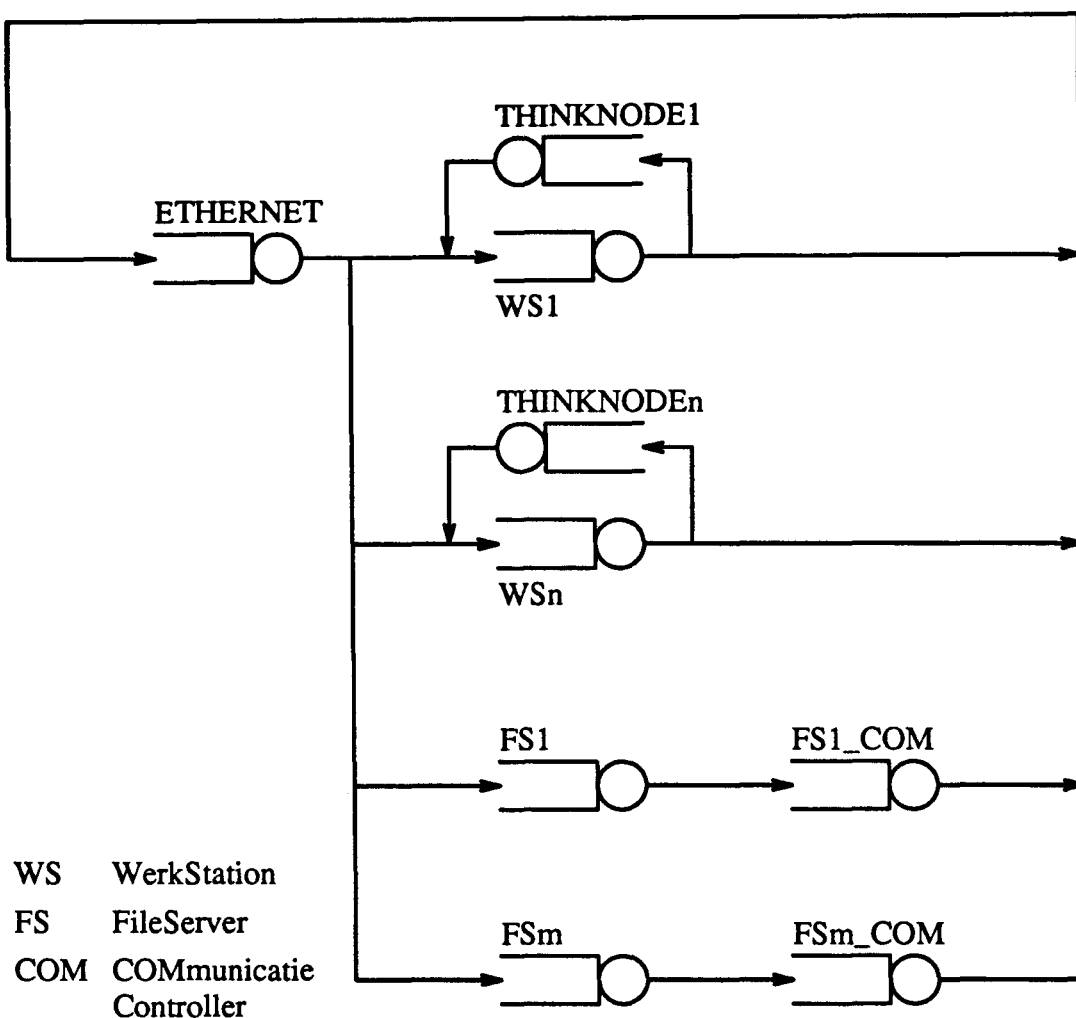


Fig. 4.3 Gebruikte wachtrijmodel voor ECHO-systemen.

In het simulatieprogramma kan eveneens een willekeurig aantal fileservers gedeclareerd worden. De fileservers zijn afzonderlijk parametrizeerbaar.

Een fileserver, die niet op het lokale segment is aangesloten, kan voorzien worden van grotere parameters oftewel langere servicetijden. Daarin zijn dan de extra tijden verdisconteerd die nodig zijn voor het transport door het netwerk, via de bridges, naar en van de betreffende fileserver alsmede de wachttijden die bij deze fileserver optreden ten gevolge van de belasting ervan door systemen die op andere segmenten van het netwerk aangesloten zijn.

De processor en de communicatie-controller van het werkstation worden in het simulatieprogramma gezamenlijk door één node voorgesteld. De servicetijden en bezettingsgraden van deze subcomponenten afzonderlijk zijn niet bijzonder belangrijk. Een werkstation wordt niet beschouwd als een systeemcomponent die door andere componenten gedeeld wordt zoals wel het geval is met het netwerk en de fileservers.

Tenslotte worden in het simulatieprogramma eveneens de processor en de disk van de fileserver gezamenlijk door één node voorgesteld. Dit is gedaan vanwege het feit dat het niet mogelijk is geweest om binnen de fileserver metingen te verrichten en aldus meetresultaten te verzamelen voor de activiteiten die binnen de fileserver optreden gedurende het afhandelen van leesacties. Dit betreft de servicetijden bij de processor en de disk afzonderlijk, de mate waarin "read-ahead" oftewel "prefetching" van diskblokken optreedt alsmede de invloed van het door de fileserver gehanteerde "disk-scheduling"-algoritme voor de binnenkomende verzoeken om blokken data.

Zonder nadere gegevens hieromtrent moet de interactie tussen de processor en de disk van de fileserver volledig gebaseerd worden op schattingen en veronderstellingen wat de betrouwbaarheid van het model niet ten goede komt.

Derhalve is ervoor gekozen de fileserver met twee nodes weer te geven waarbij de eerste node de primaire serviceverlening aan een verzoek realiseert, wat de afhandeling van communicatiesoftware en het inlezen van diskblokken omvat, en de tweede node die de afzonderlijke pakketten, waarin een blok data wordt opgesplitst, over het ethernet verzendt en aldus als representatie van de communicatie-controller kan worden opgevat.

Het resulterende systeemmodel is weergegeven in fig. 4.3. Dit model dient als basis voor het simulatieprogramma.

5. Metingen.

5.1. Doelstellingen van de metingen.

Allereerst worden hier de redenen voor het meten van performanceaspecten van het PCG-netwerk uiteengezet. Deze redenen hebben achtereenvolgens betrekking op :

- de modellering van ECHO-systemen.
- de parametrisering van het systeemmodel.
- de validering van het systeemmodel en de modelsimulaties.

5.1.1. Metingen voor de modellering van ECHO-systemen.

Voor de analyse van de performance van ECHO-systemen is er nog niet de beschikking over een bestaande implementatie van een dergelijk systeem. Inzicht in de performance van toekomstige ECHO-systemen is echter reeds nu gewenst om vooraf een oordeel te kunnen vellen over de adequaatheid en haalbaarheid van specifieke configuraties voor ECHO-systemen met betrekking tot de performance ervan alsmede voor de bepaling van de wijze waarop ECHO-systemen geïmplementeerd moeten worden. Dit inzicht kan derhalve op dit moment slechts verkregen worden door de performance analyse te verrichten op basis van een model voor ECHO-systemen. De waarde van het op deze wijze verworven inzicht wordt volledig bepaald door de adequaatheid waarmee het model werkelijke ECHO-systemen, voor wat betreft de performanceaspecten daarvan, weergeeft.

De in ECHO-systemen aanwezige computersystemen, de hardware en zeker ook de software waaruit deze zijn samengesteld, alsmede de onderlinge interactie van deze computersystemen via het netwerk zijn uitermate complex van aard. De performance van het gehele systeem wordt derhalve bepaald door een veelheid van aspecten en factoren. Deze aspecten zijn echter niet alle in gelijke mate van invloed op de performance.

Bij de modellering moet een zekere mate van abstractie toegepast worden om de complexiteit van de modellen binnen redelijke grenzen te houden. Op grond hiervan komen de aspecten, die het sterkst op de performance van invloed zijn, in aanmerking om bij de modellering te worden meegenomen terwijl bepaalde aspecten, die daarop in mindere mate van invloed zijn, bij de modellering eventueel weggelaten kunnen worden.

De modellering van een ECHO-systeem geschiedt in de eerste plaats op basis van kennis die verkregen is door het bestuderen van beschrijvingen van de aspecten die bepalend zijn voor de performance van ECHO-systemen. Dit kan bijvoorbeeld betrekking hebben op manuals, technische specificaties en de broncode van het besturingssysteem. De relatieve invloed, die de afzonderlijke aspecten op de performance hebben, is echter niet altijd uit beschrijvingen af te leiden. Metingen

bieden dan aanvullende informatie bij de keuze of een bepaald aspect wel of niet in een model ondergebracht moet worden en met welke mate van detail dat eventueel moet geschieden.

5.1.2. Metingen voor de parametrisering van het systeemmodel.

Metingen kunnen dus een bijdrage leveren bij de modellering van ECHO-systemen. Voor de parametrisering van de daaruit resulterende modellen is het verrichten van metingen echter een absolute noodzaak.

Zo vereist de gegevensoverdracht tussen verschillende computersystemen zowel bij de bron als de bestemming van het gegevensverkeer de executie van verscheidene softwaremodules. De hiermee gepaard gaande executietijden kunnen alleen door middel van metingen worden vastgesteld.

Bij het lezen van of schrijven naar een fileserver van een blok data door een werkstation zijn deze executietijden bepalend voor de van de processoren van de beide systemen benodigde servicetijden. En daarmee eveneens bepalend voor de totale daarvan benodigde servicetijden bij het inlezen van een gehele casefolder.

Bij de gegevensoverdracht tussen verschillende computersystemen zijn de totale responsetijden evenzeer afhankelijk van de servicetijden bij de hardwarecomponenten zoals de disk, de communicatie-controller, het ethernet en eventuele bridges. Deze servicetijden kunnen eveneens door metingen worden vastgesteld. De resultaten hiervan zijn, afhankelijk van de betreffende hardwarecomponent, in meer of mindere mate voorspelbaar.

Zo is de tijd, die nodig is voor het transport over het ethernet van een pakket met een bepaalde grootte, een deterministische grootheid vanwege de vaste transmissiesnelheid van het ethernet. De wachttijd, die ligt tussen het moment waarop een ethernet-controller een pakket klaar heeft voor verzending en het moment waarop deze controller er in slaagt het net te verwerven en het pakket daadwerkelijk te verzenden, is echter een belastingafhankelijke, stochastische grootheid vanwege de eigenschappen van het binnen Ethernet gebruikte CSMA/CD-toegangsprotocol.

Bij de disk van de fileserver is de tijd, die vereist is voor het lezen van een reeks blokken uit een bestand, afhankelijk van de mate waarin fragmentatie optreedt bij de plaatsing van de verschillende blokken van een bestand op de disk alsmede van het aantal processen dat gelijktijdig blokken (van verschillende bestanden) inleest. Indien dit door meerdere processen gelijktijdig geschiedt, zullen twee na elkaar bij de fileserver binnekomende verzoeken om een blok in het algemeen van verschillende processen afkomstig zijn. Deze na elkaar gevraagde blokken uit verschillende bestanden zullen zeker niet na elkaar op de disk zijn opgeslagen. In dit geval treden relatief lange servicetijden op in vergelijking met het door één proces aanvragen van (opeenvolgende) blokken uit één bepaald bestand. In dit laatste geval zullen de blokken zich namelijk, afhankelijk van de mate waarin fragmentatie optreedt doch relatief vaak, wel na elkaar op de disk bevinden waardoor tijdens het inlezen van deze blokken relatief weinig bewegingen van

de leeskop van de disk vereist zijn. Deze "seek"-tijd bepaalt in sterke mate de van de disk vereiste servicetijd.

Bij de voor een specifieke leesactie vanaf de disk vereiste leestijd speelt tevens het door de fileserver gehanteerde algoritme voor het in een wachtrij plaatsen van de verzoeken om diskblokken een rol. De invloed hiervan kan onderzocht worden door de volgorde waarin deze verzoeken bij de fileserver aankomen te vergelijken met de volgorde waarin ze door de fileserver met de diskblokken beantwoord worden. Hierbij moet echter ook weer rekening gehouden worden met de door de fileserver zelf gegenereerde verzoeken om diskblokken in het kader van de "prefetching"- oftewel "read-ahead"-taktiek die de fileserver toepast.

Metingen aan de ethernet-controller leveren informatie op over de snelheid waarmee deze in staat is opeenvolgende pakketten over het ethernet te verzenden. Hierbij dient er voortdurend data in de wachtrij van de controller aanwezig te zijn en moet het ethernet vrij zijn opdat deze meetresultaten alleen afhankelijk zijn van de beschouwde ethernet-controller.

In het algemeen is een ethernet-controller niet in staat om opeenvolgende pakketten direkt na elkaar te verzenden maar bevindt zich tussen de pakketten een klein tijdsinterval. Dit betekent dat via één enkele controller (over meerdere pakketten gemiddeld) nooit een 100% belasting van het ethernet gerealiseerd kan worden. In de praktijk ligt de belasting, die door een enkel aangesloten systeem op het ethernet gecreëerd wordt, overigens aanzienlijk lager door overhead bij de processor en eventuele disk-I/O.

Tenslotte geven metingen aan bridges informatie over de daar optredende vertragingen bij de overdracht van een pakket van het ene naar het andere op de bridge aangesloten ethernet-segment.

Steeds geldt bij deze voor de parametrisering van de modellen te verrichten metingen dat het type en de nauwkeurigheid ervan afhankelijk is van het niveau waarop, oftewel de mate van detail waarmee, de modellering zich voltrekt.

Hier dient direkt vermeld te worden dat we niet in staat zijn geweest ook al deze mogelijke metingen daadwerkelijk uit te voeren. Deels vanwege het niet aanwezig zijn van de betreffende hardwarecomponenten, bijvoorbeeld bridges, deels omdat er niet de beschikking was over de daarvoor vereiste meetinstrumenten. Dit geldt bijvoorbeeld voor het meten van de executietijden van de softwaremodules en van de servicetijden van de disk. Deze metingen zijn echter wel van cruciaal belang voor het inzicht in wat er zich met name binnen de fileservers afspeelt in het bijzonder indien deze zwaar belast worden door meerdere, gelijktijdig leesacties uitvoerende processen.

5.1.3. Metingen voor de validering van de modelsimulaties.

Behalve voor de parametrisering van het model voor ECHO-systemen zijn metingen evenzeer van belang voor de validering van de op basis van het model verrichte simula-

ties.

De validering geschiedt door het model toe te snijden op het PCG-netwerk en zowel het netwerk als de modelsimulaties te onderwerpen aan overeenkomstige belastingen en vervolgens de resultaten van de metingen en de simulaties te vergelijken.

De voor validering verrichte metingen hebben betrekking op de totale responsetijden, die optreden bij het inlezen van bepaalde hoeveelheden data, bij een bepaalde belasting van het gehele systeem.

De aard van de hierbij gebruikte belastingen is zoveel mogelijk analoog genomen aan die welke in ECHO-systemen zullen voorkomen. Dit betreft het herhaaldelijk vanaf fileservers inlezen van grote bestanden door werkstations.

De frequentie waarmee afzonderlijke werkstations bestanden inlezen is bij deze metingen (aanzienlijk) hoger dan wat in de praktijk bij ECHO-systemen voor individuele caseworkers verwacht wordt. Dit is gedaan om ondanks het beperkte aantal werkstations in het PCG-netwerk toch de middelmatige of hoge belastingen op met name de fileservers te kunnen creëren, zoals die in ECHO-systemen zullen optreden.

Voor de validering is het van belang dat het netwerk en de fileservers in het model niet alleen korrekte responsetijden opleveren indien er slechts één enkele leesactie in het systeem optreedt, maar ook realistisch reageren op een middelmatige of hoge belasting in het geval meerdere leesacties simultaan plaatsvinden.

Het is echter uitdrukkelijk niet het geval dat de performanceresultaten, zoals de responsetijden bij het inlezen van een casefolder, die worden vastgesteld wanneer de werkstations bijvoorbeeld een vijfmaal zo hoge leesfrequentie hanteren, representatief zijn voor een ECHO-systeem met vijfmaal zoveel werkstations c.q. caseworkers.

Zo wordt bijvoorbeeld de bovengrens voor de responsetijd bij een leesactie benaderd indien alle actieve werkstations tegelijk een leesactie vanaf één en dezelfde fileserver starten. De responsetijd is dan ongeveer gelijk aan het aantal actieve werkstations maal de servicetijd per leesactie. Deze maximale responsetijd ligt bij een vijfmaal zo groot aantal werkstations dus ook vijfmaal zo hoog. De frequentieverdeling van de responsetijd zal bij een vijfmaal zo groot aantal werkstations, waarbij ieder afzonderlijk werkstation wel een vijfmaal lagere leesfrequentie hanteert, toch bij grote responsetijden aanzienlijk hogere waarden hebben en daardoor ook een significant hogere gemiddelde waarde.

Verschillen tussen de meetresultaten en de overeenkomstige simulatieresultaten hebben herhaaldelijk geleid tot een onderzoek naar mogelijke oorzaken en vervolgens aanpassingen van en aanvullingen op het simulatieprogramma.

5.2. De meetinstrumenten.

Voor het verrichten van de metingen aan het PCG-netwerk hebben de volgende middelen ter beschikking gestaan :

- het meetprogramma Simulation
- een Excelan network analyzer.
- de utilities traffic en etherfind.

5.2.1. Het meetprogramma Simulation.

Voor de validering van de modelsimulaties is gebruik gemaakt van het meetprogramma Simulation. Dit programma is tijdens de eerste maanden van dit afstudeerwerk geschreven door een collega-afstudeerder, zie [ZEG87]. De doelstelling, structuur en mogelijkheden ervan zijn in onderling overleg bepaald.

Dit meetprogramma biedt onder meer de mogelijkheid om op tot twaalf verschillende werkstations een leesproces op te starten. Ieder proces gaat er vervolgens toe over om op bepaalde tijdstippen een bepaalde hoeveelheid data te lezen uit een op een fileserver aanwezig bestand. Ieder leesproces symboliseert aldus de activiteiten van een caseworker.

De met de leesacties gepaard gaande responsetijden worden door ieder proces gemeten en na iedere leesactie naar een eigen logfile op de fileserver echo weggeschreven. Het besturingsproces van het meetprogramma, dat de verschillende meetprocessen opstart, draait zelf normaliter op één van de niet gebruikte werkstations.

De gebruiker dient het meetprogramma van de nodige waarden voor de invoerparameters te voorzien. Dit betreft in de eerste plaats :

- de identificatie van de werkstations waarop een leesproces moet worden opgestart.
- het tijdstip waarop deze processen moeten beginnen.
- de tijdsduur gedurende welke ze actief moeten zijn.

Verder dient voor ieder meetproces apart te worden opgegeven :

- de grootte van de hoeveelheid data die bij een leesactie door het meetproces vanaf de fileserver moet worden ingelezen.
- het tijdsinterval, de thinktime, tussen opeenvolgende leesacties.
- de fileserver , echo of galm, waarop de leesactie moet worden uitgevoerd.

Bij de hoeveelheid data en de thinktime moet een type verdeling, een verwachtingswaarde en eventueel een standaarddeviatie, worden gespecificeerd. Bij de hoeveelheid data kan onder meer een normale verdeling en bij de thinktime een negatief-exponentiële of eveneens een normale verdeling opgegeven worden.

Het is ook mogelijk om bij een specifieke leesactie een bepaald percentage van de totale hoeveelheid in te lezen data van echo te betrekken en het resterende deel vervolgens van galm.

Het invoeren van de parameters voor het meetprogramma geschiedt via een aantal invoerschermen. Er kunnen aldus tot twaalf processen, ieder voor een bepaald werksta-

tion, gespecificeerd worden en eventueel twee leesprocessen op de beide fileservers.

Bij de metingen zijn de leesprocessen echter alleen op de werkstations opgestart en niet ook nog op de fileservers zelf. Dit laatste gaat, indien een fileserver van zijn eigen achtergrondgeheugen leest, niet gepaard met datatransport over het netwerk. Het is in het algemeen bij zwaar belaste fileservers, zoals in ECHO-systemen het geval zal zijn, raadzaam gebruikers niet lokaal op een fileserver werkzaam te laten zijn, vanwege

- 1 de extra belasting die deze gebruikers op de fileserver zouden veroorzaken met de daarmee samenhangende extra vertragingen voor de overige in het systeem aanwezige gebruikers, voor zover deze gebruik maken van de betreffende fileserver.
- 2 de relatief grote responsetijden die de gebruikers, die lokaal op de fileserver werkzaam zijn, zullen ervaren bij zware belasting van deze gemeenschappelijk gebruikte systeemcomponent.

