APPROXIMATE ANALYSIS OF OPEN QUEUEING NETWORKS WITH BLOCKING

by

Elias Yannopoulos

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical and Industrial Engineering
University of Manitoba
Winnipeg, Manitoba R3T 2N2 Canada

ISBN 0-315-86101-0

Name ELIAS YANNOPOULOS

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

~~INDUSTRIAL ENGINEERING~~ ~~0546~~ **U·M·I**

OPERATIONS RESEARCH

SUBJECT TERM

SUBJECT CODE 0796

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences
Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984
Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

Nom ____

Dissertation Abstracts International est organisé en catégories de sujets. Veuillez s.v.p. choisir le sujet qui décrit le mieux votre thèse et inscrivez le code numérique approprié dans l'espace réservé ci-dessous.

____ SUJET

☐☐☐☐ CODE DE SUJET

U·M·I

Catégories par sujets

HUMANITÉS ET SCIENCES SOCIALES

COMMUNICATIONS ET LES ARTS
- Architecture ... 0729
- Beaux-arts ... 0357
- Bibliothéconomie ... 0399
- Cinéma ... 0900
- Communication verbale ... 0459
- Communications ... 0708
- Danse ... 0378
- Histoire de l'art ... 0377
- Journalisme ... 0391
- Musique ... 0413
- Sciences de l'information ... 0723
- Théâtre ... 0465

ÉDUCATION
- Généralités ... 515
- Administration ... 0514
- Art ... 0273
- Collèges communautaires ... 0275
- Commerce ... 0688
- Économie domestique ... 0278
- Éducation permanente ... 0516
- Éducation préscolaire ... 0518
- Éducation sanitaire ... 0680
- Enseignement agricole ... 0517
- Enseignement bilingue et multiculturel ... 0282
- Enseignement industriel ... 0521
- Enseignement primaire. ... 0524
- Enseignement professionnel ... 0747
- Enseignement religieux ... 0527
- Enseignement secondaire ... 0533
- Enseignement spécial ... 0529
- Enseignement supérieur ... 0745
- Évaluation ... 0288
- Finances ... 0277
- Formation des enseignants ... 0530
- Histoire de l'éducation ... 0520
- Langues et littérature ... 0279
- Lecture ... 0535
- Mathématiques ... 0280
- Musique ... 0522
- Orientation et consultation ... 0519
- Philosophie de l'éducation ... 0998
- Physique ... 0523
- Programmes d'études et enseignement ... 0727
- Psychologie ... 0525
- Sciences ... 0714
- Sciences sociales ... 0534
- Sociologie de l'éducation ... 0340
- Technologie ... 0710

LANGUE, LITTÉRATURE ET LINGUISTIQUE
- Langues
  - Généralités ... 0679
  - Anciennes ... 0289
  - Linguistique ... 0290
  - Modernes ... 0291
- Littérature
  - Généralités ... 0401
  - Anciennes ... 0294
  - Comparée ... 0295
  - Mediévale ... 0297
  - Moderne ... 0298
  - Africaine ... 0316
  - Américaine ... 0591
  - Anglaise ... 0593
  - Asiatique ... 0305
  - Canadienne (Anglaise) ... 0352
  - Canadienne (Française) ... 0355
  - Germanique ... 0311
  - Latino-américaine ... 0312
  - Moyen-orientale ... 0315
  - Romane ... 0313
  - Slave et est-européenne ... 0314

PHILOSOPHIE, RELIGION ET THÉOLOGIE
- Philosophie ... 0422
- Religion
  - Généralités ... 0318
  - Clergé ... 0319
  - Études bibliques ... 0321
  - Histoire des religions ... 0320
  - Philosophie de la religion ... 0322
- Théologie ... 0469

SCIENCES SOCIALES
- Anthropologie
  - Archéologie ... 0324
  - Culturelle ... 0326
  - Physique ... 0327
- Droit ... 0398
- Économie
  - Généralités ... 0501
  - Commerce-Affaires ... 0505
  - Économie agricole ... 0503
  - Économie du travail ... 0510
  - Finances ... 0508
  - Histoire ... 0509
  - Théorie ... 0511
- Études américaines ... 0323
- Études canadiennes ... 0385
- Études féministes ... 0453
- Folklore ... 0358
- Géographie ... 0366
- Gérontologie ... 0351
- Gestion des affaires
  - Généralités ... 0310
  - Administration ... 0454
  - Banques ... 0770
  - Comptabilité ... 0272
  - Marketing ... 0338
- Histoire
  - Histoire générale ... 0578
  - Ancienne ... 0579
  - Médiévale ... 0581
  - Moderne ... 0582
  - Histoire des noirs ... 0328
  - Africaine ... 0331
  - Canadienne ... 0334
  - États-Unis ... 0337
  - Européenne ... 0335
  - Moyen-orientale ... 0333
  - Latino-américaine ... 0336
  - Asie, Australie et Océanie ... 0332
- Histoire des sciences ... 0585
- Loisirs ... 0814
- Planification urbaine et régionale ... 0999
- Science politique
  - Généralités ... 0615
  - Administration publique ... 0617
  - Droit et relations internationales ... 0616
- Sociologie
  - Généralités ... 0626
  - Aide et bien-àtre social ... 0630
  - Criminologie et établissements pénitentiaires ... 0627
  - Démographie ... 0938
  - Études de l' individu et de la famille ... 0628
  - Études des relations interethniques et des relations raciales ... 0631
  - Structure et développement social ... 0700
  - Théorie et méthodes. ... 0344
  - Travail et relations industrielles ... 0629
- Transports ... 0709
- Travail social ... 0452

SCIENCES ET INGÉNIERIE

SCIENCES BIOLOGIQUES
- Agriculture
  - Généralités ... 0473
  - Agronomie. ... 0285
  - Alimentation et technologie alimentaire ... 0359
  - Culture ... 0479
  - Élevage et alimentation ... 0475
  - Exploitation des péturages ... 0777
  - Pathologie animale ... 0476
  - Pathologie végétale ... 0480
  - Physiologie végétale ... 0817
  - Sylviculture et faune ... 0478
  - Technologie du bois ... 0746
- Biologie
  - Généralités ... 0306
  - Anatomie ... 0287
  - Biologie (Statistiques) ... 0308
  - Biologie moléculaire ... 0307
  - Botanique ... 0309
  - Cellule ... 0379
  - Écologie ... 0329
  - Entomologie ... 0353
  - Génétique ... 0369
  - Limnologie ... 0793
  - Microbiologie ... 0410
  - Neurologie ... 0317
  - Océanographie ... 0416
  - Physiologie ... 0433
  - Radiation ... 0821
  - Science vétérinaire ... 0778
  - Zoologie ... 0472
- Biophysique
  - Généralités ... 0786
  - Medicale ... 0760

SCIENCES DE LA TERRE
- Biogéochimie ... 0425
- Géochimie ... 0996
- Géodésie ... 0370
- Géographie physique ... 0368
- Géologie ... 0372
- Géophysique ... 0373
- Hydrologie ... 0388
- Minéralogie ... 0411
- Océanographie physique ... 0415
- Paléobotanique ... 0345
- Paléoécologie ... 0426
- Paléontologie ... 0418
- Paléozoologie ... 0985
- Palynologie ... 0427

SCIENCES DE LA SANTÉ ET DE L'ENVIRONNEMENT
- Économie domestique ... 0386
- Sciences de l'environnement ... 0768
- Sciences de la santé
  - Généralités ... 0566
  - Administration des hipitaux ... 0769
  - Alimentation et nutrition ... 0570
  - Audiologie ... 0300
  - Chimiothérapie ... 0992
  - Dentisterie ... 0567
  - Développement humain ... 0758
  - Enseignement ... 0350
  - Immunologie ... 0982
  - Loisirs ... 0575
  - Médecine du travail et thérapie ... 0354
  - Médecine et chirurgie ... 0564
  - Obstétrique et gynécologie ... 0380
  - Ophtalmologie ... 0381
  - Orthophonie ... 0460
  - Pathologie ... 0571
  - Pharmacie ... 0572
  - Pharmacologie ... 0419
  - Physiothérapie ... 0382
  - Radiologie ... 0574
  - Santé mentale ... 0347
  - Santé publique ... 0573
  - Soins infirmiers ... 0569
  - Toxicologie ... 0383

SCIENCES PHYSIQUES

Sciences Pures
- Chimie
  - Genéralités ... 0485
  - Biochimie ... 487
  - Chimie agricole ... 0749
  - Chimie analytique ... 0486
  - Chimie minérale ... 0488
  - Chimie nucléaire ... 0738
  - Chimie organique ... 0490
  - Chimie pharmaceutique ... 0491
  - Physique ... 0494
  - PolymÇres ... 0495
  - Radiation ... 0754
- Mathématiques ... 0405
- Physique
  - Généralités ... 0605
  - Acoustique ... 0986
  - Astronomie et astrophysique ... 0606
  - Electronique et électricité ... 0607
  - Fluides et plasma ... 0759
  - Météorologie ... 0608
  - Optique ... 0752
  - Particules (Physique nucléaire) ... 0798
  - Physique atomique ... 0748
  - Physique de l'état solide ... 0611
  - Physique moléculaire ... 0609
  - Physique nucléaire ... 0610
  - Radiation ... 0756
- Statistiques ... 0463

Sciences Appliqués Et Technologie
- Informatique ... 0984
- Ingénierie
  - Généralités ... 0537
  - Agricole ... 0539
  - Automobile ... 0540
  - Biomédicale ... 0541
  - Chaleur et ther modynamique ... 0348
  - Conditionnement (Emballage) ... 0549
  - Génie aérospatial ... 0538
  - Génie chimique ... 0542
  - Génie civil ... 0543
  - Génie électronique et électrique ... 0544
  - Génie industriel ... 0546
  - Génie mécanique ... 0548
  - Génie nucléaire ... 0552
  - Ingénierie des systämes ... 0790
  - Mécanique navale ... 0547
  - Métallurgie ... 0743
  - Science des matériaux ... 0794
  - Technique du pétrole ... 0765
  - Technique minière ... 0551
  - Techniques sanitaires et municipales ... 0554
  - Technologie hydraulique ... 0545
- Mécanique appliquée ... 0346
- Géotechnologie ... 0428
- Matières plastiques (Technologie) ... 0795
- Recherche opérationnelle ... 0796
- Textiles et tissus (Technologie) ... 0794

PSYCHOLOGIE
- Généralités ... 0621
- Personnalité ... 0625
- Psychobiologie ... 0349
- Psychologie clinique ... 0622
- Psychologie du comportement ... 0384
- Psychologie du développement ... 0620
- Psychologie expérimentale ... 0623
- Psychologie industrielle ... 0624
- Psychologie physiologique ... 0989
- Psychologie sociale ... 0451
- Psychométrie ... 0632

APPROXIMATE ANALYSIS OF OPEN QUEUEING NETWORKS WITH BLOCKING

BY

ELIAS YANNOPOULOS

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Abstract

The analysis of production systems using queueing techniques has received considerable attention in recent years. Production systems can be represented as queueing networks. Many queueing networks are analysed using simulation. However, the limitations of simulation have resulted in the development of queueing methods which are much faster than simulation and give reasonably good results. The objective of this thesis is to develop new and improved approximation methods for the estimation of system performance measures for queueing networks.

Real life systems consist of arbitrary configurations of tandem, split, and merge queueing networks. The non-exponentiallity of traffic and service processes and buffer capacity constraints make the analysis of real life systems difficult. Existing methods work only for specific arbitrary systems. An approximation method, which can be used to model any type of arbitrary configurations is presented. Existing approximation methods for the analysis of tandem, split, and merge systems, are used as components of the proposed approximation. The method was tested in the modelling of a part of a conveyor system installed in a manufacturing company. It is shown that the proposed method gives good results for low and moderate traffic.

From this analysis, the limitations of existing methods for the modelling of the three basic queue configurations (tandem, split, and merge) became evident. New approximation methods are needed, which will be more accurate, simpler, faster, and capable of modelling a wider variety of tandem, split, and merge systems, than the existing methods. In this respect, two approximations are developed.

i) An approximation method for the analysis of exponential tandem networks is presented. It is shown that it gives improved results when compared with those

obtained by other existing methods. This method provides the joint queue length probability distributions for triplets of adjacent nodes information which can not be obtained by other existing methods.

ii) A simple, and quick approximation method for tandem, split, and merge systems is developed. This is the first method to report results for split, and merge configurations consisting of more than three nodes (with general arrival and service times). This method is a very useful tool when used as part of optimization procedures for which execution time is very important.

Acknowledgements

I wish to express my grateful thanks and appreciation to my advisor Dr. A.S.Alfa, for his invaluable aid, encouragement, enthusiasm, criticism, and excellent advising during the course of this work.

I am also indebted to the other members of the Advisory Committee, Prof. O.Hawaleshka, Dr. D.H.Burn and Dr. J.MacGregor Smith for their valuable suggestions and advice.

Funding provided by the Natural Sciences and Engineering Research Council of Canada through a grant to Dr. A.S.Alfa is gratefully acknowledged. Funding provided by Bell-Northern Research through a grant to Dr. A.S.Alfa and by the University of Manitoba Transport Institute through a grant to Dr. A.S.Alfa is also gratefully acknowledged.

TABLE OF CONTENTS

CHAPTER PAGE

CHAPTER PAGE

CHAPTER PAGE

LIST OF FIGURES

FIGURE	PAGE

FIGURE PAGE

LIST OF TABLES

TABLE		PAGE

CHAPTER 1

Introduction

1.1 Analysing Production Systems using Queueing Techniques

The importance of analysing production systems using queueing techniques, and the difficulties encountered in the analysis are discussed in this section.

1.1.1 The Importance of Analysing Production Systems

Production systems constitute the basic components of any industrial activity. The role of production systems is to process raw material in order to produce finished items. A production system consists of a set of workstations linked by a material handling system and a number of storage areas. Items (customers) arrive at the system from outside, receive service (get processed) at the workstations, and finally depart from the system.

In this highly competitive global economy, a manufacturing company can only survive if it is efficient. This thus calls for the optimal design and operation of its production system. An item can be produced by employing many different combinations of design parameters (parameters such as workstations configurations, available storage space, processing rates, etc.). The engineer is then faced with the following (among other) problems:

- Which is the best system design among a large number of alternatives ?

- What is the effect of a change in the processing rates on the performance of the system ?

Thus the problem is to find the optimal combination of the design parameters to minimize the operating cost of the system.

Considering the high costs associated with modern automated production systems, a slight improvement in efficiency can result in significant reductions in total costs. Thus, there is the need for the development of techniques which can be used for the optimal design and control of production systems. We should be able to use these techniques for: i) estimating the performance of the system and ii) providing answers to the "what-if" questions (such as what is the effect of a specific system parameter on the system performance ?). These techniques can then be used as components of optimization procedures in order to find the optimal design of the system.

The objective of this thesis is to develop new and improved methods for predicting the system performance measures (such as average sojourn time through the system, average inventory levels, etc). All the models developed in this thesis are of the analytical types based on queueing theory.

1.1.2 Representation of Production Systems as Queueing Networks

Every production system can be viewed as a queueing network, where machines and items routes are represented by nodes and arcs, respectively. Queueing networks consist of a number of nodes/queues linked together to form tandem, split, and merge configurations where items/customers travel through the network receiving service at all or some of the service facilities. Processing times at individual nodes can be constant or can have any type of probability distributions. Each item has its own route through the network and its own processing times. Buffers with limited or unlimited capacities for temporary storage of unprocessed items are located behind each node. Queueing networks are generally classified into two types i) open and ii) closed networks. In open networks items enter the system at some specific points and after they receive service, they depart from the network at some specific exit points. In closed networks a fixed and finite number of items are considered to be in the system and are trapped in the system in the sense that no others may enter and none of these may leave.

In this research we consider open queueing networks, and assume that each node is attended by a single server and that all items belong to the same class. Open queueing networks can be considered as consisting of four classes of queue configurations. These four classes of networks are:

- Tandem networks. Tandem networks consist of queues in tandem (Figure 1.1). Buffers (for temporary storage of items) with limited or unlimited capacities are placed behind each node. Items (from outside) arrive only at the first node, receive service at all nodes of the system and finally depart from the last node.

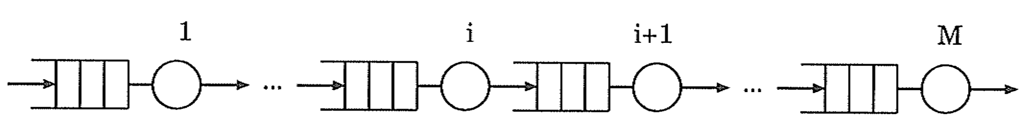

Figure 1.1: Queues in Tandem

- Split configurations. Queueing networks consisting of a single node (first level) linked to n parallel single nodes (second level) are called split configurations

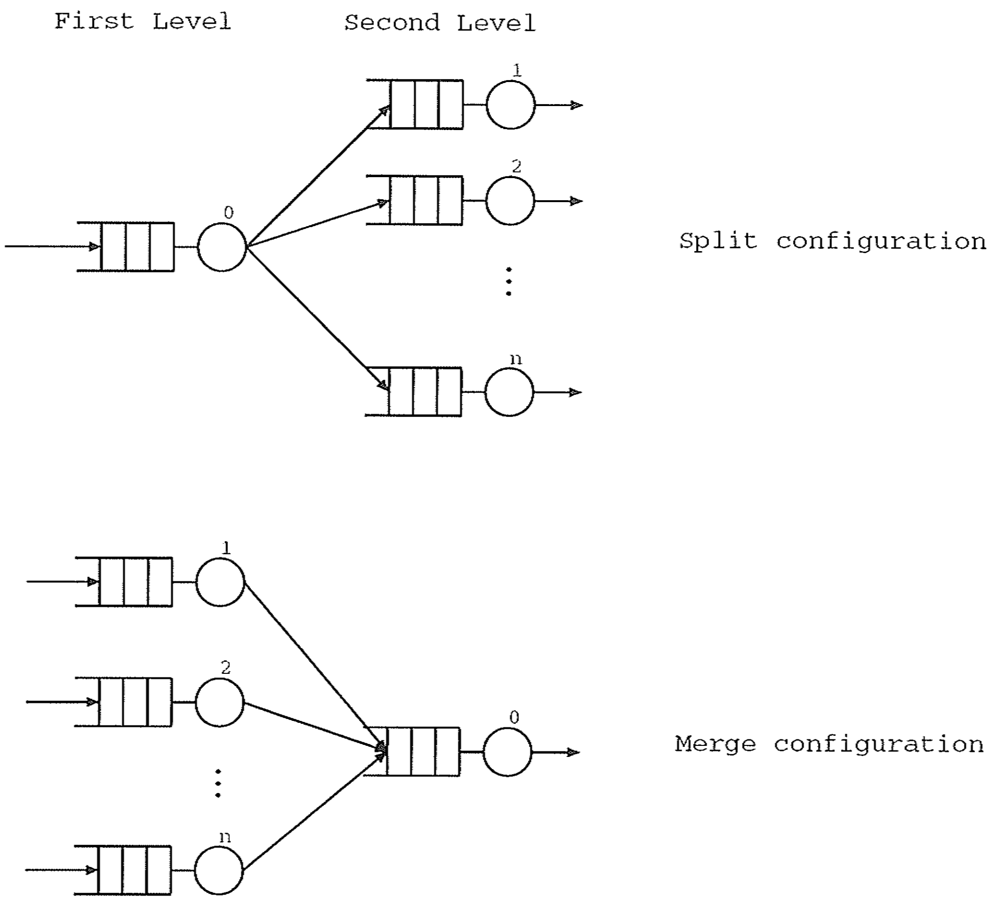

Figure 1.2: Split and Merge Configurations

(Figure 1.2). Items arrive at the first level node and then proceed towards the second level nodes which are the departing points of the system. Buffers are placed behind each node and can have limited or unlimited space.

- Merge configurations. Merge configurations consist of n parallel first level single nodes linked to a single second level node. Items arrive at the first level nodes and depart from the second level node (Figure 1.2). All buffers can have limited or unlimited size.

- Arbitrary configurations. Arbitrary configurations are the networks that can not be classified as pure tandem, split, or merge systems but consist of combinations of the first three classes of networks. One arbitrary configuration is shown in Figure 1.3.

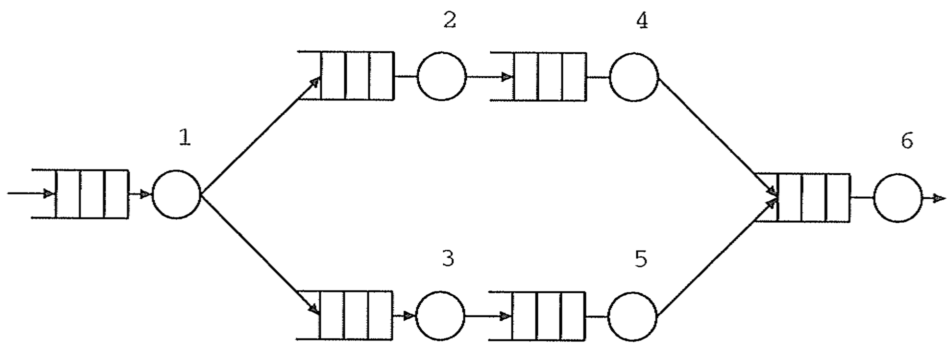

Figure 1.3: Arbitrary Configurations

Queueing networks can further be categorized according to the nature of their traffic and service processes. Thus, networks with exponential service processes and Poisson external arrivals are called exponential networks, and networks consisting of any types of queue configurations with service times and interarrival times following general distributions are called general networks.

1.1.3 Difficulties Encountered in the Analysis of Queueing Networks

Queueing networks are often very difficult to analyse, especially in those cases where a large number of nodes are involved and arrival and service processes have general distributions. To analyse a queueing network one has to consider the following:

- The blocking phenomenon. Blocking occurs when there is an interruption of the flow of items from one node to the next one as a result of the downstream buffer being full. Different processing rates at different nodes and limited storage space at intermediate buffers are the main reasons for the occurence of blocking. When a node is blocked it can not process any new items. The two most common types of blocking are: i) manufacturing type of blocking and ii) communication type of blocking. According to the manufacturing type definition of blocking, node i (Figure 1.1) gets blocked when an item that has just completed its service at node i finds the buffer of node $i+1$ full. In this case, the blocked item is forced to wait in front of node i, and node i can not process any new items. In the second type of blocking node i gets blocked when the downstream buffer becomes full. Node i is not allowed to serve new items for as long as the downstream buffer is full. The detailed differences and analysis of the different types of blocking mechanisms are well discussed by Onvural and Perros [39]. As a result of blocking we need to revise the service times in order to include possible additional delays the items may have to undergo because of blocking. The procedure of effectively revising the service times is one of the key aspects in the system modelling of queueing networks.

- The selection of appropriate interarrival and service times distributions. Very often interarrival times and/or service times of items are distributed according to empirical distributions for which the mathematical formulas that describe them are not known. To deal with this, probability distributions such as the Erlang and the Hyperexponential families of distributions are used, or for distributions with rational Laplace transforms the Coxian family of distributions can be used. In recent years, the Phase type distributions (which are the general form of the above mentioned distributions) have become very popular.

More about Phase type distributions can be found in Neuts [38]. It is obvious that inapropriate approximations of empirical distributions may lead to poor approximation of the system performance.

- The nonrenewal nature of most stochastic processes. All existing approximation methods assume that all stochastic processes (traffic and service processes) are renewal processes. This assumption is made in order to simplify the analysis. Nevertheless, often this assumption is not true, and as Patuwo et al. [40] showed, it usually results in the underestimation of the system performance measures.

1.2 Techniques used for the Analysis of Queueing Networks

There are two solution approaches used for the analysis of queueing networks. These two approaches are:

- Simulation. Simulation has been used for the analysis of queueing networks for the last three decades. However, the appearance of new simulation packages with animation capabilities together with the availability of personal computers versions made the use of simulation very popular in recent years. This is because animation (with its powerful visual effects) helped managers to understand and trust the simulation models.

- Analytical methods. These are mathematical models which use formulas or algorithms, depending on the type of application, for the estimation of the system performance measures.