5.2.1.1. De meetmethode en de meetomstandigheden.

Het PCG-netwerk, waaraan de metingen verricht worden, is een operationeel systeem. Overdag wordt dit netwerk door de gebruikers belast terwijl 's nachts op bepaalde tijden zware systeemprocessen actief zijn onder andere voor backups.

Gedurende deze tijden zijn metingen niet zinvol vanwege de vele overige belastingen in het systeem. De metingen zijn daarom buiten deze tijden, voornamelijk 's nachts, uitgevoerd.

Na de parametrisering van de meting wordt de meetjob, samen met het tijdstip waarop deze moet beginnen, door middel van het "at"-commando in de "at"-queue van het werkstation geplaatst. Deze queue wordt ieder kwartier door het besturingssysteem onderzocht op jobs die voor activering in aanmerking komen. Aldus wordt het besturingsproces van het meetprogramma ongeveer op het door de gebruiker gespecificeerde tijdstip op het werkstation geactiveerd.

Het besturingsproces verzamelt allereerst gegevens over de toestand van de verschillende systemen in het PCG-netwerk.

Van ieder werkstation, dat actief, "up", is, wordt de belasting vastgesteld. Deze belasting wordt opgegeven als het aantal jobs dat op een werkstation actief is. Aldus kan worden vastgesteld of op sommige werkstations nog applicatieprocessen aanwezig zijn. Veelal betreft dit processen die niet korrekt beëindigd zijn bij het uitloggen door de gebruiker. Soms bevinden deze processen zich in een oneindige lus.

De aanwezigheid van dergelijke processen kan de waarde van de meetresultaten sterk beperken of geheel teniet doen met name indien deze processen een niet-verwaarloosbare activiteit over het netwerk of op de fileserver veroorzaken.

Hierna vraagt het besturingsproces de processtatussen op van de beide fileservers om nauwkeuriger te kunnen vaststellen of en zo ja welke ongewenste processen en activiteiten daarop aanwezig zijn.

Deze statusinformatie wordt eveneens na afloop van de metingen verzameld.

De aldus verkregen informatie over de toestand van het systeem voor en na de metingen wordt voor latere analyse in een apart logfile opgeslagen. Er kan op basis van deze gegevens niet worden vastgesteld of er tijdens de metingen nog eventuele stoorprocessen zijn opgetreden.

De metingen beginnen nadat het besturingsproces op alle door de gebruiker opgegeven (en tevens actieve) werkstations een meetproces heeft opgestart. De meetprocessen krijgen de door de gebruiker opgegeven tijdsduur voor de meting als argument mee. Deze processen bepalen aan de hand van de systeemtijd van het werkstation of deze tijdsduur verstreken is en zullen in dat geval de leesacties staken. Het besturingsproces zelf slaapt gedurende deze tijdsduur. Na afloop daarvan zal het besturingsproces de meetprocessen afbreken, voorzover die dan nog niet uit zichzelf geëindigd zijn. Aldus wordt verzekerd dat er geen meetprocessen, om welke reden dan ook, actief blijven en daardoor eventuele volgende meetjobs kunnen verstoren.

Een cyclus binnen het meetproces begint met het bepalen van de bij de eerstvolgende leesactie in te lezen hoeveelheid data. Dit geschiedt op basis van de door de gebruiker ingevoerde gegevens en de random number generator van het besturingssysteem. Deze hoeveelheid data wordt gelezen uit een voor ieder werkstation specifiek bestand op de door de gebruiker opgegeven fileservers door middel van een "read"-systeemprocedure waaraan deze hoeveelheid als parameter wordt meegegeven. (Het in twee delen vanaf beide fileservers inlezen van de totale hoeveelheid data is ook mogelijk.)

De tijd die met de leesactie gepaard gaat wordt gemeten door zowel vlak ervoor als erna de systeemtijd op te vragen. Deze tijd heeft een resolutie van 20 ms. Direct na afloop van de leesactie wordt de wachttijd tot aan het begin van de volgende leesactie berekend. Dit geschiedt eveneens op basis van door de gebruiker ingevoerde gegevens en de random number generator.

Vervolgens wordt een string opgebouwd met gegevens over de afgelopen leesactie. Dit betreft onder meer de zojuist besproken grootheden als de responsetijd en de volgende wachttijd. Deze logregel wordt naar een voor ieder werkstation afzonderlijk logfile geschreven. Na afloop van de metingen worden deze aparte logfiles tot één groot logfile samengevoegd om statistische resultaten voor responsetijden en belasting af te leiden. Het is niet mogelijk gebleken een voor alle werkstations gemeenschappelijk logfile te gebruiken daar het locken van dit bestand, dat voor het wegschrijven van een logregel nodig is, teveel tijd vereist en een te grote vertraging van de meetprocessen veroorzaakt.

Voor het loggen is ook bij afzonderlijke logfiles in het bijzonder bij zware belastingen van de fileservers enige tijd vereist. Deze tijd wordt op dezelfde wijze bepaald als de responsetijden van de leesacties en van de zojuist berekende wachttijd afgetrokken. Het meetproces slaapt gedurende het resterende gedeelte van deze wachttijd indien deze positief is.

5.2.1.2. De cache-problematiek.

In de praktijk zullen de casefolders, die een caseworker achtereenvolgens leest, steeds verschillend zijn. In onze meetomgeving bestaat echter niet de mogelijkheid om ettelijke honderden megabytes aan dummy-casefolders te creëren. Ieder proces betreft daarom de data bij de opeenvolgende leesacties steeds uit hetzelfde bestand op de benaderde fileserver. Hiertoe is op beide aanwezige fileservers voor elk van de twaalf werkstations afzonderlijk een 1 MB-groot bestand aangebracht.

De reden voor het gebruik van tamelijk grote en voor ieder werkstation afzonderlijke bestanden is het feit dat zowel de werkstations als de fileservers beschikken over een cachegeheugen. Tijdens de leesacties door de processen zijn daarin bij de fileserver de laatste van disk gelezen blokken en bij de werkstations de laatste via het netwerk ontvangen blokken aanwezig.

Bij het gebruik van een gemeenschappelijk bestand voor de leesacties van de verschillende processen bestaat er een bepaalde kans dat blokken waarom verzocht wordt zich reeds in de cache van de fileserver bevinden vanwege eerdere, door andere processen verrichte verzoeken om dezelfde blokken. De fileserver hoeft in deze gevallen de disk niet te benaderen. Deze situatie zal zich in de praktijk nooit in een ECHO-systeem voordoen en dient dus ook bij de metingen vermeden te worden.

Bij het gebruik van kleine bestanden bestaat de kans dat dit gehele bestand of delen daarvan zich ten gevolge van voorgaande leesacties reeds in de cache van het werkstation bevinden waardoor het kan voorkomen dat volgende verzoeken om blokken data niet of niet altijd over het netwerk naar de fileserver geleid worden maar direkt vanuit de cache worden vervuld. Deze situatie dient eveneens vermeden te worden.

De grootte van het cachegeheugen bedraagt bij de aanwezige Sun systemen ongeveer tien procent van het (vrij beschikbare) werkgeheugen. De werkstations zijn standaard voorzien van een 4 MB-groot werkgeheugen. Deze cache heeft dan een omvang van ruim 300 kB. Er zijn echter ook machines met een 8 MB of zelfs 12 MB-groot werkgeheugen. In dat eerste geval is een cache van 703 kB vastgesteld, zie [ZEG87].

De omvang van de cache bij de werkstations is bepaald door tweemaal achter elkaar een bepaalde hoeveelheid data vanaf de fileserver in te lezen en de tijden die met deze beide leesacties gepaard gaan te meten. Vervolgens is dit in stappen van 1 kB herhaald met grotere hoeveelheden data.

Indien het werkstation bij de eerste leesactie de hoeveelheid data in zijn geheel in de cache kan plaatsen, wordt bij de tweede leesactie de data uit de eigen cache en niet via het netwerk vanaf de fileserver gelezen. In dit geval is voor de tweede leesactie slechts een fractie van de voor de eerste leesactie benodigde tijd vereist.

Indien echter de hoeveelheid data niet in de cache past, zal het laatst ingelezen datablok een eerder ingelezen blok overschrijven. Het resultaat is dat bij de tweede leesactie alle blokken opnieuw vanaf de fileserver betrokken worden. In dit geval is de werking van de cache bij het werkstation uitgeschakeld en blijken de responsetijden bij beide leesacties van dezelfde orde grootte te zijn.

De overgang tussen deze beide situaties is scherp en blijkt zich volledig te voltrekken

indien de ingelezen hoeveelheid data één kB kleiner of groter is dan het cachegeheugen.

In de praktijk zullen de achtereenvolgens door de caseworkers ingelezen casefolders steeds van de fileservers betrokken worden (indien ze zich tenminste daarop bevinden). Dit kan bij de metingen dus zonder meer gerealiseerd worden indien de in te lezen hoeveelheid data groter is dan de cache van het werkstation.

Het is echter ook gewenst dat de processen kleinere hoeveelheden data in kunnen lezen. Hierbij dient eveneens het lezen uit de cache vermeden te worden. Dit is gerealiseerd door op de fileserver een 1 MB-groot bestand te creëren en bij opeenvolgende leesacties steeds uit een ander deel van dit bestand te lezen. Hiertoe is het bestand in een aantal logische blokken ter grootte Blocksize opgedeeld. Blocksize is als constante gedefinieerd en kan door de gebruiker naar wens aangepast worden.

Indien bij een leesactie de (gemiddelde) hoeveelheid in te lezen data, stel Size, kleiner is dan Blocksize, wordt bij iedere leesactie een ander logisch blok van het bestand benaderd. Hierbij mag Size echter niet minder dan ongeveer 35 procent van Blocksize bedragen daar anders wederom het risico bestaat dat uit de lokale cache van het werkstation gelezen wordt.

Stel namelijk n het aantal logische blokken waarin het 1 MB-grote bestand is opgedeeld, oftewel $n * \text{Blocksize} = 1 \text{ MB}$. Dan zal na n leesacties door een proces een hoeveelheid data ter grootte van ongeveer $n * \text{Size}$ zijn ingelezen. Een volgende leesactie zal daarna beginnen in een reeds eerder gebruikt logisch blok. Derhalve dient $n * \text{Size}$ groter te zijn dan de omvang van de cache teneinde te voorkomen dat de eerstvolgende gevraagde blokken zich reeds daarin bevinden.

Het herhaald lezen van blokken uit de lokale cache van het werkstation doet zich minder snel voor indien de totale omvang van de bestanden, nu 1 MB, groter genomen wordt. Teneinde niet al te veel ruimte op de disk te bezetten met dummy-bestanden is dit echter niet gedaan. Derhalve kunnen ook de werkstations met een cache, die aanzienlijk groter is dan de standaard 300 kB, niet voor de metingen gebruikt worden.

Indien Size groter is dan Blocksize is het risico van het uit de lokale cache van een werkstation lezen van blokken niet aanwezig. Het is echter op grond van het volgende niet aan te raden voor Blocksize een waarde te gebruiken die veel kleiner is dan Size.

Indien Size even groot of iets kleiner is dan Blocksize bestaat de kans dat de fileserver na afloop van een bepaalde leesactie de eerste blokken aan het begin van het volgende logische blok inleest. Dit is een zogenaamde "read-ahead"-actie die om performance redenen door de fileserver wordt uitgevoerd en gebaseerd is op de veronderstelling dat indien een proces verzoekt om een blok uit een bepaald bestand er waarschijnlijk verzoeken om één of meerdere daaropvolgende blokken zullen volgen. Hierdoor zou het mogelijk zijn dat bij het begin van de volgende leesactie van hetzelfde proces reeds enige blokken in de cache van de fileserver aanwezig zijn. De verschillende leesacties dienen echter onafhankelijk van elkaar plaats te vinden.

Om dit te voorkomen worden de logische blokken van een bestand in achterwaartse richting benaderd. Hierdoor wordt de read-ahead als zodanig niet voorkomen maar wordt wel het effect daarvan teniet gedaan. De volgende leesactie zal namelijk niet be-

ginnen bij de eerste diskblokken van het volgende logische blok, die zich dan ten gevolge van een read-ahead mogelijk reeds in de cache van de fileserver bevinden, maar bij de eerste diskblokken van het vorige logische blok van het bestand.

Indien nu echter Blocksize veel kleiner genomen wordt dan Size, zal binnen één door een proces om een bepaalde hoeveelheid data uitgevoerde leesactie meerdere malen in achterwaartse richting bewogen worden naar een nieuw logisch blok in het bestand. Het effect van de read-ahead wordt hierbij steeds teniet gedaan. In de praktijk zal een bestand echter normaliter in voorwaartse richting gelezen worden waarbij de read-ahead een performance verbetering oplevert. Dit effect dient bij de metingen niet onnodig teniet gedaan te worden.

Bij de leesacties van 512 kB is steeds voor Blocksize een waarde van 256 kB gebruikt. Bij leesacties van bijvoorbeeld 40 kB een waarde van 64 kB .

5.2.2. De Excelan Ethernet Network Analyzer.

De meetresultaten die verkregen worden met het meetprogramma Simulation hebben voor wat betreft de responsetijden betrekking op de totale tijd die benodigd is voor het inlezen van een casefolder of image door een werkstation vanaf een fileserver via het netwerk waarbij zowel de fileserver als het netwerk belast kunnen zijn met gegevenstransporten van of naar overige werkstations.

Deze meetresultaten liggen dus op een macro-niveau en kunnen gebruikt worden voor de validering van het systeemmodel en het simulatieprogramma door ze te vergelijken met de resultaten die dit programma produceert indien bij de metingen en bij de simulaties vergelijkbare belastingen van het systeem worden aangebracht.

Het programma moet echter voor het runnen van de simulatie voorzien worden van getalwaarden voor een groot aantal parameters. De simulatieresultaten zijn direct afhankelijk van de grootte van deze parameters.

De types van de parameters, die in het programma aanwezig zijn, volgen uit de structuur van het model, uit de gekozen modellering. Daar de modellering van ECHO-systemen hier plaatsvindt op het lage ethernetpakket-niveau dient een aanzienlijk aantal tijdsparameters bepaald te worden die betrekkelijk klein zijn ten opzichte van de totale responsetijd bij de overdracht van een casefolder of image.

Voor een aantal van deze parameters is het mogelijk geweest de waarden ervan vast te stellen met behulp van een Excelan Ethernet Network Analyzer.

Dit apparaat stelt de gebruiker in staat om het pakketverkeer op het ethernet te onderzoeken. Het kan onder meer gebruikt worden voor het meten van performanceaspecten van het netwerk.

De analyzer leest de over het ethernet getransporteerde pakketten in. De pakketten, die voldoen aan (eventueel) door de gebruiker gespecificeerde selectiecriteria, worden vervolgens voorzien van een timestamp en opgeslagen in een buffergeheugen.

Tijdens de metingen worden voortdurend statistische resultaten voor het verkeer op het ethernet berekend, op het beeldscherm getoond en eventueel op de disk opgeslagen.

Verder kan het apparaat ook gebruikt worden om pakketverkeer in het ethernet te injecteren.

De network analyzer bestaat in wezen uit een speciale ethernet-adapter-kaart met een Intel 82586 Ethernet LAN-controller en een op de kaart aanwezige 80186-microprocessor die belast is met het selecteren en verwerken van de ingelezen pakketten en het berekenen van statistische resultaten omtrent het pakketverkeer op het net. Verder is er op de kaart een 700 kB-groot ringbuffergeheugen aanwezig voor de (tijdelijke) opslag van geselecteerde pakketten.

De kaart kan in een willekeurige IBM-compatible PC, bij voorkeur een AT, geplaatst worden. In dit geval betreft dit een Compaq Portable II. Het apparaat wordt direkt via een transceiver op de ethernet-kabel aangesloten.

De processor van de PC wordt belast met de besturing van een test, het naar het scherm uitvoeren van de resultaten alsmede het eventueel naar de disk schrijven van (gedeeltes van) geselecteerde pakketten en/of statistieken omtrent het netwerkverkeer.

Het gebruik van de analyzer gaat in de vorm van het opzetten en vervolgens uitvoeren van een test waarbij alle over het ethernet getransporteerde pakketten door de analyzer worden ingelezen.

Een test is een programma waarbij de gebruiker (alleen) de criteria dient in te voeren voor het selecteren van pakketten voor opslag in het buffergeheugen en/of voor door de analyzer te verzenden pakketten.

Tot de aspecten waarop het filteren van pakketten kan plaatsvinden behoren :

- de pakketlengte,
- het al dan niet opgetreden zijn van transmissiefouten, zoals CRC-fouten,
- de inhoud van bepaalde velden van het pakket zoals het bron- en/of bestemmingsadres en de waarde van maximaal 14 op willekeurige plaatsen in het dataveld van het pakket aanwezige bytes. Hiervoor komen met name de protocolheaders in aanmerking.

Er staan 8 onafhankelijke kanalen voor de specificatie van selectiecriteria ter beschikking. Ieder kanaal filtert pakketten met bepaalde eigenschappen uit het verkeer op het ethernet.

De selectiecriteria kunnen zelf als een specifieke test worden opgeslagen zodat de test op een later tijdstip eenvoudig herhaald kan worden.

De geselecteerde pakketten worden voorzien van een timestamp met een resolutie van 1 μ s en daarna opgeslagen in het ringbuffergeheugen van 700 kB. Daarin kunnen ongeveer 450 pakketten geplaatst worden met de binnen ethernet geldende maximale lengte voor het dataveld van 1500 bytes.

Deze in het ringbuffer aanwezige pakketten kunnen vervolgens geheel of gedeeltelijk (bijvoorbeeld alleen de protocolheaders) opgeslagen worden in een tracefile op de disk. De snelheid, waarmee dit naar disk loggen geschiedt, is echter beperkt. In de specificaties wordt hiervoor een waarde van 50 tot 100 pakketten per seconde opgegeven. Dit betekent dat, indien tijdens een test veel grote pakketten snel na elkaar ingelezen en in het ringbuffer opgeslagen worden, het risico bestaat dat eerder geselecteerde pakketten die nog niet naar het tracefile zijn weggeschreven, overschreven worden en verloren gaan. Het optreden van dergelijke verliezen wordt wel in het tracefile aangegeven.

De pakketten in het ringbuffer of in een tracefile kunnen na afloop van een test door de

gebruiker geanalyseerd worden.

Aan de op het ethernet aangesloten systemen kan een symbolische naam worden toegerekend die bijvoorbeeld bij het tonen van de inhoud van een tracefile de gemakkelijk te verwarren ethernetadressen vervangt.

Verder is er nog een aparte utility aanwezig waarmee de protocolheaders van de in een tracefile opgeslagen pakketten geanalyseerd kunnen worden voor TCP/IP- en ISO-protocollen. Deze headers worden dan op een meer inzichtelijke wijze gepresenteerd dan in de anders gebruikte hexadecimale code.

Tijdens het uitvoeren van een test worden statistische resultaten omtrent de verkeersactiviteit op het ethernet berekend en in de vorm van tabellen en grafieken op het scherm getoond. De schermuitvoer wordt iedere seconde aangepast met de resultaten over de afgelopen seconde.

Tot deze resultaten behoren onder meer :

- de netwerkbezettingsgraad gemiddeld over de afgelopen 1, 5, 10 en 20 seconden,
- de verdeling van de pakketgrootte,
- het pakkettransport in pakketten per seconde en in kByte per seconde,
- het totaal aantal ingelezen pakketten alsmede het aantal geselecteerde pakketten per selectiekanaal.
- het aantal opgetreden transmissiefouten.

Deze statistieken kunnen eveneens in een apart bestand worden opgeslagen.

Behalve voor het onderzoeken van het verkeer op het ethernet kan de analyzer (eventueel gelijktijdig) gebruikt worden om verkeer in het net te injecteren.

Dit kan een enkel pakket zijn dat bijvoorbeeld een verzoek aan een bepaald systeem bevat om vast te stellen wat de responsetijd van het systeem daarop is. De inhoud van het pakket kan de gebruiker naar believen vaststellen.