Simulation is used primarily for the analysis of complicated networks for which analytical methods can not be employed satisfactorily. Although simulation is a very useful analysis tool, it is time consuming and hence is not very suitable for inclusion in optimization procedures. Furthermore, simulation models are susceptible to sampling errors. Nevertheless, as it was noted before, simulation is very suitable for the analysis of very complicated networks and also for the evaluation of approximation methods.

Analytical methods can be grouped into two classes. The first class of methods yields exact results for the system analysed. These methods work for relatively simple networks (for example two node systems) for which computer memory and computational time requirements are not very high. The first step of analysis of the exact methods, is to use an appropriate system state description which will allow the system to be solved as a markovian system. Then, if we work in discrete (continuous) time the transition probability matrix P (infinitesimal generator W) is developed. The system is solved using the known formulas $\Pi = \Pi P$ for discrete time and $\Pi W = 0$ for continuous time where Π is the steady state probability vector and $\sum_i \Pi_i = 1$. For more details the reader should refer to Kleinrock [31] and Gross and Harris [22]. The second class of methods can be further classified into two subclasses. The methods of the first subclass provide the upper and lower bounds on system performance measures. The methods of the second subclass do not give exact results but give estimates (approximations) of the performance measures of the system. Approximation methods are used in problems where exact analysis is very difficult. The approximation methods relax the assumptions for some of the system parameters and decompose the network into individual nodes or pairs of nodes.

The focus of this thesis is in the development of new and improved approximation methods.

1.3 Major Contributions of this Research

Three approximation methods for the analysis of i) real life systems ii) exponential tandem networks and iii) tandem, split, and merge networks with general arrival and service processes are presented in this thesis. The major contributions of this research can be listed as follows:

- For real life systems:
  - an approximation method for the analysis of real life systems is developed and it is applied to the modelling of a conveyor system. The method contains, as its components, existing approximations for tandem, split, merge, and single node systems.
  - this method can be used in systems with general traffic and service times and finite buffers. It provides estimates of the average sojourn time through the system and the average queue lengths behind each workstation.
  - the modelling of the conveyor system reveals the limitations and weaknesses of existing approximation methods, thus providing directions for future research.
- For a tandem system with exponential service times, Poisson external arrivals, and finite buffers:

  - a new improved approximation method is developed and it is shown that it gives improved results when compared with those obtained by other existing methods.
  - the method provides the joint queue length probability distributions for triplets of adjacent nodes information which can not be obtained by other existing methods.

- For tandem, split, and merge systems with general arrival and service processes and finite buffers:
  - a simple and quick approximation method is developed for tandem, split, and merge systems. This method is a very useful tool when used as part of optimization procedures for which execution time is very important.
  - the method is the first to present results for split, and merge configurations consisting of more than three nodes (with general arrival and service times).
  - the method has a very simple structure and therefore is very easy to use.
  - the method can be used to reduce a large number of alternatives ; then more accurate methods can be used to find the best alternative.
  - there are no limitations on the type of the probability distributions used for the traffic and service processes.

1.4 Organization of this Thesis

The rest of this thesis is organized as follows:

Chapter Two is the Literature Review. The most recent works in the area of analysis of queueing networks are briefly discussed.

Chapter Three presents an approximation method for real life systems. The method can be used to model arbitrary configurations of queues with finite buffers, traffic and service processes having general distributions, and blocking. The approximation was applied to the analysis of a part of a conveyor system installed in a manufacturing company.

Chapter Four presents an approximation method for queues in tandem. Service times and external interarrival times have exponential distributions and finite buffers are placed in between adjacent nodes. The method is based on the work of Brandwajn and Jow and it consists of an iterative scheme.

Chapter Five presents a simple, and quick approximation algorithm for the analysis of tandem, split, and merge configurations with traffic and service processes having general distributions.

Chapter Six presents the concluding remarks of this thesis, and a brief discussion of this thesis' topics. Also, possible future research possibilities are given.

CHAPTER 2

Literature Review

2.1 Introduction

The literature is quite extensive in the area of the analysis of open queueing networks. Several exact and approximation methods have been developed. Systems consisting of two nodes are solved exactly, because in most cases the state space is of reasonable size thus allowing the numerical solution of the system to be feasible. As the network becomes larger (more than two nodes in the system) approximation methods become more attractive as exact methods become numerically infeasible. In this literature review, exact and approximation methods for single server open finite queueing systems will be discussed. Emphasis will be on the approximation methods. A brief review of papers for closed queueing networks will also be presented.

2.2 Classification of Open Queueing Networks

The literature is grouped into three parts: i) the first part consists of those papers that deal with tandem networks ii) in the second part, the methods developed for split and merge configurations will be reviewed and iii) in the third part methods developed for arbitrary configurations will be reviewed.

2.2.1 Tandem Configurations

Hunt [27] was one of the first researchers to explore the effect which the buffer size has on the efficiency of a production line. A line consisting of two machines/nodes was considered and the maximum possible utilization was calculated. Service times are exponentially distributed and external arrivals are Poisson. Four cases were considered in it: i) buffers with unlimited capacities, ii) no queues allowed with the exception that the first node may have an infinite queue, iii) buffers with limited capacity placed between adjacent nodes with the exception that the first node may have infinite waiting space, and vi) no queues and no vacant facilities allowed with the exception that the first node may have an infinite queue ; the line moves all at once as a unit. The results obtained for the four cases were compared with each other. This work became the basis for research by other people in the years that followed. In this review, our attention will be focused on the works published during the last fifteen years.

Gershwin and Berman [19] considered a system that consists of two machines and one buffer with limited capacity placed between the machines. It is assumed that the first machine is never starved and the second machine is never blocked. Processing times, times to breakdowns, and repair times at both machines are exponential random variables with parameters μ_i, p_i and r_i, $i = 1, 2$, respectively. Each machine can be in one of two states: operational or under repair. The binary variable a_i denotes the status of machine i. If $a_i = 1$, machine 1 is operational and if $a_i = 0$, machine i is under repair. The state of the system is denoted by $s = (n, a_1, a_2)$ where n $(0 \leq n \leq N)$ is the number of items at the buffer and at machine 2. The states of the system are distinguished in internal and boundary states. Internal states are the states for which $1 \leq n \leq N-1$. The rest are the boundary states. A solution of the form

$$P(n, a_1, a_2) = cX^n Y_1^{a_1} Y_2^{a_2}, \quad 1 \leq n \leq N-1 \tag{2.1}$$

is proposed for the internal states. X, Y_1, and Y_2 are parameters to be determined, and have no physical meaning. It is shown that X, Y_1, Y_2 must satisfy the following three nonlinear equations:

$$\sum_{i=1}^{2}(p_i Y_i - r_i) = 0 \tag{2.2}$$

$$\mu_1(\frac{1}{X} - 1) = (p_1 Y_1 - r_1)(1 + \frac{1}{Y_1}) \tag{2.3}$$

$$\mu_2(X - 1) = (p_2 Y_2 - r_2)(1 + \frac{1}{Y_2}) \tag{2.4}$$

Equations (2.2)-(2.4) lead to a fourth degree polynomial in Y_1. This results in four triples (X_j, Y_{1j}, Y_{2j}), $j = 1, 2, 3, 4$. Thus (2.1) can be written as :

$$P(n, a_1, a_2) = \sum_{j=1}^{4} c_j X_j^n Y_{1j}^{a_1} Y_{2j}^{a_2}, \quad 1 \leq n \leq N-1 \tag{2.5}$$

where $c_j, j = 1, 2, 3, 4$ are computed from the boundary equations.

Altiok and Stidham [7] solved a problem similar to the above but they considered a line with more than two machines. The assumptions are the same as the ones in [19]. Thus, service times, repair and failure times are exponentially distributed for all machines. Buffers with limited capacity are placed between machines and it is assumed that the first server is always busy. The effective service time of node i is expressed in terms of its Laplace transform. It is shown that the Laplace transform has a rational form. In order to transform the system to a Markovian one, the Coxian-2 distribution is used to express the effective service times of each machine. It is known that the Coxian distribution is used for the representation of general distributions with rational Laplace transform. The states of the system are defined using a vector of the form

$$\boldsymbol{x} = (k_1, s_1; k_2, n_2, s_2; k_3, n_3, s_3; ...; k_M, n_M)$$

where

$$k_i = \begin{cases} 0 & \text{if the } i\text{th station is idle} \\ 1 & \text{if the server is in fictitious stage 1} \\ 2 & \text{if the server is in fictitious stage 2 } (i = 2, ..., M) \end{cases}$$

$$s_i = \begin{cases} 1 & \text{if the station is blocked} \\ 0 & \text{otherwise } (i = 1, ..., M-1) \end{cases}$$

n_i is the number of items at machine i (both in the buffer and in service) $i = 1, ..., M$. The solution is straightforward. The infinitesimal generator matrix Q is constructed and the equations $Q^T\boldsymbol{p} = 0$, $\boldsymbol{ep} = 1$ (where $\boldsymbol{p}$, and $\boldsymbol{e}$ are the steady state probability column vector, and a row vector with all its elements equal to 1, respectively) needs to be solved. This method cannot be applied to long lines with many nodes because the number of states becomes so large that the construction of the generator matrix becomes cumbersome.

Gershwin and Schick [20] obtained exact solutions for queueing networks with queues in tandem and blocking. Two networks were considered: i) a two node system and ii) a three node system. Their method can be applied to the analysis of longer lines but the great dimensionality of the resulting Markov chain limits the applicability of their method. The systems considered consist of machines in tandem separated by buffers with limited capacity. The following assumptions are made: The first server always has raw material available for processing, processing times are constant and equal for all machines, machines fail only when they operate on an item, and times between succesive failures and repairs are geometrically distributed. The state of the system at time t is defined as:

$$s(t) = (n_1(t), ..., n_{k-1}(t), \alpha_1(t), ..., \alpha_k(t))$$

where n_{k-1} is the number of items at buffer $k-1$, α_k is the state of machine k ($\alpha_k = 0, 1$ if machine k is under repair or operational, respectively), and t is the time in machine cycles. The proposed method decreases the number of linear equations that have to be solved in order to calculate the steady state probability distribution of the system. It is shown that it is possible to find l vectors that satisfy at least $M-l$ transition equations (M is the number of the system's states). The real contribution of this method is in the reduction of the computations required for the solution of a system with M linear equations. The solution analysis is conceptually the same one used in Gershwin and Berman [19]. The system's transitions equations are grouped into internal and boundary equations and a solution in product form is found that satisfies the internal equations. Then, with the use of the boundary equations an easy and efficient way is found for calculating the steady state probability vector. The drawback of this method is that the number of computations depends on the dimensionality of the system. Thus, for long lines and large buffers the method is inefficient. However, the solutions of the three and two nodes networks can be used as building blocks for other approximation techniques.

Altiok [2] considered production lines with Phase-type processing and repair times, and finite buffers. Machines are in tandem and the time until a breakdown occurs is exponentially distributed. It is assumed that the first server always has items available for processing. The system is defined using a rather complicated description and the steady state probabilities are obtained solving the equations $Q^T\boldsymbol{p} = 0$, $\boldsymbol{ep} = 1$ (where Q, $\boldsymbol{p}$, and $\boldsymbol{e}$ are the generator matrix, the steady state probability column vector, and a row vector with all its elements equal to 1, respectively). The drawback of this approach is that it cannot be used for long lines because the number of states becomes very large. The method was applied to a two node, one buffer production system.

All approximation methods can be classified into two categories. In the first category, the system is decomposed into individual nodes. The arrival and service processes of individual nodes are modified in order to include the influence of the other parts of the network on the analysed node. Then, information obtained from the analysis of the individual node is used in the analysis of the rest of the network.

Altiok [1] developed an approximation method for the analysis of an open network that consists of M servers in tandem. Buffers with limited capacity are placed between adjacent nodes. Service processes have exponential distributions and external arrivals occur at the first node and have Poisson distribution. Service times are revised to include the effect of blocking on the time that an item spends at each node. Each service process is approximated by a Coxian-2 distribution. An item at node i first receives exponential service with rate μ_i, and then with probability a_{i+1} it receives an additional exponential service with parameter μ_{i+1}. We define a_{i+1} as the probability that upon service completion at node i, the queue of node $i+1$

is full. Two assumptions are made: i) the input process to each queue is a Poisson process and ii) a queue may get blocked only by the immediate successor queue. The network is decomposed into $M/C_2/1/N$ queues and the individual queues are solved using the embedded Markov chain approach.

Perros and Altiok [41] considered a tandem queueing network where service times at all queues are exponentially distributed and external arrivals occur at the first node according to a Poisson distribution. Because of the limited buffer space between succesive nodes, blocking occurs. Node i may get blocked by any node j, $i < j \leq M$ (M is the last node). In this case, all nodes k ($i < k < j$) are blocked. This assumption is an extension to Altiok [1] where it was assumed that node i can get blocked only by its immediate successor node $i + 1$. An item that starts its service at node i will first receive an exponential service with parameter μ_i. Then with probability $1 - a_{i0}$ it will receive an additional delay depending on the status of the downstream nodes. Let us assume that at the moment of service completion at node i , nodes $i + 1$ to $i + k$ are blocked while node $i + k + 1$ is serving. The additional delay that the blocked item will undergo is the residual service time of the $i+k+1$ node plus an additional delay in case that this node upon service completion will get blocked. Thus node i is modelled as an $M/C_{M-i+1}/1/N_i + 1$ queue. The proposed algorithm starts analysing the last node of the tandem network as an $M/M/1/N_M + 1$ queue. Using information obtained from the analysis of the Mth node the algorithm proceeds to the analysis of node $M - 1$. The analysis proceeds backwards and stops when it reaches node 1.

Altiok [3] considered a tandem queueing network with phase type service times and blocking. The first server may always be busy or customers may arrive according to a Poisson arrival process. The network is decomposed into individual queues with revised service times but the assumption that the arrival process at each node is a Poisson process is made. Assume that the service time Y at node i has a phase type distribution characterized by the pair (α, S) with K phases. If node i gets blocked then the blocked item at node i will have to wait for the remaining service time of node $i+1$ which is of the phase type. Let us denote by V the period of time that node i is blocked. Then, the distribution of the variable V is characterized by a phase type distribution described by the pair (β, B) with order L. Thus, the effective service time at node i denoted by U is given by:

$$U = \begin{cases} Y & \text{with probability } 1-\pi \\ Y+V & \text{with probability } \pi \end{cases}$$

where π is the probability that upon service completion at node i the buffer of node $i+1$ is full. Each queue is analysed in isolation using the matrix-geometric method of Neuts. The algorithm starts with the last node and proceeds backwards. This process is repeated several times until it adequately approximates the throughput of the system. The steady state probability distribution of the number of items at each node is computed.

Another approximation for tandem queueing network with blocking, was developed by Jun and Perros [29]. It is assumed that all service processes have Coxian-2 distributions and the arrival processes at each queue, except the first one, is a Coxian-2 distribution. Buffers with limited capacity are placed between adjacent nodes. The approximation algorithm revises the service and arrival processes at each node in order to accomodate the effects of blocking. Then each queue is analysed in isolation as a $C_2/C_2/1/N$ queue. For the analysis of the single $C_2/C_2/1/N$ queue the iterative procedure of Yao and Buzacott [55] is used. It was shown that it gives significantly better results when compared to results obtained by Altiok's [3] approximation. Cases with finite or infinite first nodes were considered.

In the second category of approximation methods, the analysis considers cells that consist of pairs of queues with revised arrival and service processes.

Gershwin [16] extended the work of Gershwin and Schick [20] to include systems consisting of more than three nodes in series. The assumptions about the modelled networks characteristics are exactly the same as the ones considered in Gershwin and Schick [20]. Consider a line L that consists of k machines in series and $k-1$ buffers (each buffer is placed between a pair of machines). The line L is decomposed into $k-1$ sublines consisting of two machines each. Denote by $L(i)$ the two node line that contains a buffer B_i which has the same capacity with the buffer B_i in the original line L. Both machines in the two node lines have a geometric working time distribution and their repair times are also assumed to be geometrically distributed. Machine U_i (D_i) matches the line upstream (downstream) the buffer i in the original line L. The rate of flow into the buffer B_i in the $L(i)$ approximates that of buffer B_i in line L. The probability of resumption of flow into and out of buffer B_i in line $L(i)$ in a time unit after the period during which it was interrupted is close to the probability of the corresponding event in L. Let us denote by $p_u(i)$, and $p_d(i)$ the parameters of the working time distribution for machines U_i and D_i respectively. Also, let $r_u(i)$ and $r_d(i)$ be the parameters of the repair times for machines U_i and

D_i, respectively.

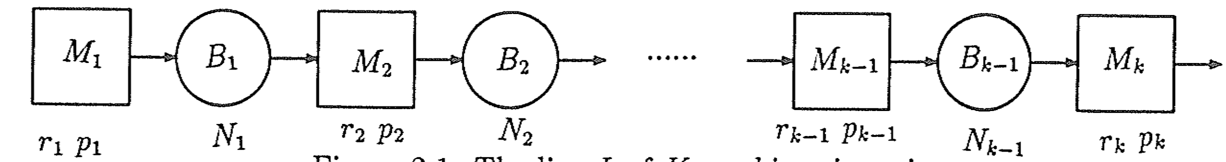

Figure 2.1: The line L of K machines in series

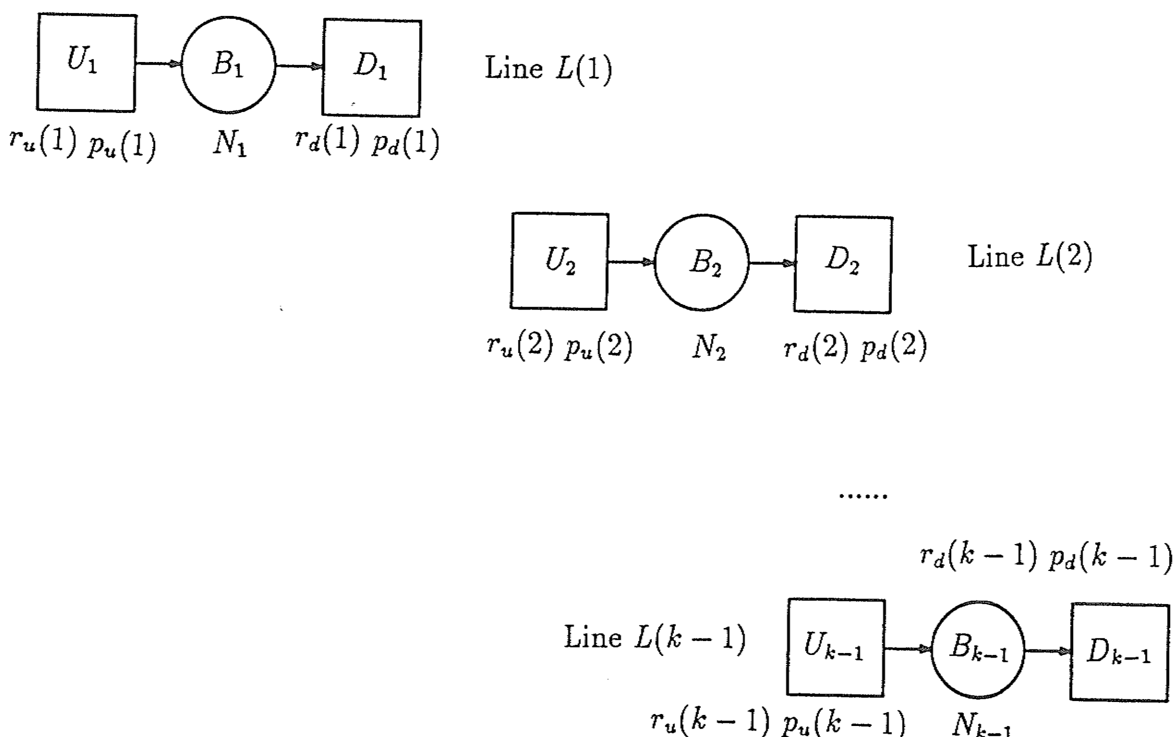

Figure 2.2: The decomposed line L

Then, a system of equations relates these parameters with the corresponding parameters of neighboring sublines and with the corresponding parameters in the original line L. A total of $4(k-1)$ equations in $4(k-1)$ unknowns $r_u(i), r_d(i), p_u(i), p_d(i)$ for $i = 1, 2, ..., k$ describe the system. The system is solved using an iterative procedure. The two node sublines are evaluated using Gershwin and Schick's [20] method.

The algorithm is relatively complicated and in some cases does not converge. However, for the cases reported, it gave good results.

Choong and Gershwin [13] extended the work of Gershwin [16] to include random processing times. It is assumed that processing times, failure, and repair times have exponential distributions. In this work, machines may have different processing times as well as different mean times to repair and failure. The analysis is similar to the one in Gershwin [16]. The two node sublines are solved using the method of Gershwin and Berman [19]. The algorithm yields good results although it does not always converge.

Dallery, David, and Xie [14] simplified the algorithm proposed by Gershwin [16] by replacing some of the original equations with equivalent ones. The new algorithm is much simpler and faster than Gershwin's, and converges for all the cases considered. From the experiments performed it was found that the new algorithm in some cases was ten times faster than the one proposed by Gershwin.

Brandwajn and Jow [11] considered a tandem queueing network where all service processes are exponentially distributed. Two cases were considered: i) arrivals occur at the first node and are Poisson processes and ii) the first server is assumed to always have raw material available for processing (i.e the first server is never idle). Buffers are placed between adjacent nodes and have limited capacities. Service rates are allowed to be state dependent. The approximation method considers subsystems of two adjacent nodes and it computes equivalent arrival and departure rates for each subsystem. The system is modelled in continuous time and it is defined by the number of items at each queue. The approximation method starts with the development of the balance equations of the system, and then the balance equations are modified using conditional probabilities for nodes i and $i+1$. The resulting balance equations are considered to be the balance equations of an equivalent system that consists of the two finite queues i and $i+1$. Applying this procedure the line is decomposed into subsystems consisting of pairs of nodes with revised Poisson arrival processes and revised exponential service times for the second nodes of each subsystem. The algorithm starts with the first pair of nodes and it proceeds torwards the last pair. The steady state probabilities for each subsystem are numerically calculated. Then, the analysis proceeds to the next pair of nodes and, using information obtained from the previously analysed pair, the new pair of nodes is solved. The algorithm stops when a convergence criterion is satisfied. The method gives good results mainly because it gives a good representation of the blocking within each pair of nodes. For a tandem network of N nodes, $N-1$ solutions of the two nodes subsystems have to be solved at each iteration. The approximation method was compared with the method developed by Perros and Altiok [41] and it was shown that it gives more accurate results.

Altiok and Ranjan [6] developed an approximation method for a queueing network with queues in tandem using a two-node decomposition approach. Service times have phase-type distributions and buffers between adjacent nodes have limited capacity. The first server is assumed to have an unlimited supply of raw material. The solution approach is very similar to the one in [16]. It was shown that the two node decomposition approach gives better estimates of the system's performance measures (for individual nodes) when compared to estimates obtained by Altiok [3].

Gun and Makowski [24] developed an approximation method for the evaluation of the performance measures of a tandem queueing network with blocking. Service times are allowed to be of the Phase-type and the first server may always be busy or it can be allowed to generate arrivals according to a Phase-type distribution. The blocking mechanism considered here is of the communication type, that is the node i gets blocked at a time of service completion if the buffer of the successive node $i+1$ becomes full as a result of this service completion. The analysis of the system is carried out in discrete time and a two node decomposition approach is adopted. A line that consists of N nodes, is decomposed into $N-1$ two node subsystems. Each subsystem consists of a buffer having the same capacity with the corresponding buffer in the original line, and an upstream and a downstream node. The upstream and downstream nodes have revised service times in order to incorporate the effects of idling and blocking, respectively. To illustrate the method consider node $i+1$ at service completion time. Node $i+1$ is either blocked with probability W^{i+1} or with probability $1-W^{i+1}$ it starts service on a new item. If node $i+1$ is blocked, assume that node k $(i+1 < k < j)$ is blocked and node j is serving and is in phase l $(1 \leq l \leq m_j)$ where m_j is the number of transient phases of the Phase-type distribution of the service time of node j. Thus, the effective service time of node $i+1$ is the remaining service time of node j and the sum of service completions at nodes k. Therefore, if node $i+1$ is blocked, then its service time is initialized at phase l with probability $W^{i+1,j}$ (where $W^{i+1,j}$ is the probability that all nodes k with $i+1 \leq k < j$ are blocked) and transitions occur in the set S_j (where S_j is the set of states of the Phase service distribution for node j) with matrix Q_j (where the matrix Q_j is used for the representation of the Phase type service of node j). Upon service completion (at node j), node $(j-1)$ initializes service according to α_{j-1}, thus causing a transition from the set S_j to the set S_{j-1} in the representation of the $i+1$ node. This procedure is repeated moving upstream from node j to node $i+1$. Hence, the effective service time distribution of the second node of the two nodes subsystem can be represented by the pair $(\alpha_2(i+1), Q_2(i+1))$ where

$$Q_2(i+1) = \begin{pmatrix} Q_{i+1} & & & & \\ p_{i+2}\alpha_{i+1} & Q_{i+2} & & & \\ & . & . & & \\ & & . & . & \\ & & & p_N\alpha_{N-1} & Q_N \end{pmatrix}, \quad p_2(i+1) = \begin{pmatrix} p_{i+1} \\ 0^T_{m_{i+2}} \\ . \\ . \\ 0^T_{m_N} \end{pmatrix}.$$

where $p_2(i+1)$ is the column vector of absorption probabilities and $\alpha_2(i+1)$ is given by:

$$\alpha_2(i+1) = ((1 - W^{i+1})\alpha_{i+1}, W^{i+1,i+2}, ..., W^{i+1,N})$$

Employing a similar analysis, the effective service time of the first node is derived. The system is solved using an iterative method. For the analysis of each pair of nodes the two node analysis of Gun and Makowski [25] is used. Information obtained for a pair of nodes is used for the analysis of the next one. The algorithm stops when a convergence criterion is satisfied. Results obtained from this method were compared with results obtained by the method of Jun and Perros [29]. The two methods seem to be of about the same accurate but Jun and Perros's algorithm is 2-3 times faster than of Gun and Makowski.

Another approximation is that of Gershwin [18]. Gershwin applied Dallery, David and Xie's [14] modifications to the work of Choong and Gershwin [13] and this resulted in a faster and simpler algorithm.

2.2.2 Split and Merge Configurations

Very few papers have appeared dealing with the modelling of split and merge configurations. Boxma and Konheim [9] were the first to develop an approximation method for the analysis of such systems. This method works for two-node tandem systems, split systems with two second level nodes and for merge systems with two first level nodes. Service times and external interarrival times have exponential distributions. Buffers are finite. The authors produced results for a large number of these three types of networks. An extension of this method was proposed for the analysis of arbitrary configurations of queues. However, results were provided only for a three-node tandem system.

The next step in the analysis of split and merge configurations is to consider networks that consist of more than three nodes. The first paper was published by Altiok and Perros [4]. All service times are exponentially distributed and external arrivals occur according to a Poisson fashion. The idea is to find an efficient way to replace the actual service times with effective service times in order to accomodate the delays that the items might undergo due to blocking. Then, the system is decomposed into individual queues and each queue is studied in isolation. Let us first consider the split configuration. Items that have completed service at the first level node join the second level queue i with probabilities q_i. If the buffer of the second level queue i is full at that time the item has to wait in front of the first level node. It is assumed that each second level node cannot get blocked. The node is thus viewed as a $M/M/1/N_i+1$ queue. The first level queue is modelled as a $M/PH_k/1$ queue, and the effective service time has a Phase type distribution. Consider now the merge configuration. Two or more queues merge into one queue. When an item completes its service at the ith first level queue it may have to undergo an additional delay due to the fact that the buffer of the second level queue is full. The duration of the additional delay depends on how many other items are already blocked. Thus, a blocking line is formed consisting of all the blocked items. A FIFO (first come, first served) release rule is assumed for the blocking line. The second level queue does not get blocked and is modelled as a $M/M/1/N+n$ queue. Each first level queue i can be viewed as a $M/PH_{k_i}/1$ queue. For the analysis of the $M/PH_{k_i}/1/1$ queue the matrix-geometric method of Neuts is employed.

Lee and Pollock [34] developed an algorithm for the analysis of the same merge configurations considered by Altiok and Perros [4]. Lee and Pollock's algorithm approximates the steady state probabilities for each queue of the system. The method decomposes the network into individual queues which are then analysed as $M/M/1/N$ or $M/G/1/N$ queues in isolation. This method is similar to the one developed by Altiok and Perros but differs in that it describes the state of the merged queue by considering the sequence (rather than only the number) of blocked units. This algorithm seems to give better results when compared with the results obtained by Altiok and Perros.

Kerbache and Smith [30] presented an approximation method that can be used for the approximation of tandem, split, and merge queueing networks. The method is called Generalized Expansion Method and it is an extension of the Expansion Method which was developed for exponential networks. The GEM is same in principle as the approximation methods proposed by Kuehn [32] and Labetoulle and Pujolle [33]. The GEM is based on the assumptions that all processes are renewal processes and it uses the first two moments for the description of probability distributions. The basic idea of the GEM is to place an artificial node between adjacent nodes i and j to collect all items that upon their service completion at node i find the buffer of node j full. The GEM consists of three major stages ; network reconfiguration, parameter estimation, and feedback elimination.

1. Network reconfiguration. For each node j with limited waiting space, an artificial node of the type $GI/G/\infty$ is placed between nodes i and j to collect all blocked items. A blocked item will receive service at the artificial node and then it will try to join the queue of node j. If the buffer of node j is still full, the item will receive an additional service at the artificial node.

2. Parameter estimation. At this stage all parameters are evaluated using existing techniques. The service time at the artificial node is taken to be the remaining service time of node j. For the approximation of performance measures of single queues existing methods for general queues are used. That is, if node j is of the $G/G/1/N$ type, Labetoulle and Pujolle's [33] or Yao and Buzacott's [54] methods can be used for the estimation of single queue parameters such as probability that the queue is full. Also, for the approximation of the squared coefficient of variation of the departure process from nodes of the $G/G/1$ type, Marshall's [37] formula can be used.

3. Feedback elimination. Because of the repeated visits (feedback) to the artificial node, there is strong dependence in the arrival processes. In order to eliminate these dependencies, it is assumed that each item receives all its service during its first passage. Thus, the service rate and the squared coefficient of variation are revised.

A system of simultaneous nonlinear equations needs to be solved for the determination of the systems parameters. The GEM was tested for the three node split, three node merge, and two node tandem configurations with Erlang-2 (E_2) and Hyperexponential-2 (H_2) service times and Poisson external arrivals. It gives good approximations for as long as the traffic intensity is less than 0.70. The results obtained are in the form of the average sojourn time in the system.

2.2.3 Arbitrary Configurations

Kuehn [32] considered open queueing networks with general service and arrival times, infinite buffers, and feedback. All processes (service and arrival) are considered to be renewal processes and are represented by their first two moments. Kuehn provides formulas for the decomposition and superposition of renewal processes and approximates the mean waiting time at each queue. The arrival process at each queue is revised in order to accomodate the feedback problem. Finally, the expected network flow times are calculated.

Labetoulle and Pujolle [33] developed the isolation method for the analysis of general open queueing networks with finite buffers. As was the case in Kuehn [32], they considered only the first two moments of each general process and assumed that all processes are renewal processes. The arrival and service processes were revised in order to accommodate the phenomenon of blocking. Then, the network was decomposed into individual queues and each queue was analysed in isolation. The information necessary for the analysis of the individual queue i can be obtained only through the analysis of the other queues that are connected to queue i. An iterative scheme is developed for the calculation of the parameters of the system.

Their method was tested for a computer network with a particular structure and it was found that it gives good results.

Takahashi, Miyahara, and Hasegawa [47] considered an open queueing network with exponential servers, Poisson arrivals and finite buffers. The system was decomposed into individual queues with revised arrival and service processes. Each individual queue was analysed as a $M/M/1$ queue. They assumed that the revised service process at each queue is still exponentially distributed and that the revised arrival process to each node is a Poisson process. The desired parameters of the system are obtained from the solution of a system of simultaneous equations. The method seems to yield good approximations of quantities such as blocking probabilities and output rates. The method was tested for a network consisting of three nodes and for a tandem network of four nodes. The amount of calculations increases only linearly with the number of nodes.

Altiok and Perros [5] studied two arbitrary configurations of exponential queueing networks with blocking. The first configuration is of the triangle type and the second configuration consists of four nodes. The service times at all nodes for both configurations are exponentially distributed and the external arrivals occur only at one node of each configuration and the arrivals occur according to a Poisson distribution. Buffers with limited capacities are placed in between nodes, but the node which receives external arrivals can have buffer with limited or unlimited size. The proposed algorithm decomposes the network into individual queues, which are then analysed in isolation as $M/PH/1/K$ queues. The service times are considered to be of the phase-type in order to accomodate the effects of blocking. This method was shown to give good results but its extension to networks with a large number of nodes is a difficult task. This is because the phase-type mechanisms become very large and thus the repeated analysis of $M/PH/1/K$ queues may require a lot of computational time. The algorithm gives the marginal queue length distributions of individual nodes.

Perros and Snyder [42] presented a computationally efficient version of the Altiok and Perros [5] method. Perros and Snyder considered the same arbitrary configuration that Altiok and Perros had examined. The difference between this new algorithm and the previous one is that the individual nodes (after the network has been decomposed) are not treated as $M/PH/1/K$ queues but rather as $M/C_2/1/K$ queues. More specifically, once all the parameters of the phase type distributions are known, the phase type distributions are collapsed to two phase Coxian distributions with parameters determined by the method of moments. It was shown that this method had the same accuracy with the method of Altiok and Perros but it was much faster and now larger networks could be analysed.

Another paper in this area of research is that of Jun and Perros [28]. Jun and Perros considered the system of Altiok and Perros [5] but with deadlock. Deadlock may occur as a result of the arbitrary interconnection of nodes. Suppose node i is blocked by node j, and an item in node j, upon service completion, chooses to go to node i. If node i is full at that time, a deadlock occurs. It is assumed that a deadlock is resolved by simultaneously exchanging blocking units between nodes i and j. The algorithm developed decomposes the network into individual queues with revised service processes. The service processes are characterized by phase type distributions whose parameters are obtained iteratively. Results were obtained for two networks consisting of three and five nodes respectively. The results were shown to be good. However, this method requires the construction of very detailed phase type mechanisms which is time consuming.

Gershwin [17] analysed tree-structured assembly/disassembly networks with finite buffers and unreliable machines. This method is an extension of the transfer line algorithm of Gershwin [16]. Machines are assumed to spend a random amount of time processing each item due to the failures and repairs of machines. An approximate decomposition method for the evaluation of the performance measures of the system is developed and it is shown that gives good results.

Whitt [48] developed a software package called QNA (Queueing Network Analyzer) to analyse open queueing networks with multiserver nodes with the first-come first-served discipline and no capacity constraints. Service and external interarrival times can have general distributions. All stochastic processes are assumed to be renewal processes. The analysis approach uses only two parameters to characterize the arrival and service processes. These two parameters describe the rate and the variability of each of the processes. The nodes are analysed as $GI/G/m$ queues each, and approximations of the congestion measures of the system are obtained. QNA is a very useful tool for the analysis of queueing networks and it was shown that it gives good results. However, its applications are limited to the networks with buffers with unlimited capacities.

Pourbabai and Sonderman [45] developed an approximation method for the analysis of a stochastic recirculation system with randomly accessed multiple heterogeneous servers. The items are assigned to one of the servers upon arrival randomly. All service times are exponentially distributed and there is no waiting space. Items which find all servers busy, recirculate into the system and form an overflow process.

The overflow processes combine with the external arrival process to form the new input into the system. Two parameters are used to represent traffic processes: the mean and the squared coefficient of variation (scv). Each workstation is treated as a $GI/M/1/1$ queueing system and the algorithm provides estimates of the rates and scvs of all traffic processes of the system.

Pourbabai [43] considered a system similar to the previous conveyor system. Incoming items enter the system and travel towards the first workstation. If the workstation is not full, items receive service and then depart from the system. In case the workstation is full, the items bypass it and arrive at the second workstation. If the items find the second workstation full, they bypass it and merge with the external arrival process to form the new input process to the system. The model can be generalized to include N workstations. Each workstation is modelled as a $GI/M/S/K$ queue and two performance measures are calculated: the efficiency of each workstation and the average congestion along each conveyor.

The same analysis approach was applied by Pourbabai [44] to analyse a slightly different conveyor system. The system consists of L workstations. Items are first going towards the $(i)th$ workstation. If the $(i)th$ workstation is full, then the item is routed towards the $(i+1)st$ workstation, for $i = 1$ to $L-1$. It is assumed that the $(L)th$ workstation has a very large buffer to accomodate all items that were not processed at the other $L-1$ workstations. Each workstation is modelled as a $GI/M/N_i/K_i$ queue. Approximations of the server utilization of each workstation and the average congestion along every conveyor of the material handling system are provided.

2.3 Closed Queueing Networks

Closed queueing networks are networks in which a fixed and finite number of customers are considered to be in the system. No customers are allowed to enter into the network and none of the customers inside is allowed to leave the network. Closed loop material handling systems (where transportation carriers, rather than products/items, are viewed as customers), and systems of operating machines-repair facilities are two examples of closed queueing networks applications.

Gordon and Newell [21] derived the equilibrium distribution of customers in a closed queueing network with exponential servers and infinite buffers. Let us consider a network with M nodes and N circulating customers. The state of this network can be described by the vector $\boldsymbol{n} = (n_1, n_2, ..., n_M)$ where n_i is the number of customers at the ith node and $\sum_{i=1}^{M} n_i = N$. Denote by μ_i, and p_{ij} the service rate at node i and the probability a customer will go to the jth node after completes its service at node i.

The equilibrium distribution of customers in the network is given as:

$$p(n_1, n_2, ..., n_M) = \frac{1}{G(N)} \Pi_{i=1}^{M} (X_i)^{n_i} \tag{2.6}$$

where $(X_1, ..., X_M)$ is a real positive solution to the equations

$$\mu_j X_j = \sum_{i=1}^{M} \mu_i X_i p_{ij}, \quad 1 \leq j \leq M \tag{2.7}$$

and $G(N)$ is a normalizing constant given as

$$G(N) = \sum_{\boldsymbol{n} \in S(N,M)} \Pi_{i=1}^{M} (X_i)^{n_i} \tag{2.8}$$

where

$$S(N, M) = \{(n_1, n_2, ..., n_M) | \sum_{i=1}^{M} n_i = N \text{ and } n_i \geq 0 \ \forall\, i\} \tag{2.9}$$

The development of an efficient computational algorithm by Buzen [12] for the calculation of $G(N)$ made Gordon and Newell's method easy to implement. Baskett, Chandy, Muntz, and Palacios [8] extended Gordon and Newell's work to include closed networks with different classes of customers. They assumed service times having probability distributions with rational Laplace transforms. They derived the equilibrium distribution of states of a model which consists of four different types of service centers and R different classes of customers.

Gordon and Newell's result was used in the work of other researchers. Two of theses works are those of Gross, Miller, and Soland [23] and Madu [35]. Gross, Miller, and Soland studied a multi-echelon system in which a finite number of items is desired to be operational at any given time, and in which queueing may occur at the repair facilities when all channels -finite in number- are busy. Their aim was to determine the optimal spares levels and repair capacities of this system.

Madu used Gordon and Newell's formula and Buzen's algorithm to analyse a maintenance network with loaded-independent servers. The problem is to determine the maintenance float needed to support an operating system with N circulating units. When a unit fails, it is sent into the service facility for repair. The failure and repair times distributions are assumed to be exponentially distributed. This closed system is a special case of a Jacksonian (an exponential network with no buffer capacity constraints) network. The number of units to maintain in standby status in order to maximize the system availability is easily obtained using Gordon and Newell's formula.

2.4 Conclusions

Open queueing networks have been analysed by a variety of approximation algorithms over the past fifteen years. Most of these approximations are specialised to work only for specific queueing configurations and cannot be used to solve other different systems. From the above discussion (Sections 2.2.1, 2.2.2, and 2.2.3) it can be seen that the majority of the existing methods work for tandem queueing networks. This is partly, because production lines are modelled as tandem networks and partly because it is not very difficult to analyse tandem configurations. Although, there are many algorithms designed to work for tandem systems, still there is room for improvement. The improvement in these methods can come from two directions: i) to develop methods which are as accurate as the best of the existing ones but are faster and ii) to develop algorithms which give more accurate results when compared to results obtained by existing methods.

The progress in the research for the analysis of split and merge systems, as it was discussed in Section 2.2.2, is far behind when compared with the number of papers appeared that deal with tandem systems. Approximation methods for split and merge configurations with general arrival and service processes and consisting of more than three nodes have not received much attention. We must also add that for exponential split and merge configurations, only three papers ([9], [4], [34]) have been published. Therefore, future research in this area should be focused first on the improvement of the existing methods for exponential networks and secondly in the development of approximations for networks with general arrival and service processes.

Most of the algorithms that have been designed to analyse arbitrary configurations work only for these specific configurations and their generalization to include a wider variety of systems is difficult. Whitt's QNA is a method that can be used for the analysis of any type of network configurations but it requires buffers with unlimited storage space. Thus, there is the need for the development of methods which are designed to work for any type of queueing networks without any capacity or other constraints.

As was mentioned earlier, exact analysis is very difficult for systems with very large state spaces because of the large amount of required computational effort. On the other hand, approximation methods are much more simpler and faster. Nevertheless, there are cases where approximation methods require a considerable amount of computer memory and computational time. The use of phase type distributions to represent the "effective" service times, although it helps for the detailed description of the effects of blocking, it is time consuming. This problem becomes more obvious when optimization is involved. Thus, there are times, the engineer would like to have in his use simple and quick approximation methods. These methods could be used in optimization procedures and in cases where it is desirable to reduce a large variety of alternatives before more accurate methods could be used.

CHAPTER 3

An Approximation Method for Real Life Systems

3.1 Introduction

Most real life systems are arbitrary configurations of queues with general arrival and service processes and finite buffers. As was pointed out in Chapter 2, such systems have not received much attention in the literature. Arbitrary configurations have been studied but the methods developed are designed to work for the specific systems studied, rather than for any general system. Thus, in this chapter, our effort is focused on the development of an approximation method which can be applied to any real life system independent of the configuration. The proposed method is designed to work for systems with general arrival and service processes and finite buffers. The approximation method is applied to the modelling of a part of a conveyor system and the results obtained are compared to simulation results. It is shown that the approximation performs well for light and moderate traffic. The algorithm gives estimates of performance measures of the system such as average sojourn time through the network and average queue lengths at each workstation.

3.2 The Approximation Method

3.2.1 Model Assumptions

Let us consider any arbitrary configuration of queues. Service times and external interarrival times have general distributions, and buffers with finite capacities are placed behind each node. Buffer capacity constraints lead to blocking. Blocking occurs when an item that has just completed its service at node i tries to join node j and finds the buffer of node j full. The item is forced to wait at node i until space becomes available at node j. Meanwhile node i can not process any new items.

3.2.2 The Basic Concepts of the Approximation Method

As was mentioned in Section 3.1 most of the existing methods work for specific systems (for example tandem, split, merge, exponential etc) and their generalization to include other systems is very difficult. The approximation method presented in this chapter deals with this problem. This method can be used to analyse any type of system. The method is conceptually very simple. We must find a way to transform the system studied into a new equivalent system which will be easier to analyse. This new equivalent system will not have buffer capacity constraints, and the service times of all nodes will be revised in order to include additional delays the items may have to undergo due to blocking. Hence, the proposed approximation consists of two passes. In the first pass the effective rates and effective variances of the service times of all nodes are derived. In the second pass the effective rates and effective variances are used as rates and variances of the service times of all nodes and then we assume the network is free of blocking. Finally, using a single node approximation method and Little's result we estimate the performance measures of the entire system.

3.2.3 The First Pass

In the first pass we have to use an approximation method to analyse tandem, split, and merge systems. Kerbache and Smith's [30] Generalized Expansion Method (GEM) is the only existing method that can be used to model tandem, split, and merge systems with general stochastic processes.

The First Pass consists of two Steps.

- **Step 1.** The GEM is first applied to the last node(s) of the system. To do this we must first approximate the arrival process to the last node(s) of the system. It is assumed that all stochastic processes are renewal processes, and two parameters (mean and scv) are used to represent probability distributions. The following formulas (Whitt [48]) give the rate and scv of the traffic process which results from the superposition of renewal traffic processes. The rate λ_j of the arrival process to node j is given as:

$$\lambda_j = \lambda_{0j} + \sum_{i=1}^{n} \lambda_i q_{ij} \tag{3.1}$$

where λ_{0j} is the total external arrival rate to node j, q_{ij} is the routing probability from node i to node j, n is the number of streams that merge, and λ_i is the rate of the ith merging stream. The scv c_j^2 of the process that results from the superposition of n streams is given as:

$$c_j^2 = \sum_{i=1}^{n} (\lambda_i / \sum_{k=1}^{n} \lambda_k) c_i^2 \tag{3.2}$$

where λ_i, and c_i^2 are the rate and scv of the ith merging stream, respectively. If a traffic process with parameters λ and c^2 splits into k streams, with each being selected independently according to probabilities p_i, then the ith stream obtained from the splitting (Whitt [48]) has rate λ_i and scv c_i^2 given by

$$\lambda_i = \lambda p_i \tag{3.3}$$

$$c_i^2 = p_i c^2 + 1 - p_i \tag{3.4}$$

We also need to approximate the scv of the departure process from node i. Marshall's [37] formula gives the scv of the interdeparture time c_d^2 in a $GI/G/1$ queue:

$$c_d^2 = c_a^2 + 2\rho^2 c_s^2 - 2\rho(1-\rho)\mu EW \tag{3.5}$$

where ρ is the traffic intensity of node i, c_s^2 is the scv of service times of node i, c_a^2 is the scv of the arrival process at node i, μ is the service rate of node i, and EW is the mean waiting time in the queue of node i.

In order to approximate the arrival process to the last node(s) of the system, we do not revise the service times at each of the nodes of the system. We use formulas (3.1)-(3.4) to approximate the processes that result from the merging and splitting of traffic processes. To calculate the average waiting time in the queue (which is needed in Marshall's formula) we use a $G/G/1/N$ single node approximation method. We use Marshall's formula to approximate the scv of the departure process from $GI/G/1/N$ queues. The rate λ_d of the departure process from node i is given as

$$\lambda_d = \lambda_a(1-P) \tag{3.6}$$

where λ_a is the rate of the arrival process at node i, and P is the probability that node i is full.

- **Step 2.** Having approximated the arrival processes of the last node(s) of the system, we apply the GEM to calculate the effective service times of the nodes which are linked to the last nodes of the system. It is assumed that the last nodes of the system cannot get blocked. Using now the effective service times as service times for the nodes that were already analysed, the method applies the GEM to the nodes that are linked to the already analysed nodes. By the end of the first pass the effective service times of all nodes have been calculated.

3.2.4 The Second Pass

In the second pass, it is assumed the network is free of blocking. This is now equivalent to a network with no buffer capacity constraints and with service times which have been replaced by the effective service times calculated at the First Pass. Whitt's formulas are used to approximate the processes that result from the merging and splitting of traffic processes. Marshall's formula is used to calculate the scv of the departure process from each node. The single node approximation method of Kraemer and Langenbach-Belz (Whitt [48]) is used to analyse each of the $GI/G/1/\infty$ queues. The mean queue length MN of node i is given from Little's result as

$$MN = \rho_i + \lambda_a MW \tag{3.7}$$

where

$$MW = \tau_i \rho_i (c_a^2 + c_i^2) g / 2(1 - \rho_i) \tag{3.8}$$

where MW is the mean waiting time in the queue of node i and g is defined as

$$g = \begin{cases} \exp\left(-\frac{2(1-\rho_i)}{3\rho_i}\frac{(1-c_a^2)^2}{c_a^2+c_i^2}\right), & c_a^2 < 1 \\ 1, & c_a^2 \geq 1. \end{cases} \tag{3.9}$$

where λ_a, c_a^2 are the rate and scv of the arrival process, respectively, ρ_i, τ_i, c_i^2 are the traffic intensity, mean, and scv of the service time of node i, respectively.