Pakketten kunnen met een bepaalde door de gebruiker op te geven frequentie herhaald worden waardoor een bepaalde achtergrondbelasting in het netwerk aangebracht kan worden tot ruim 90 procent van de bruto ethernet-capaciteit van 10 Mbit/s. Hiervoor dienen (bij voorkeur) pakketten gebruikt te worden die door geen enkel systeem ingelezen worden, bijvoorbeeld pakketten met bestemmingsadres nul.

5.2.3. De Sun-utilities traffic en etherfind.

Op de Sun-systemen zijn eveneens enige utilities aanwezig, die gebruikt kunnen worden voor algemeen netwerk-diagnostische doeleinden. Metingen met een acceptabele nauwkeurigheid kunnen hiermee echter niet worden verricht.

De programma's traffic en etherfind leveren ieder in beperkte mate de faciliteiten van een netwerk analyzer als de Excelan. Een belangrijk nadeel is echter dat deze programma's op een component van het systeem draaien terwijl het sterk te verkiezen is over meetapparatuur te beschikken die volkomen onafhankelijk is van het meetobject.

Het programma traffic biedt op het beeldscherm van een werkstation een grafisch overzicht van het op het ethernet aanwezige verkeer. Hierbij kan het verkeer vooraf eventueel gefilterd worden op basis van de bron, bestemming, lengte en het transmissieprotocol van de pakketten.

Het uitvoerscherm bestaat uit maximaal vijf vensters die ieder een bepaald aspect van het netwerkverkeer tonen. Dit betreft de momentane bezettingsgraad van het ethernet in de vorm van een grafiek alsmede de verdeling van het verkeer naar de bron, bestemming, lengte en het transmissieprotocol van de over het net getransporteerde pakketten in de vorm van histogrammen. Zowel de bezettingsgraad als de verschillende verdelingen worden gemiddeld over tijdsintervallen met door de gebruiker gespecificeerde tijdsduren.

Traffic levert alleen grafische en geen alfanumerieke resultaten. De enigste mogelijkheid om de resultaten op te slaan is in de vorm van een screendump. De meetfout in de resultaten is onbekend en het aflezen van de histogrammen gaat tamelijk onnauwkeurig.

Traffic is alleen geschikt om snel een globale indruk te verkrijgen over de omvang en de samenstelling van het verkeer op het ethernet.

Het programma etherfind levert op het beeldscherm of in een bestand een trace van de over het net getransporteerde pakketten.

Deze trace bevat voor ieder pakket een regel met de lengte, het type, de bron en de bestemming en de logische poorten van het pakket.

Ook bij etherfind kan de gebruiker een filter specificeren in de vorm van een boolean expressie met als termen voorwaarden voor wederom de bron, bestemming, lengte, en het protocol alsmede de waarde van bepaalde bytes van het pakket.

Hoewel etherfind intern ieder van het net ingelezen pakket voorziet van een timestamp, voert de oorspronkelijke versie van etherfind deze timestamp niet uit naar de trace. Deze versie is daarom aangepast zodat dit nu wel gebeurt. De timestamp wordt echter afgeleid van de systeemklok die een resolutie heeft van slechts 20 ms. Dit is te grof voor de hier benodigde meetresultaten.

Etherfind is derhalve ongeschikt voor het bepalen van de tijdsparementen van het systeemmodel maar kan wel gebruikt worden om bijvoorbeeld te onderzoeken op welke wijze de interactie tussen een werkstation en een fileservers plaatsvindt tijdens het inlezen van een bestand.

5.3. Traces van het pakketverkeer op het ethernet.

Zowel met de network analyzer als met het programma etherfind zijn traces verzameld van het pakketverkeer over het ethernet. Dit verkeer is hierbij gegenereerd met behulp van het programma Simulation.

De traces van de network analyzer en die van etherfind zijn vergelijkbaar van aard. In beide gevallen geven de traces een overzicht van pakketten, die gedurende een bepaalde periode over het ethernet getransporteerd zijn, waarbij voor een pakket in ieder geval de bron en bestemming en de lengte ervan worden weergegeven, alsmede het aan de hand van een interne klok bepaalde tijdstip, waarop het pakket op het net aanwezig is

geweest. De tijdsintervallen tussen twee opeenvolgende (groepen van) pakketten kunnen vaak als responsetijden bij de fileserver of verblijfstijden bij het werkstation worden opgevat. De timestamps zijn echter alleen bij de network analyzer van een voldoende hoge resolutie om als basis voor de parametrisering van het simulatieprogramma te dienen.

Reeds in een vroeg stadium is het mogelijk geweest met (een aangepaste versie van) etherfind traces te verzamelen. De informatie uit deze traces is gebruikt bij de bepaling van de structuur van het simulatieprogramma. Voor een korrekte parametrisering van dit programma is minimaal de beschikking over de network analyzer vereist.

De network analyzer biedt echter slechts beperkte faciliteiten voor een statistische analyse van de tracebestanden. Deze is nodig vanwege het stochastische karakter van de tijdsintervallen tussen de opeenvolgende (groepen van) pakketten in de trace. Daarnaast blijkt ook de wijze waarop een leesactie verloopt afhankelijk te zijn van de responsetijden bij de fileserver voor de afzonderlijke NFS-verzoeken en daardoor eveneens stochastisch van aard te zijn.

De uitvoering van de gewenste statistische analyses van de traces is derhalve slechts mogelijk door zelf programmatuur op te stellen die, eventueel na een bestandsconversie, de tracebestanden van de network analyzer direct benadert.

Vanwege het feit dat de analyzer pas in een laat stadium ter beschikking is gekomen, is het opstellen van deze programmatuur niet meer mogelijk geweest. Het "handmatig" uitvoeren van een dergelijke statistische analyse is een zeer tijdsintensieve aangelegenheid gebleken. Als gevolg hiervan is het niet mogelijk geweest om alle relevante parameters in het simulatieprogramma met een redelijke mate van nauwkeurigheid te bepalen. In meerdere gevallen zijn gemiddelden en distributies geschat op basis van een beperkt aantal waarnemingen.

Hierbij geldt dat de tijdsaspecten van verschijnselen, die herhaaldelijk optreden binnen één leesactie, zoals de reeks NFS-verzoeken, nauwkeuriger kunnen worden vastgesteld dan de tijdsaspecten van eenmalig tijdens een leesactie optredende verschijnselen waarbij voor meerdere waarnemingen veelal evenzovele traces vereist zijn.

De traces, die zijn verzameld, kunnen onderscheiden worden in traces waarbij slechts één werkstation bestanden inleest en traces waarbij meerdere werkstations tegelijkertijd leesacties uitvoeren.

Bij de eerste groep bevat een trace het verloop van één of enkele leesacties van het werkstation. In dit verloop is een vrij sterke regelmaat in het optreden van bepaalde (groepen van) pakketten te ontwaren die de analyse van deze traces vereenvoudigt. Verder is het bij deze groep mogelijk de traces te vergelijken met de resultaten van het programma Simulation, die betrekking hebben op de totale tijdsduur van de leesacties, omdat het begin en einde van een leesactie in de trace gelokaliseerd kunnen worden.

Bij de tweede groep traces bevat een trace meestal slechts een deel van ieder van de simultaan optredende leesacties. Deze traces vertonen in veel mindere mate een duidelijke structuur vanwege het tussen elkaar door optreden van pakketten afkomstig van of bestemd voor verschillende werkstations waardoor de analyses van deze traces

aanzienlijk gecompliceerder en minder vruchtbaar zijn. Er kunnen uit deze traces slechts weinig resultaten voor specifieke leesacties worden afgeleid.

De analyses, die verricht zijn, hebben om deze redenen hoofdzakelijk betrekking op de eerste groep traces. Er mag echter niet zonder meer aangenomen worden dat simultaan optredende leesacties opgevat kunnen worden als een superpositie van meerdere afzonderlijk plaatsvindende leesacties. De gemiddelde servicetijden bij de fileserver voor de afzonderlijke NFS-verzoeken blijken namelijk bij de tweede groep traces groter te zijn dan bij de eerste groep.

Dit kan verklaard worden door het feit dat NFS-blokken uit verschillende bestanden zeker niet fysiek na elkaar op de disk liggen, dit in tegenstelling tot de blokken uit één bestand waarbij dit wel vaak het geval zal zijn. Bij simultaan optredende leesacties van meerdere werkstations hebben de na elkaar door de fileserver ontvangen NFS-verzoeken veelal betrekking op blokken uit verschillende bestanden waardoor meer tijdsintensieve bewegingen van de kop van de disk vereist zijn dan in het geval na elkaar blokken uit hetzelfde bestand ingelezen moeten worden.

5.3.1. Traces van afzonderlijke leesacties.

In het hiernavolgende zal het verloop van een leesactie door een workstation besproken worden zoals deze door de eerste groep traces zijn vastgelegd. Hierbij wordt eerst de meest regelmatige vorm, waarin een leesactie kan optreden, behandeld terwijl daarna mogelijke afwijkingen daarvan beschreven worden.

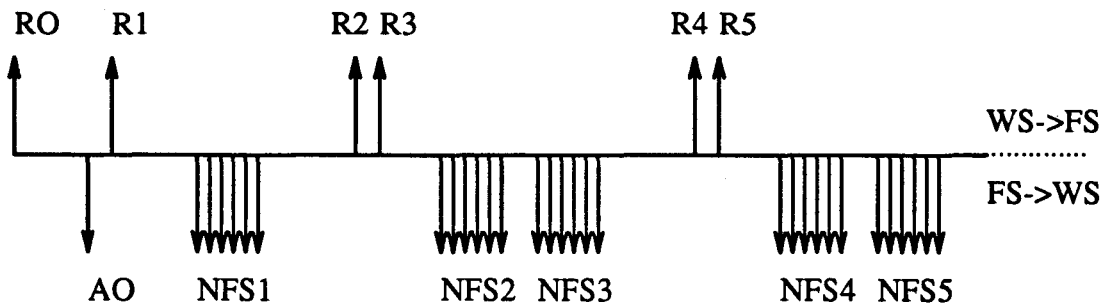


Fig. 5.1 Regelmatig verloop van het inlezen van een bestand vanaf de NFS-fileserver.

Een leesactie wordt geopend met het versturen van een enkel klein pakket door het workstation dat de fileserver eveneens met een enkel klein pakket beantwoordt, zie fig. 5.1.

Vervolgens begint de eigenlijke bestandsoverdracht met het verzenden van het eerste NFS-verzoek door het workstation waarop de fileserver antwoordt met het eerste NFS-blok in de vorm van 6 vrijwel direct na elkaar verstuurd pakketten. Daarvan hebben er 5 de maximaal voor ethernet toegestane lengte van 1518 bytes. Het zesde pakket is 938 bytes lang. Samen vormen ze een 8 kB NFS-blok plus de overhead van de UDP/IP- en Ethernet-protocollen. Alle NFS-blokken worden op deze wijze door de fileserver verstuurd.

(De vermelde pakketlengtes hebben betrekking op de door de network analyzer weer-

gegeven waarden die de lengte van het dataveld plus die van de headers en trailer van een ethernet-pakket omvatten.)

Na de volledige ontvangst van het eerste NFS-blok verstuurt het werkstation na elkaar twee verzoeken voor de twee eerstvolgende NFS-blokken uit het bestand. Het tweede verzoek van dit paar kan worden opgevat als een door het werkstation over het netwerk verrichte prefetch of read-ahead van een volgend NFS-blok.

Dit mechanisme is in NFS om performanceredenen ingebouwd en vermindert het deel van de tijd dat de fileserver tijdens een leesactie idle is in afwachting van een volgend NFS-verzoek. Er wordt hierdoor door de fileserver een groter aantal blokken per tijdseenheid aangeleverd waardoor de tijdsduur van met name een afzonderlijk optredende leesactie verkleind wordt.

Deze beide verzoeken worden door de fileserver beantwoord met de beide gevraagde NFS-blokken. Hierop volgt wederom een paar van NFS-verzoeken van het werkstation waarna de cyclus wordt herhaald totdat alle benodigde blokken uit het bestand door het werkstation zijn ontvangen.

Op dat moment is de eigenlijke leesactie teneinde en komt de read-systeemaanroep in het programma Simulation terug. Dit blijkt uit de vergelijking van de uit de traces afgeleide tijdsduren van een reeks leesacties met de resultaten van Simulation voor dezelfde reeks van leesacties.

Hierna volgen tussen het werkstation en de fileserver enige uitwisselingen van pakketten die onder andere betrekking hebben op het door Simulation wegschrijven van een logregel naar een bestand op de fileserver echo. Deze interactie is relatief klein van omvang ten opzichte van het er aan voorafgaande inlezen van een bestand, verloopt weinig gestructureerd en valt buiten de eigenlijke leesactie. Deze pakketuitwisselingen zijn niet in het simulatieprogramma ondergebracht en worden hier niet verder besproken.

Wel wordt het feit besproken dat een werkstation vervolgens steeds een bepaalde hoeveelheid data van de ND-fileserver betreft. De omvang hiervan hangt samen met de omvang van het zojuist ingelezen bestand.

Bij leesacties van 512 kB-grote bestanden betreft dit 57 of 65 pakketten van 1072 bytes. Bij 192 kB-bestanden betreft het nog steeds 57 pakketten. Bij 40 kB-bestanden meestal 9 pakketten. Bij 120 kB-bestanden worden in ruim de helft van de gevallen 9 pakketten en in de overige gevallen 57 pakketten ingelezen.

In het algemeen wordt een enkel pakket plus een veelvoud van 8 pakketten overgestuurd. Ieder pakket heeft een dataveld dat 1 kB aan data plus de ND-protocoloverhead bevat. Bij het transport van deze pakketten mogen echter niet meer dan 6 pakketten tegelijkertijd uitstaan voordat het werkstation de ontvangst ervan bevestigd.

Het werkstation verstuurt tijdens dit gegevenstransport steeds 66-bytes pakketten als verzoek- of acknowledgebericht naar de ND-fileserver.

Ook bij dit gegevenstransport komt het voor dat twee verzoeken tegelijk in het systeem uitstaan. De structuur van het verloop van dit transport is echter veel minder regelmatig dan bij het inlezen van het bestand het geval is.

Het is niet mogelijk gebleken de aard van deze informatie nader vast te stellen maar het

betreft waarschijnlijk informatie die vanwege de leesactie bij het werkstation verloren is gegaan, bijvoorbeeld pagina's in het werkgeheugen die overschreven zijn.

De informatie van de ND-fileserver wordt door het besturingssysteem ingelezen. Deze interactie met de ND-fileserver heeft geen invloed op de tijdsduur van de zojuist uitgevoerde leesactie.

Soms komt het voor dat tijdens een leesactie reeds een gegevenstransport tussen het werkstation en de ND-fileserver optreedt. In dat geval werkt de tijd, die met dit gegevenstransport gepaard gaat, wel door in de tijdsduur van de leesactie.

Tijdens het inlezen van een bestand treden in sommige gevallen afwijkingen op van het zojuist beschreven regelmatige gedrag.

Zo wordt bijvoorbeeld voorafgaand aan de eigenlijke bestandsoverdracht het uitwisselen van een enkel klein pakket tussen het werkstation en de fileserver in sommige gevallen enkele malen herhaald. Blijkbaar wordt alleen het laatste pakket van de fileserver door het werkstation opgevat als een positief antwoord dat met de eigenlijke leesactie aangevangen kan worden.

Tijdens de bestandsoverdracht ligt tussen twee direkt na elkaar door de fileserver verstuurd NFS-blokken in driekwart van de gevallen een klein tijdsinterval van 2 tot 6 ms. In dat geval verloopt de leesactie op de zojuist beschreven regelmatige wijze. In ongeveer 25 % van de gevallen ligt tussen deze beide blokken een aanzienlijk groter tijdsinterval van meer dan 10 ms. In dat geval verstuurt het werkstation een enkel NFS-verzoek tussen door. Na de ontvangst van het tweede NFS-blok van het paar en van dit afzonderlijke NFS-blok hervat het werkstation het paarsgewijs aanvragen van NFS-blokken, zie fig. 5.2.

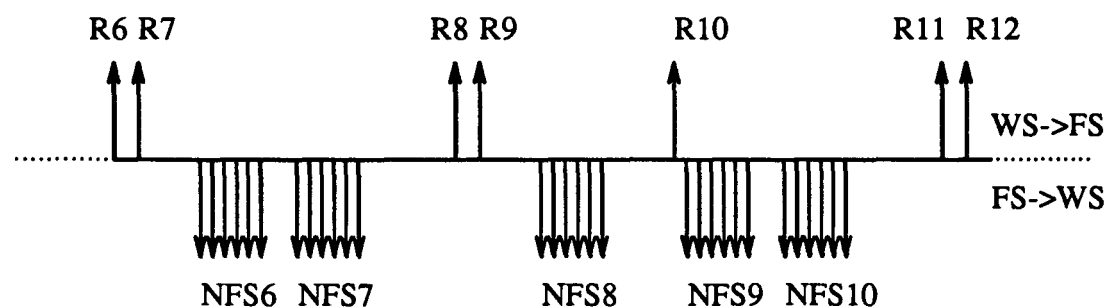


Fig. 5.2 Afzonderlijk NFS-verzoek R10 vanwege traag antwoordblok NFS9.

De tijd van 10 ms wordt door het werkstation gebruikt om de communicatie-software af te handelen voor het zojuist ontvangen NFS-blok alsmede voor het opstellen van een verzoek om het eerstvolgende nog niet ingelezen NFS-blok.

5.3.2. Traces van simultaan plaatsvindende leesacties.

Bij traces uit de tweede groep komt het voor dat dit synchrone, paarsgewijze verrichten van NFS-verzoeken (tijdelijk) verdwijnt indien ook het antwoord op dit afzonderlijke

NFS-verzoek langer dan 10 ms uitblijft. In dat geval wordt ook een afzonderlijk tweede NFS-verzoek verricht. Er is dan geen sprake meer van het paarsgewijs aanvragen van NFS-blokken maar meer van twee processen die afwisselend NFS-verzoeken verrichten en NFS-blokken ontvangen, maar waartussen verder geen relatie aanwezig is. Na verloop van tijd wordt evenwel veelal weer het paarsgewijs verrichten van NFS-verzoeken hervat, zie fig. 5.3.

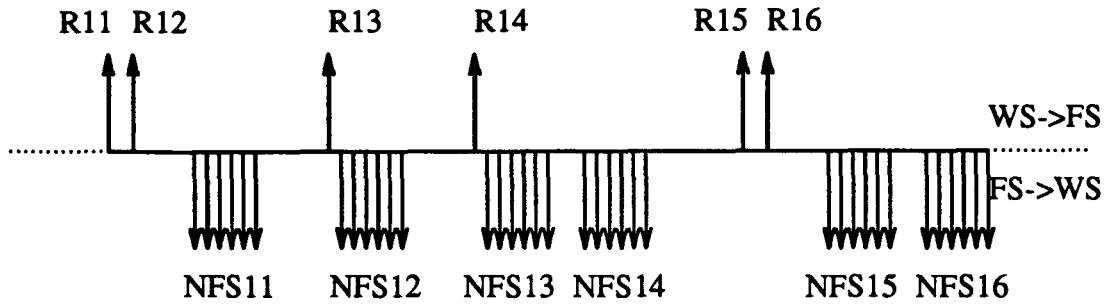


Fig. 5.3 Meerdere afzonderlijke NFS-verzoeken R13 en R14.

De beide NFS-verzoeken van een paar alsmede de beide daaropvolgende NFS-blokken worden via verschillende logische poorten van het UDP-protocol overgedragen en kunnen op basis daarvan onderscheiden worden.

Het is niet mogelijk geweest om na te gaan in welke mate dit verschijnsel optreedt. Dit verschijnsel is slechts bij hoge uitzondering bij de eerste groep traces opgetreden.

De beide fileservers blijken afwisselend de toegang tot het ethernet te verkrijgen om een enkel pakket te versturen wanneer ze beide tegelijkertijd proberen een blok data te verzenden dat uit meerdere ethernet-pakketten bestaat. Dit kan verklaard worden door het feit dat een enkele ethernet-controller meerdere pakketten niet onmiddellijk na elkaar kan versturen, terwijl een controller die een pakket klaar heeft voor verzending en wacht op het vrijkomen van het ethernet wel in staat is om indien dit gebeurt het pakket direct te verzenden. Afgezien van eventueel bij de beide eerste pakketten van de beide blokken, zullen zich in dit geval op het ethernet geen botsingen voordoen.