3.2.5 The Generalized Expansion Method

The basic idea of the Generalized Expansion Method (GEM) is to add an artificial node to each capacitated queue to collect any blocked customers (see Figure 3.1). It should be noted that the GEM uses only the first two moments of the arrival and service processes for the estimation of all other parameters of the system. In case that node j is full the blocked customer of node i will be routed to the artificial node with probability P_N (where N is the maximum number of customers at node j). After the customer receives service at the artificial node he/she tries again to rejoin node j. There is a probability $P_N^{'}$ the customer has to stay at the artificial node for one more service. The artificial node is modelled as $GI/G/\infty$ queue. The probability $P_N^{'}$ is calculated using the approximation procedure of Labetoulle and Pujolle [33].

To determine the service rate of the artificial node h the following formula from Kleinrock [31] (page 173, equation (5.16)) is used

$$\mu_h = (2\mu_j)/(1 + \mu_j^2\sigma_j^2) \tag{3.10}$$

where μ_j and σ_j^2 are the service rate and variance of the service time of the blocking node, respectively.

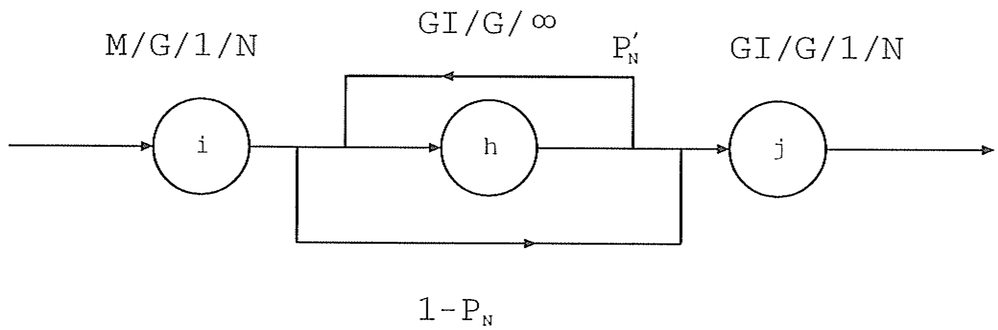

Figure 3.1: The GEM for a tandem system

In order to calculate the C^2_{di} which is the squared coefficient of variation (scv) of the departure process from node i, we use the following formula from Marshall [37]

$$C^2_{di} = C^2_{ai} + 2\rho_i^2 C^2_{si} - 2\rho_i(1-\rho_i)\mu_i W_{qi} \tag{3.11}$$

where C^2_{ai} is the scv of the arrival process at node i, ρ_i is the server's traffic intensity of node i, C^2_{si} is the scv of service time of node i, μ_i is the service rate of node i, and W_{qi} is the expected waiting time in the queue at node i.

To determine the squared coefficient of variation C^2_{ih} of the arrival process at the artificial node h, it is assumed that the departure process from node i splits into two processes. The first process is the arrival process of nonblocked customers at node j and the second process is the arrival process of blocked customers at node h. The scv C^2_{ih} is given from the following formula (Whitt [48])

$$C^2_{ih} = P_N C^2_{di} + 1 - P_N \tag{3.12}$$

The squared coefficient of variation of the arrival process of nonblocked customers at node j is given by

$$C^2_{ij} = (1-P_N)C^2_{di} + P_N \tag{3.13}$$

The squared coefficient of variation of the service time of the artificial node h is determined using the following formula from Kleinrock [31]

$$m(s^n) = [u(s^{n+1})]/[(n+1)u(s)] \tag{3.14}$$

where $m(s^n)$ is the n-th moment of the remaining service time, and $u(s^{n+1})$ is the (n+1)th moment of the service time.

In general, most non-negative distributions whose coefficient of variation is less than one can be approximated by an Erlang distribution, and most non-negative distributions whose coefficient of variation is greater than 1 can be approximated by an Hyperexponential distribution. The Erlang-2 (E_2) distribution is a special case of the more general Erlang-k family of probability distributions (see Appendix A). The third moment of the Erlang-2 distribution is given by

$$u(s^3) = 3(1/\mu)^3 \tag{3.15}$$

where $1/\mu$ is the mean service time.

The third moment of the Hyperexponential-2 (H_2) distribution is given by

$$u(s^3) = 6(a_1/\mu_1^3 \; + \; a_2/\mu_2^3) \tag{3.16}$$

The scv of the service time of the artificial node is

$$C_h^2 = [m(s^2) - (1/\mu_h)^2]/[1/(\mu_h)^2] \tag{3.17}$$

It is assumed that the customer receives only one service at the artificial node. The revised service rate of the artificial node is given by (Whitt [48])

$$\mu_h^{'} = (1 - P_N^{'})\mu_h \tag{3.18}$$

and the revised scv of the service time is

$$C_{sh}^2 = P_N' + (1 - P_N')C_h^2 \tag{3.19}$$

The effective service time at node i that precedes node j is

$$\mu_{i_{eff}}^{-1} = \mu_i^{-1} + P_N \mu_h^{'-1} \tag{3.20}$$

A system of nonlinear equations in an equal number of unknowns needs to be solved to determine the values of the parameters of the system. The number of equations depends on the type of the queues and on the approximations that are used for the estimation of some of the parameters of the system.

3.3 Application of the Approximation Method to a Real Life System

The approximation method was applied to the modelling of a real life system. The system is part of a conveyor installed in a manufacturing company. This conveyor system is a typical example of material handling system. Material handling systems costs in the manufacturing industry constitute about 30% - 80% of typical operating costs (Hill [26]). Material handling systems (MHS) integrate the production system, by moving items/customers through the workstations. Conveyor systems, are very common MHS in manufacturing and the need for their analysis arises everyday. The analysis of material handling systems with the intent of applying methods that optimize the operation of MHS play a key role in the effort to increase the productivity of the whole production facility. There are many different types of MHS. The most common ones are: manually driven carts, forklifts, cranes, AGV's, robots, and conveyor systems.

The flow of material through the production line can be studied as a stochastic process in a network of queues. The output from a specific workstation is the input of another workstation. The presence of blocking at the nodes and the non-exponential nature of the distributions of the service times at the nodes make the analysis difficult. The conveyor system under study, moves/transfers the parts to be processed through several workstations. There are buffers with limited capacities between workstations and each workstation has its own service time distribution. The conveyor system uses carriers for the transportation of the parts through the line. The whole system is a closed network, if we consider the transportation units-carriers as customers. Therefore, the whole conveyor system is a closed queueing network with a fixed number of carriers. The analysis of the conveyor system with analytical methods is very difficult. For this reason, a part of the system was isolated and it was examined as an open queueing network. Analysing the isolated network as an open queueing network, we can make inferences about the same system when it operates as a part of the closed network. The analysis will give us the average queue lengths behind workstations and control points and the average sojourn time through the network. These results will be compared with results obtained from simulation.

3.3.1 The system

The system to be modelled is part of a larger conveyor system. The whole conveyor system is a painting line which integrates the production departments with the assembly line. This painting line (see Figure 3.2) consists of six conveyors which run through the following areas: i) the loading area ii) the shot blast iii) the washer

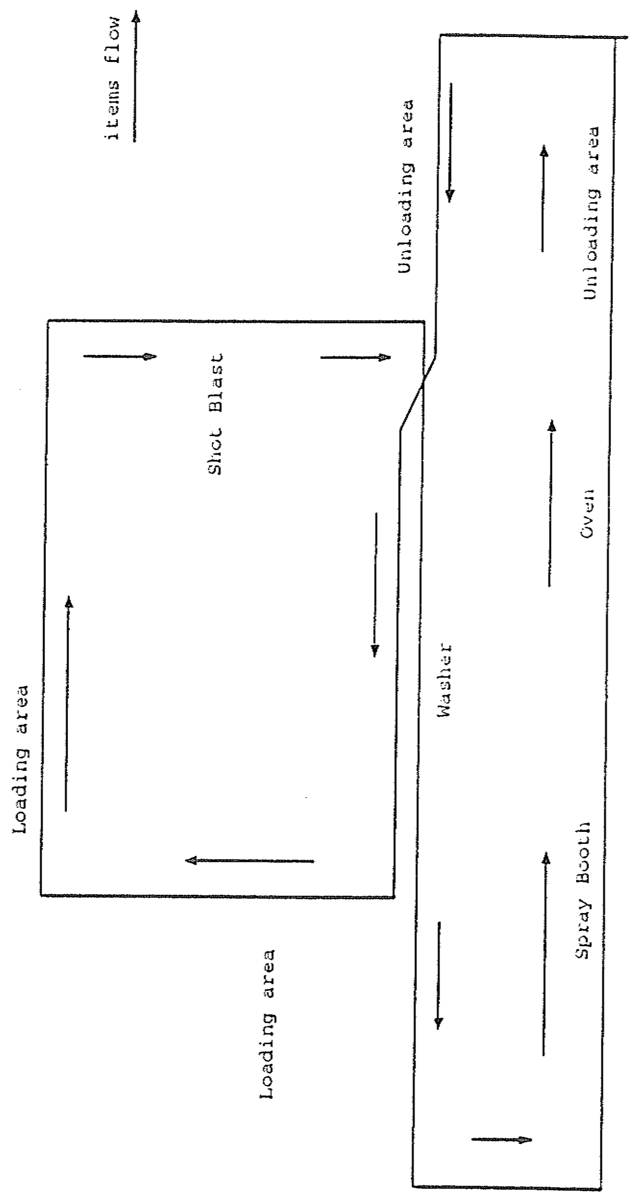

Figure 3.2: Diagrammatic representation of the conveyor system

iv) the spray booth v) the oven and vi) the unloading area. The loading area consists of ten loading stations and the unloading area consists of four unloading stations. There are points in the painting line that are called STOPS which are used for the loading, unloading and control of the movement of the transportation units which are called carriers. The transportation units-carriers are loaded with parts and circulate through the painting line moving from one conveyor to the other and from one workstation to the next one. The loading and unloading stations are considered to be workstations and their service times have general distributions. Every STOP has a limited space for waiting carriers, therefore each STOP is considered as having a buffer with limited capacity. The six conveyors run independently of each other and can have different speeds. The whole system is a closed network and the number of the circulating carriers-customers is fixed. Yannopoulos, Jenness and Hawaleshka [53] developed a simulation model using the language PCMODEL to analyse this closed network painting line. Knowing the limitations of simulation models we attempt here to develop an analytical model for the same problem. The analysis of the closed system with analytical methods is very difficult because of the complexity of the system. For this reason we isolated a part of the conveyor system and after some simplifications we analysed it as an open queueing network with general arrival and service processes. The input to the system to be modelled is known from the simulation model of the closed network and the results from the analysis will be compared with the results obtained from another simulation model of the open network developed in parallel with the analytical model. The analysis of the isolated open network will give us estimates of the parameters of the closed network and also it is possible to make inferences about the behaviour of the same part of the closed conveyor system. By solving this isolated part of the larger conveyor system, we show that it is possible to use queueing techniques to analyse real life systems.

The whole closed system can be analysed by decomposing it into smaller networks which will subsequently be modelled as open queueing networks. These smaller networks will be analysed in isolation by considering the output of one network to be the input of the next one. The system to be modelled consists of five STOPS and two conveyors (see Figure 3.3). More specifically in the system to be modelled there are three STOPS that serve as loading stations (STOPS #3,#4,and #5) and there are two STOPS (STOPS #1,and #2) that serve as control points. The service times have general distributions and their empirical distributions are known from historical data. The system operates under the following rules:

— STOP#1 sends carriers alternately towards STOPS#2, and #3. If STOP#2 AND STOP#3 are full then STOP#1 gets blocked. In case that one of the two STOPS is full, then STOP#1 sends carriers towards the STOP which is not full. We permit seven carriers to accumulate behind STOP#1. In other words, STOP#1 is a control point in the line.

—STOP#2 releases carriers towards STOP#4, when STOP#4 is not full. If STOP#4 is full, then STOP#2 gets blocked. We permit seven carriers to accumulate behind STOP#2.

—STOP#3 is a loading station. After the completion of the service the carrier is routed towards STOP#5. If STOP#5 is full, STOP#3 gets blocked. We permit ten carriers to accumulate behind STOP#3.

—STOP#4 is a loading station. After the completion of the loading, the carrier is released and exits the system. Two carriers are permitted to accumulate behind

STOP#4.

—STOP#5 is a loading station. After the service, the carrier exits the system. Two carriers are permitted to accumulate behind STOP#5.

—The distances between adjacent STOPS in the line are known.

—External arrivals occur only at STOP#1.

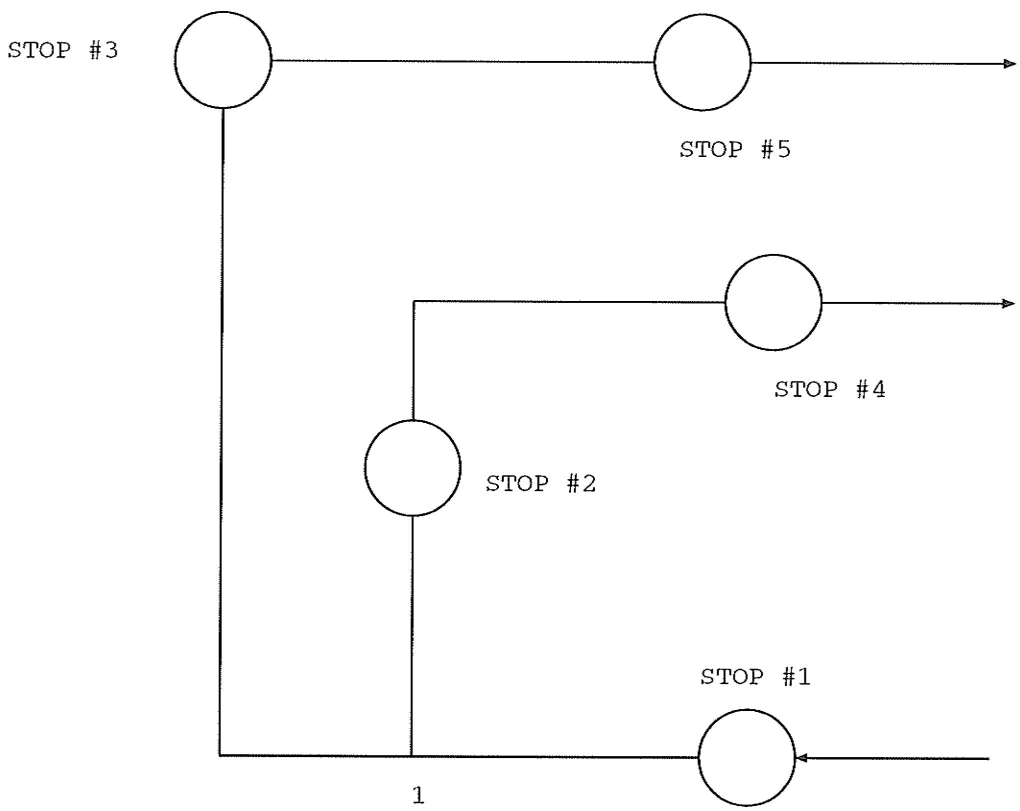

Figure 3.3: The system

3.3.2 The model

The system is modelled as an open queueing network that consists of five servers (STOPS) with general service times. The arrival process to the system at STOP#1 follows a general distribution and all queues have limited capacity. All stochastic processes are represented by two parameters: the mean and the squared coefficient of variation defined as the variance divided by the squared mean. In the first pass of the approximation method the effective rates and effective variances of the service times at all nodes are derived using the GEM of Kerbache and Smith. In the second pass we use the effective rates and effective variances as rates and variances of the service times at all nodes and then assume that the network is free of blocking. Then using the approximation method of Kraemer and Langenbach-Belz we estimate the queueing performance measures of the system.

3.3.3 The System as a Queueing Network

All STOPS are represented by their corresponding nodes (see Figure 3.4). The STOPS are modelled as follows:

—STOP#1 is modelled as $G/G/1/7$ queue (Node 1).

—STOP#2 is modelled as $G/G/1/7$ queue (Node 2).

—STOP#3 is modelled as $G/G/1/10$ queue (Node 3).

—STOP#4 is modelled as $G/G/1/2$ queue (Node 4).

—STOP#5 is modelled as $G/G/1/2$ queue (Node 5).

Travel times between nodes are estimated as waiting times at some imaginary nodes that are placed between the two real nodes. These imaginary nodes are modelled as $G/D/\infty$ queues where their service times are deterministic and equal to the corresponding travel times. More specifically:

The travel times from STOP#2 to STOP#4, from STOP#3 to STOP#5, from STOP#1 to POINT1, from POINT1 to STOP#2, and from POINT1 to STOP#3, are obtained as the waiting time at the $G/D/\infty$ queue type, at nodes 13,14,10,11, and 12 respectively.

All artificial nodes are modelled as $G/G/\infty$ queues.

Node h_1 collects blocked carriers when STOP#2 is full, node h_2 collects blocked carriers when STOP#3 is full, node h_3 collects blocked carriers when STOP#4 is full, and node h_4 collects blocked carriers when STOP#5 is full. Throughout the rest of this chapter the following notations will be used

Notation	Description
λ_a, C_a^2	arrival rate, and squared coefficient of variation of the external arrival process, respectively
λ_i, C_i^2 ,	rates, and squared coefficients of variation of the processes that result from the splitting of the external arrival process, respectively, $i = b, c$
λ_{ij},	the component of the departure process from node i (which preceeds node j) that goes directly towards node j, $i = 1, 2, 3,\ j = 2, 3, 4, 5$
λ_{h_j},	rate of the arrival process at the artificial node h_j, $j = 1, 2, 3, 4$
$P_{i,j}$,	probability of having i customers at queue j, $i = 2, 7, 10$ $j = 1, 2, 3, 4, 5$
λ, r_1, r_2, z	auxilliary variables
$P'_{i,j}$,	probability a carrier to find the jth queue full having stayed at the artificial node at least once, $i = 2, 7, 10$ $j = 1, 2, 3, 4, 5$

C^2_{ai},	squared coefficient of variation of the arrival process at node i, $i = 2, 3, 4, 5$
$C^2_{ah_j}$,	squared coefficient of variation of the arrival process at the artificial node h_j, $j = 1, 2, 3, 4$
C^2_{ij},	the squared coefficient of variation of the component of the departure process from node i (which precedes node j) that arrives directly with probability $1 - P_N$ at node j, $i = 1, 2, 3$ $j = 2, 3, 4, 5$
$C^2_{dh_i}$,	the squared coefficient of variation of the departure process from the artificial node h_i, $i = 1, 2, 3, 4$
ρ_{h_i},	traffic intensity of artificial node h_i, $i = 1, 2, 3, 4$
$\hat{\rho}_i$,	auxilliary variables, $i = 1, 2, 3, 4, 5$
μ'_{h_i},	revised service rate of artificial node h_i, $i = 1, 2, 3, 4$
C^2_{hi},	scv of the service time of the artificial node hi, $i = 1, 2, 3, 4$
$C^2_{sh_i}$,	revised scv of the service time of artificial node h_i, $i = 1, 2, 3, 4$
C^2_i,	scv of the service time of node i, $i = 1, 2, 3, 4, 5$
μ_{h_i},	service rate of the artificial node h_i, $i = 1, 2, 3, 4$
μ_i,	service rate of node i, $i = 1, 2, 3, 4, 5$
ρ_i,	traffic intensity of node i, $i = 1, 2, 3, 4, 5$
W_{qi},	mean waiting time in the queue at node i, $i = 1, 2, 3, 4, 5$

W_i,	mean waiting time at node i, $i = 1, 2, 3, 4, 5$
λ_{di}, C_{di}^2,	rate, and scv of the departure process from node i, respectively, $i = 1, 2, 3, 4, 5$
σ_i^2,	variance of the service time of node i, $i = 1, 2, 3, 4, 5$
$\mu_{i_{eff}}$,	effective service rate of node i, $i = 1, 2, 3, 4, 5$
$\sigma_{i_{eff}}^2, C_{i_{eff}}^2$,	effective variance, and effective scv of the service time of node i, respectively, $i = 1, 2, 3, 4, 5$

3.3.4 The Analysis

We will show the application of the approximation method at the part of the line that consists of the nodes $1, 10, 12, h2, 3, 14, h4$ and 5. The analysis of the part of the line that consists of the nodes $1, 10, 11, h1, 2, 13, h3$ and 4 is carried out in the same manner.

First Pass

We will apply the GEM starting from the last nodes of the network (nodes 4 and 5) and then the analysis will move backwards. The aim is to estimate the effective service rates and effective variances of the service times of all nodes.

Step 1. In this step we calculate the departure process from node 3. We assume that nodes 1, and 3 do not get blocked and also that all traffic processes are renewal processes. The service time of node 1 is zero. The arrival process described by λ_a, C_a^2 to node 1 is known and because of the zero service time the departure process from node 1 has the same λ_a, C_a^2. The queueing nodes of the type $G/D/\infty$ do not affect the traffic processes. Point 1 is a splitting point for the process described by λ_a, C_a^2. The component processes that result from the splitting have parameters that are given from the following formulas

$$\lambda_b = 0.5\lambda_a \tag{3.21}$$

$$\lambda_c = 0.5\lambda_a \tag{3.22}$$

$$C_b^2 = 0.5C_a^2 + 1 - 0.5 \tag{3.23}$$

$$C_c^2 = 0.5C_a^2 + 1 - 0.5 \tag{3.24}$$

The above equations (3.23) and (3.24) are valid when the splitting of the process described by λ_a, C_a^2 is done randomly. When carriers are split alternately, the squared coefficients of the component processes are 0.5 of the squared coefficient of the λ_a, C_a^2 process. In our problem, the carriers are split alternately only when both buffers of nodes 3 and 2 are not full. When one of the two buffers is full, the carriers are not split alternately but are routed towards the node which is not full. Thus, we assume that our splitting process is closer to a random process than an alternate process as traffic intensity increases.

The GEM has not been applied yet, therefore, node $h1$ does not exist yet. The process with rate λ_c and scv C_c^2 reaches node 3 which is modelled as a $GI/G/1/10$ queue. The diffusion approximation of Yao and Buzacott [54] together with Marshall's formula are used to approximate the departure process from node 3. Using the diffusion approximation we obtain the probability $P_{10,3}$ and the mean queue length. The rate of the departure process is

$$\lambda_{d3} = \lambda_c(1 - P_{10,3}) \tag{3.25}$$

Knowing the mean queue length, the mean waiting time in the queue can be obtained from the formulas

$$W_3 = J/\lambda_c \tag{3.26}$$

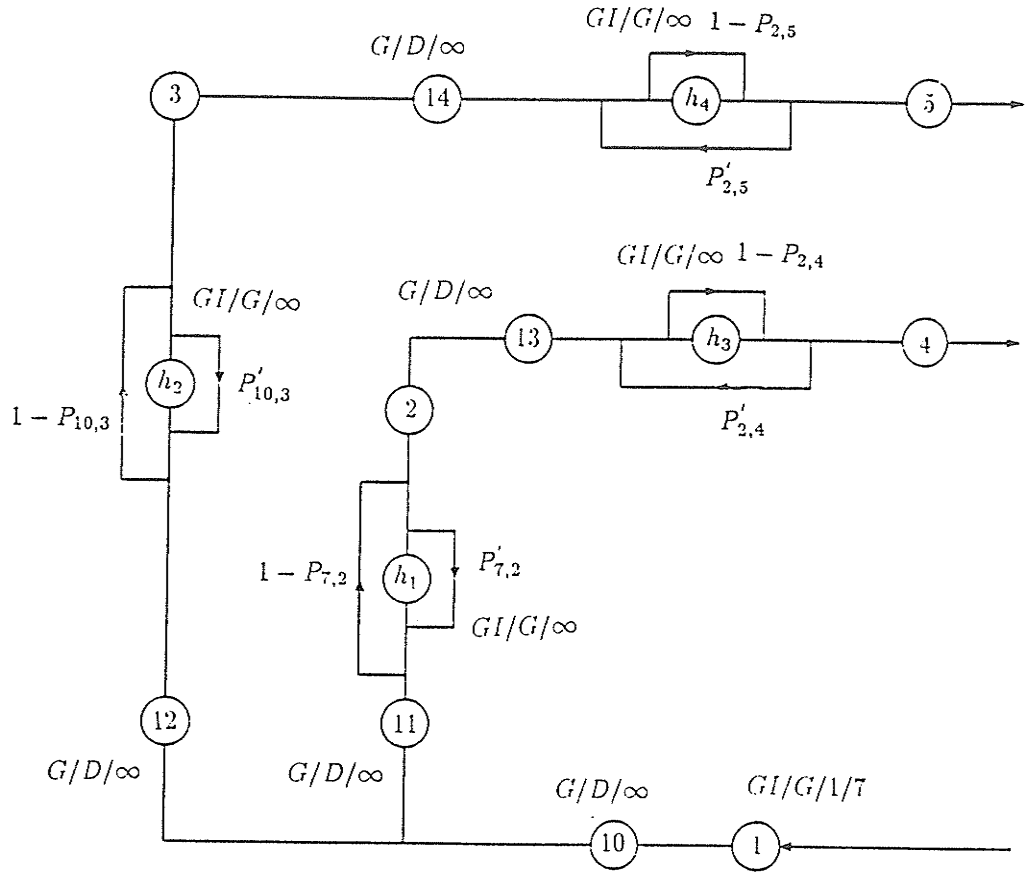

Figure 3.4: The reconfigured network

where J is the mean queue length and

$$W_{q3} = W_3 - 1/\mu_3 \tag{3.27}$$

The squared coefficient of variation of the departure process is given by

$$C_{d3}^2 = C_c^2 + 2(\lambda_c/\mu_3)^2 C_3^2 - 2(\lambda_c/\mu_3)(1 - \lambda_c/\mu_3)\mu_3 W_{q3} \tag{3.28}$$

Having obtained the rate and the scv of the departure process from node 3 we are ready to apply the GEM to node 5.

Step 2. We now apply the GEM to node 5 (see Figure 3.4). The service rate and the scv of the service time of node 5 are known. The service rate of the artificial node h_4 is given by

$$\mu_{h_4} = (2\mu_5)/(1 + \mu_5^2\sigma_5^2) \tag{3.29}$$

The traffic intensity of node 5 is

$$\rho_5 = \lambda_{d3}/\mu_5 \tag{3.30}$$

In order to estimate the squared coefficient of variation $C_{h_4}^2$ of the service time of the artificial node h_4 we approximate the service times distribution of node 5 as E_2 or H_2 depending on the value of C_5^2. To calculate the values of the rest of the parameters we have to solve the following system of nonlinear equations

$$\lambda_{d3} = \lambda_{35} + \lambda_{h_4} \tag{3.31}$$

$$\lambda_{35} = \lambda_{d3}(1 - P_{2,5}) \tag{3.32}$$

$$\lambda = \lambda_{35} - \lambda_{h_4}(1 - P_{2,5}^{'}) \tag{3.33}$$

$$P_{2,5}^{'} = [((\mu_5 + \mu_{h_4})/(\mu_{h_4})) - (\lambda(r_2^2 - r_1^2) - (r_2 - r_1))/(\mu_{h_4}(r_2^3 - r_1^3) - (r_2^2 - r_1^2))]^{-1} \tag{3.34}$$

$$z = (\lambda + 2\mu_{h_4})^2 - 4\lambda\mu_{h_4} \tag{3.35}$$

$$r_1 = (\lambda + 2\mu_{h_4} - z^{1/2})/(2\mu_{h_4}) \tag{3.36}$$

$$r_2 = (\lambda + 2\mu_{h_4} + z^{1/2})/(2\mu_{h_4}) \tag{3.37}$$

$$P_{2,5} = [\rho_5(1 - \rho_5)]/(\hat{\rho}_5^{-1} - \rho_5^2) \tag{3.38}$$

$$\hat{\rho}_5 = \exp\{-2(1 - \rho_5)/(\rho_5 C_{a5}^2 + C_5^2)\} \tag{3.39}$$

$$C_{ah_4}^2 = P_{2,5}C_{d3}^2 + 1 - P_{2,5} \tag{3.40}$$

$$C_{35}^2 = (1 - P_{2,5})C_{d3}^2 + P_{2,5} \tag{3.41}$$

$$C_{a5}^2 = (\lambda_{35}C_{35}^2 + \lambda_{h_4}C_{dh_4}^2)/(\lambda_{35} + \lambda_{h_4}) \tag{3.42}$$

$$C_{dh_4}^2 = C_{ah_4}^2 + 2\rho_{h_4}^2 C_{h_4}^2 \tag{3.43}$$

$$\mu_{h_4}^{'} = (1 - P_{2,5}^{'})\mu_{h_4} \tag{3.44}$$

$$C_{sh_4}^2 = P_{2,5}^{'} + (1 - P_{2,5}^{'})C_{h_4}^2 \tag{3.45}$$

$$\rho_{h_4} = \lambda_{h_4}/\mu_{h_4} \tag{3.46}$$

Solving the simultaneuos system of the nonlinear equations (3.31)-(3.46) we obtain $P_{2,5}$,$P_{2,5}^{'}$,$\mu_{h_4}^{'}$ and $C_{sh_4}^2$. It is now possible to calculate the effective service time and effective variance of node 3 using the following expressions

$$\mu_{3_{eff}}^{-1} = \mu_3^{-1} + P_{2,5}\mu_{h_4}^{'} \tag{3.47}$$

$$\sigma_{3_{eff}}^2 = \sigma_3^2 + P_{2,5}^2\sigma_{sh_4}^2 \tag{3.48}$$

where

$$\sigma_{sh_4}^2 = C_{sh_4}^2\mu_{h_4}^{'-2} \tag{3.49}$$

To solve the simultaneous system of nonlinear equations, we used the multi-variable Newton method [49] (see Appendix B). Using the effective service rate and the effective variance for node 3 the analysis moves back one node and applies the GEM to node 3. Following exactly the same approach we can determine the effective service rate and effective variance of node 1. By the end of the first pass we have estimated all effective service times and effective variances of the nodes of the system.

Second Pass

In the second pass, we assume that no blocking is taking place, therefore we consider the effective service rates and the effective variances as service rates and variances for the nodes of the system. This is now equivalent to a system consisting of infinite capacity queues. The revised network (see Figure 3.5) is a queueing network that consists of infinite capacity queues with general service times and general external arrival process. To calculate the mean queue lengths, we apply the approximation method of Kraemer and Langenbach-Belz. It is assumed that all traffic processes are renewal processes.

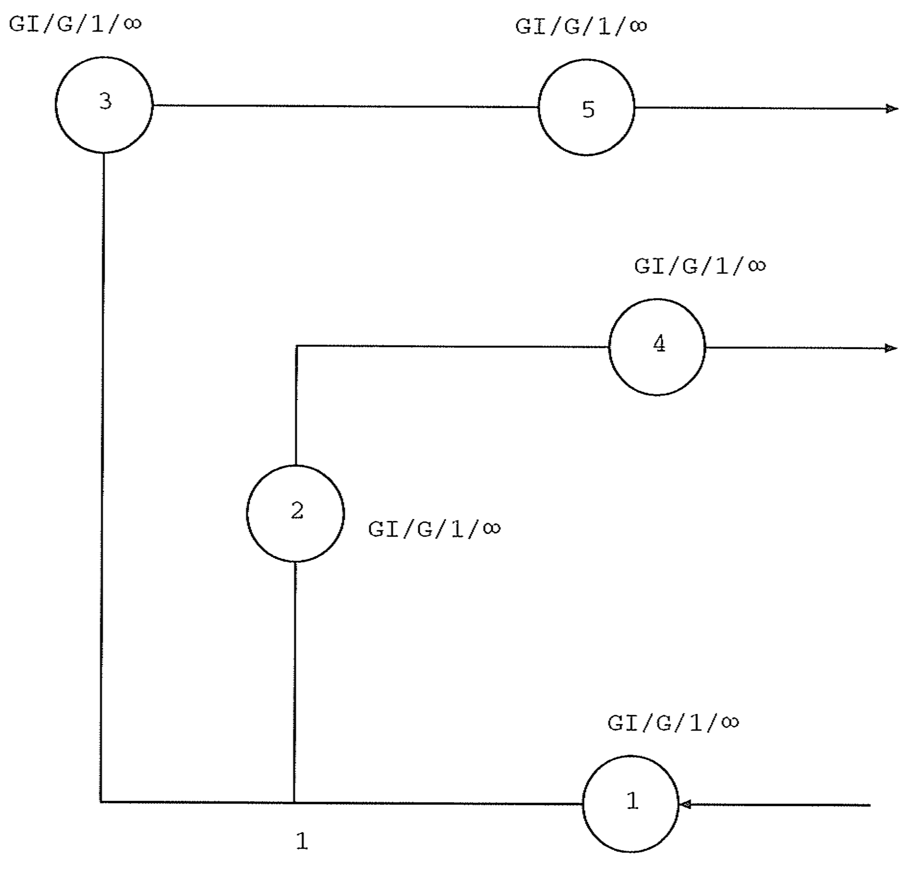

Figure 3.5: System configuration at the second pass

The external arrival process described by λ_a, C_a^2 and the first two moments of the service times distribution of node 1 are known. To find the mean queue length of node 1 we apply the following formulas based on the approximation method of Kraemer and Langenbach-Belz

$$EW = \tau_1 \rho_1 (C_a^2 + C_1^2) g / 2(1 - \rho_1) \tag{3.50}$$

where EW is the mean waiting time at the queue and g is defined as

$$g = \begin{cases} \exp(-\frac{2(1-\rho_1)}{3\rho_1} \frac{(1-C_a^2)^2}{C_a^2+C_1^2}), & C_a^2 < 1 \\ 1, & C_a^2 \geq 1. \end{cases} \tag{3.51}$$

The mean number of carriers at node 1, EN is obtained from Little's formula

$$EN = \rho_1 + \lambda_a EW \tag{3.52}$$

The rate of the departure process from node 1 is λ_a. To calculate the squared coefficient of variation of the departure process we use Marshall's formula. Point 1 is a splitting point for the departure process from node 1. Applying the above analysis we are able to approximate the mean queue lengths of all the queues of the network.

3.3.5 Numerical results

The number of carriers that circulate through the real system (closed queueing network) can be varied. The relationship between the throughput of the system and the number of carriers is of the hysteresis type, i.e the throughput of the system increases as the number of carriers increases but past a point the throughput starts to decrease as the number of carriers increases. We selected eight sets of data that were

Table 3.1: External Arrival Process

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
λ_a	0.144	0.248	0.510	0.623	0.720	0.760	0.770	0.770
C_a^2	0.627	0.536	0.420	0.213	0.194	0.100	0.135	0.180

collected when the closed system was operating with 105,110,120,125,130,135,140 and 150 carriers, referred to as Cases 1,2,3,4,5,6,7, and 8, respectively. It should be noted that the real closed system is set up to run with a minimum of 103 carriers. The external arrival times distributions and service times distributions for nodes 3,4, and 5 are given in Tables 3.1 and 3.2 respectively. The rate of the arrival process λ_a is expressed in carriers/min and the mean service times are expressed in minutes.

Two performance measures are calculated. These are the average sojourn time through the network and the average queue lengths at each node. Both these performance measures are very important in the efficient design of the system. Knowing the average sojourn time, the designer is able to select the combination of the system design parameters that improves the system efficiency ; the second performance measure indicates the work-in process levels, information that can be used to prevent high inventory costs and to reduce the congestion along the conveyor. The results obtained from the method are then compared with those obtained by simulation. For the comparisons we use the midpoints of 95% confidence intervals obtained from five simulation runs (for each case).

Table 3.2: Mean and Variance of Service Times

	Node 1		Node 2		Node 3		Node 4		Node 5	
	Mean	Var	Mean	Var	Mean	Var	Mean	Var	Mean	Var
Case 1	0	0	0	0	2.15	1.43	1.100	0.700	0	0
Case 2	0	0	0	0	2.15	1.43	1.100	0.700	0	0
Case 3	0	0	0	0	2.15	1.43	1.390	0.890	0.425	0.330
Case 4	0	0	0	0	2.15	1.43	1.580	0.970	0.660	0.453
Case 5	0	0	0	0	2.15	1.43	1.884	1.110	1.060	0.610
Case 6	0	0	0	0	2.15	1.43	2.030	1.110	1.200	0.593
Case 7	0	0	0	0	2.15	1.43	2.210	1.150	1.530	0.515
Case 8	0	0	0	0	2.15	1.43	2.230	1.040	1.600	0.486

Average Sojourn Times

The proposed algorithm gives good estimates of the average sojourn time for the first six cases and poor estimates for the last two cases when the system operates under heavy traffic. Average sojourn time is defined as the average time a carrier spends in the system. For each of the eight cases considered, two average sojourn times are calculated ; one for the carriers that travel through Part I of the system and one for the carriers that travel through Part II of the system. Part I consists of nodes 1,3, and 5 and Part II consists of nodes 1,2, and 4. Both the sojourn times (ST) and "non-travel sojourn times" (NTST) are calculated. The sojourn time calculated is the actual time it takes for a carrier to travel through the system. The non-travel sojourn time is the actual sojourn time minus the travel delays due to the conveyors (the time it takes for a carrier to travel from one node to the next one). This was done as a result of our introduction of the $G/D/\infty$ type of queues (which model the travel times on the conveyor). It gives exact results for the part of the sojourn time that consists of the travel delays on the conveyor. Therefore, we felt that we should provide estimations for the times that the carriers spend waiting at the queues plus the service time at the nodes i.e the "non-travel sojourn times". Thus, the actual sojourn time consists of the travel delays plus the time spent waiting to receive service at each node plus the service time at each node. Tables 3.3 and 3.4 illustrate the average sojourn times (in minutes) obtained from the approximation and from simulation for Part I, and Part II, respectively. All percent relative errors (% RE) are absolute relative errors.

As observed in Tables 3.3, and 3.4 the approximation provides good estimates for the actual ST and for the NTST for the first six cases and poor estimates for the

Table 3.3: Average Sojourn Times (Part I)

	Approx.				Simulation	
Case	ST	% RE	NTST	% RE	ST	NTST
1	10.02	2.1	2.35	9.8	9.81 ± 0.10	2.14 ± 0.10
2	10.20	3.0	2.53	13.5	9.90 ± 0.08	2.23 ± 0.08
3	11.66	7.0	3.99	23.5	10.90 ± 0.11	3.23 ± 0.11
4	12.68	8.4	5.01	24.3	11.70 ± 0.08	4.03 ± 0.08
5	14.90	5.2	7.23	10.1	15.71 ± 1.06	8.04 ± 1.06
6	16.15	20.5	8.48	33.0	20.32 ± 1.68	12.65 ± 1.68
7	17.10	38.7	9.43	53.4	27.90 ± 3.59	20.33 ± 3.59
8	17.46	44.5	9.79	58.8	31.44 ± 2.08	23.77 ± 2.08

Table 3.4: Average Sojourn Times (Part II)

	Approx.				Simulation	
Case	ST	% RE	NTST	% RE	ST	NTST
1	7.21	3.2	1.15	17.2	7.45 ± 0.08	1.39 ± 0.08
2	7.26	3.6	1.20	18.4	7.53 ± 0.07	1.47 ± 0.07
3	7.88	0.1	1.82	0.5	7.89 ± 0.04	1.83 ± 0.04
4 4	8.35	2.0	2.29	7.5	8.19 ± 0.03	2.13 ± 0.03
5	9.54	4.8	3.48	14.5	9.10 ± 0.06	3.04 ± 0.06
6	10.35	0.7	4.29	1.6	10.42 ± 0.21	4.36 ± 0.21
7	12.10	42.4	6.04	59.5	20.99 ± 4.64	14.93 ± 4.64
8	12.38	46.3	6.32	62.8	23.06 ± 2.87	17 ± 2.87

Table 3.5: Average Queue Lengths of Nodes 3 and 2

	Node 3			Node 2		
Case	Approx.	% RE	Simulation	Approx.	% RE	Simulation
1	0.17	13.3	0.15 ± 0.02	0		0
2	0.31	3.3	0.30 ± 0.02	0		0
3	0.90	23.3	0.73 ± 0.06	0.01		0
4	1.33	18.8	1.12 ± 0.07	0.03	200	0.01 ± 0.01
5	2.14	26.2	2.90 ± 0.40	0.07	22.2	0.09 ± 0.02
6	2.66	47.5	5.07 ± 1	0.11	78.4	0.51 ± 0.16
7	2.84	63.5	7.79 ± 1.18	0.16	96.5	4.59 ± 1.64
8	2.93	67.1	8.89 ± 0.46	0.16	97.2	5.64 ± 0.74

last two cases. The quality of the approximation is similar for Part I and Part II. The reason for the poor performance of the algorithm in the last two cases is that the system operates under heavy traffic (see Table 3.7).

Average Queue Lengths

Now we compare the average queue lengths at the nodes as obtained by the approximation method and by simulation. Table 3.5 illustrates the average queue lengths for nodes 3 and 2 and Table 3.6 illustrates the average queue lengths for nodes 4 and 5.

We observe that the results from our algorithm for nodes 3 and 2 match those from simulation for the first three cases when the network operates under light traffic

Table 3.6: Average Queue Lengths of Nodes 4 and 5

	Node 4			Node 5		
Case	Approx.	% RE	Simulation	Approx.	% RE	Simulation
1	0.08	0	0.08 ± 0.02	0		0
2	0.15	6.25	0.16 ± 0.02	0		0
3	0.46	15	0.40 ± 0.02	0.12	14.3	0.14 ± 0.01
4	0.69	13.1	0.61 ± 0.03	0.23	14.8	0.27 ± 0.02
5	1.18	19.2	0.99 ± 0.03	0.46	20.7	0.58 ± 0.08
6	1.52	18.8	1.28 ± 0.03	0.57	6.6	0.61 ± 0.03
7	2.17	29.9	1.67 ± 0.07	0.79	8.1	0.86 ± 0.02
8	2.27	34.3	1.69 ± 0.03	0.84	3.4	0.87 ± 0.03

(see Table 3.7). As the number of carriers in the system is increased and the traffic intensity increases (cases 4 to 8) the algorithm tends to underestimate the queue lengths (as compared with the simulation results). More specifically, for node 3, the algorithm deteriorates as we go from case 1 to case 8. In general, the algorithm gives very good estimates while the traffic intensities of nodes 3 and 5 are not greater than 0.55 and 0.11 respectively, and poor estimates as the traffic intensities of nodes 3 and 5 start to exceed 0.83 and 0.59 respectively. This same behaviour is observed for node 2. The algorithm gives good results as long as the traffic intensity of node 4 is less than 0.68. As the traffic intensity tends towards the value of 0.86 the algorithm fails to give acceptable results.

As we can see from Table 3.6, the algorithm gives good estimates for the average queue length behind node 5 and good estimates for the average queue length behind node 4 for the first four cases and it overestimates the average queue length for node 4 in the last four cases. The underestimation of the queue length for node 2 is partly due to the fact that the algorithm overestimates the queue length for node 4 (in the last four cases). There are many reasons for the algorithm's poor performance under heavy traffic. They are as follows:

- The assumption that all traffic processes are renewal processes is not realistic.
- The use of only two parameters (mean and squared coefficient of variation) for the description of general distributions is an oversimplification. It is known that it is possible for two completely different probability distributions to have the same mean and scv. Therefore, using only two parameters we lose some of the distribution's characteristics which may cause deviations from our estimations.

Table 3.7: Traffic Intensity

Case	Node 3	Node 4	Node 5
1	0.16	0.08	0
2	0.27	0.14	0
3	0.55	0.35	0.11
4	0.67	0.49	0.21
5	0.77	0.68	0.38
6	0.82	0.77	0.46
7	0.83	0.85	0.59
8	0.83	0.86	0.61

- The behaviour of the GEM itself under heavy traffic is another factor. Kerbache and Smith tested the GEM for three topologies (serial queues, merging queues, and splitting queues) and used two types of service times distributions (E_2 and H_2). They also, assumed that the arrival process to the first node is Poisson distributed. Their results indicate that the GEM underestimates the simulation results as the traffic intensity of the second node increases towards unity (large deviations from simulation results are observed when the traffic intensity becomes greater than 0.70). In our application, all the arrival processes have general distributions and the service times have also general distributions. The network operates under heavy traffic in cases 5-8.

- The approximations that have been used have produced errors that accumulate. More specifically:

Marshall's formula estimates the scv of the departure process from a $GI/G/1/\infty$ type queue. In our application Marshall's formula is used for the estimation of the scv of the departure process from a $GI/G/1/N$ type queue. The use of Whitt's formula for the approximation of the scvs of the processes that result from the splitting of a renewal process, also adds some errors to the already assumption that all the traffic processes are renewal processes.

- The model assumes that node 1 sends alternately carriers towards nodes 2 and 3. But in reality this is not the case, because only when both nodes 2 and 3 are not full node 1 sends alternately carriers towards nodes 2 and 3. When one of the nodes 2 and 3 is full, node 1 sends carriers towards the node which is not full.

- Approximating the general service time distributions with E_2 or H_2 also contributes errors to the analysis.

The time required for the execution of the approximation method cannot be compared to the simulation run time because of the different types of computers used. The program of the approximation method was run on a Sun workstation and the simulation program was run on a PC 386 computer.

3.4 Summary

In this Chapter an approximation method for the analysis of real life systems was presented. This method can be used to model arbitrary configurations of queues with general stochastic processes and finite buffers. The method transforms the examined system into a new equivalent system. The new system is assumed to be free of blocking and the service times of all nodes are revised to incorporate additional delays the items may have to undergo due to blocking. The method was applied to the modelling of a real problem. A conveyor system with general traffic and service processes, limited buffer capacities and splitting of the traffic process was considered. The approximation method developed seems to yield good results when the performance measure of interest is the average sojourn time through the system. On the other hand the method does not seem to yield as accurate results when the performance measure of interest is the average queue lengths. Thus, this algorithm can be very useful for systems that operate under light and moderate traffic (traffic intensity < 0.70) and when a good estimation of the sojourn time is more important than the accurate estimation of the average queue lengths. Moreover, by developing a method for real life systems, we provide a solution approach that can be employed in the analysis of any arbitrary system. It is also expected that by improving the approximation methods that were used as components of the algorithm, the performance of the whole algorithm will improve. It should be noted that this approximation method is the first to appear that was tested for a real life system. The results of this Chapter were submitted for publication to the Journal of Manufacturing Systems (Yannopoulos and Alfa [50]).

We now proceed to develop improved models for subsystems of general configurations i.e. tandem, split, and merge. Because tandem is the most common system, we attach more effort to that.

CHAPTER 4

An Approximation Method for Queues in Tandem with Blocking

4.1 Introduction

Queueing networks with queues in tandem appear frequently in manufacturing and in communication networks. The analysis of queues in tandem is not an easy task when there is blocking. Blocking occurs when there is an interruption of the flow of items from one node to the next node. Different processing rates at different nodes and limited storage space at intermediate buffers are the main reasons for the appearance of blocking. When a node is blocked then it can not process any new items.

Numerous papers have dealt with the problem of analysing tandem queueing networks with blocking. Exact analysis is possible only for small configurations (2-3 nodes). For larger networks, the state space becomes very large, and computationally unmanageable. This thus calls for the development of approximation methods that estimate the performance measures of the system. All existing approximation techniques have limitations in the amount of information that they provide and in the accuracy of their results. For instance, most of the approximations give us estimates of the performance measures of individual nodes, but they do not give the steady state information about groups of nodes (2-3 or more nodes). In addition, most of the approximations perform very well for lines that have a moderate number of queues (3-6) and low probability of blocking. However, in real life there are cases where lines with queues in manufacturing or communication may consist of a large number of nodes (10-20) with low or high probabilities of blocking combined with large buffer sizes. The goal in this chapter, is to provide information about the steady state of groups of nodes (groups of three nodes) and at the same time to give better estimates of the performance measures of queues that belong to long lines or belong to lines with high blocking probabilities and with large buffer sizes. The price that we pay, however, is an increase in the computational effort. Thus, there is a trade off between computational effort and information obtained.