Bij de tweede groep traces zijn vaak meerdere, van verschillende werkstations afkomstige NFS-verzoeken in de wachtrij bij de fileserver aanwezig. Een specifiek NFS-verzoek wordt in dat geval pas enige honderden milliseconden na ontvangst door de fileserver beantwoord. Gedurende deze tijd worden eerst de reeds aanwezige, eerder ontvangen NFS-verzoeken afgehandeld. Het bepalen van de servicetijd van een specifiek NFS-verzoek is derhalve niet mogelijk. Op basis van deze traces kan alleen de gemiddelde servicetijd berekend worden uit het aantal NFS-verzoeken, dat gedurende een bepaalde periode wordt afgehandeld, mits er tijdens de betreffende periode ook alleen maar NFS-verzoeken worden afgehandeld.

5.3.3. Meetresultaten van de traces.

In de traces kunnen de tijdsintervallen tussen opeenvolgende (groepen van) pakketten vaak opgevat worden als servicetijden bij de fileserver of verblijfstijden bij het werkstation. De waarden, die hiervoor zijn vastgesteld, worden gebruikt bij de parametrisering van het simulatieprogramma.

De in het hiernavolgende vermelde tijden zijn afgeleid uit traces waarbij slechts een enkel werkstation leesacties uitvoert, tenzij anders is vermeld.

Bij een leesactie blijkt het tijdsinterval tussen het allereerste door het werkstation verzonden pakket en het eerste NFS-verzoek sterkt te variëren van ongeveer 10 tot 60 ms.

De tijd tussen het eerste NFS-verzoek en het eerste pakket van het daaropvolgende NFS-pakket varieert van ongeveer 20 to 50 ms. Deze tijd wordt hier opgevat als de servicetijd van het NFS-verzoek bij de fileserver.

Het is niet duidelijk of (de processor van) de fileserver ook tijdens het verzenden van het NFS-verzoek door de ethernet-controller, dat in totaal 8 ms duurt, nog belast is met de verwerking van communicatie-software voor de latere pakketten van het NFS-blok. Tussen de transmissietijden van de 6 pakketten van een NFS-blok ligt een tijdsinterval van gemiddeld 0.2 ms. Indien ervanuitgegaan wordt dat de verwerking door (de processor van) de fileserver eindigt bij de verzending van het eerste pakket van het NFS-blok, moet dit interval worden toegeschreven aan acties binnen de ethernet-controller gericht op het prepareren van het volgende pakket voor verzending. Daar voor de transmissie van een maximum-lengte-pakket over het ethernet 1.25 ms vereist is, betekent dit dat de ethernet-controller in theorie in staat is een maximale bezettingsgraad van het ethernet te creëren van ruim 80 %. Hiervoor is het natuurlijk vereist dat er voortdurend pakketten in de wachtrij van de controller voor verzending aanwezig zijn. In de praktijk is dit bij de fileserver niet het geval daar voor disk-I/O en verwerking van de communicatie-software samen aanzienlijk meer tijd is vereist dan voor het verzenden van het blok door de ethernet-controller. In de traces wordt dit weergegeven door de tijdsintervallen tussen de na elkaar door de fileserver verzonden NFS-blokken. De ethernet-controller is derhalve een groot deel van de tijd idle.

Indien het tijdsinterval van 0.2 ms echter te wijten zou zijn aan het feit dat de processor van de fileserver het volgende pakket nog niet aan de controller voor verzending heeft toegekend, waardoor de controller op de processor moeten wachten, is de waarde van 80 % een conservatieve schatting voor de maximale bezettingsgraad van het ethernet die met de controller in theorie gecreëerd kan worden.

Bij het paarsgewijs verzenden van NFS-verzoeken ligt de servicetijd voor het eerste verzoek van het paar tussen 15 en 35 ms. De per leesactie (van 512 kB) gemiddelde tijd hiervoor varieert van 22 tot 25 ms.

Tussen de beide NFS-verzoeken van een paar ligt (in driekwart van de gevallen) een tijdsinterval van 2 tot 6 ms.

Wederom ervanuitgaand dat de serviceverlening door (de processor van) de fileserver

eindigt bij het verzenden van het eerste pakket van een NFS-blok, betekent dit dat de servicetijd van het tweede NFS-blok 10 tot 14 ms bedraagt daar de transmissie van het eerste NFS-blok 8 ms duurt.

Deze servicetijd is lager dan die van het eerste NFS-verzoek van het paar. Dit zou verklaard kunnen worden door het door de fileserver verrichten van een prefetch van volgende diskblokken na de ontvangst van het eerste NFS-verzoek van het paar zodat op het moment van de ontvangst van het tweede NFS-verzoek van dit paar het inlezen van de daarmee aangevraagde blokken in een meer of minder vergevorderd stadium verkeert.

In 25 % van de gevallen ligt tussen de beide NFS-verzoeken van het paar echter een tijdsinterval van 10 tot 30 ms.

Dit kan veroorzaakt worden door het niet na elkaar op de disk liggen van de na elkaar gevraagde blokken, waardoor tijdsintensieve bewegingen van de kop van de disk optreden. De servicetijden van het tweede NFS-verzoek zijn dan vergelijkbaar met die van het eerste NFS-verzoek.

In deze gevallen verricht het werkstation vervolgens een afzonderlijk NFS-verzoek. Het NFS-blok, dat met een afzonderlijk verstuurd NFS-verzoek is aangevraagd, volgt eveneens in ongeveer driekwart van de gevallen snel op het tweede NFS-blok van het paar. De servicetijd ligt dan eveneens in de orde van 10 tot 14 ms. In een kwart van de gevallen zijn de servicetijden vergelijkbaar met die van het eerste NFS-verzoek van het paar en liggen tussen 15 en 30 ms.

De ontvangst van de NFS-blokken door het werkstation veroorzaakt de executie van communicatie-software voor de afhandeling van de ontvangst van de NFS-blokken en voor het opstellen van volgende NFS-verzoeken.

Het eerste verzoek van het volgende paar vindt bij een Sun-3/50, -3/110 en 3/260 gemiddeld respectievelijk 23, 14 en 7 seconde na de volledige ontvangst van het tweede NFS-blok van het vorige paar plaats.

Hieruit blijkt duidelijk de invloed van het verschil in snelheid van de drie typen werkstations. De snelheden van deze typen werkstations verhouden zich echter niet als de drie vermeld tijden. Dit zijn ook niet de totale servicetijden die bij de werkstations optreden voor het afhandelen van de communicatie-software. Deze servicetijden beginnen reeds bij de ontvangst van het eerste pakket van het eerste NFS-blok van het paar.

In deze gevallen verricht het werkstation vervolgens een afzonderlijk NFS-verzoek. Het NFS-blok, dat met een afzonderlijk verstuurd NFS-verzoek is aangevraagd, volgt eveneens in ongeveer driekwart van de gevallen snel op het tweede NFS-blok van het paar. De servicetijd ligt dan eveneens in de orde van 10 tot 14 ms. In een kwart van de gevallen zijn de servicetijden vergelijkbaar met die van het eerste NFS-verzoek van het paar en liggen tussen 15 en 30 ms.

De interactie met de ND-fileserver treedt 100 tot 300 ms na afloop van de eigenlijke leesactie op. De voor de overdracht van 57 of 65 pakketten benodigde tijd bedraagt in totaal 200 tot 300 ms. Ook hierbij is voor de beantwoording van het eerste ND-verzoek van het werkstation relatief veel tijd vereist, namelijk 30 to 50 ms. In de gevallen dat een verzoek en 6 antwoord-pakketten als zodanig in de trace geïdentificeerd kunnen

worden bedraagt de servicetijd 10 tot 25 ms.

Bij het optreden van een CRC-fout bij de overdracht van een NFS-blok wordt het verzoek om het betreffende blok na 700 ms herhaald, zie fig. 5.4. Nadat het werkstation dit blok vervolgens korrekt heeft ontvangen, wordt de leesactie gecontinueerd.

De mate waarin CRC-fouten optreden varieert ook gedurende langere traces nog van 1 CRC-fout per 4 overgedragen casefolders tot 1 per 20 casefolders oftewel 1 CRC-fout per 2 tot 10 MByte aan getransporteerde informatie.

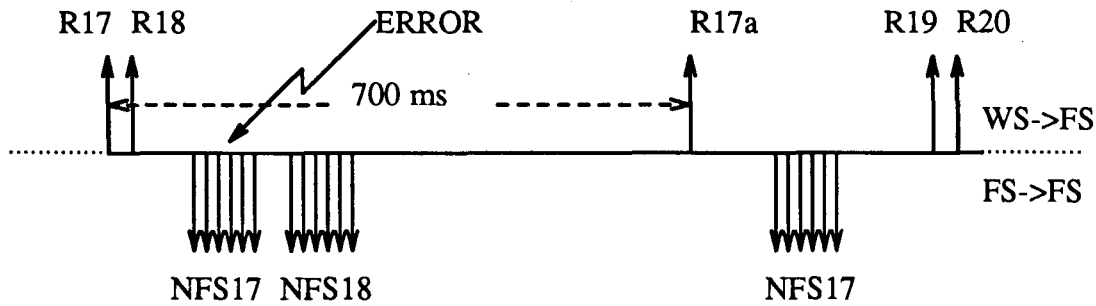


Fig. 5.4 Hertransmissie van NFS-verzoek R17 vanwege transmissiefout in blok NFS17.

Bij de traces uit de tweede groep worden gedurende een seconde dertig tot veertig van verschillende werkstations afkomstige NFS-verzoeken door één fileserver beantwoord. Minder naarmate er meer werkstations simultaan leesacties uitvoeren.

De maximale belasting van het ethernet, die aldus gemiddeld over een periode van minimaal een seconde door een enkele fileserver gerealiseerd kan worden, bedraagt 27 %. Dit komt overeen met 40 NFS-blokken van bruto 8,5 kB per seconde oftewel 340 kB/sec oftewel 2,7 Mbit/s. Deze maximale waarde wordt (alleen) gerealiseerd indien er voortdurend NFS-verzoeken in de wachtrij van de fileserver aanwezig zijn.

Bij traces waarbij tien werkstations simultaan actief zijn, worden de NFS-blokken worden in ongeveer 80 % van de gevallen in dezelfde volgorde verstuurd als de NFS-verzoeken door de fileserver zijn ontvangen. Indien slechts drie werkstations tegelijk actief zijn, geldt dit echter in vrijwel alle gevallen.

6. Het simulatieprogramma ECHOSIM.

Voor de performance analyse van ECHO-systemen is een systeemmodel in de vorm van een wachtrijnetwerk opgesteld. Op basis daarvan is het simulatieprogramma ECHOSIM geschreven. ECHOSIM simuleert het gegevenstransport in ECHO-systemen en bepaalt performancegrootheden bij een specifieke configuratie en belasting van het systeemmodel.

De structuur en de werking van ECHOSIM is voor een groot deel gebaseerd op kennis die is verkregen door de analyse van traces van het pakketverkeer op het ethernet. Deze traces zijn verzameld met de network analyzer en met het programma etherfind. ECHOSIM is geschreven in Vax-Pascal binnen de simulatieomgeving SIMENV.

6.1. Algemene beschrijving van ECHOSIM.

ECHOSIM is een programma dat door middel van discrete simulatie het gedrag van ECHO-systemen simuleert bij de overdracht van grote bestanden van de fileservers via het netwerk naar de werkstations. Het inlezen van grote bestanden, zoals casefolders, dossiers en images, door werkstations vanaf fileservers is de verkeerscomponent die het gegevensverkeer in ECHO-systemen in overwegende mate kenmerkt.

ECHOSIM concentreert zich op het gegevenstransport op een afzonderlijk ethernet-segment in het hiërarchisch gestructureerde netwerk. Op een dergelijk segment zal een bepaald aantal werkstations zijn aangesloten die voor hun informatiebehoefte voortdurend de op het netwerk aangesloten fileservers benaderen.

De overdracht van een bestand gaat binnen ECHOSIM als in een ECHO-systeem in de vorm van een reeks NFS-verzoeken van het workstation naar de fileserver die ieder met een 8 kB NFS-blok uit het bestand beantwoord worden. Ieder NFS-blok wordt in de vorm van 6 ethernet-pakketten door de fileserver naar het workstation verzonden.

ECHOSIM slaat bij iedere leesactie van een workstation de tijd op die verloopt vanaf het moment waarop het eerste door het workstation aan de fileserver verstuurd verzoekbericht het workstation verlaat tot aan het moment waarop het laatste ethernet-pakket van het laatste NFS-blok uit het bestand bij het workstation aankomt.

Deze bij een leesactie optredende responsetijd is afhankelijk van het aantal overige werkstations dat tegelijkertijd leesacties uitvoert en daardoor gemeenschappelijke systeemcomponenten belast. De afzonderlijke leesacties van ieder workstation treden op willekeurige momenten op. De responsetijden zijn derhalve stochastische grootheden die binnen ECHOSIM statistisch worden verwerkt.

Van deze tijdens een simulatie opgetreden responsetijden wordt in een rapport, dat na afloop van de simulatie wordt geproduceerd, het aantal, maximum, minimum, gemiddelde, de standaarddeviatie en een histogram opgenomen. Dit rapport bevat verder voor de statische systeemcomponenten, zoals de fileservers en het ethernet, het aantal verzoeken, waaraan door deze systeemcomponenten service is verleend, de gemiddelde

bezettingsgraad van deze componenten alsmede het maximale aantal verzoeken dat tijdens de simulatie in de wachtrij voor deze componenten op serviceverlening heeft gewacht en de gemiddelde wachttijd die de verzoeken daar hebben ondergaan.

6.2. De parametrisering van ECHOSIM.

Voor het uitvoeren van een specifieke simulatie kan c.q. moet de gebruiker voorafgaand aan de simulatie een groot aantal parameters invoeren. Deze parameters hebben betrekking op de configuratie, op de in het model geïntroduceerde belasting, op de tijdsduur van de simulatie en op een groot aantal tijdparameters op ethernet-pakketniveau die de interne werking van de simulatie bepalen.

Voor alle parameters, die niet expliciet door de gebruiker gespecificeerd worden, zijn defaultwaarden gedefinieerd. De in dit hoofdstuk vermelde getalwaarden hebben steeds betrekking op deze defaultwaarden, tenzij anders is aangegeven.

Voor de in het simulatieprogramma aanwezige stochastische grootheden, zoals bijvoorbeeld de servicetijden bij de fileservers, worden uniforme distributies aangenomen. Alleen voor de thinktimes, bijvoorbeeld tussen de opeenvolgende leesacties van een werkstation, worden negatief-exponentiële verdelingen gebruikt. Indien voor een bepaalde grootheid een interval wordt aangegeven, betreft dit de intervalgrenzen van de uniforme distributie van deze grootheid.

De parameters die betrekking hebben op de configuratie zijn :

- het aantal werkstations.
- het aantal fileservers.
- de relaties tussen afzonderlijke werkstations en fileservers.

In een ECHO-systeem zal bij het inlezen van een bestand door een werkstation in de meeste gevallen een fileserver aangesproken worden die op hetzelfde segment is aangesloten als het betreffende werkstation. Het is echter ook mogelijk dat bij een leesactie een op een ander segment aangesloten fileserver benaderd wordt. De mate, waarin dit optreedt, is afhankelijk van de distributie van de bestanden over de verschillende fileservers in het systeem.

Een fileserver, die op een ander segment is aangesloten, kan bij een simulatie voorzien worden van een afwijkende parametrisering. Alle voor de simulatie gedeclareerde werkstations worden van dezelfde parameters voorzien.

De gebruiker van ECHOSIM kan, indien er meerdere fileservers gedeclareerd worden, voor ieder werkstation één NFS-fileserver en één ND-fileserver selecteren.

Op deze eerste fileserver is de voor case-handling relevante informatie opgeslagen, die door middel van NFS en derhalve conform het UDP-protocol over het netwerk wordt getransporteerd, terwijl zich op de tweede fileserver permanente en tijdelijke systeeminformatie specifiek voor dit werkstation bevindt, respectievelijk de root van het bestandsstelsel en de swap area van het werkstation. Deze systeeminformatie wordt con-

form het ND-protocol tussen werkstation en fileserver uitgewisseld. Beide fileservers kunnen één en dezelfde (fysieke) fileserver zijn.

Deze hoeveelheid informatie, die na afloop van een leesactie nog wordt ingelezen, heeft een niet-verwaarloosbare invloed op de performance van een NFS-fileserver indien deze tevens als ND-fileserver dienst doet.

De interactie met de ND-fileserver heeft geen invloed op de tijd die voor de overeenkomstige leesactie benodigd is, maar eventueel wel op gelijktijdig plaatsvindende leesacties van andere werkstations indien de ND-fileserver van dit werkstation ook als NFS-fileserver van deze andere werkstations fungeert.

De grootheden die de in het model geïntroduceerde belasting bepalen zijn :

- de hoeveelheid data waarop afzonderlijke leesacties betrekking hebben.
- de hoeveelheid data die na een leesactie van de ND-fileserver betrokken wordt.
- de frequentie waarmee de leesacties door ieder werkstation worden uitgevoerd. Deze frequentie wordt mede bepaald door de thinktime-parameter. Dit is de verwachtingswaarde van de negatief-exponentieel verdeelde tijdsduur die tussen de beëindiging van de vorige en het begin van de volgende leesactie ligt. Het is de tijd gedurende welke een token in het wachtrijnetwerk bij de THINK-node verblijft, zie fig. 4.3.

De omvang van de belasting wordt aldus door de gebruiker ingevoerd. De wijze waarop de belasting gerealiseerd wordt, is in het programma vastgelegd.

De tijdsduur van de simulatie wordt bepaald door de runtime-parameter die de maximale simulatietijd voorstelt. Deze tijd heeft betrekking op de tijdschaal die in de simulatie gehanteerd wordt en niet op de werkelijke tijdsduur van de simulatie.

Behalve de simulatietijd kan tevens een reset-tijd ingevoerd worden. Ook deze tijd heeft betrekking op de in de simulatie gehanteerde tijdschaal. Bij het bereiken van deze reset-tijd worden de datastructuren, waarin de statistische resultaten omtrent de simulatie worden opgeslagen, gereset. Aldus kunnen de aanloopverschijnselen, die bij de simulatie optreden, buiten de statistische resultaten gehouden worden. Deze hebben dan, indien de reset-tijd tenminste voldoende groot genomen wordt, alleen betrekking op het gedeelte van de simulatie waarin deze stationair is.

Verder kan de gebruiker nog de totale breedte en de klassebreedte van het histogram invoeren, dat in het simulatierapport voor de weergave van de bij de leesacties opgetreden responsetijden gebruikt wordt.

De modellering van ECHOSIM speelt zich af op ethernet-pakketniveau waardoor alle servicetijden in het model in de orde van milliseconden liggen. Deze servicetijden hebben betrekking op het werkstation, het netwerk en de fileserver. De gebruiker kan de defaultwaarden hiervoor, die berekend, gemeten of geschat zijn, eventueel vervangen door andere waarden.

6.3. De structuur van ECHOSIM.

ECHOSIM is een procesgeoriënteerd simulatieprogramma. In het programma zijn meerdere typen processen aanwezig en van ieder type kunnen tijdens een simulatie meerdere processen gelijktijdig optreden.

De verschillende typen processen zijn :

- het "read-control"-besturingsproces.
- het token-proces.
- het watchdog-proces.
- het fileserver-ruis-proces.

De eerste drie typen processen hebben direkt betrekking op het inlezen van bestanden door werkstations. Het laatste type proces staat in wezen los van deze leesacties.

Een read-control-proces stuurt alle acties die optreden bij het herhaaldelijk inlezen van bestanden door een enkel werkstation. Dit besturingsproces initieert tijdens een leesactie voortdurend token-processen. Het besturingsproces zelf blokkeert na iedere initiatie van een token-proces.

Een token-proces verricht de acties die optreden bij het verzenden van een verzoekpakket door een werkstation via het netwerk aan een fileserver en het antwoorden daarop door de fileserver, bijvoorbeeld bij het door een werkstation vanaf een fileserver inlezen van een 8 kB NFS-blok uit een bestand. Na het uitvoeren van deze acties verstuurt het token-proces een bericht naar het overeenkomstige, geblokkeerde besturingsproces en beëindigt zelf. Het besturingsproces wordt hierdoor weer geactiveerd en kan een volgend token-proces initiëren. Dit wordt herhaald totdat alle acties, die met het inlezen van een bestand door een werkstation gepaard gaan, zijn uitgevoerd.