In this research, we deal with a tandem queueing network. Buffers with limited storage space are located between adjacent nodes. External arrivals occur only at the first node and follow a Poisson distribution. Service times at all nodes are of the exponential type. The analysis is based on the work done by Brandwajn and Jow [11]. Brandwajn and Jow's method is built upon the ideas of equivalence and decomposition, the two steps involved in the analysis of queueing systems (Brandwajn [10]). At the first step (equivalence) the state equations for a chosen marginal probability are obtained and at the second step (decomposition) the conditional probabilities introduced through the equivalence are computed. More specifically, Brandwajn and Jow obtained the state equations for the marginal joint probabilities of the queue lengths of all possible pairs of adjacent nodes and they approximated the conditional probabilities introduced through the equivalence. These pairs of nodes are used as building blocks in the analysis, and their solutions (numerically or with the use of other computing techniques) generate information that is subsequently used in the analysis of the next pair of nodes. Their method provides estimates of the performance measures of individual nodes and approximations of the joint queue length probability distribution for pairs of neighboring nodes.

Their method seems to yield good estimates of the performance measures of the system when compared with other approximations and simulation. The objective of this study is to develop a method based on the work of Brandwajn and Jow, which will give more accurate results and more information about the joint steady state probability distributions of the queue lengths.

We consider cells that consist of three nodes with revised arrival and service processes. We provide the joint queue length probability distributions for triplets of adjacent nodes and we compare our results to those obtained by Brandwajn and Jow and those obtained by simulation. Each cell of three nodes at each step of the recursive scheme developed is solved numerically. This method allows the service rates to be state dependent (dependent on the number of customers at each node). In our analysis we have assumed communication type of blocking, however, the method can be modified to include manufacturing type of blocking.

4.2 Equivalence and Decomposition

Many of the techniques developed for the analysis of queueing networks consist of two main steps: i) equivalence and ii) decomposition. What these techniques have in common is the replacement of the solution of a single system by the solutions of simpler subsystems and then to combine these solutions of the subsystems to solve the whole system.

Brandwajn [10] tried to put the solution methods of these techniques under one unified approach. This unified approach consists of two steps: The first step is referred to as equivalence. In this step, the original state description is replaced by a suitably chosen marginal probability. In the second step referred to as decomposition, the conditional probabilities introduced in the first step are computed. The two steps in more details are presented in the following two subsections.

4.2.1 Equivalence

Suppose that we have developed the state equations for our system using the state description $s = (s_1, ..., s_i, ..., s_k)$. The next step is to consider a marginal state description $s^* = (s_1, ..., s_i)$ which contains a subset of s. Let $\hat{s} = (s_{i+1}, ..., s_k)$. The equations for the reduced state s^* can be obtained by summing the original state equations over all values of $\hat{s}$:

$$p(s) = p(s^*)p(s|s^*)$$

and

$$p(s) = p(s^*)p(\hat{s}|s^*)$$

so that

$$p(s^*) = \sum_{\hat{s}} p(s)$$

and

$$\sum_{\hat{s}} p(\hat{s}|s^*) = \sum_{\hat{s}} p(s|s^*) = 1$$

If $a(s)$ is the coefficient of $p(s)$ in the original state equations, it will be replaced by

$$\sum_{\hat{s}} a(s)p(s|s^*)$$

as a corresponding coefficient for $p(s^*)$ in the reduced set of equations.

4.2.2 Decomposition

Brandwajn used the term decomposition to refer to all those techniques that are used to determine simpler (decomposed) systems from the original single complex system. Brandwajn proposes to view these decomposition methods as ways to evaluate approximately the conditional probabilities $p(\hat{s}|s^*)$. Some of these decomposition methods are discussed in Brandwajn [10]. These methods differ in the way that approximate the conditional probabilities $p(\hat{s}|s^*)$.

4.3 The proposed method

Let us consider the queueing network that appears in Figure 4.1. The queueing network consists of K queues in tandem with buffers of limited space placed between adjacent nodes. Denote by M_i the maximum number of items (including the one in service) that can be stored in the buffer of node i.

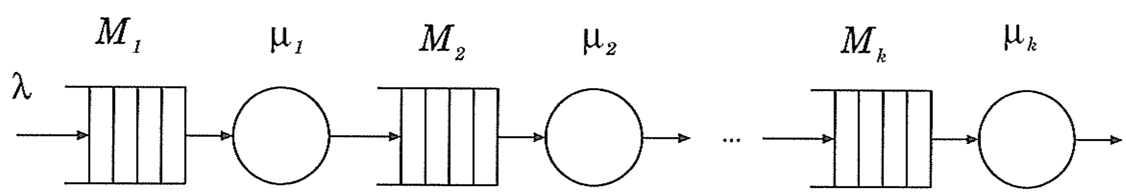

Figure 4.1: Queues in Tandem

Arrivals occur at the first node of the network according to the Poisson process with rate $\lambda(n_1)$ which depends on the number of items currently at the first node. When the first node is full the source that generates arrivals shuts down and it starts again as soon as the number of items at the first node drops below M_1. Thus,

$\lambda(n_1) = 0$ if $n_1 = M_1$. The service times at all nodes have exponential distributions with rates $\mu_i(n_i, n_{i+1})$. The service rate at node i depends on the number of items currently at node i and $i+1$. In this study we employ the communication type of blocking, that is $\mu_i(n_i, n_{i+1}) = 0$ if $n_{i+1} = M_{i+1}$, where n_i denotes the current number of items at node i. Also, $\mu_i(n_i, n_{i+1}) = 0$ if $n_i = 0$.

The system is completely characterized by its steady state probability distribution $p(n_1, n_2, ..., n_k)$ of the number of items at each node. The balance equations for this network are as follows:

$\{\lambda(n_1) + \sum_{i=1}^{k-1} \mu_i(n_i, n_{i+1}) + \mu_k(n_k)\}p(n_1, ..., n_i, n_{i+1}, n_{i+2}, ..., n_k) =$
$\lambda(n_1-1)p(n_1-1, ..., n_i, n_{i+1}, n_{i+2}, ..., n_k)+\mu_k(n_k+1)p(n_1, ..., n_i, n_{i+1}, n_{i+2}, ..., n_k+1)+$
$\sum_{i=1}^{k-1} \mu_i(n_i+1, n_{i+1}-1)p(n_1, ..., n_i+1, n_{i+1}-1, n_{i+2}, ..., n_k)$

Let us consider the three nodes $i, i+1, i+2$ and let us denote by $p(n_i, n_{i+1}, n_{i+2})$ the steady state probability distribution of the number of items at each of these three nodes. We can express this probability in terms of the joint probability distribution of the number of items at each of the nodes by writing

$p(n_i, n_{i+1}, n_{i+2}) = \sum_{j\neq i,i+1,i+2} \sum_{n_j=0}^{M_j} p(n_1, ..., n_i, n_{i+1}, n_{i+2}, ..., n_k)$.

The joint steady state probability distribution of the number of items at each node can be written as:

$p(n_1, ..., n_k) = Pr\{n_1, ..., n_{i-1}, n_{i+3}, ..., n_k | n_i, n_{i+1}, n_{i+2}\}p(n_i, n_{i+1}, n_{i+2})$

Performing the summation

$\sum_{j\neq i,i+1,i+2} \sum_{n_j=0}^{M_j}$

on the balance equations, we get

$$[\sum_{j\neq i,i+1,i+2} \sum_{n_j=0}^{M_j} \lambda(n_1)Pr\{n_1, ..., n_k | n_i, n_{i+1}, n_{i+2}\} + \sum_{j\neq i,i+1,i+2} \sum_{n_j=0}^{M_j}$$

$$\mu_1(n_1,n_2)Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\} + \sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_2(n_2,n_3)$$

$$Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\} + ... + \sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i-1}(n_{i-1},n_i)$$

$$Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\} + \sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_i(n_i,n_{i+1})$$

$$Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\} + \sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i+1}(n_{i+1},n_{i+2})$$

$$Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\} + \sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i+2}(n_{i+2},n_{i+3})$$

$$Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\} + \sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i+3}(n_{i+3},n_{i+4})$$

$$Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\} + ... + \sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_k(n_k)$$

$$Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\}]p(n_i,n_{i+1},n_{i+2}) =$$

$$\sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\lambda(n_1-1)Pr\{n_1-1,...,n_k|n_i,n_{i+1},n_{i+2}\}p(n_i,n_{i+1},n_{i+2})+$$

$$\sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_1(n_1+1,n_2-1)Pr\{n_1+1,n_2-1,...,n_k|n_i,n_{i+1},n_{i+2}\}$$

$$p(n_i,n_{i+1},n_{i+2}) + ... + \sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i-2}(n_{i-2}+1,n_{i-1}-1)$$

$$Pr\{n_1,...,n_{i-2}+1,n_{i-1}-1,...,n_k|n_i,n_{i+1},n_{i+2}\}p(n_i,n_{i+1},n_{i+2})+$$

$$\sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i-1}(n_{i-1}+1,n_i-1)Pr\{n_1,...,n_{i-1}+1,...,n_k|n_i-1,n_{i+1},n_{i+2}\}$$

$$p(n_i-1,n_{i+1},n_{i+2}) + \sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_i(n_i+1,n_{i+1}-1)$$

$$Pr\{n_1,...,n_k|n_i+1,n_{i+1}-1,n_{i+2}\}p(n_i+1,n_{i+1}-1,n_{i+2})+$$

$$\sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i+1}(n_{i+1}+1,n_{i+2}-1)Pr\{n_1,...,n_k|n_i,n_{i+1}+1,n_{i+2}-1\}$$

$$p(n_i,n_{i+1}+1,n_{i+2}-1)+\sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i+2}(n_{i+2}+1,n_{i+3}-1)$$

$$Pr\{n_1,...,n_{i+3}-1,...,n_k|n_i,n_{i+1},n_{i+2}+1)+$$

$$\sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i+3}(n_{i+3}+1,n_{i+4}-1)$$

$$Pr\{n_1,...,n_{i+3}+1,n_{i+4}-1,...,n_k|n_i,n_{i+1},n_{i+2}\}p(n_i,n_{i+1},n_{i+2})+...+$$

$$\sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_k(n_k+1)Pr\{n_1,...,n_k+1|n_i,n_{i+1},n_{i+2}\}p(n_i,n_{i+1},n_{i+2})$$

The above expression can be written as

$$[a_i(n_i,n_{i+1},n_{i+2})+\mu_i(n_i,n_{i+1})+\mu_{i+1}(n_{i+1},n_{i+2})+u_{i+2}(n_i,n_{i+1},n_{i+2})]$$

$$p(n_i,n_{i+1},n_{i+2})=a_i(n_i-1,n_{i+1},n_{i+2})p(n_i-1,n_{i+1},n_{i+2})+\mu_i(n_i+1,n_{i+1}-1)$$

$$p(n_i+1,n_{i+1}-1,n_{i+2})+\mu_{i+1}(n_{i+1}+1,n_{i+2}-1)p(n_i,n_{i+1}+1,n_{i+2}-1)+$$

$$u_{i+2}(n_i,n_{i+1},n_{i+2}+1)p(n_i,n_{i+1},n_{i+2}+1)$$

where

$$a_i(n_i,n_{i+1},n_{i+2})=\begin{cases}\sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i-1}(n_{i-1},n_i) & \\ Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\}, & i=2,...,K-2\\ \lambda(n_i), & i=1\end{cases}$$

and

$$u_{i+2}(n_i,n_{i+1},n_{i+2})=\begin{cases}\sum_{j\neq i,i+1,i+2}\sum_{n_j=0}^{M_j}\mu_{i+2}(n_{i+2},n_{i+3}) & \\ Pr\{n_1,...,n_k|n_i,n_{i+1},n_{i+2}\}, & i=1,...,K-3\\ \mu_k(n_i), & i=K\end{cases}$$

The conditional probability $Pr\{n_1, ...n_k | n_i, n_{i+1}, n_{i+2}\}$ can be written as

$$Pr\{n_1, ..., n_k | n_i, n_{i+1}, n_{i+2}\} =$$

$$Pr\{n_{i-1} | n_i, n_{i+1}, n_{i+2}\} Pr\{n_1, ..., n_k | n_{i-1}, n_i, n_{i+1}, n_{i+2}\}$$

$$Pr\{n_1, ..., n_k | n_i, n_{i+1}, n_{i+2}\} =$$

$$Pr\{n_{i+3} | n_i, n_{i+1}, n_{i+2}\} Pr\{n_1, ..., n_k | n_i, n_{i+1}, n_{i+2}, n_{i+3}\}$$

Substituting the above expressions into a_i and u_{i+2} we get

$$a_i(n_i, n_{i+1}, n_{i+2}) = \begin{cases} \sum_{n_{i-1}=1}^{M_{i-1}} \mu_{i-1}(n_{i-1}) & \\ Pr\{n_{i-1} | n_i, n_{i+1}, n_{i+2}\}, & i = 2, ..., K-2 \\ \lambda(n_i), & i = 1 \end{cases}$$

$n_i = 0, ..., M_i - 1$; $n_{i+1} = 0, ..., M_{i+1}$; $n_{i+2} = 0, ..., M_{i+2}$

$$u_{i+2}(n_i, n_{i+1}, n_{i+2}) = \begin{cases} \sum_{n_{i+3}=0}^{M_{i+3}-1} \mu_{i+2}(n_{i+2}) & \\ Pr\{n_{i+3} | n_i, n_{i+1}, n_{i+2}\}, & i = 1, ..., K-3 \\ \mu_k(n_i), & i = K \end{cases}$$

$n_i = 0, ..., M_i$; $n_{i+1} = 0, ..., M_{i+1}$; $n_{i+2} = 1, ..., M_{i+2}$

The cell that consists of the three nodes $i, i+1, i+2$ behaves as the tandem system in Figure 4.2 where the external arrival process has a state dependent rate $a_i(n_i, n_{i+1}, n_{i+2})$ and the service rate of the third node is $u_{i+2}(n_i, n_{i+1}, n_{i+2})$. The capacities of the buffers placed between the three nodes in the cell are the same with the ones in the original system. If we know the equivalent rates $a_i(n_i, n_{i+1}, n_{i+2})$ and $u_{i+2}(n_i, n_{i+1}, n_{i+2})$, we are able to solve the tandem system of Figure 4.2 using a numerical method. Once we have solved the three node system we can compute the joint steady state probability $p(n_i, n_{i+1}, n_{i+2})$. So far, the approach that we have followed is an exact solution without any approximations. The approximation element is now entering our analysis. We have to approximate the equivalent rates and to do so we assume that

$Pr\{n_{i-1}|n_i, n_{i+1}, n_{i+2}\} = Pr\{n_{i-1}|n_i, n_{i+1}\}, \quad i = 2, ..., K-1$ and

$Pr\{n_{i+3}|n_i, n_{i+1}, n_{i+2}\} = Pr\{n_{i+3}|n_{i+1}, n_{i+2}\}, \quad i = 1, ..., K-3$

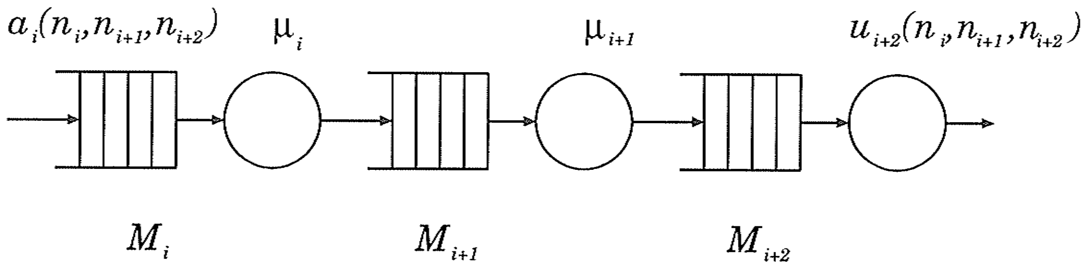

Figure 4.2: Equivalent three node cell

Using the above approximations the equivalent rates now become

$$a_i(n_i, n_{i+1}, n_{i+2}) \simeq a_i^*(n_i, n_{i+1}) = \begin{cases} \sum_{n_{i-1}=1}^{M_{i-1}} \mu_{i-1}(n_{i-1}) & \\ Pr\{n_{i-1}|n_i, n_{i+1}\}, & i = 2, ..., K-2 \\ \lambda(n_i), & i = 1 \end{cases} \tag{4.1}$$

$n_i = 0, ..., M_i - 1 \; ; \; n_{i+1} = 0, ..., M_{i+1} \; ; \; n_{i+2} = 0, ..., M_{i+2}$

$$u_{i+2}(n_i, n_{i+1}, n_{i+2}) \simeq u^*_{i+2}(n_{i+1}, n_{i+2}) = \begin{cases} \sum_{n_{i+3}=0}^{M_{i+3}-1} \mu_{i+2}(n_{i+2}) & \\ Pr\{n_{i+3}|n_{i+1}, n_{i+2}\}, & i = 1, ..., K-3 \\ \mu_k(n_k), & i = K \end{cases} \tag{4.2}$$

$n_i = 0, ..., M_i$; $n_{i+1} = 0, ..., M_{i+1}$; $n_{i+2} = 1, ..., M_{i+2}$

The solution cell (the cell that consists of nodes $i, i+1, i+2$ with revised arrival and service processes) in Figure 4.2 is now being transformed to the cell that is illustrated in Figure 4.3. The proposed method begins the analysis of the queueing network from the first triplet of nodes and proceeds to the next cell of three nodes until it reaches the last cell. The analysis consists of successive iterations through the network. Each iteration starts at the first cell of nodes and ends at the last cell of nodes. At each iteration the arrival and service processes of each cell are revised using the updated information. At each iteration of the algorithm $K-2$ cells have to be solved numerically. The updated information about the joint probability distribution of the queue lengths of the nodes of a cell is used as an input to the analysis of the next cell. The resulting joint probability distribution of a triplet of nodes is used for the evaluation of conditional probabilities that are then used in the analysis of neighboring cells.

4.4 The Algorithm

Consider a network with K nodes in series. The steps of the algorithm can be described as follows:

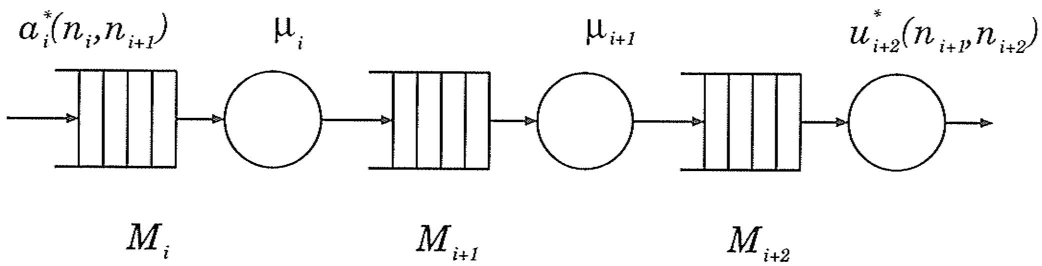

Figure 4.3: The solution cell

STEP 1.

a. Initialize the joint probabilities $p^0(n_i, n_{i+1}, n_{i+2})$ for $i = 1, ..., K-2$, such that $\sum_{n_i=0}^{M_i} \sum_{n_{i+1}=0}^{M_{i+1}} \sum_{n_{i+2}=0}^{M_{i+2}} p^0(n_i, n_{i+1}, n_{i+2}) = 1$.

b. Using the above joint probabilities compute the conditional probabilities $p^0(n_{i+2}|n_i, n_{i+1})$ for $i = 2, ..., K-2$.

c. Set $j = 1$.

STEP 2.

At iteration j, starting from the first three nodes of the system, solve $K-2$ triplets of adjacent nodes (n_i, n_{i+1}, n_{i+2}) for $i = 1, ..., K-2$.

For the triplet consisting of nodes $i, i+1, i+2$

a. Calculate the rate of the arrival process $a_i^*(n_i, n_{i+1})$ at node i, using Equation (4.1). The conditional probabilities $p^j(n_{i-1}|n_i, n_{i+1})$ involved in Equation (4.1) computed from the joint probabilities $p^j(n_{i-1}, n_i, n_{i+1})$.

b. Calculate the rate of the service process of the node $i+2$ using

Equation (4.2). The conditional probabilities $p^{j-1}(n_{i+3}|n_{i+1}, n_{i+2})$ involved in Equation (4.2) are computed from the joint probabilities $p^{j-1}(n_{i+1}, n_{i+2}, n_{i+3})$.

c. Solve the triplet of nodes numerically or by using some other computing method. The solution provides the joint probabilities $p^j(n_i, n_{i+1}, n_{i+2})$.

d. Compute the conditional probabilities $p^j(n_i|n_{i+1}, n_{i+2})$ and $p^j(n_{i+2}|n_i, n_{i+1})$ (which will be used in the solution of other triplets of nodes) from the joint probabilities $p^j(n_i, n_{i+1}, n_{i+2})$.

STEP 3.

If the convergence criterion (a difference between steady state probabilities of each node calculated at successive iterations less than a prespecified value) is not satisfied, set $j = j + 1$ and go to STEP 2, otherwise go to STEP 4.

STEP 4.

Provide the performance measures of the system (for example average queue lengths).

All the examples run using this algorithm seem to converge and the required computational effort depends on the number of nodes, the buffer sizes, and the service rates. The algorithm stops when the difference between the steady state probabilities of each node calculated at successive iterations is less than 10^{-5}.

4.5 Numerical results

The proposed algorithm was tested for different combinations of number of nodes, buffer sizes, and service rates. The results were compared to those obtained by using the two-node approach developed by Brandwajn and Jow [11]. We use the average queue lengths to compare our algorithm's performance against discrete simulation. Five simulation runs were performed for each experiment and the 95% confidence intervals were calculated. For each simulation run 35,000 items were used. The service rate μ_i of node i is considered to be independent of the number of items at node i. The proposed algorithm gives improved results when compared to the ones obtained by Brandwajn and Jow [11] especially in cases with large systems (10 nodes or more) and in cases where there is a combination of large buffer sizes and large changes in service rates between adjacent nodes. More specifically, we examined networks consisting of 5,6,10 and 15 nodes with various service rates and buffer sizes. In all the experiments we calculate the total error which is the sum of the absolute values of the errors for the average queue lengths with respect to the simulation results. For the calculation of the errors, the midpoints of the confidence intervals were used.

4.5.1 Small systems

In this section, we perform a set of experiments for systems consisting of a small number of nodes and small buffer sizes.

Experiment 1

We consider five nodes in tandem, and their buffer sizes are $M_1 = 4, M_2 = 5, M_3 = 5, M_4 = 4, M_5 = 3$ with service rates $\mu_1 = 2.5, \mu_2 = 0.3, \mu_3 = 2, \mu_4 = 1, \mu_5 = 0.5$ and

Table 4.1: Exp 1. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	3.6699	0.0124	3.6692	0.0117	3.6575 ± 0.0053
2	4.8623	0.0027	4.8621	0.0025	4.8596 ± 0.0005
3	0.2560	-0.0344	0.2359	-0.0545	0.2904 ± 0.0069
4	0.7720	-0.1786	0.7564	-0.1942	0.9506 ± 0.0161
5	1.1110	-0.1302	1.1080	-0.1332	1.2412 ± 0.0065
Total Error		0.3583		0.3961	

arrival rate $\lambda = 1.2$. Table 4.1 illustrates the results of Experiment 1. The total error from our algorithm is 0.3583. This is less than the total error of Brandwajn and Jow's method which is 0.3961. BJ stands for Brandwajn and Jow's method.

Experiment 2

In the second experiment we consider five nodes with buffer sizes $M_1 = M_2 = M_3 = M_4 = M_5 = 3$ and with service rates $\mu_1 = 1.7, \mu_2 = 2, \mu_3 = 1.5, \mu_4 = 1.7, \mu_5 = 0.8$ and with arrival rate $\lambda = 1.2$. We observe from Table 4.2 that the total error of our algorithm is 0.1811 while Brandwajn and Jow's method error is 0.2382.