Een watchdog-proces is een eenvoudig proces dat als een watchdog-timer voor het besturingsproces dienst doet door het geblokkeerde besturingsproces uiterlijk na een bepaalde tijd weer te activeren indien het bericht van een token-proces te lang uitblijft.

Het fileserver-ruis-proces is gericht op het verbruiken van een bepaald deel van de servicecapaciteit van een fileserver en fungeert aldus als een ruis- of stoor-proces. Dit proces plaatst met negatief-exponentieel verdeelde tussenpozen een verzoek bij de fileserver dat gedurende een uniform verdeelde servicetijd de fileserver belast. De verzoeken van dit proces hebben bij de fileserver prioriteit boven die van de token-processen teneinde een zo konstant mogelijk percentage van de servicecapaciteit van de fileserver te verbruiken onafhankelijk van de momentane belasting van de fileserver door de token-processen. Het fileserver-ruis-proces kan gebruikt worden voor het modelleren van overige op de fileserver aanwezige processen of van belastingen van de fileserver die van andere ethernet-segmenten afkomstig zijn. Deze fileserver-ruis-processen kunnen naar wens wel of niet gebruikt worden.

In eerdere versies van het simulatieprogramma werd voor de leesacties met één enkel proces volstaan waardoor er derhalve ook geen sprake was van een berichtuitwisseling

tussen verschillende processen, die bedoeld is voor de onderlinge synchronisatie van deze processen.

De reden voor het gebruik in de uiteindelijke versie van ECHOSIM van verscheidene processen bij het realiseren van een leesactie ligt in het feit dat binnen NFS een client-systeem bij het inlezen van een bestand twee NFS-verzoeken om (opeenvolgende) blokken uit het bestand gelijktijdig in het ECHO-systeem uit kan hebben staan. Deze beide verzoeken worden in principe onafhankelijk van elkaar door het netwerk en de fileserver verwerkt. Hierdoor zijn voor de modellering van de beide verzoeken in ECHOSIM afzonderlijke token-processen vereist.

6.3.1. Het read-control-besturingsproces.

Het read-control-besturingsproces representeert in een simulatie een enkel werkstation dat herhaaldelijk met negatief-exponentieel verdeelde tussenpozen via het netwerk vanaf een fileserver bestanden inleest. Bij de aanvang van een simulatie worden derhalve evenveel van deze processen gecreëerd als het aantal door de gebruiker van ECHOSIM voor de simulatie gedeclareerde werkstations.

Gedurende een leesactie door een werkstation initieert een besturingsproces herhaaldelijk een token-proces, of een tweetal token-processen direkt na elkaar, waarna het besturingsproces zelf blokkeert. De laatste actie van een token-proces bestaat steeds uit het versturen van een bericht naar het bijbehorende besturingsproces waardoor dit gereactiveerd wordt. Het besturingsproces voert vervolgens één of meerdere acties uit, waaronder veelal de initiatie van een volgend token-proces, waarna het wederom blokkeert. Deze opeenvolgende initiaties van processen tijdens een leesactie zijn grafisch weergegeven in fig. 6.1.

Bij aanvang van een leesactie wordt allereerst het begintijdstip genoteerd. De leesactie wordt ingeluid met de initiatie door het besturingsproces van een token-proces van het open-type. De acties van dit token-proces bestaan uit het routeren van een verzoek door een werkstation via het ethernet naar de fileserver en van het antwoord daarop van de fileserver terug naar het werkstation. Beide in de vorm van een enkel ethernet-pakket.

Vervolgens wordt met de eigenlijke bestandsoverdracht aangevangen. Het besturingsproces initieert hiertoe een enkel token-proces van het udp1-type. Dit token-proces verzorgt het inlezen door het werkstation van het eerste 8 kB NFS-blok uit het bestand op de NFS-fileserver.

Na afloop van het eerste udp1-type token-proces test het besturingsproces of het gehele bestand is ingelezen. Dit zal na het eerste NFS-blok in het algemeen nog niet het geval zijn waarop het besturingsproces twee token-processen initieert. Eerst een token-proces van het udp1-type en 2 tot 5 ms later een token-proces van het udp2-type. Deze beide processen lezen ieder een volgend NFS-blok uit het bestand in.

Na de beëindiging van het udp1-type token-proces van dit paar initieert het bestu-

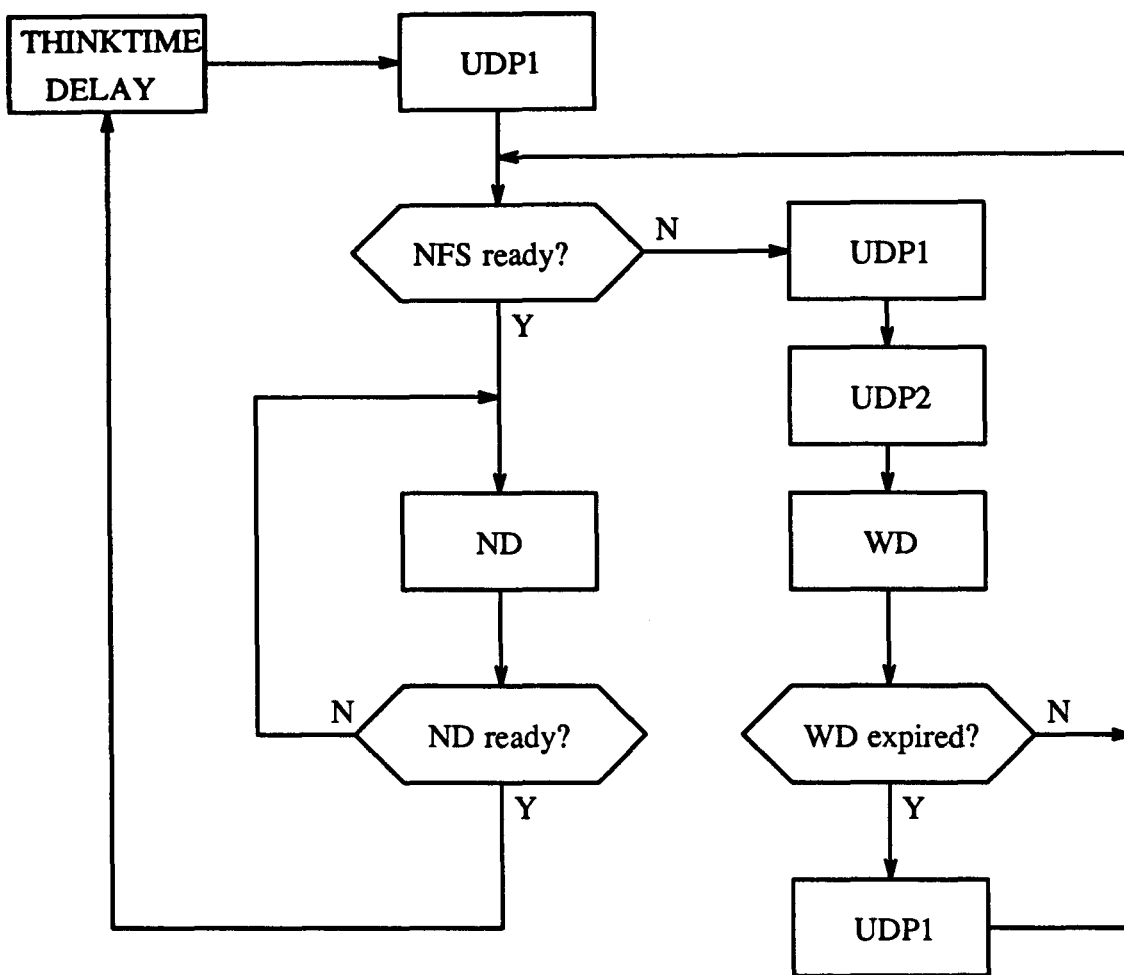


Fig. 6.1 De tijdens een leesactie door het besturingsproces geïnitieerde processen.

ringsproces een watchdog-proces en blokkeert weer. Dit watchdog-proces zal na een watchdog-tijd van 10 ms een bericht naar het besturingsproces terugsturen en beëindigen. Het udp2-type token-proces verstuurt echter een "cancel-watchdog"-bericht naar het besturingsproces op het moment dat de overdracht van het bij dit token-proces behorende NFS-blok van de fileserver via het ethernet naar het werkstation begint. Het besturingsproces breekt het watchdog-proces af indien het dit cancel-watchdog-bericht eerder ontvangt dan het bericht van het watchdog-proces. Het cancel-watchdog-bericht wordt echter genegeerd indien het ontvangen wordt na het bericht van het watchdog-proces.

Het bericht van het watchdog-proces heeft tot doel het geblokkeerde besturingsproces te activeren indien het cancel-watchdog-bericht van het udp2-type token-proces langer dan de watchdog-tijd uitblijft. In deze gevallen zal het door het watchdog-proces geactiveerde besturingsproces een enkel token-proces van het udp1-type initiëren.

Vervolgens ontvangt het besturingsproces de berichten die verstuurd worden bij de beëindiging van het udp2-type token-proces en eventueel van het afzonderlijke udp1-type token-proces.

Opnieuw test het besturingsproces of inmiddels het gehele bestand is ingelezen. Indien dat nog niet het geval is, wordt de zojuist beschreven cyclus na 14 tot 24 ms herhaald. Indien dat wel het geval is, wordt de dan geldende simulatietijd, het eindtijdstip van de leesactie, genoteerd.

Steeds indien een udp1- of udp2-type token-proces beëindigt en een NFS-blok door het werkstation is ontvangen, bepaalt het besturingsproces of hierbij al dan niet een transmissiefout is opgetreden. De gebruiker kan hiervoor een kans specificeren.

Een transmissiefout bij de overdracht van (een pakket van) een NFS-blok leidt na een retransmissietijd van 700 ms tot de initiatie van een enkel udp1-type token-proces. Dit afzonderlijke token-proces kan opgevat worden als een hertransmissie van het vorige verzoek waarbij de transmissiefout is opgetreden. Gedurende de retransmissietijd ligt de leesactie stil.

De met de leesactie gepaard gaande responsetijd, dat wil zeggen het verschil tussen het eind- en het begintijdstip van een leesactie, wordt in een datastructuur opgeslagen voor statistische verwerking.

Na afloop van de leesactie van het bestand vanaf de NFS-fileserver wordt nog een bepaalde hoeveelheid data van de ND-fileserver ingelezen. Het besturingsproces initieert hiervoor een reeks van token-processen van het nd-type die ieder 6 kB vanaf de ND-fileserver inlezen totdat de gehele hiervoor door de gebruiker opgegeven hoeveelheid data door het werkstation is ontvangen.

Hiermee is een cyclus van het besturingsproces teneinde. Het besturingsproces wacht vervolgens een negatief-exponentieel verdeelde thinktime tot aan het begin van een volgende leesactie van het bijbehorende werkstation.

6.3.2. Het token-proces.

Het token-proces komt in ECHOSIM in 4 versies voor, namelijk het open-, udp1-, udp2- en nd-type. Het open-type treedt bij een leesactie als eerste token-proces op en gaat aan de eigenlijke bestandsoverdracht vooraf. De bestandsoverdracht zelf wordt vervolgens in de vorm van afzonderlijke 8 kB NFS-blokken door de udp1- en udp2-typen token-processen gerealiseerd. Na afloop van de eigenlijke bestandsoverdracht verzorgen de nd-type token-processen het inlezen van een bepaalde hoeveelheid data vanaf de ND-fileserver.

Alle token-processen doorlopen dezelfde reeks van toestanden en kunnen derhalve in een enkele procesbeschrijving worden ondergebracht. De aard van de door de 4 typen token-processen in de verschillende toestanden te verrichten acties vertoont eveneens sterke overeenkomsten. Afhankelijk van het type worden in sommige toestanden bepaalde acties wel of niet uitgevoerd en/of bij de bepaling van de servicetijd een specifieke verdeling gebruikt.

Alle token-processen routeren eerst een verzoekpakket van een werkstation via het eth-

ernet naar de fileserver. Het antwoord hierop van de fileserver, dat uit één of meerdere pakketten bestaat, wordt vervolgens via de ethernet-controller van de fileserver en het ethernet zelf terug naar het werkstation geleid.

Alle servers in het model, dat wil zeggen het ethernet, de fileserver en de ethernet-controller van de fileserver, hanteren de FCFS-queueing discipline bij de eventuele selectie van het verzoek in de rij van wachtenden voor de server dat als eerstvolgende voor serviceverlening in aanmerking komt.

De servicetijd van een pakket bij het ethernet, oftewel de tijd gedurende welke het ethernet bezet wordt voor de overdracht van een pakket, vormt een deterministische grootheid die direkt bepaald wordt door de lengte van het pakket. De bepaling van de servicetijden bij de fileserver is gecompliceerd en wordt in de volgende paragraaf besproken.

Ieder type token-proces probeert direkt na de initiatie ervan het ethernet te bemachtigen en een relatief klein verzoekpakket naar de fileserver te versturen. Indien het ethernet op dit moment bezet is, treedt er voor dit pakket een wachttijd bij de ethernet-node op. Na de beëindiging van de serviceverlening bij het ethernet is het pakket bij de fileserver-node aangekomen en probeert het pakket aldaar service te verkrijgen. Ook hier kunnen wachttijden optreden indien de fileserver-node reeds bezet is.

De verwerking van een verzoek door de fileserver levert na een bepaalde servicetijd een antwoordbericht op, dat uit één of meer pakketten bestaat, en aan de ethernet-controller van de fileserver wordt overgedragen ter verzending over het ethernet naar het betreffende werkstation. Dit bericht bezet na een eventuele wachttijd de server van de ethernet-controller-node waarop de serviceverlening door deze node aan dit bericht begint. De controller zal proberen het eerste c.q. enige pakket van dit bericht te verzenden. Hiertoe wordt geprobeerd het ethernet eveneens te bemachtigen. Indien het ethernet vrij is, lukt dit direkt. Bij een bezet ethernet wordt het pakket in de wachtrij van de ethernet-node geplaatst.

Op een gegeven moment begint de serviceverlening door het ethernet aan het betreffende pakket en wordt dit naar het werkstation overgedragen. Gedurende deze tijd bezet het token-proces dus zowel de controller-node als de ethernet-node en is er dus sprake van "multiple-resource-possession" door dit token-proces.

Na de beëindiging van de serviceverlening door het ethernet wordt ook de controller weer vrijgegeven indien het antwoordbericht slechts één enkel pakket omvat. Indien het antwoordbericht echter uit meerdere pakketten bestaat, wacht het token-proces gedurende 0.2 ms voordat geprobeerd wordt het volgende pakket te verzenden. Dit herhaalt zich dan totdat alle pakketten van het antwoordbericht zijn verstuurd, waarna ook de ethernet-controller wordt vrijgegeven. Het antwoordbericht is dan in zijn geheel bij het werkstation gearriveerd.

Aldus wordt gerealiseerd dat er een tijdsinterval aanwezig is tussen de afzonderlijke pakketten van een bericht, indien slechts één werkstation een bestand inleest, en dat, in-

dien meerdere fileservers tegelijkertijd NFS-blokken versturen, de pakketten op het ethernet afwisselend van de verschillende fileservers afkomstig zullen zijn. Dit alles in overeenstemming met de resultaten van traces van het pakketverkeer op het ethernet.

Als laatste actie verstuurt ieder token-proces een bericht naar het bijbehorende, geblokkeerde besturingsproces waarna het hierdoor weer geactiveerde besturingsproces de eerstvolgende tot zijn repertoire behorende akties zal uitvoeren. Het token-proces beëindigd daarna definitief.

Alleen het udp2-type token-proces verstuurt ook nog een bericht naar het besturingsproces op het moment dat het eerste pakket van het bijbehorende NFS-blok over het ethernet naar het werkstation verzonden wordt. Dit zogenaamde cancel-watchdog-bericht aktiveert het besturingsproces om het watchdog-proces af te breken. Dit bericht wordt door het besturingsproces genegeerd indien het bericht van het watchdog-proces eerder ontvangen is en het watchdog-proces derhalve reeds beëindigd is.

6.3.3. De servicetijden bij de fileserver.

De bepaling van de servicetijden bij de fileserver verloopt gecompliceerd.

Voor de udp1- en udp2-type token-processen samen staan drie uniforme distributies ter beschikking, respectievelijk UDP, UDP1 en UDP2, met in deze volgorde afnemende gemiddelde servicetijden, namelijk respectievelijk de intervallen [23,35], [15,30] en [10,14] met alle waarden in milliseconden.

Voor het nd-type token-proces staan 2 uniforme distributies ter beschikking, namelijk ND1 en ND2, respectievelijk [10,30] en [20,30] met wederom alle tijden in milliseconden.

De servicetijd van het open-type token-proces wordt op eenvoudige wijze bepaald, namelijk op basis van de UDP-distributie.

Voor de overdracht van het eerste NFS-blok uit een bestand, wat steeds door een udp1-type token-proces wordt gerealiseerd, wordt een gemiddeld relatief lange servicetijd bepaald op basis van de UDP-distributie.

Volgende NFS-blokken uit het bestand worden steeds ingelezen door een paar van direkt na elkaar door het besturingsproces geïnitieerde token-processen. Het paar bestaat uit een udp1- en udp2-type token-proces. Dit paar kan eventueel gevolgd worden door een afzonderlijk udp1-type token-proces indien het antwoord op het udp2-type token-proces van het paar relatief lang uitblijft.

Steeds geldt dat voor een token-proces bij de fileserver een gemiddeld relatief lange servicetijd wordt bepaald op basis van de UDP-distributie indien het aan het betreffende token-proces voorafgaande token-proces bij de fileserver niet van hetzelfde besturingsproces afkomstig is. Dit kan het geval zijn indien meerdere besturingsprocessen gelijktijdig leesacties uitvoeren en de verzoeken van token-processen van de verschillende besturingsprocessen tussen elkaar door bij de fileserver aankomen.

Indien het voorafgaande token-proces van hetzelfde besturingsproces afkomstig is,

wordt de servicetijd voor het udp1-type token-proces, dat tot het paar van token-processen behoort, bepaald op basis van de UDP1-distributie. De servicetijd van het udp2-type token-proces wordt dan in 75 % van de gevallen bepaald op basis van de UDP2-distributie terwijl in de overige gevallen gebruik gemaakt wordt van de UDP-distributie. De servicetijd van een afzonderlijk udp1-type token-proces wordt dan in eveneens 75 % van de gevallen bepaald door de UDP2-distributie en in de overige gevallen door de UDP1-distributie.

Tenslotte wordt bij een nd-type token-proces de servicetijd bij de fileserver bepaald op basis van de ND1-distributie indien het voorafgaande token-proces bij de fileserver eveneens een nd-type token-proces en van hetzelfde besturingsproces afkomstig is, terwijl in de overige gevallen een gemiddeld langere servicetijd op basis van de ND2-distributie bepaald wordt.

6.4. Een evaluatie van ECHOSIM.

ECHOSIM simuleert het gegevenstransport in ECHO-systemen. Op dit gegevens-transport zijn verscheidene factoren van invloed. Echter niet al deze factoren zijn in ECHOSIM geïmplementeerd. Sommige factoren zijn niet of op een benaderende wijze in ECHOSIM ondergebracht. Dit is het geval indien een (nauwkeurige) implementatie ervan onnodig geacht wordt of indien niet over voldoende inzicht in of niet over voldoende statistische gegevens omtrent een bepaalde faktor beschikt wordt.

Zo is het bij Ethernet gebruikte CSMA/CD-toegangsprotocol niet in ECHOSIM geïmplementeerd. De performance van een ethernet, waarover in de praktijk verwarring blijkt te bestaan, is afhankelijk van verscheidene factoren en niet in de laatste plaats van de werking van dit toegangsprotocol. Uit de literatuur zijn meetresultaten voor de performance van een ethernet bekend, zie [SCH80], waaruit blijkt dat de maximaal haalbare bezettingsgraad van een ethernet sterk afhankelijk is van de (gemiddelde) grootte van de over het ethernet verzonden pakketten en tot 97 % kan oplopen bij het gebruik van pakketten met de maximaal bij ethernet toegestane lengte van 1500 bytes voor het dataveld. In dat geval gaan dus slechts enkele procenten van de transmissiecapaciteit van het ethernet verloren met de voor dit "contention"-protocol karakteristieke selectie van het aangesloten station dat als eerstvolgende een pakket mag verzenden indien er daarvoor meerdere kandidaten zijn.

Voor zeer kleine pakketten daalt de maximaal haalbare bezettingsgraad tot de waarde 1/e van het "slotted-ALOHA"-protocol, zie [TAN81]. In dat geval is het "collision detection"-mechanisme van CSMA/CD niet of nauwelijks meer effectief.

(Weliswaar wordt in [SCH80] gemeten aan een inmiddels verouderde versie van Ethernet met een bitsnelheid van 2.94 in plaats van 10 Mbit/s en een maximale lengte van het dataveld van een pakket van 512 in plaats van 1500 bytes, maar met hetzelfde CSMA/CD-toegangsprotocol zodat ervanuitgegaan mag worden dat resultaten voor de latere, definitieve versie van Ethernet niet wezenlijk zullen verschillen.)

In ECHO-systemen treden bulktransporten van gegevens op waarbij hoofdzakelijk relatief grote ethernet-pakketten, veelal met een maximale lengte, gebruikt worden. Zo

wordt een NFS-blok als 5 maximum-lengte pakketten plus een pakket met een dataveld van 920 bytes verstuurd, die samen het 8 kB-datablok plus de headers van het UDP/IP-protocol vormen. Het gegevenstransport van of naar de ND-fileserver geschiedt met pakketten met een dataveld van 1072 bytes zijnde 1 kB data plus de ND/IP-headers. Pakketten, die een verzoek bevatten, hebben slechts een omvang van 60 tot 160 bytes maar komen bij het bulktransport relatief weinig voor. De gemiddelde pakketgrootte zal in ECHO-systemen derhalve groot zijn.

Een consequentie van het gebruik van een eenvoudige FCFS-queueing discipline voor het ethernet is dat eventuele wachttijden anders over de afzonderlijke pakketten verdeeld worden dan bij het CSMA/CD-protocol zal geschieden. Dit is echter niet of nauwelijks relevant daar het inlezen van een bestand met de overdracht van een groot aantal pakketten gepaard zal gaan waardoor alleen de gemiddelde wachttijd van de pakketten van belang zal zijn.

Op grond van het bovenstaande is afgezien van een implementatie van het CSMA/CD-protocol die het reeds grote aantal events, dat bij een afzonderlijke leesactie optreedt, nog verder zou vergroten.

De implementatie van ECHOSIM is onder meer gebaseerd op de analyse van traces van het gegevenstransport binnen het PCG-netwerk. ECHOSIM is aldus sterk toegesneden op het PCG-netwerk en derhalve systeemafhankelijk. Dit heeft zowel betrekking op de structuur van ECHOSIM, zoals bijvoorbeeld de implementatie van het voor NFS karakteristieke paarsgewijs aanvragen van blokken data uit een bestand, als op de parametrisering van ECHOSIM.

NFS zal deel uitmaken van alle ECHO-systemen. Het paarsgewijs verrichten van NFS-verzoeken, waarbij het tweede verzoek kan worden opgevat als een prefetch oftewel read-ahead van een blok data over het netwerk, vormt een wezenlijk aspect van NFS. Het is in NFS geïmplementeerd om verbetering van de performance van NFS te realiseren. De implementatie ervan in ECHOSIM is derhalve eveneens wezenlijk. De resultaten van ECHOSIM kunnen niet nauwkeuriger zijn dan de nauwkeurigheid waarmee het simulatieprogramma de realiteit modelleert.

Voor de simulatie van ECHO-systemen, waarin systemen aanwezig zijn, die sneller of langzamer zijn dan de in het PCG-netwerk aanwezige systemen, is het opnieuw vaststellen van de daarin op ethernet-niveau optredende servicetijden vereist. Dit zal dan leiden tot een afwijkende parametrisering en derhalve tot afwijkende resultaten. Dit is echter onvermijdelijk. De resultaten van de simulatie zijn van toepassing op het gesimuleerde systeem en niet op wezenlijk daarvan afwijkende systemen.

Bij de traces kan onderscheid gemaakt worden tussen traces waarbij slechts één enkel werkstation bestanden inleest en traces waarbij meerdere werkstations tegelijkertijd leesacties uitvoeren.

ECHOSIM is voornamelijk gebaseerd op de eerste groep traces, waarin een specifieke leesactie van een enkel werkstation geanalyseerd kan worden zonder dat in de trace verstoringen aanwezig zijn door het optreden van pakketten die afkomstig zijn van of

bestemd zijn voor andere werkstations.

In deze eerste groep traces vertonen (groepen van) pakketten een repeterend gedrag bij het inlezen van opeenvolgende NFS-blokken uit een bestand. Tevens variëren de tijdsintervallen tussen opeenvolgende (groepen van) pakketten vrij gelijkmatig over een begrensd interval. Om deze reden is er voor de servicetijden steeds gebruik gemaakt van uniforme distributies.

Het is niet mogelijk geweest de interne werking van de fileserver tijdens het afhandelen van de NFS-verzoeken van één of meer werkstations te onderzoeken. Het is ook om deze reden dat de fileserver in het wachtrijnetwerk, dat als model van ECHO-systemen dienst doet, door een enkel node wordt voorgesteld die zowel de processor als de disk van de fileserver representeert.

Het "disk-scheduling"-algoritme, dat de processor hanteert bij het in een wachtrij voor de disk plaatsen van verzoeken om diskblokken, kan ertoe leiden dat deze verzoeken in een andere volgorde door de disk worden afgehandeld dan de volgorde waarin ze door de processor zijn ontvangen. Dit algoritme is niet in ECHOSIM geïmplementeerd daar gebleken is dat NFS-verzoeken vrijwel steeds in dezelfde volgorde beantwoord worden als waarin ze door de fileserver ontvangen zijn.

In ECHOSIM wordt een leesactie ingeluid met één token-proces van het open-type, dat een enkel ethernet-pakket van het werkstation naar de fileserver en weer terug leidt. In de praktijk treden met name bij hoge belastingen ook wel meerdere van dergelijke vraag-en-antwoord-pakketten op voorafgaand aan de eigenlijke bestandsoverdracht in de vorm van NFS-blokken. In dit geval kan blijkbaar alleen het laatst ontvangen antwoord als een positief antwoord worden opgevat.

In ECHOSIM treedt de interactie met de ND-fileserver steeds na afloop van de eigenlijke leesactie op. In werkelijkheid treden dergelijke gegevensoverdrachten, waarschijnlijk swap-operaties, ook wel tijdens de leesacties op. Dit is niet in ECHOSIM geïmplementeerd voornamelijk omdat de omvang van deze gegevensoverdrachten en de kans er op niet met enige mate van nauwkeurigheid kon worden vastgesteld.

De interactie met de ND-fileserver is in ECHOSIM minder nauwkeurig gemodelleerd dan de interactie met de NFS-fileserver. Bij de interactie met de ND-fileserver is een korrekte omvang van de belasting van de ND-fileserver belangrijker dan een korrekte implementatie van deze belasting of een korrekte interactietijd.

De in ECHOSIM aanwezige servicetijden, die alle op ethernetpakket-niveau liggen, zijn berekend, gemeten of geschat.

In het geval van het ethernet zijn de servicetijden berekend op basis van de lengte van de pakketten en de vaste bitsnelheid van het ethernet.

De servicetijden bij het werkstation zijn gemeten. ECHOSIM bevat voor de representatie van de werkstations slechts één proces-type terwijl in het PCG-netwerk verschillende typen werkstations aanwezig zijn. Voor de waarden van de parameters die direct betrekking hebben op een werkstation is daarom een gewogen gemiddelde gebruikt van de voor de verschillende werkstations gemeten waarden. Dit betreft bijvoorbeeld het

tijdsinterval vanaf de ontvangst van een NFS-blok tot aan het verrichten van een volgend NFS-verzoek.

Ook de servicetijden bij de fileserver, in het geval er slechts één leesactie tegelijkertijd afgehandeld wordt, zijn gemeten. De servicetijden voor een NFS-verzoek zijn groter indien er meerdere leesacties gelijktijdig door de fileserver afgehandeld moeten worden. Er worden in dat geval in ECHOSIM regelmatig relatief lange servicetijden bepaald op basis van de UDP-distributie in plaats van op de UDP1- of UDP2-distributie die relatief kortere servicetijden opleveren. De UDP-distributie is geschat op basis van de traces waarbij meerdere werkstations tegelijkertijd leesacties uitvoeren.

De servicetijden bij de fileserver voor nd-type token-processen zijn eveneens op basis van de traces geschat. Een meer nauwkeurige bepaling kan in dit geval moeilijk plaatsvinden omdat de verzoeken bij de interactie met de ND-fileserver betrekking hebben op 8 kB-blokken terwijl het ND-protocol slechts 6 pakketten van 1 kB na elkaar verstuurt voordat een acknowledge-pakket ontvangen moet worden. In de traces kunnen om deze reden de 8 kB-blokken slecht gelokaliseerd worden waardoor de bijbehorende servicetijden moeilijk bepaald kunnen worden.

Ook bij de interactie met de ND-fileserver is aangenomen dat de servicetijden groter zijn indien de ND-verzoeken van een bepaald besturingsproces niet direkt na elkaar door de ND-fileserver afgehandeld worden.

7. Resultaten van metingen en simulaties.

7.1. De verrichte metingen.

De metingen, die met het programma Simulation aan het PCG-netwerk zijn verricht, hebben statistische resultaten opgeleverd voor de responsetijden van leesacties door werkstations als functie van de door deze werkstations in het netwerk geïntroduceerde belasting. De omvang van deze belasting wordt bepaald door het aantal werkstations dat leesacties uitvoert, de frequentie waarmee de leesacties door ieder werkstation uitgevoerd worden en de grootte van de ingelezen bestanden.

Voor de hoeveelheid data, die tijdens een meetperiode bij iedere leesactie wordt ingelezen, is steeds een vaste waarde van 512 kB gebruikt, de verwachte gemiddelde omvang van een casefolder.

Voor het verrichten van de metingen staan maximaal 12 werkstations en 2 fileservers ter beschikking. Bij 6 van deze 12 werkstations fungeert de fileserver echo als ND-fileserver, bij de overige 6 betreft dit de fileserver galm. In de grafieken wordt na het aantal werkstations tussen haakjes de naam van de ND-fileserver van de werkstations vermeld.

Het is vanwege de diversiteit van de aanwezige typen werkstations niet mogelijk geweest bij de metingen alleen identieke typen te gebruiken. De 6 werkstations met echo als ND-fileserver zijn 1 25 MHz-, 2 20 MHz- en 3 15 MHz-werkstations. Bij de 6 werkstations met galm als ND-fileserver betreft dit 1 20 MHz- en 5 15 MHz-werkstations.

Het aantal werkstations dat gedurende een meetperiode actief is geweest, bedraagt 1, 6 of 12. In het eerste geval wordt voor de thinktime tussen de opeenvolgende leesacties van een werkstation een vaste waarde gebruikt. In de laatste twee gevallen wordt voor de thinktime een negatief-exponentiële verdeling aangenomen.

Bij de metingen met 6 werkstations worden steeds 6 werkstations met dezelfde ND-fileserver gebruikt.

Bij de metingen zijn 1 of 2 fileservers betrokken. De beide fileservers echo en galm zijn identieke systemen.

Het maakt voor de belasting van de NFS-fileserver, en daardoor ook voor de tijdsduren van de leesacties, een verschil of deze fileserver tijdens de metingen wel of niet tevens als ND-fileserver moet dienen voor de werkstations die de leesacties uitvoeren. Dit verschil is onderzocht door 6 werkstations met dezelfde ND-fileserver tijdens een meting de eigen ND-fileserver als NFS-fileserver te laten gebruiken en tijdens een andere meting bij de leesacties de andere fileserver te laten benaderen. Dit is zowel gedaan voor de werkstations met echo als voor die met galm als ND-fileserver.

De metingen zijn verricht voor 4 verschillende waarden voor de gemiddelde thinktime. Aldus wordt de frequentie waarmee de leesacties optreden gevarieerd en daarmee de in

het systeem geïntroduceerde belasting.

De duur van een meetperiode is steeds 40 minuten geweest. Er is afgezien van een langere tijdsduur, waardoor meetwaarden met een grotere statistische nauwkeurigheid verkregen zouden zijn, vanwege het grote aantal te verrichten metingen en de hoge belastingen die daarbij met name bij de fileserver optreden. Tevens is er bij een langere tijdsduur een grotere kans op het optreden van stoorprocessen tijdens de metingen die de waarde van de resultaten van de meting sterk verminderen maar waarvan het al dan niet opgetreden zijn vaak moeilijk te detecteren is.

In de grafieken wordt de gemiddelde responsetijd van de tijdens een meetperiode verrichte leesacties uitgezet tegen het gemiddelde aantal leesacties dat gedurende de meetperiode per tijdseenheid heeft plaatsgevonden. Dit aantal wordt na afloop van de meetperiode berekend uit het totale aantal leesacties dat tijdens de meetperiode is opgetreden.

De reden dat de gemiddelde responsetijd in de grafieken niet wordt uitgezet tegen de door de gebruiker ingevoerde waarde voor de thinktime, of het omgekeerde daarvan, is dat de werkelijk gerealiseerde thinktime met name bij hoge belastingen van het systeem en dienovereenkomstige lage waarden in de orde van enkele seconde voor de thinktime, tot enkele seconde groter kan zijn dan de door de gebruiker daarvoor opgegeven waarde. Dit wordt veroorzaakt door het te laat terugkomen van de "sleep"-systeemprocedure waarmee de individuele thinktimes binnen het meetprogramma Simulation gerealiseerd worden.

7.2. De verrichte simulaties.

De verrichte simulaties kunnen onderscheiden worden in enerzijds de simulaties die tot doel hebben het simulatieprogramma te valideren door de resultaten te vergelijken met de meetresultaten die het meetprogramma Simulation oplevert bij overeenkomstige belastingen tijdens de simulaties en de metingen, en anderzijds de uiteindelijke simulaties van systemen waarvoor nog geen fysiek equivalent bestaat maar waarvan performanceaspecten door middel van de simulaties onderzocht kunnen worden.

Tot de eerste groep behoren de simulaties waarbij 1, 6 of 12 werkstations tegelijkertijd actief zijn. Hierbij wordt een (onrealistisch) hoge frequentie voor de leesacties van de afzonderlijke werkstations gebruikt teneinde met het beperkte aantal werkstations toch de voor ECHO-systemen karakteristieke middelmatige en hoge belastingen van het netwerk en de fileserver te realiseren.

Bij 6 actieve werkstations is dit zowel gedaan voor het geval dat de NFS-fileserver tevens als ND-fileserver dienst doet als voor het geval dat een andere fileserver als ND-fileserver optreedt. In het eerste geval zijn de responsetijden hoger omdat de fileserver zwaarder belast wordt.

Bij 12 actieve werkstations fungeert de NFS-fileserver voor 6 van de werkstations als ND-fileserver terwijl de overige 6 werkstations dezelfde NFS-fileserver maar een andere ND-fileserver hebben. Dit in overeenstemming met de metingen die met het programma Simulation zijn verricht voor het geval dat 12 werkstations tegelijkertijd

leesacties uitvoeren.

Deze simulaties zijn verricht voor een reeks van belastingen door de gemiddelde thinktime tussen de opeenvolgende leesacties van de afzonderlijke werkstations te variëren.

Bij de uiteindelijke simulaties is voor de gemiddelde thinktime tussen de opeenvolgende leesacties van de afzonderlijke werkstations een realistische waarde gebruikt. Bij deze simulaties wordt het aantal werkstations gevarieerd om de responsetijd bij verschillende belastingen van het systeem te bepalen.

Voor de waarde van de gemiddelde thinktime is uitgegaan van het aantal van 80 casefolders dat een caseworker per dag inleest. Bij een daadwerkelijke werktijd van 7 uur komt dit overeen met een gemiddelde thinktime van 315 seconden. Bij deze groep van simulaties zijn echter waarden van 200, 150 en 100 seconden voor de gemiddelde thinktime gebruikt. Enerzijds om ook de overige grote gegevensstromen zoals het inlezen van dossiers door caseworkers en het inlezen van casefolder door casecheckers te representeren. Anderzijds vanwege het feit dat dat ECHO-systemen een verbetering van de efficiency en derhalve van de arbeidsproductiviteit van de medewerkers moeten opleveren.

De twee kortste thinktimes zijn bedoeld om de invloed van piekbelastingen op de responsetijd van de leesacties te bepalen.

Deze simulaties zijn verricht voor 1 en voor 2 op het ethernet-segment aangesloten fileservers. In het laatste geval treden beide fileserver ieder voor de helft van het aantal werkstations als NFS- en als ND-fileserver op.

Bij de simulaties hebben de leesacties betrekking op bestanden met een vaste omvang van 512 kB, waarbij achteraf nog 60 kB van de ND-fileserver wordt betrokken.

In ECHOSIM kan slechts een enkel type werkstation gerepresenteerd worden. Voor de parametrisering daarvan is uitgegaan van een gewogen gemiddelde van de aanwezige werkstations zodat een werkstation is verkregen dat in snelheid ligt tussen het standaard 15 MHz- en het 20 MHz-werkstation. Dit is gedaan om de heterogene samenstelling van de aanwezige werkstations in het PCG-netwerk, maar mogelijk eveneens in ECHO-systemen, op een gemiddelde wijze in het simulatieprogramma onder te brengen.

Bij alle simulaties is gebruik gemaakt van een kans op het optreden van transmissiefouten bij de overdracht van een 8 kB-NFS-blok van 0.002. Dit betekent dat gemiddeld bij 1 op de 8 leesacties van 512 kB, oftewel 64 NFS-blokken, een transmissiefout optreedt, die een vertraging van de leesactie van 700 ms oplevert.

Verder wordt bij alle simulaties 2 procent van de servicecapaciteit van de fileserver door het fileserver-ruis-proces in beslag genomen.

Op basis van de door de simulaties verkregen resultaten voor de responsetijd en de eisen die daaraan gesteld worden, kan het aantal werkstations bepaald worden dat per segment kan worden aangesloten.

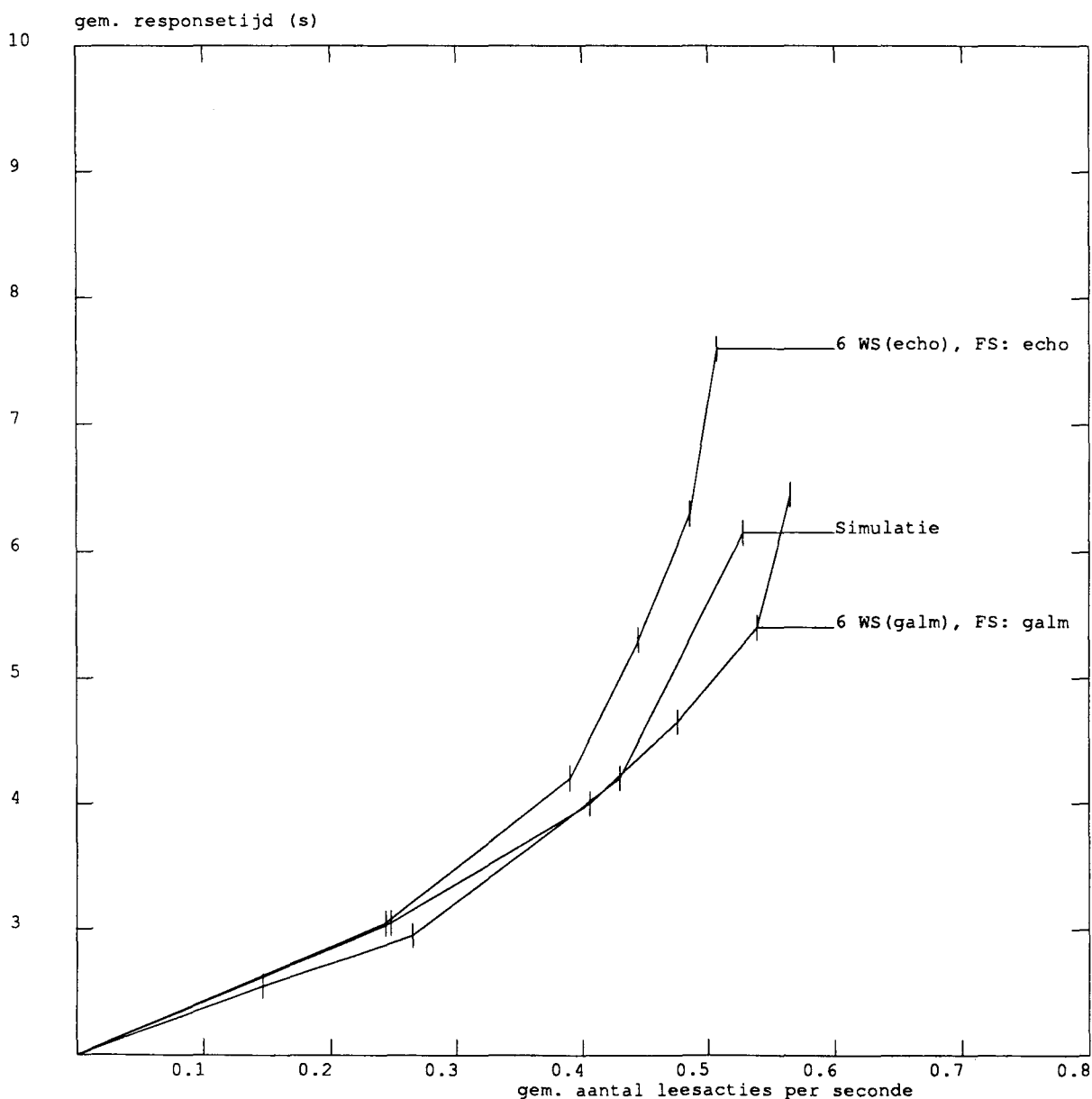
7.3. Resultaten van metingen en overeenkomstige simulaties.