Experiment 3

We consider five nodes with buffer sizes $M_1 = 4, M_2 = 5, M_3 = 5, M_4 = 4, M_5 = 3$ and service rates $\mu_1 = 2.1, \mu_2 = 1.5, \mu_3 = 0.9, \mu_4 = 0.55, \mu_5 = 0.7$ and arrival rate $\lambda = 2$. In this example BJ's method yields a total error of 0.0966 and our algorithm yields a total error of 0.1057 (Table 4.3). Thus, BJ's method for this experiment

Table 4.2: Exp 2. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	1.8882	0.0323	1.8888	0.0329	1.8559 ± 0.0227
2	2.0531	0.0451	2.0673	0.0593	2.0080 ± 0.0153
3	2.2939	0.0501	2.3112	0.0674	2.2438 ± 0.0266
4	2.1001	0.0326	2.1173	0.0498	2.0675 ± 0.0357
5	2.1902	0.0210	2.1980	0.0288	2.1692 ± 0.0264
Total Error		0.1811		0.2382	

gave better results than our algorithm but the difference is very small.

In experiments four and five we considered six nodes in tandem. In both cases our algorithm and BJ's method give fairly similar results. More specifically, our algorithm seems to perform better in the fourth experiment and BJ's method seems to perform better in the fifth experiment. But in both cases the differences between the total errors of the two methods are very small.

Experiment 4

In this experiment the buffer sizes of the six nodes are $M_1 = M_2 = M_3 = M_4 = M_5 = M_6 = 4$ and the service rates are $\mu_1 = 2.5, \mu_2 = 2, \mu_3 = 1.5, \mu_4 = 0.8, \mu_5 = 1.25, \mu_6 = 0.5$ and the arrival rate $\lambda = 2$. Our algorithm's total error is 0.0989 and BJ's method total error is 0.1017 (Table 4.4).

Experiment 5

The buffer sizes of the six nodes are $M_1 = M_2 = M_3 = M_4 = M_5 = M_6 = 4$ and

Table 4.3: Exp 3. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	3.7150	0.0244	3.7119	0.0213	3.6906 ± 0.0076
2	4.7319	0.0100	4.7287	0.0068	4.7219 ± 0.0052
3	4.5885	0.0313	4.5852	0.0280	4.5572 ± 0.0059
4	3.1463	-0.0178	3.1458	-0.0183	3.1641 ± 0.0148
5	1.1715	-0.0222	1.1715	-0.0222	1.1937 ± 0.0334
Total Error		0.1057		0.0966	

Table 4.4: Exp 4. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	3.6837	0.0182	3.6826	0.0171	3.6655 ± 0.0090
2	3.7601	0.0084	3.7594	0.0077	3.7517 ± 0.0040
3	3.6848	0.0139	3.6843	0.0134	3.6709 ± 0.0156
4	3.5368	0.0171	3.5383	0.0186	3.5197 ± 0.0186
5	2.7383	0.0259	2.7410	0.0286	2.7124 ± 0.0913
6	3.2251	0.0154	3.2260	0.0163	3.2097 ± 0.0390
Total Error		0.0989		0.1017	

Table 4.5: Exp 5. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	3.7322	0.0219	3.7308	0.0205	3.7103 ± 0.0040
2	3.7959	0.0061	3.7949	0.0051	3.7898 ± 0.0045
3	3.7337	0.0068	3.7329	0.0060	3.7269 ± 0.0022
4	3.5060	0.0052	3.5075	0.0067	3.5008 ± 0.0116
5	1.5912	-0.0668	1.5959	0.0599	1.6580 ± 0.0706
6	2.3723	-0.0507	2.3746	0.0484	2.4230 ± 0.0464
Total Error		0.1575		0.1466	

the service rates are $\mu_1 = 2.5, \mu_2 = 2, \mu_3 = 1.25, \mu_4 = 0.5, \mu_5 = 1, \mu_6 = 0.5$ and the arrival rate is $\lambda = 2$. BJ's method gives better results than our algorithm and the total error of BJ' method is 0.1466 and the total error of our algorithm is 0.1575 (Table 4.5).

4.5.2 Large systems

In the next three experiments we consider longer lines. We consider ten and fifteen nodes in tandem. Our algorithm seems to perform consistently better than BJ's method although the computational effort required is two to three times the effort required by BJ's method.

Experiment 6

In experiment six we consider ten nodes with buffer sizes $M_1 = 3, M_2 = 4, M_3 =$

Table 4.6: Exp 6. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	2.6849	0.0108	2.6757	0.0016	2.6741 ± 0.0051
2	3.6996	0.0049	3.6888	-0.0059	3.6947 ± 0.0020
3	2.5563	0.0062	2.5508	0.0007	2.5501 ± 0.0122
4	1.1968	-0.0172	1.1872	-0.0268	1.2140 ± 0.0142
5	1.4862	-0.0709	1.4732	-0.0839	1.5571 ± 0.0313
6	1.2780	-0.0308	1.3185	0.0097	1.3088 ± 0.0267
7	1.7753	0.0227	1.8540	0.1014	1.7526 ± 0.0322
8	2.8102	0.0798	2.9261	0.1957	2.7304 ± 0.0573
9	1.8968	0.0623	1.9502	0.1157	1.8345 ± 0.0218
10	1.2179	0.0182	1.2356	0.0359	1.1997 ± 0.0197
Total Error		0.3238		0.5773	

$3, M_4 = 2, M_5 = 4, M_6 = 3, M_7 = 3, M_8 = 4, M_9 = 3, M_{10} = 2$, and service rates $\mu_1 = 2.1, \mu_2 = 1.5, \mu_3 = 0.9, \mu_4 = 0.75, \mu_5 = 1, \mu_6 = 1.5, \mu_7 = 1.2, \mu_8 = 0.9, \mu_9 = 1.1, \mu_{10} = 0.65$ and the arrival rate $\lambda = 2$. From Table 4.6 we see that our algorithm performs much better than BJ's algorithm and the total error of our algorithm is 0.3238 while BJ's method total error is 0.5773.

Experiment 7

We consider fifteen nodes in series with buffer sizes $M_1 = M_2 = M_3 = M_4 = M_5 =$

$M_6 = M_7 = M_8 = M_9 = M_{10} = M_{11} = M_{12} = M_{13} = M_{14} = M_{15} = 3$ and service rates $\mu_1 = 2.1, \mu_2 = 1.5, \mu_3 = 0.9, \mu_4 = 0.65, \mu_5 = 1.5, \mu_6 = 1.5, \mu_7 = 1.2, \mu_8 = 0.9, \mu_9 = 1.1, \mu_{10} = 0.4, \mu_{11} = 1, \mu_{12} = 1, \mu_{13} = 1, \mu_{14} = 1, \mu_{15} = 1$ and the arrival rate $\lambda = 2$. As we see in Table 4.7 our algorithm's total error is 0.1162 and BJ's method total error is 0.1990.

Experiment 8

In this experiment we consider fifteen nodes with buffer sizes $M_1 = M_2 = M_3 = M_4 = M_5 = M_6 = M_7 = M_8 = M_9 = M_{10} = M_{11} = M_{12} = M_{13} = M_{14} = M_{15} = 3$ and service rates $\mu_1 = 2.5, \mu_2 = 2, \mu_3 = 1.2, \mu_4 = 1.9, \mu_5 = 2, \mu_6 = 0.9, \mu_7 = 1.3, \mu_8 = 1.7, \mu_9 = 1.1, \mu_{10} = 2, \mu_{11} = 0.95, \mu_{12} = 1, \mu_{13} = 1.5, \mu_{14} = 2, \mu_{15} = 0.8$ and the arrival rate $\lambda = 2$. As we see (Table 4.8) our algorithm gives better results than BJ's ones and more specifically the total error of our algorithm is 0.1327 and the total error of BJ's method is 0.2216.

4.5.3 Systems with high blocking probabilities and large buffer sizes

Brandwajn and Jow's method tends not to perform well in cases where there is a large difference between the service rates of adjacent nodes there by leading to high blocking probabilities and the first node has a relatively large buffer size. Thus, in the next six experiments we consider lines consisting of five nodes with the first node having a large buffer size and service rates that result in high blocking probabilities.

Experiment 9

In this experiment, we consider five nodes in tandem with buffer sizes $M_1 = 8, M_2 = 3, M_3 = 6, M_4 = 3, M_5 = 3$ and service rates $\mu_1 = 1.5, \mu_2 = 0.7, \mu_3 = 1, \mu_4 = 0.5, \mu_5 = 1$, respectively. The external arrival rate is $\lambda = 0.4$. The results are

Table 4.7: Exp 7. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	2.7758	0.0126	2.7747	0.0115	2.7632 ± 0.0074
2	2.7871	0.0077	2.7863	0.0069	2.7794 ± 0.0030
3	2.6813	0.0051	2.6817	0.0055	2.6762 ± 0.0031
4	2.3674	0.0036	2.3721	0.0083	2.3638 ± 0.0217
5	1.8629	0.0071	1.8844	0.0286	1.8558 ± 0.0274
6	2.4257	0.0037	2.4528	0.0308	2.4220 ± 0.0169
7	2.6084	0.0188	2.6302	0.0406	2.5896 ± 0.0093
8	2.5549	0.0108	2.5642	0.0201	2.5441 ± 0.0108
9	2.3509	0.0104	2.3529	0.0124	2.3405 ± 0.0086
10	2.4769	0.0098	2.4739	0.0068	2.4671 ± 0.0118
11	0.5777	-0.0032	0.5796	-0.0013	0.5809 ± 0.0100
12	0.5852	-0.0034	0.5860	-0.0026	0.5886 ± 0.0071
13	0.5868	-0.0038	0.5860	-0.0057	0.5917 ± 0.0131
14	0.5798	-0.0108	0.5786	-0.0120	0.5906 ± 0.0231
15	0.5405	-0.0054	0.5400	-0.0059	0.5459 ± 0.0121
Total Error		0.1162		0.1990	

Table 4.8: Exp 8. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	2.5508	0.0174	2.5422	0.0088	2.5334 ± 0.0138
2	2.6470	0.0013	2.6391	-0.0066	2.6457 ± 0.0056
3	2.5449	-0.0026	2.5381	-0.0094	2.5475 ± 0.0090
4	2.0422	-0.0105	2.0262	-0.0265	2.0527 ± 0.0209
5	2.3615	0.0026	2.3423	-0.0166	2.3589 ± 0.0204
6	2.5017	-0.0045	2.4897	-0.0165	2.5062 ± 0.0094
7	1.5125	-0.0159	1.4950	-0.0334	1.5284 ± 0.0356
8	1.5782	-0.0042	1.5726	-0.0048	1.5824 ± 0.0381
9	1.9829	-0.0028	1.9911	0.0054	1.9857 ± 0.0292
10	1.6104	-0.0066	1.6311	0.0141	1.6170 ± 0.0314
11	2.1884	0.0022	2.1939	0.0077	2.1862 ± 0.0212
12	1.5005	-0.0061	1.5016	-0.0050	1.5066 ± 0.0143
13	1.0202	-0.0291	1.0239	-0.0254	1.0493 ± 0.0379
14	1.1887	-0.0101	1.2178	0.0190	1.1988 ± 0.0410
15	1.7817	-0.0168	1.8159	0.0174	1.7985 ± 0.0285
Total Error		0.1327		0.2216	

Table 4.9: Exp 9. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	1.3172	-0.1216	0.9854	-0.4534	1.4388 ± 0.0479
2	1.4056	-0.0077	1.2851	-0.1282	1.4133 ± 0.0180
3	2.4651	-0.0245	2.2681	-0.2215	2.4896 ± 0.0252
4	1.8207	-0.0046	1.8230	-0.0023	1.8253 ± 0.0040
5	0.5761	-0.0112	0.5790	-0.0083	0.5873 ± 0.0015
Total Error		0.1696		0.8137	

illustrated in Table 4.9. We observe that our method performs much better than BJ's method for the first three nodes. The total error of our method is 0.1696 while the total error of BJ's method is 0.8137.

Experiment 10

We now consider the same line of Experiment 9 but with arrival rate $\lambda = 0.6$. As can be seen in Table 4.10, our method still performs better than BJ's method but the difference between the total errors is now less than the difference of the total errors in Experiment 9.

Experiment 11

In this experiment, we consider the same line of Experiments 9 and 10 but with arrival rate $\lambda = 1.2$. We observe (Table 4.11) that the total error of our method is still less than the total error of BJ's method.

Next, we consider a line with five nodes with the first node having a buffer size

Table 4.10: Exp 10. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	5.6318	-0.0843	5.5888	-0.1273	5.7161 ± 0.0258
2	2.5343	0.0163	2.5312	0.0132	2.5180 ± 0.0052
3	4.4145	0.0167	4.4041	0.0063	4.3978 ± 0.0174
4	2.2686	0.0293	2.2669	0.0276	2.2393 ± 0.0060
5	0.6770	-0.0212	0.6768	-0.0214	0.6982 ± 0.0040
Total Error		0.1678		0.1958	

Table 4.11: Exp 11. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	7.4332	0.0595	7.4268	0.0531	7.3737 ± 0.0076
2	2.6026	0.0239	2.6014	0.0227	2.5787 ± 0.0078
3	4.4805	-0.0063	4.4563	-0.0305	4.4868 ± 0.0153
4	2.2764	0.0484	2.2712	0.0432	2.2280 ± 0.0210
5	0.6785	-0.0160	0.6776	-0.0169	0.6945 ± 0.0154
Total Error		0.1541		0.1664	

Table 4.12: Exp 12. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	0.9831	0.1224	0.7895	-0.0712	0.8607 ± 0.0215
2	0.5017	0.0111	0.3624	-0.1282	0.4906 ± 0.0135
3	1.9293	-0.0596	1.8108	-0.1781	1.9889 ± 0.0189
4	0.6754	-0.0110	0.6644	-0.0220	0.6864 ± 0.0038
5	0.5988	0.0089	0.5971	0.0072	0.5899 ± 0.0026
Total Error		0.2130		0.4067	

of 10.

Experiment 12

The buffer sizes and service rates of the five nodes are $M_1 = 10, M_2 = 3, M_3 = 4, M_4 = 2, M_5 = 2$ and $\mu_1 = 1, \mu_2 = 5, \mu_3 = 0.7, \mu_4 = 1, \mu_5 = 0.9$ respectively. The external arrival rate is $\lambda = 0.4$. As can be seen in Table 4.12, the proposed method performs better than BJ's method.

Experiment 13

In this experiment, we use the system of Experiment 12 but with arrival rate $\lambda = 0.6$. The results are illustrated in Table 4.13. We observe that the proposed method performs better than BJ's method and the total error of our method is 0.3246 while BJ's method total error is 0.4484.

Experiment 14

In this last experiment we change the arrival rate of the system of Experiment 12

Table 4.13: Exp 13. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	6.6702	-0.3078	6.5903	-0.3877	6.9780 ± 0.0272
2	2.1167	0.0006	2.0774	-0.0387	2.1161 ± 0.0034
3	3.7901	0.0008	3.7859	-0.0034	3.7893 ± 0.0034
4	0.9291	0.0078	0.9312	0.0099	0.9213 ± 0.0022
5	0.7644	0.0076	0.7655	0.0087	0.7568 ± 0.0006
Total Error		0.3246		0.4484	

to $\lambda = 0.8$. As we see in Table 4.14 the total error of the proposed method is 0.1073 while the total error of BJ's method is 0.1987.

4.6 Summary

From the experiments performed, we conclude that the proposed method yields results that show improvement over the ones obtained by BJ's method. The two methods appear to be equally good as long as the network examined is of small size (5-6 nodes) with low or moderate probabilities of blocking and small buffer sizes, but our method requires more computational effort. For larger lines (10 nodes or more) and for lines with large buffer sizes and high blocking probabilities the proposed method seems to perform better than BJ's method. This is understandable since, as the line becomes larger (more nodes in the system), the interdependencies between

Table 4.14: Exp 14. Average Queue Lengths

Queue	Approximate	Error	BJ	Error	Simulation
1	8.5646	-0.0134	8.5323	-0.0457	8.5780 ± 0.0089
2	2.2169	-0.0605	2.1770	-0.1004	2.2774 ± 0.0114
3	3.8407	-0.0209	3.8215	-0.0401	3.8616 ± 0.0034
4	0.9323	-0.0094	0.9323	-0.0094	0.9417 ± 0.0027
5	0.7662	-0.0031	0.7662	-0.0031	0.7974 ± 0.0053
Total Error		0.1073		0.1987	

the nodes increase (dependencies that result from the blocking process) resulting in the arrival-service processes to individual nodes not being of the exponential type. Because the proposed method considers cells consisting of three nodes instead of two, these deviations from the exponential assumptions are well accounted for. As was pointed out earlier the improvement in the accuracy that we obtained with our method, is paid for by an increase in the computational effort. As can be seen in Table 4.15 our algorithm requires in most of the cases 1.5 - 3 times more iterations than BJ's method. Moreover, because our algorithm solves cells consisting of three instead of two nodes, the time required to solve a cell is 1.5 times the time required by BJ's method. On the other hand, for a system with K nodes, $K-2$ $(K-1)$ cells are solved at each iteration of our algorithm (BJ's method). Thus, there is a trade off between accuracy and computational effort. We should also mention that another advantage of our method is that it provides the joint steady state probability distribution of the number of items at each node for triplets of nodes, information that can not be obtained by other approximation methods. Simulation programs were run on PC 386 computers and the programs of the approximation methods were run on Sun workstations. Thus, comparisons of run times required by the simulation and by the approximation methods cannot be made. The results of this Chapter were submitted for publication to Performance Evaluation (Yannopoulos and Alfa [51]).

Table 4.15: Number of Iterations

	Approximation	BJ
Experiment 1	35	31
Experiment 2	57	25
Experiment 3	15	6
Experiment 4	11	10
Experiment 5	43	28
Experiment 6	283	132
Experiment 7	52	13
Experiment 8	194	79
Experiment 9	277	114
Experiment 10	144	41
Experiment 11	28	19
Experiment 12	399	64
Experiment 13	233	68
Experiment 14	83	33

CHAPTER 5

A Simple and Quick Approximation Algorithm for Tandem, Split, and Merge Queueing Networks

5.1 Introduction

Most of the existing approximation methods are not very easy to use. Implementing most of these approximation algorithms is very involved and requires considerable computational effort and computer memory. Some of the algorithms consist of iterative procedures, with each iteration involving the solution of one or two node systems. Other algorithms involve solving systems of nonlinear equations. Computational time requirements can sometimes be excessive depending on the examined system and the approximation technique used. For example, the development of phase-type distributions for lines with many nodes/machines leads to higher dimensional phase-type distributions. This could require prohibitive computational time. Very often, an approximation method for evaluating the performance measures of a queueing system is used as an integral part of an optimization procedure. The efficiency and effectiveness of an approximation method, in terms of computational requirements and quality of estimation of performance measures, will affect the overall performance of the optimization procedure itself.

Also, most of the existing methods are designed to work only for specific queue configurations (for example tandem, split or merge) or for specific probability distributions. Thus, a different method must be used, each time a different system is examined. As was pointed out in Section 2.2.2, only very few papers have appeared that deal with split, and merge configurations. Moreover, results for split, and merge configurations consisting of more than three nodes with general processes have not been reported yet.

Thus, there are times the analyst would like to have a method which would be simple, and fast, with good accuracy, which could be used for the analysis of a wide variety of queueing networks. The aim of this research is to present an approximation algorithm which is very simple, very fast, and provides reasonably accurate results. The method is designed to be used for the analysis of tandem, split, and merge configurations with general service and arrival processes and blocking. We show that the relative errors are in most of the cases within 10% (20%) for moderate (heavy) traffic when compared to simulation results. The examples used to demonstrate the quality of the approximation consist of cases with external interarrival and service times having Exponential, Erlang-2, Erlang-4, and Coxian-2 distributions. This is the first study to report results for split, and merge configurations consisting of more than three nodes with general stochastic processes and blocking. The algorithm provides estimates of the average sojourn time through the network.

5.2 The Basic Concept of the Approximation Method

The approximation method proposed in this Chapter applies a decomposition approach. It decomposes the system into several single-node cells with revised arrival and service processes. Once the system is decomposed, single-node approximations are used for the analysis of the individual single node-cells. To develop a method which is simple and at the same time fast, we have to find an efficient way of revising the traffic and service processes of the individual nodes. Let us consider the two node system in Figure 5.1. The departure process from node i (which is the arrival process of node $i+1$) contains information about the status of node $i+1$. This information has to do with the saturation of node $i+1$ which may lead to the blocking of node i. The blocked items at node i in a loss system would be lost customers, but in a delay system (like ours) the blocked items wait at node i until node $i+1$ becomes unsaturated. The fact that the departure process from node i is not a renewal process affects the average queue length of node $i+1$ (Patuwo et al. [40]). To deal with this problem, we introduce an equivalent loss node $i+1$ with revised arrival process having a rate greater than the real one. More specifically, for this two-node system we assume that the arrival rate at node $i+1$ is equal to the external arrival rate and node $i+1$ is now treated as a loss node rather than a delay node.

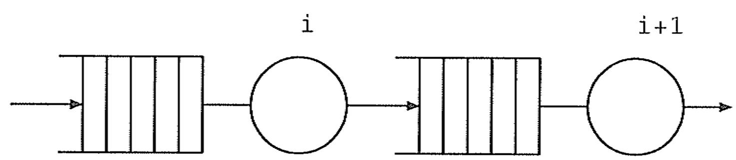

Figure 5.1: A two-node system

The next step is the revision of the service processes of individual nodes. Using information obtained from the analysis of node $i+1$ we revise the mean and variance of the service times of node i. This is in order to include possible additional delays for the item currently in node i that could result if upon completion of its (the item's) service at node i, node $i+1$ is full.

The method begins from the last node(s) of the network and proceeds towards the first(s) nodes decomposing the network, revising the arrival and service processes of each node at each stage using information obtained from the analysis of the last examined node. We now show how to use this approach for the analysis of networks with queues in series, split, and merge configurations.

5.3 Tandem Configurations

5.3.1 Model Assumptions

Consider the tandem network shown in Figure 1.1. External arrivals occur only at the first node. These arrivals have general distributions with rate and squared coefficient of variation λ, and c_a^2, respectively. Service times at all nodes have general distributions. Let μ_i, V_i, c_i^2, and N_i be the service rate, the service time variance, the service time squared coefficient of variation, and queue capacity (including the one in service) for node i, where $i = 1, 2, ..., M$. Departures from the system occur only at the last node (node M). Items finding the first node full upon their arrival are considered to be lost. An item that completed its service at node i proceeds towards node $i+1$. If at time of service completion at node i, node $i+1$ is full, then the item is forced to remain at node i until space becomes available at the buffer of node $i+1$. During the time when node i is occupied by the blocked item, it cannot process any new items. We assume node M cannot get blocked.

5.3.2 The Approximation Algorithm for Tandem Configurations

Consider the tandem system shown in Figure 1.1. The method starts with the last node M and then proceeds towards the first node. Node M does not get blocked so its service process is not revised. We distinguish two cases: (i) $c_a^2 \leq 1$ and (ii) $c_a^2 > 1$. In case (i) the arrival process at node i, $(i = 2, ..., M)$ is considered to be Poisson. In case (ii) the scv of the arrival process at node i, $(i = 2, ..., M)$ is assumed to be equal to c_a^2. Node M is treated as a loss system and analysed using a single-node approximation method. For more information about single-node approximations see Springel and Makens [46]. In this study we use two single node approximation methods, and compare how each one of them affects the performance of our algorithm. The two single node approximations used in this study are: i) Yao and Buzacott's [54] and ii) Gelenbe's [15]. These approximation methods are presented in Appendix C .