De responsetijden van afzonderlijk optredende leesacties, waarbij door een werkstation een hoeveelheid data van 512 kB uit een bestand gelezen wordt, bedraagt voor een :

- 15 MHz-werkstation 2.10 ± 0.05 s.
- 20 MHz-werkstation 1.90 ± 0.05 s.
- 25 MHz-werkstation 1.70 ± 0.05 s.

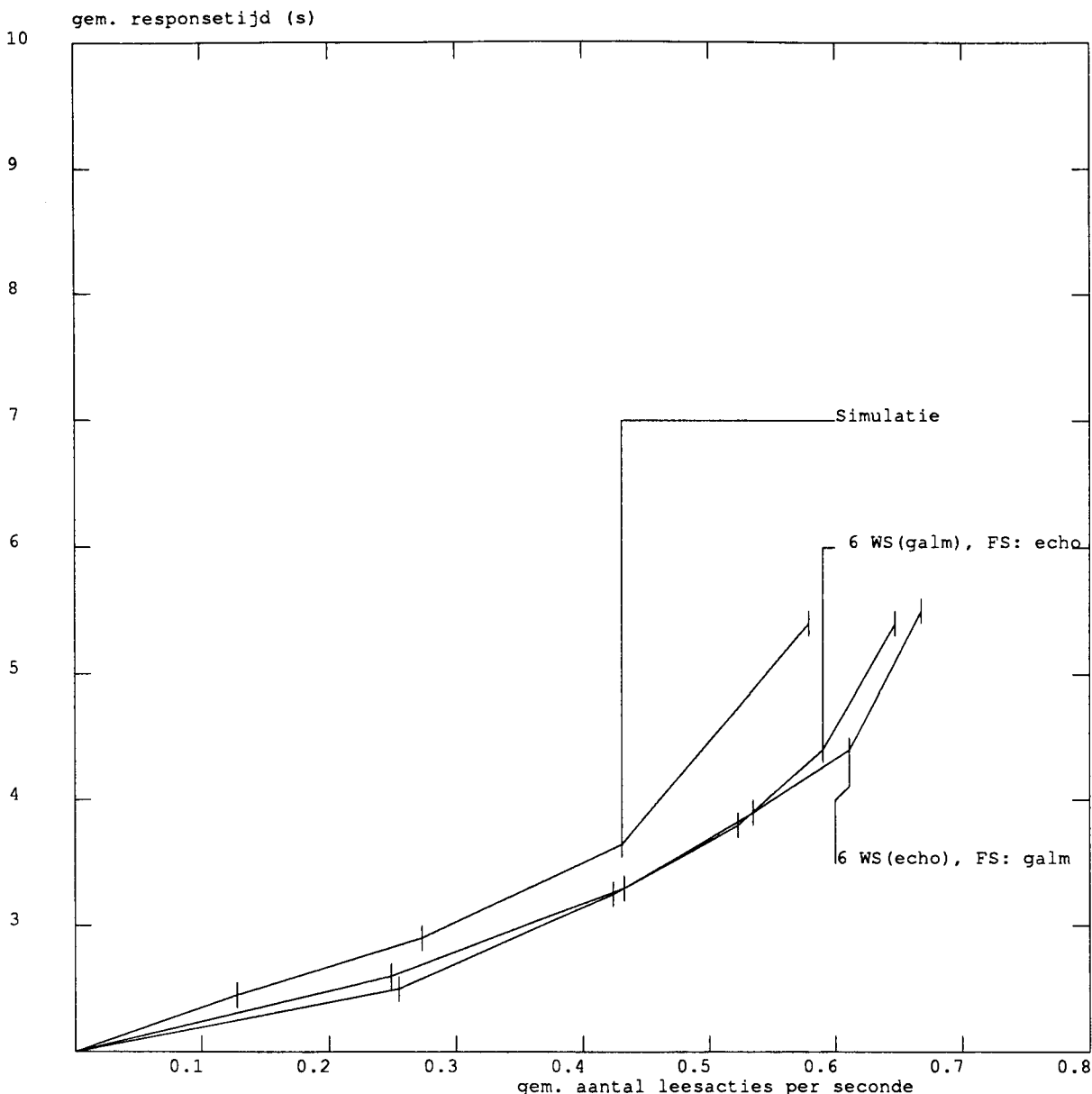
Er is hierbij geen verschil opgetreden tussen leesacties vanaf de fileserver echo of vanaf de fileserver galm.

In deze waarden zijn resultaten van leesacties waarbij CRC-fouten zijn opgetreden, die een vertraging van 700 ms veroorzaken, niet meegenomen.



Grafiek 7.1 Responsetijd als functie van de belasting, NFS- \equiv ND-fileserver.

Evenmin leesacties waarbij vertragingen zijn opgetreden van meer dan 200 ms ten gevolge van tijdens de eigenlijke leesacties optredende interacties met de ND-fileserver.



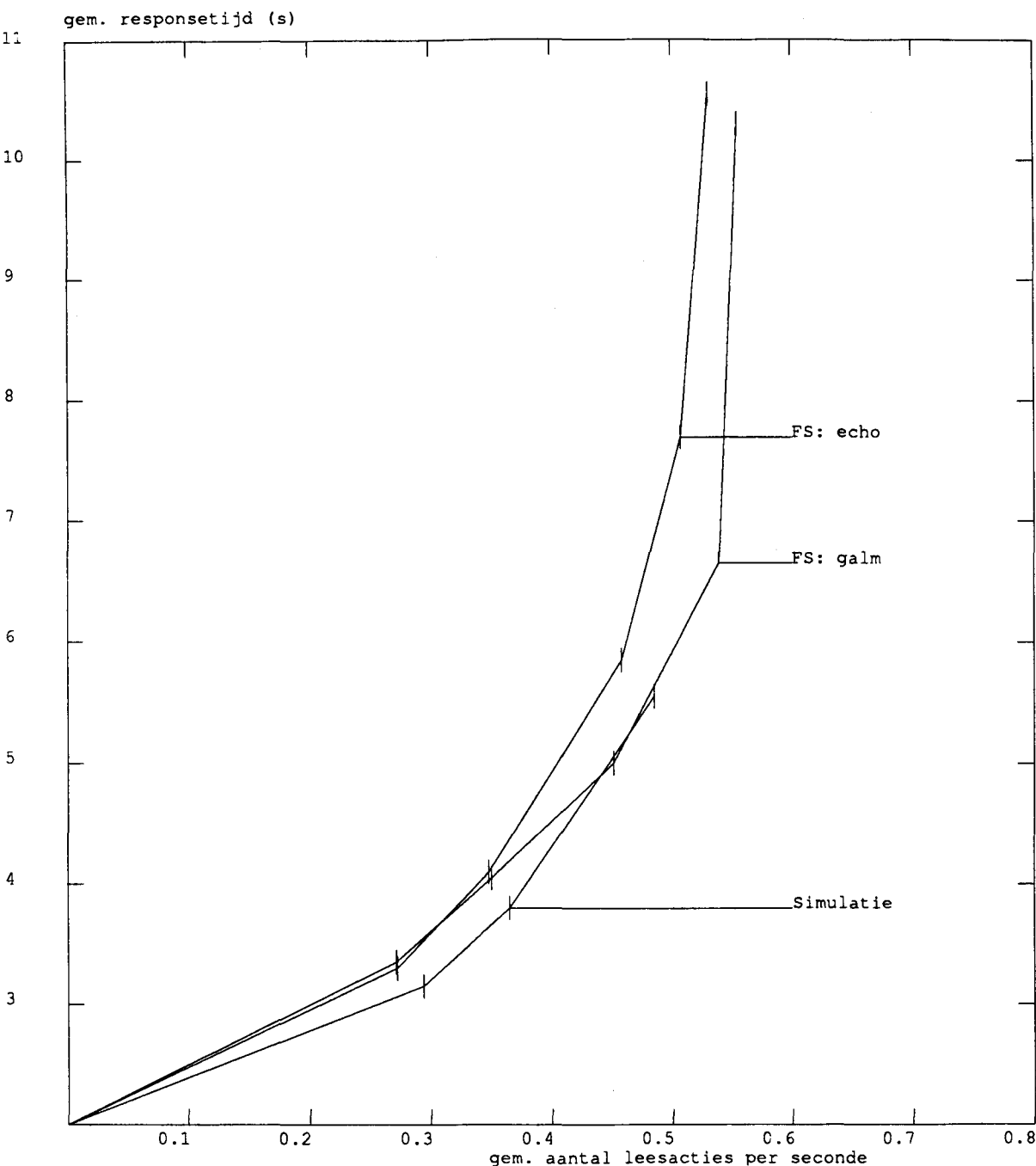
Grafiek 7.2 Responsetijd als functie van de belasting, NFS- ≠ ND-fileserver.

In grafiek 7.1 is de gemiddelde responsetijd weergegeven voor 6 werkstations en 1 fileserver, die zowel als NFS- als als ND-fileserver fungeert. Dit zowel voor de gemeten resultaten met echo dan wel galm als fileserver als de simulatieresultaten.

Uit de grafiek blijkt dat voor de responsetijden bij de fileserver echo grotere waarden optreden dan bij de fileserver galm.

In principe is het mogelijk dat de bestanden, waaruit tijdens de metingen de data gelezen wordt, bij echo meer verspreid op de disk terecht gekomen zijn dan bij galm. Dat zou dan aanleiding geven tot hogere servicetijden bij echo dan bij galm. Bij de afzonderlijk plaatsvindende leesacties is echter geen verschil vastgesteld tussen

leesacties vanaf echo of vanaf galm waardoor dit aspect niet als verklaring kan worden aangevoerd.

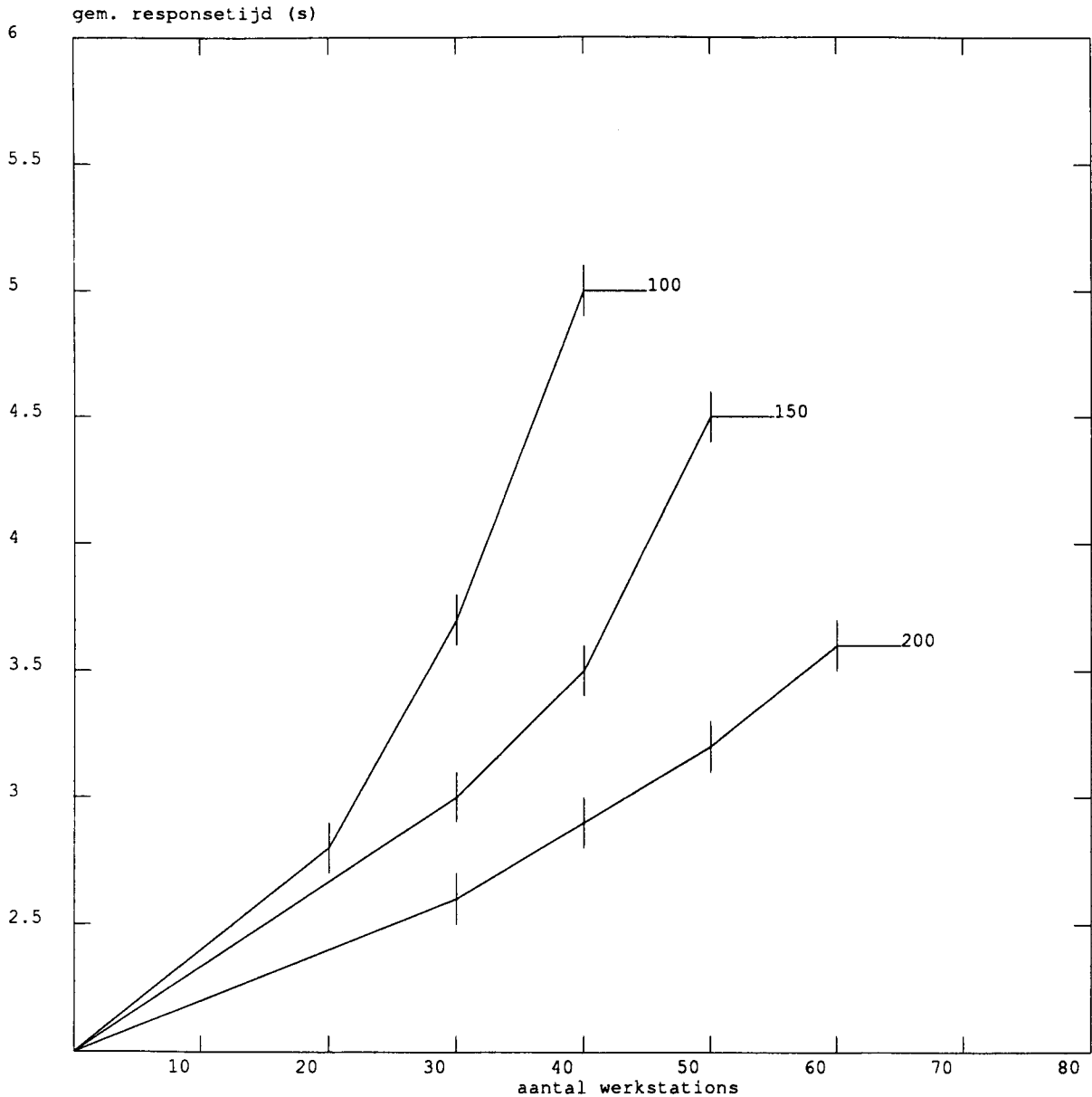


Grafiek 7.3 Responsetijd als functie van de belasting, 12 werkstations.

Een gedeeltelijke verklaring is dat bij de metingen na iedere leesactie door het programma Simulation een logregel met informatie over de zojuist verrichte leesactie wordt weggeschreven naar een bestand op de fileserver echo onafhankelijk of galm dan wel echo als NFS-fileserver dienst doet. Dit gaat bij de meting gepaard met de

uitwisseling van gemiddeld een half NFS-blok tussen de fileserver en het betrokken werkstation. Hierdoor wordt bij een meting echo altijd enigszins zwaarder belast dan galm. De omvang van deze interactie kan echter niet meer dan ongeveer 3 procent bedragen van het inlezen van de hoeveelheid data van 512 kB.

De simulatieresultaten in grafiek 7.1 komen redelijk overeen met de gemeten waarden voor de responsetijd.



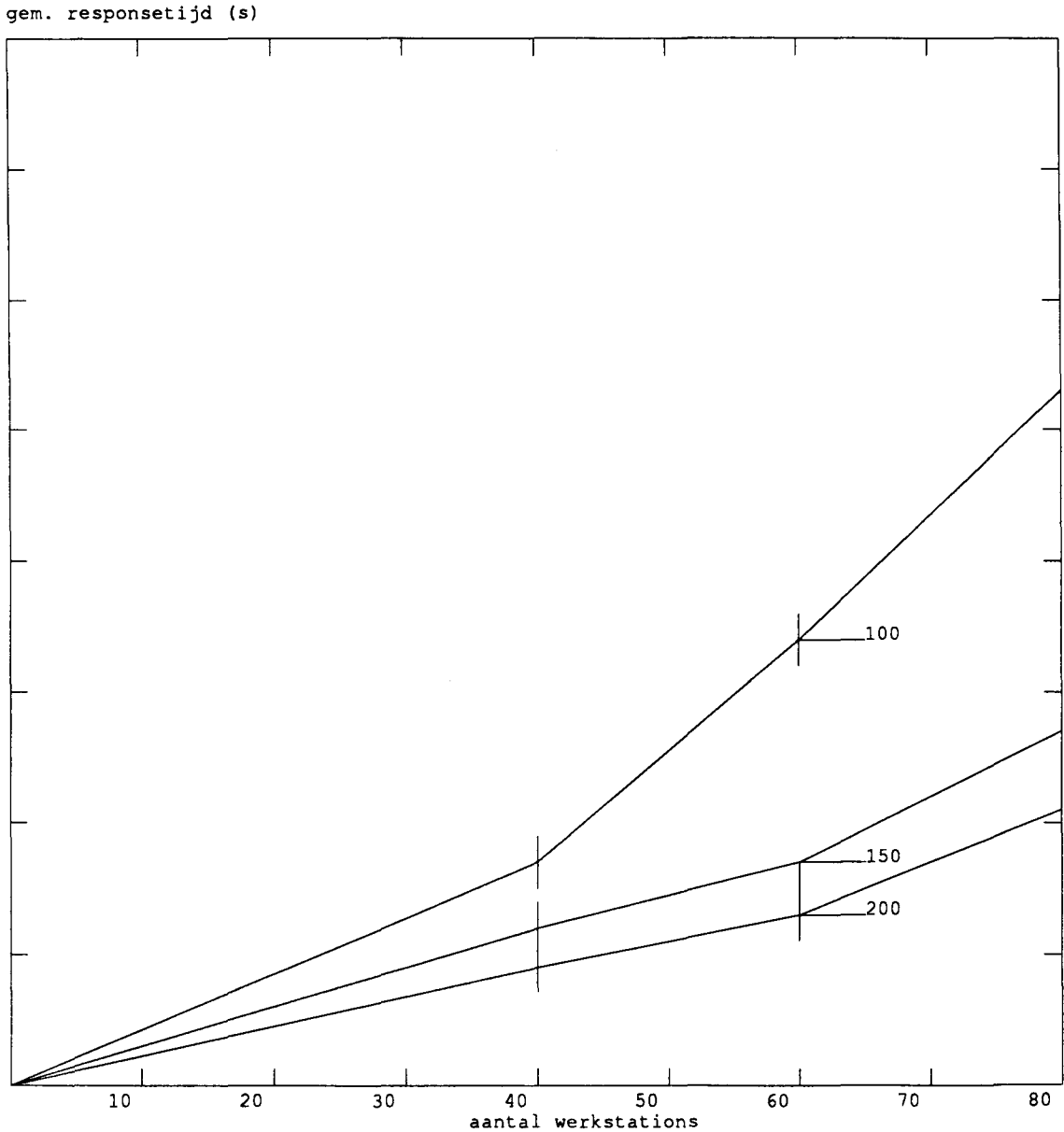
Grafiek 7.4 Simulatieresultaten voor de responsetijd, 1 fileserver.

In grafiek 7.2 wordt de gemiddelde responsetijd gegeven voor 6 werkstations en verschillende NFS- en ND-fileservers. De gemeten resultaten voor echo dan wel galm als NFS-fileserver vertonen een sterke overeenkomst.

Opmerkelijk zijn de veel lagere gemeten responsetijden in vergelijking met grafiek 7.1

bij dezelfde omvang van de belasting. Indien de NFS-fileserver niet tevens als ND-fileserver hoeft te fungeren, treedt de verdubbeling van de responsetijd tot 4 seconde bij galm pas bij een 30 % hogere belasting op terwijl bij echo dit zelfs pas bij een 50 % hogere belasting gebeurt.

De simulatieresultaten voor de responsetijden liggen in dit geval te hoog in vergelijking met de gemeten resultaten. Blijkbaar is de overhead, die de interactie met de ND-fileserver oplevert, niet voldoende zwaar in het simulatieprogramma ondergebracht. Het niet hoeven te fungeren als ND-fileserver door de NFS-fileserver zou een grotere vermindering van de responsetijden hebben moeten opleveren.

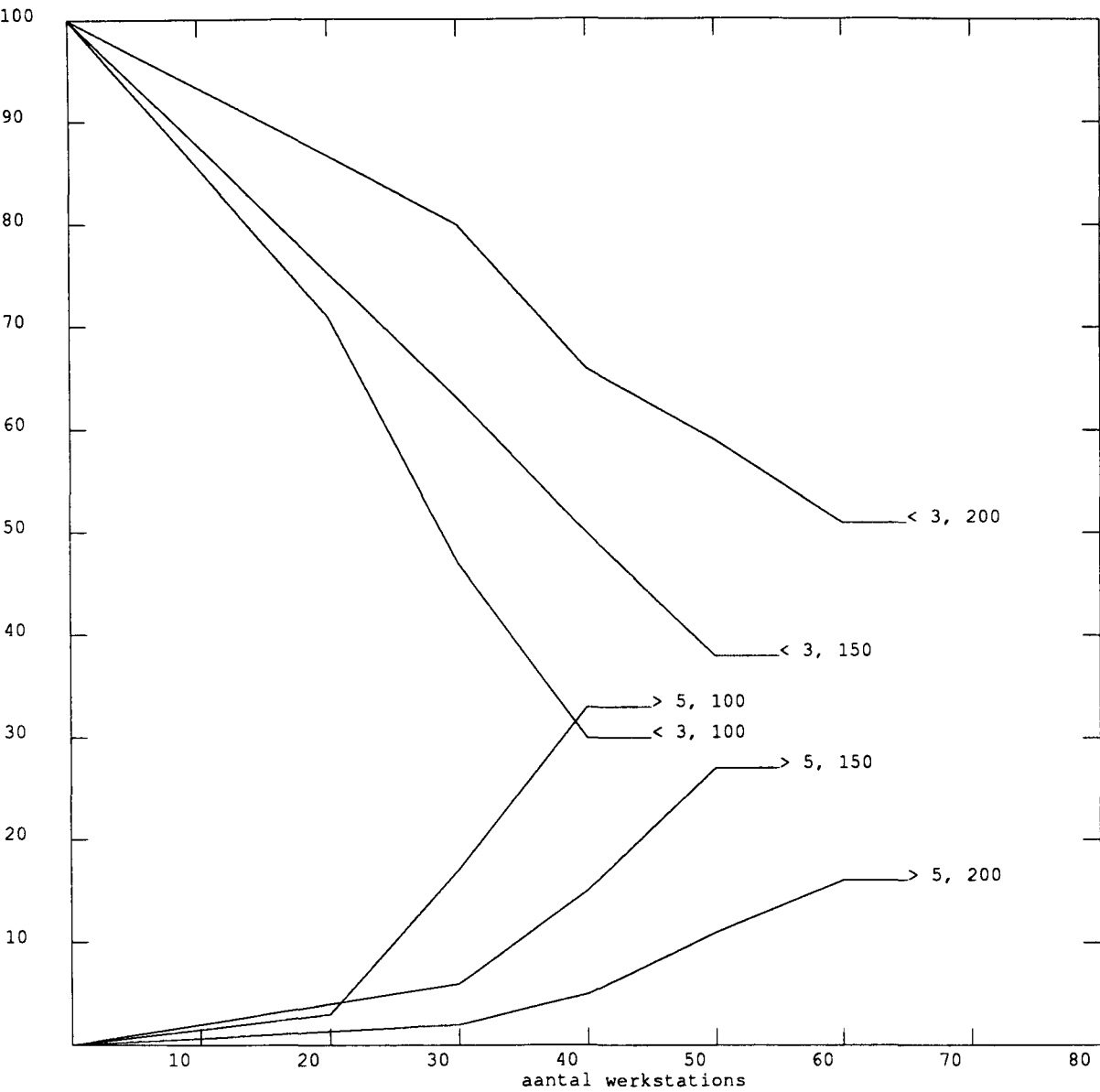


Grafiek 7.5 Simulatieresultaten voor de responsetijd, 2 fileserver.

In grafiek 7.3 is de gemiddelde responsetijd weergegeven indien 12 werkstations actief zijn, waarbij 2 fileservers ieder voor de helft van dit aantal als ND-fileserver fungeren

terwijl één van deze beide fileservers als NFS-fileserver voor alle werkstations optreedt. Ook hier zijn de responsetijden bij echo als NFS-fileserver iets hoger dan bij galm als NFS-fileserver.

De simulatieresultaten liggen hier iets lager dan de gemeten resultaten maar beide komen nog wel in redelijke mate overeen.

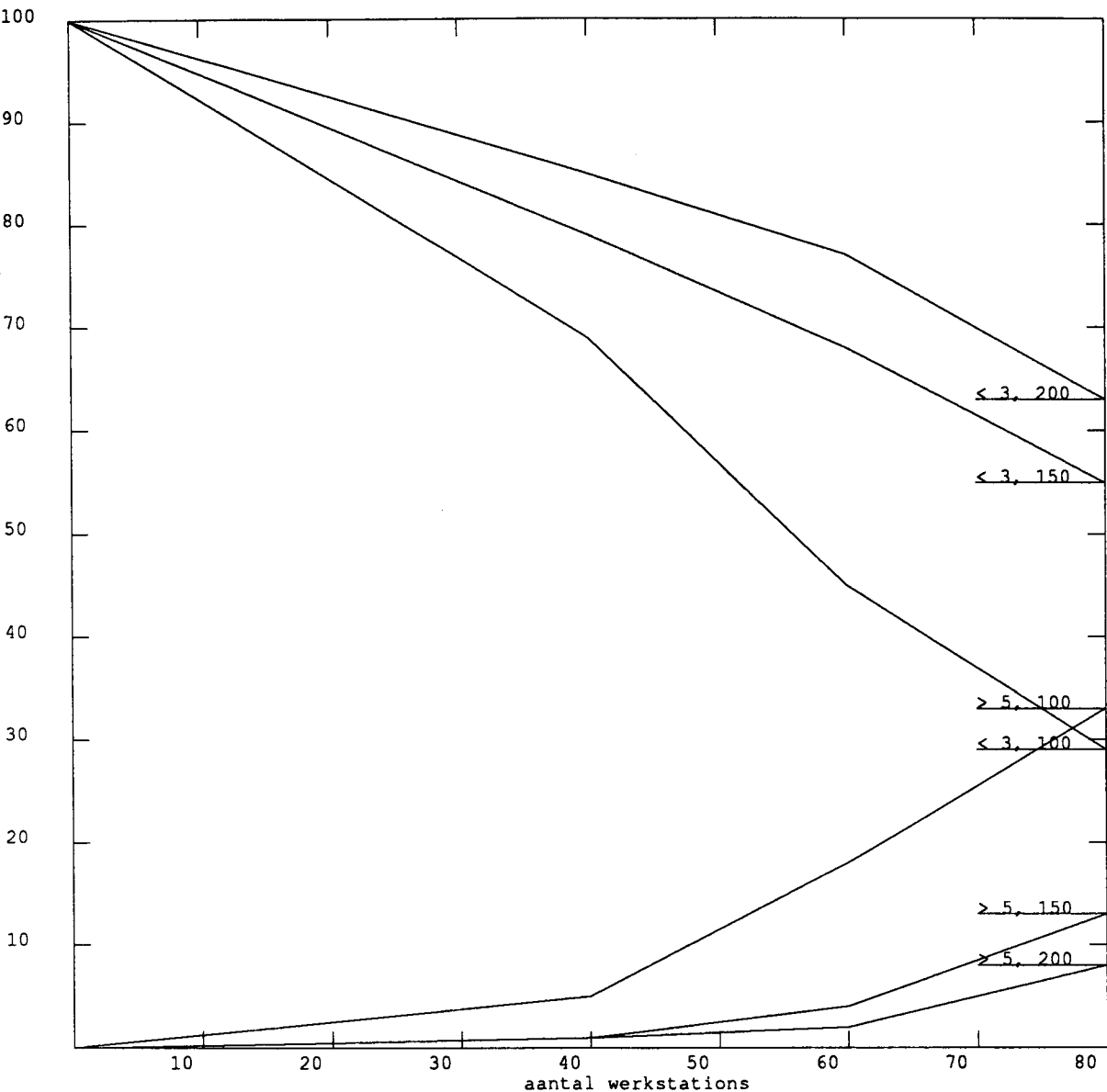


Grafiek 7.6 Perc. van de responsetijd < 3 en > 5 seconde, NFS- ≡ ND-fileserver.

7.4. Resultaten van overige simulaties.

In grafiek 7.4 zijn de simulatieresultaten weergegeven voor de gemiddelde responsetijd bij grotere aantallen werkstations met de thinktime als parameter en 1 fileserver, die voor alle werkstations zowel als NFS- als als ND-fileserver dienst doet.

In grafiek 7.5 is hetzelfde weergegeven voor 2 fileservers die ieder voor de helft van het aantal werkstations zowel als NFS- als als ND-fileserver fungeren.



Grafiek 7.7 Perc. van de responsetijd < 3 en > 5 seconde, NFS- ≠ ND-fileserver.

Uit deze grafieken kan geconcludeerd worden dat een tweemaal zo groot aantal werkstations bij een tweemaal zo grote thinktime, oftewel bij de halve frequentie van de leesacties, vergelijkbare resultaten oplevert. Resultaten voor de responsetijd in tussenvallende situaties, voor wat betreft aantallen werkstations en thinktimes, kunnen derhalve via interpolatie verkregen worden en ook extrapolatie kan in beperkte mate worden toegepast.

Verder blijkt dat de resultaten voor de responsetijd bij het gebruik van 2 fileservers overeenkomen met die bij de helft van het aantal werkstations en slechts 1 fileserver.

Dit duidt erop dat het ethernet niet de bottleneck van het systeem vormt. Dit is mede een gevolg van de modellering van het ethernet in de vorm van een eenvoudige FCFS-wachtrij.

Bij 80 werkstations en een thinktime van 100 seconde is de bezettingsgraad van het ethernet 38 % terwijl deze bij een thinktime van 200 seconde 19 % bedraagt. Deze bezettingsgraden voor het ethernet zijn bij het gebruikte type van het pakketverkeer als respectievelijk piekwaarde en doorsneewaarde voor de gemiddelde belasting zeker haalbaar.

Dit blijkt uit metingen waarbij met de network analyzer met dummy-pakketten een belasting van 50 % van het ethernet gecreëerd wordt waarbij de responsetijden voor de afzonderlijke leesacties van een enkel werkstation slechts 5 % blijkt toe te nemen namelijk van 1.9 naar 2.0 seconde.

Tenslotte wordt in de grafieken 7.6 en 7.7 de fractie van de responsetijden onder 3 seconde en boven 5 seconde weergegeven voor de metingen waarvan de gemiddelde responsetijden in de grafieken 7.4 en 7.6 getoond worden.

De relatieve fout ligt zowel bij de metingen als bij de simulaties in de orde van 5 %. Bij de metingen is dit vastgesteld door het herhalen van bepaalde metingen, bij de simulaties door bepaalde simulaties in subrun's op te delen en de resultaten van de afzonderlijke subrun's statistisch te verwerken alsmede eveneens door bepaalde simulaties te herhalen met een andere waarde voor de seed van de random-number-generator.

8. Conclusie.

Er is een simulatieprogramma opgesteld dat performanceaspecten bepaalt voor het gegevensverkeer op een enkel ethernet-segment van het hiërarchisch gestructureerde computernetwerk van een ECHO-systeem.

Zowel voor de parametrisering als voor de validering van het simulatieprogramma zijn metingen aan het op het PCG aanwezige netwerk verricht.

Voor de parametrisering zijn traces verzameld en geanalyseerd van het pakketverkeer dat op het ethernet tijdens leesacties aanwezig is. De verschillende servicetijden, die bij de verschillende systeemcomponenten optreden bij het door NFS realiseren van de bestandsoverdrachten over het netwerk, zijn handmatig op basis van deze traces bepaald en hebben derhalve een beperkte nauwkeurigheid. Voor het verkrijgen van meer nauwkeurige waarden voor deze parameters is een programmatische analyse van een groot aantal traces vereist.

Voor de validering zijn het PCG-netwerk en het simulatieprogramma onderworpen aan overeenkomstige belastingen waarna de meetresultaten en de simulatieresultaten zijn vergeleken. Hieruit kan geconcludeerd worden dat deze beide typen resultaten in redelijke mate overeenstemmen.

Zowel bij de metingen als bij de simulaties hebben de leesacties betrekking gehad op bestanden van 512 kB, de verwachte gemiddelde omvang van een casefolder.

De gemeten tijdsduur van afzonderlijk optredende leesacties van een bestand van 512 kB bedraagt 2.1, 1.9, en 1.7 seconde voor een werkstation met een klokfrequentie van respectievelijk 15, 20 en 25 MHz.

Bij de metingen is gebleken dat bij iedere leesactie tevens een interactie van het werkstation met de ND-fileserver optreedt. Dit is de fileserver waarop zich specifieke systeem informatie en de swap area van een werkstation bevindt. Zo wordt vrijwel altijd na afloop van het inlezen van een bestand van 512 kB nog ongeveer 60 kB van de ND-fileserver betrokken.

Het maakt voor de responsetijden een verschil of de fileserver, die voor de leesacties benaderd wordt, tevens als ND-fileserver voor de actieve werkstations dienst doet.

Uit metingen blijkt dat bij een fileserver onder belasting van de leesacties van meerdere werkstations de verdubbeling van de responsetijd tot 4 seconde pas optreedt bij een 30 tot 50 % groter gemiddeld aantal door de fileserver verwerkte leesacties indien deze fileserver niet tevens als ND-fileserver hoeft te fungeren ten opzichte van de situatie waarbij dit wel het geval is.

Op grond hiervan lijkt het raadzaam om de imagebestanden op te slaan op een andere fileserver dan de ND-fileserver. Temeer daar in een ECHO-systeem bij de executie van de omvangrijke ECHO-applicatie een bepaalde mate van swapping kan optreden waardoor de ND-fileserver belast wordt. Dit zal dan leiden tot vertragingen van de leesacties indien ook de imagebestanden op deze ND-fileserver zijn opgeslagen.

De simulatieresultaten verschillen enigzins van de meetresultaten indien voor de

leesacties een andere fileserver benaderd wordt dan de ND-fileserver van de werkstations. De implementatie in het simulatieprogramma van de interactie met de ND-fileserver is blijkbaar nog niet optimaal.

Bij de uiteindelijke simulaties zijn grotere aantallen werkstations en meer realistische tussenpozen tussen de opeenvolgende leesacties van de afzonderlijke werkstations van 200, 150 en 100 seconde gebruikt. Dit is gedaan voor 1 en voor 2 op het ethernet-segment aangesloten fileservers.

Op basis van deze simulatieresultaten voor de responsetijden van de leesacties en de eisen die men daaraan stelt, kan het aantal werkstations bepaald worden dat op het segment kan worden aangesloten.

Stelt men als eis dat de gemiddelde tijdsduur van het inlezen van een bestand van 512 kB maximaal 3 seconde mag bedragen dan kunnen bij 1 fileserver ruim 30 werkstations aangesloten worden, bij 2 fileservers bedraagt dit ongeveer 60 werkstations.

In het algemeen zijn de simulatieresultaten voor de responsetijden bij 1 fileserver en een bepaald aantal werkstations vrijwel gelijk aan die voor 2 fileservers en het dubbele aantal werkstations. Dit duidt erop dat het ethernet niet als bottleneck fungeert, ook nog niet indien 80 werkstations actief zijn en de tussenpozen tussen de opeenvolgende leesacties van de afzonderlijke werkstations slechts 100 seconde bedragen. In dat geval is de gemiddelde bezettingsgraad van het ethernet 38 %.

Dit is mede een gevolg van het feit dat het ethernet als een eenvoudige FCFS-wachtrij is gemodelleerd en dat het CSMA/CD-toegangsprotocol van ethernet niet geïmplementeerd is teneinde de complexiteit van het simulatieprogramma te beperken. Uit de literatuur zijn meetresultaten bekend waaruit blijkt dat de maximaal haalbare bezettingsgraad van het ethernet afhankelijk is van de gemiddelde pakketlengte maar bij het gebruik van pakketten met een maximale lengte tot 98 % kan oplopen. In ECHO-systemen treden bulktransporten van gegevens op waarbij hoofdzakelijk grote pakketten gebruikt worden. Er is derhalve geen reden aan te geven voor de vaak gehoorde stelling dat het ethernet voor niet meer dan 10 % belast zou kunnen of zou mogen worden.

Ook uit metingen met een network analyzer, waarbij met dummy-pakketten een bezettingsgraad van 50 % op het ethernet gecreëerd wordt, blijkt dat de tijdsduur voor een afzonderlijk optredende leeactie hierdoor nauwelijks toeneemt (namelijk van 1.9 tot 2.0 seconde). De voor de leesactie getransporteerde pakketten kunnen blijkbaar ondanks deze bezettingsgraad zonder excessieve vertragingen te ondergaan over het ethernet getransporteerd worden.

Literatuurverwijzingen.

- [DOR86] J.B.M. van Doremalen,
"Approximate analysis of queueing network models",
proefschrift TUE, 14 maart 1986.
- [ECH86] "Progress report to the steering group ECHO october 15th 1986.",
15 oktober 1986, intern Philips rapport OA604R.
- [KLE75] L. Kleinrock,
"Queueing systems",
Vol. I en II, 1975, John Wiley & Sons.
- [KOB78] H. Kobayashi,
"Modelling and analysis",
1978, Addison Wesley.
- [LAV83] S.S. Lavenberg (ed.),
"Computer Performance Modelling Handbook",
1983, Academic Press.
- [OUD87] E.M. van der Ouderaa,
"ECHO system performance",
5 jan 1987, intern Philips memo EO/45/2.
- [SCH87] K.D. Schewe,
"A top-down approach to the configuration of ECHO systems I",
21 december 1987, intern Philips memo KDS005/1.
- [SHO80] J.F. Shoch, J.A. Hupp,
"Measured Performance of an Ethernet Local Network",
Communications of the ACM, vol.23, nr.12, december 1980, p. 711-721.
- [SNI88] W.A.M. Snijders,
"FDDI: Nieuwe standaard voor snelle glasvezel LAN's",
i2-Elektrotechniek/Elektronica, nr. 3, 1988, p. 36-42.
- [TAN81] "A.S. Tanenbaum,
"Computer networks",
1981, Prentice-Hall.
- [TUE79] "Wachttijd en stagnatie problemen",
Collegediktaat TUE nr. 5.564.2, 4e uitgave, dec. 1979.

- [WIT87] J. Witte,
"SIMENV simulation environment functional specification user guide",
8 april 1987, intern rapport PTDS Hilversum.
- [ZEG87] H. Zegels,
"Transportsnelheden van data over een Ethernet-LAN onder belasting",
24 december 1987, afstudeerverslag HTS Heerlen.

Bijlage 1 ECHO-terminologie.

Deze bijlage bevat omschrijvingen van de belangrijkste, voor het ECHO-project specifieke termen voorzover die in dit verslag gebruikt worden.

Het ECHO-project richt zich op de integrale automatisering van de administratieve routinewerkzaamheden binnen met name verzekeringsbedrijven. Dit is terug te vinden in de omschrijvingen van de binnen het ECHO-project gehanteerde termen :

case : de bij een verzekeringsbedrijf plaatsvindende op een polis betrekking hebbende administratieve routineactiviteiten.

case handling : het algemene proces van het afhandelen van cases.

caseworker : medewerker belast met case handling.

casechecker : medewerker belast met het controleren van beslissingen door caseworkers ten aanzien van individuele cases.

casefolder : map waarin zich tijdens het afhandelen van een specifieke case alle door de caseworker tot op dat moment vergaarde informatie omtrent die case bevindt.
Traditioneel geldt hiervoor een papieren implementatie,
binnen een ECHO-systeem een elektronische implementatie.