After we finish analysing node M we proceed to node $M-1$. The rate and variance of service time of node $M-1$ are revised using the following formulas

$$(\frac{1}{\mu_{M-1}})_{new} = (\frac{1}{\mu_{M-1}})_{old} + P_M E\{Y_M\}$$

and

$$(V_{M-1})_{new} = (V_{M-1})_{old} + P_M^2 V R_M$$

where P_M is the probability that the buffer at node M is full, $E\{Y_M\}$ and VR_M are the mean and variance of the residual time of the service time at node M respectively. The nth moment of the residual time is given as (Kleinrock Vol I, page 173 [31])

$$E\{Y_M^n\} = \frac{E\{X_M^{n+1}\}}{(n+1)E\{X_M\}}$$

where $E\{X_M^n\}$ is the nth moment of the service time of node M and $E\{X_M\} = 1/\mu_M$. Thus

$$E\{Y_M\} = \frac{E\{X_M^2\}}{2E\{X_M\}}$$

The variance of the residual time VR_M is given as

$$VR_M = E\{Y_M^2\} - (E\{Y_M\})^2$$

where

$$E\{Y_M^2\} = \frac{E\{X_M^3\}}{3E\{X_M\}}$$

Node $M-1$ is analysed using a single-node approximation with arrival process having rate λ and squared coefficient of variation equal to c_a^2 if $c_a^2 > 1$, or equal to 1 if $c_a^2 \leq 1$. Node $M-2$ is then analysed using the same idea. This procedure is continued up to node 2. The last node to be considered will be the first node of the system. The arrival process of the first node is not revised. After we have computed the performance measures of the first node, we then calculate the effective arrival rate to the network using the formula

$$\lambda_{eff} = \lambda(1 - P_1)$$

where P_1 is the probability that node 1 is full. To find the average sojourn time through the network we carry out another approximation using Little's formula as follows

$$ST = \frac{L_1 + L_2 + ... + L_M}{\lambda_{eff}}$$

where ST, and L_i are the average sojourn time through the network and the average queue length of the node i respectively.

The Algorithm

Consider a network with M (> 1) nodes in series. The algorithm consists of the following steps:

STEP 1.

Analyse the last node M of the system using a single-node approximation assuming the arrival process having rate λ and squared coefficient of variation equal to c_a^2 if $c_a^2 > 1$ or equal to 1 if $c_a^2 \leq 1$.

Store the blocking probability P_M and the average queue length L_M.

STEP 2.

Set $i = M - 1$

STEP 3.

a. Revise the service process of node i to take into account the blocking phenomenon. For the revision of μ_i and V_i use the following formulas:

$(\frac{1}{\mu_i})_{new} = (\frac{1}{\mu_i})_{old} + P_{i+1}E\{Y_{i+1}\}$

and

$(V_i)_{new} = (V_i)_{old} + P_{i+1}^2 VR_{i+1}$

where P_{i+1} is the probability node $i+1$ is full, $E\{Y_{i+1}\}$ and VR_{i+1} are the mean and variance of the residual time of the service time of node $i+1$.

b. Analyse node i using a single-node approximation assuming the arrival process having rate λ and squared coefficient of variation equal to c_a^2 if $c_a^2 > 1$ or equal to 1 if $c_a^2 \leq 1$.

c. Store P_i and L_i.

d. If $i = 2$ go to STEP 4 otherwise set $i = i - 1$ and go to STEP 3(a).

STEP 4.

a. Revise the service process of node 1 to take into account the blocking phenomenon. For the revision of μ_1 and V_1 use the following formulas:

$(\frac{1}{\mu_1})_{new} = (\frac{1}{\mu_1})_{old} + P_2 E\{Y_2\}$

and

$(V_1)_{new} = (V_1)_{old} + P_2^2 V R_2$

b. Analyse node 1 using a $G/G/1/N$ single-node approximation without revising the arrival process.

c. Store P_1 and L_1.

d. Use the formula $\lambda_{eff} = \lambda(1 - P_1)$ to calculate the effective arrival rate at the first node.

e. Compute the average sojourn time through the network using the following formula

$ST = \frac{L_1 + L_2 + ... + L_M}{\lambda_{eff}}$

5.4 Split Configurations

5.4.1 Model Assumptions

A split configuration consists of a single node (first level) linked to two or more (n) nodes (second level) as shown in Figure 1.2. Let μ_i, V_i, c_i^2, and N_i be the service rate, the service time variance, the service time squared coefficient of variation, and queue capacity (including the one in service) for node i where $i = 0, 1, 2, ..., n$ and $i = 0$ denotes the first level node. External arrivals with rate, and scv λ_0, and c_a^2, respectively, occur only at the first level node and the interarrival times have general distributions. Service times at all nodes have general distributions. The departure process from the first level node splits into streams which now become the arrival processes of the second level nodes. Buffers with finite capacities are placed behind each node and an item that has just completed its service at the first level node gets blocked if all second level nodes are full at this instance. The blocked item is forced to wait at the first level node (occupying it) until one of the second level nodes becomes not full. During this time the first level node cannot process any new items. It is assumed that the departure process from the first level node is equally split into n streams and that all second level nodes have equal service rates and buffer sizes. Thus the application of this model is limited to the cases where the second level consists of machines/service facilities that are of the same type.

5.4.2 The Approximation Algorithm for the Split Configuration

Consider the split configuration shown in Figure 1.2. The method analyses the network starting from the n second level nodes and then proceeds to the first level node. More specifically, the second level nodes are treated as loss systems with arrival processes having rates $\lambda_n = \lambda_0/n$. We distinguish two cases: (i) $c_a^2 \leq 1$ and (ii) $c_a^2 > 1$. In case (i) the arrival process at the second level node i, $(i = 1, ..., n)$ is considered to be Poisson. In case (ii) the scv of the arrival process at the second level node i, $(i = 1, ..., n)$ is assumed to be equal to c_a^2. It is assumed that the second level nodes cannot get blocked so their service processes are not revised. Each of the n second level nodes is solved using a single node approximation method. The rate and variance of the first level node's service times are revised using the following formulas:

$$(\frac{1}{\mu_0})_{new} = (\frac{1}{\mu_0})_{old} + \sum_{i=1}^{n} \frac{P_i}{n} E\{Y_i\}$$

and

$$V_{0,new} = V_{0,old} + \sum_{i=1}^{n} \frac{P_i^2}{n} VR_i$$

where P_i is the probability node i is full $(i = 0, 1, ..., n)$, $E\{Y_i\}$, and VR_i are the mean, and variance of the residual time of the service time of the second level node i, respectively. The arrival process of the first level node is not revised and we use a $G/G/1/N$ approximation to analyze the first level node. Then, the effective arrival rate to the network is computed using the formula $\lambda_{eff} = \lambda_0(1 - P_0)$ and finally the average sojourn time for each of the n branches of the network are calculated using the formula $ST_i = (L_0/\lambda_{eff}) + L_i/\lambda_e$ where $\lambda_e = \lambda_{eff}/n$, ST_i, and L_i are the average sojourn time for the ith branch of the network $(i = 1, 2, ..., n)$, and the average queue length of node i $(i = 0, 1, ..., n)$, respectively.

The Algorithm

Consider a split configuration with n second level nodes. The algorithm consists of the following steps:

STEP 1.

a. Set $i = 1$

b. Analyse the ith second level node using a single node approximation assuming the arrival process having rate $\lambda_i = \lambda_0/n$ and squared coefficient of variation equal to c_a^2 if $c_a^2 > 1$ or equal to 1 if $c_a^2 \leq 1$.

c. Store P_i and L_i.

d. If $i = n$ go to STEP 2, otherwise set $i = i + 1$ and go back to STEP 1 (b).

STEP 2.

a. Revise the service process of the first level node using the following formulas

$(\frac{1}{\mu_0})_{new} = (\frac{1}{\mu_0})_{old} + \sum_{i=1}^{n} \frac{P_i}{n} E\{Y_i\}$

and

$V_{0,new} = V_{0,old} + \sum_{i=1}^{n} \frac{P_i^2}{n} VR_i$

b. Analyse the first level node as a $G/G/1/N$ queue without revising its arrival process.

STEP 3.

a. Compute the effective arrival rate to the network using the formula

$\lambda_{eff} = \lambda_0(1 - P_0)$

b. Set $i = 1$

c. Calculate the average sojourn time through the network for the ith branch using the formula

$$ST_i = (L_0/\lambda_{eff}) + L_i/\lambda_e \text{ where } \lambda_e = \lambda_{eff}/n$$

d. If $i = n$ go next, otherwise set $i = i + 1$ and go to STEP 3(c).

5.5 Merge Configurations

5.5.1 Model Assumptions

Merge configurations (see Figure 1.2) consist of n parallel nodes (first level) linked to a single node (second level). Let λ_i, μ_i, V_i, c_i^2, and N_i be the arrival rate, the service rate, the service time variance, the service time squared coefficient of variation, and queue capacity (including the one in service) for node i, respectively, where $i = 0, 1, 2, ..., n$ and $i = 0$ denotes the second level node. External arrivals occur only at the first level nodes and the interarrival times have general distributions. Service times at all nodes have general distributions. The arrival process at the second level node consists of the superposition of the n departure processes from the first level nodes. All buffers are of limited size and this leads to blocking. An item that just finished its service at one of the first level nodes will get blocked if the second level node is full at that time. If there are more than one blocked items at the same time, a blocking queue is formed and the "first blocked first released" rule applies. During the time of blocking the blocked node cannot serve new items. Buffer sizes, service rates and scvs, and external arrival rates can be different for different nodes in the system.

5.5.2 The Approximation Algorithm for the Merge Configuration

Let us consider the merge configuration shown in Figure 1.2. The analysis of the system starts from the second level node and then proceeds to the n first level nodes. The second level node is treated as a loss system with arrival process having rate $\lambda_0 = \sum_{i=1}^{n} \lambda_i$. Once more we distinguish two cases: (i) all the scvs of the external arrival processes are ≤ 1 and (ii) the scvs of the external arrival processes are assumed to be equal and > 1. In case (i) the arrival process at the second level node is considered to be Poisson. In case (ii) the scv of the arrival process at the second level node is assumed to be equal to the scv of the external arrival processes. It is assumed that the second level node does not get blocked so its service process is not revised. Using information obtained from the analysis of the second level node we revise the service processes of the first level nodes to take into account the blocking effect. Having revised the service processes of the first level nodes we then use the results of $G/G/1/N$ approximations to get the performance measures of the individual nodes. Next, the effective arrival rates are computed using the formula $\lambda_{eff,i} = \lambda_i(1 - P_i)$ and then the average sojourn times for each of the n branches of the network are obtained using the formula

$$ST_i = (L_i/\lambda_{eff,i}) + L_0/(\sum_{i=1}^{n} \lambda_{eff,i})$$

where P_i is the probability that node i is full, and L_i is the average queue length of node i $(i = 0, 1, ..., n)$.

The Algorithm

Consider a merge configuration with n first level nodes. The algorithm consists of the following steps:

STEP 1.

a. Analyse the second level node as a loss system with arrival process having rate $\lambda_0 = \sum_{i=1}^{n} \lambda_i$ and scv equal to 1 if the scvs of the external arrival processes are ≤ 1. If the scv of the external processes is greater than 1, then the scv of the arrival process at the second level node is equal to the scv of the external processes.

b. Store the values of L_0 and P_0.

STEP 2.

a. Set $i = 1$

b. Revise the service process of the ith first level node and then solve this single node system using a $G/G/1/N$ approximation. To revise the service process of the ith node use the following formulas

$(\frac{1}{\mu_i})_{new} = (\frac{1}{\mu_i})_{old} + P_0 E\{Y_0\}$

$(V_i)_{new} = (V_i)_{old} + P_0^2 V R_0$

where $E\{Y_0\}$ and VR_0 are the mean and variance of the residual time of the service time of node 0.

c. Store the values of L_i and P_i.

d. Compute the effective arrival rate to node i using the following formula

$\lambda_{eff,i} = \lambda_i(1 - P_i)$

e. If $i = n$ go to STEP 3, otherwise set $i = i + 1$ and go to STEP 2(b).

STEP 3.

a. Set $i = 1$

b. Calculate the average sojourn time through the network for the ith branch using the formula $ST_i = \frac{L_i}{\lambda_{eff,i}} + \frac{L_0}{\sum_{i=1}^{n} \lambda_{eff,i}}$

c. If $i = n$ go next, otherwise set $i = i + 1$ and go to STEP 3(b).

d. Output the results.

5.6 Numerical Results

To test the performance of the approximation algorithm, we considered many different types of network configurations, buffer sizes, and arrival and service processes. The results obtained from the algorithm are compared with those obtained by discrete simulation. For each point four simulation runs totalling 60,000 items were performed. We constructed 95% confidence intervals and the midpoints of these intervals were used in our results. The range of the [(confidence interval width)/(midpoint)]x100 of all constructed confidence intervals is 0.06%-16%. The algorithm provides estimates of the average sojourn time through the network. It is shown that the approximation gives good results for all types of networks examined. The relative errors were within 10% (20%) of the simulation results for most the experiments with light to moderate traffic (heavy traffic). Three types of networks were considered: tandem, split, and merge configurations. We used Coxian-2 probability distributions to represent distributions with squared coefficients of variation ≥ 0.50. The Erlang-2 and Exponential distributions are special cases of the Coxian-2 distribution. The formulas that give the first three moments of the Coxian-2 distribution are presented in Appendix A. We use Marie's formulas (Appendix A) to approximate the revised distributions of the service times as Coxian-2 distributions.

In all experiments the two single node approximation methods of Yao-Buzacott and Gelenbe are used. All average relative errors (ARE) mentioned in the following sections are absolute average relative errors. Denote by Ap-YB and Ap-G the approximation algorithm using Yao-Buzacott's and Gelenbe's methods, respectively. The following notations are used in the next tables:

Ex:	example number
Exp:	exponential
Ex. In.:	external interarrival
$(C_2)_i(xx)$:	service times of node i have Coxian-2 distribution with scv $= xx$
$(E_2)_i$:	service times of node i have Erlang-2 distribution
$(E_4)_i$:	service times of node i have Erlang-4 distribution
$C_2(xx)$:	external interarrival times have Coxian-2 distribution with scv $= xx$
E_2:	external interarrival times have Erlang-2 distribution
E_4:	external interarrival times have Erlang-4 distribution

The scvs of the Exponential, Erlang-2, and Erlang-4 distributions are 1, 0.5, and 0.25, respectively.

5.6.1 Tandem Configurations

We present ten Examples of tandem configurations. The descriptions of the ten systems and the average relative errors (ARE) are shown in Tables 5.1, and 5.2. The results for Examples 1 to 10 are illustrated in Figures 5.2 to 5.11, respectively. We observe that the algorithm gives good estimates of the average sojourn time through the network in most cases. The results are very good for low and moderate values of traffic intensity. As traffic increases both Ap-YB and Ap-G seem to overestimate simulation results. The method was also tested for systems with processes with low scvs (Examples 5, 6). As can be seen in Figures 5.6 and 5.7, the method still gives good results. The algorithm does not perform well in systems with high variability in service and arrival processes and in systems with big differences in the values of service rates of succesive nodes. These limitations are illustrated in the results of the Examples 9, and 10. We observe that for Example 9 (Figures 5.10), the algorithm gives poor results for even low traffic. For the system of Example 10, the results are good for low traffic ($\rho_1 \leq 0.30$). More numerical results are presented in Appendix D.

Table 5.1: Tandem configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
1	$\mu_1 = \mu_2 = 0.3$	$(E_2)_i$ $i = 1,2$	Exp.	$N_i = 8$ $i = 1,2$	10.8%	18.7%
2	$\mu_1 = 0.25, \mu_2 = 0.2$ $\mu_3 = 0.3, \mu_4 = 0.25$	$(E_2)_i$ $i = 1$ to 4	Exp.	$N_i = 2,$ $i = 1,3$ $N_i = 3,$ $i = 2,4$	7.4%	18.4%
3	$\mu_i = 0.3$ $i = 1$ to 6	$(E_2)_i$ $i = 1$ to 6	Exp.	$N_i = 4$ $i = 1$ to 6	9.6%	12.7%
4	$\mu_i = 0.13443751$ $i = 1,2$	$(C_2)_1(1.5),$ $(E_2)_2$	E_2	$N_i = 4$ $i = 1,2$	11.2%	3.9%
5	$\mu_i = 0.2$ $i = 1,2$	$(E_4)_1,$ $(E_2)_2$	E_4	$N_i = 5$ $i = 1,2$	3.5%	10.7%
6	$\mu_i = 0.2$ $i = 1,2$	$(E_2)_i$ $i = 1,2$	E_4	$N_i = 5$ $i = 1,2$	5%	6.1%

Table 5.2: Tandem configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
7	$\mu_1 = 0.13443751$ $\mu_2 = 0.1$	$(C_2)_i(1.5)$ $i = 1, 2$	$C_2(1.5)$	$N_i = 2$ $i = 1, 2$	6.6%	11.1%
8	$\mu_1 = 0.1086672$ $\mu_2 = 0.08693376$	$(C_2)_i(2)$ $i = 1, 2$	$C_2(2)$	$N_i = 4$ $i = 1, 2$	10.2%	12.7%
9	$\mu_i = 0.125$ $i = 1, 2$	$(C_2)_i(5)$ $i = 1, 2$	$C_2(2)$	$N_i = 2$ $i = 1, 2$	101%	43.5%
10	$\mu_1 = 0.2$ $\mu_2 = 0.1$	$(E_2)_i$ $i = 1, 2$	E_2	$N_i = 5$ $i = 1, 2$	24.2%	22.9%

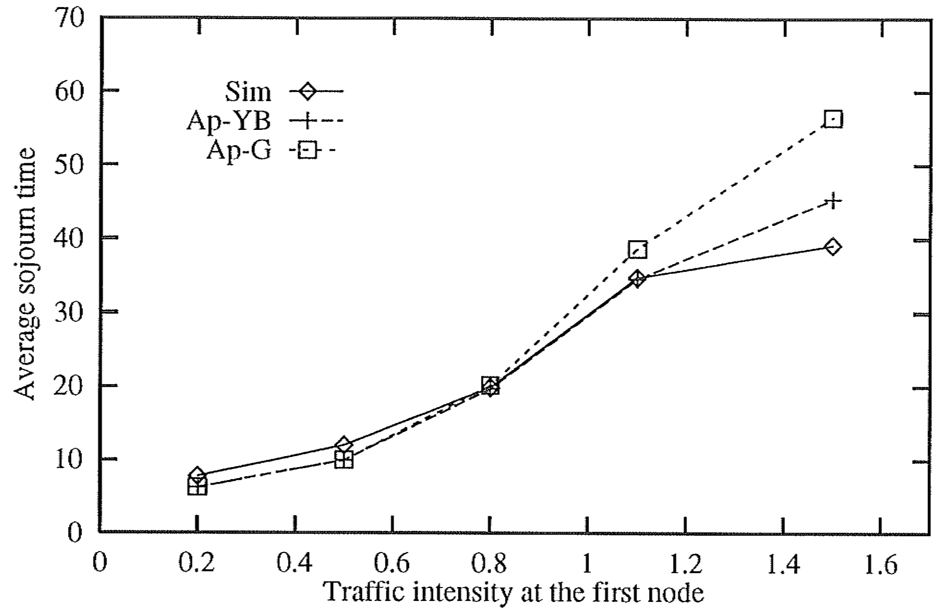

Figure 5.2: Results for a two-node tandem system

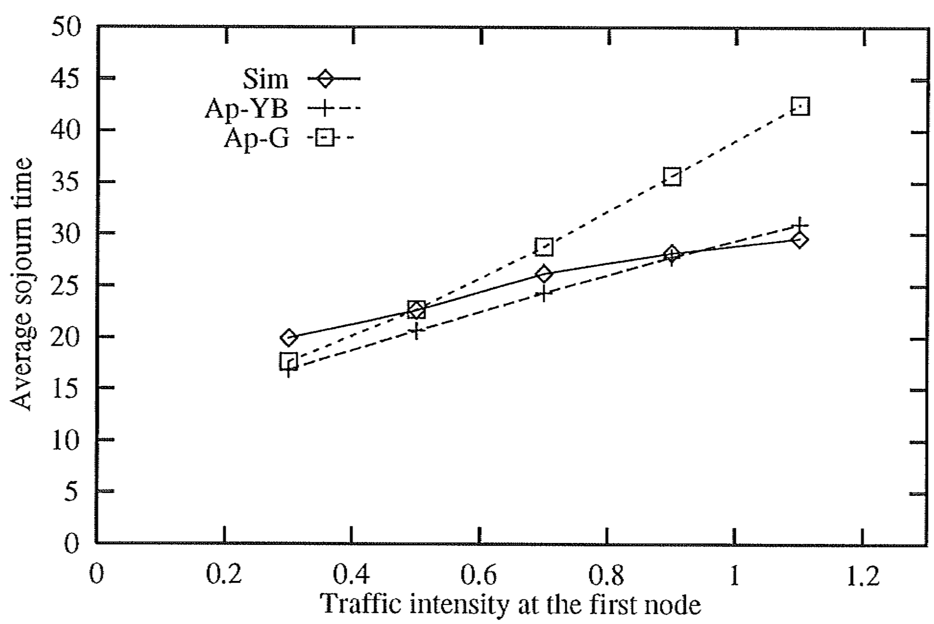

Figure 5.3: Results for a four-node tandem system

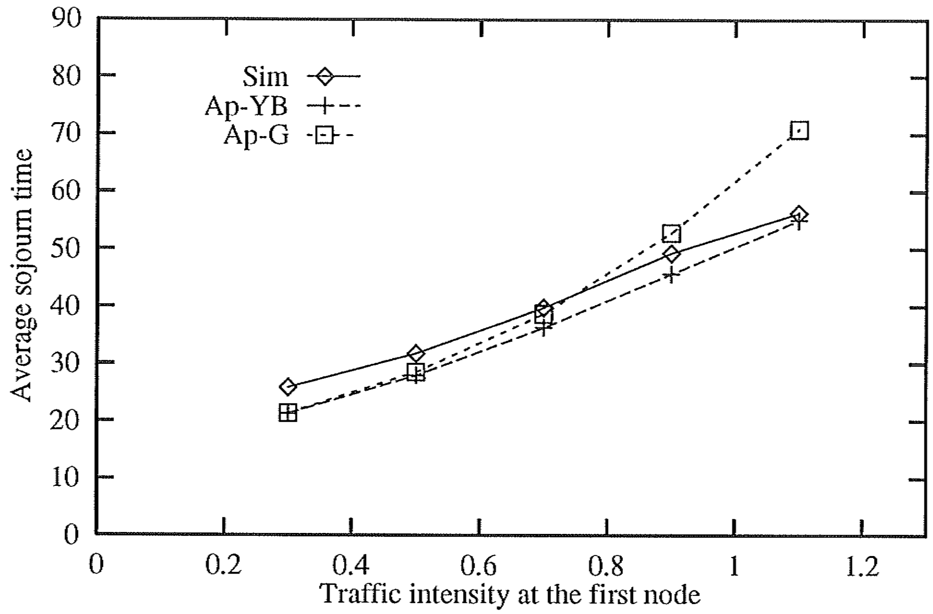

Figure 5.4: Results for a six-node tandem system

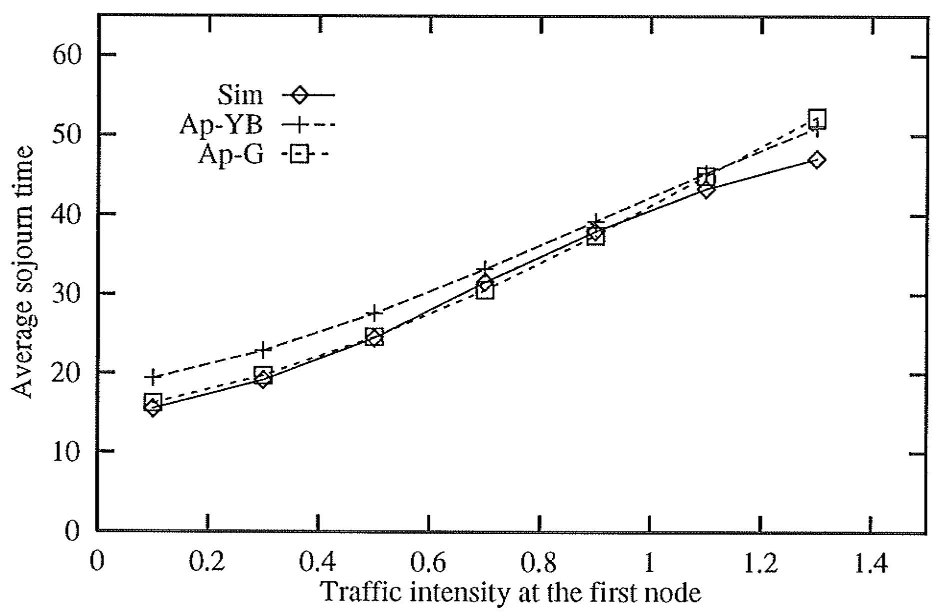

Figure 5.5: Results for a two-node tandem system

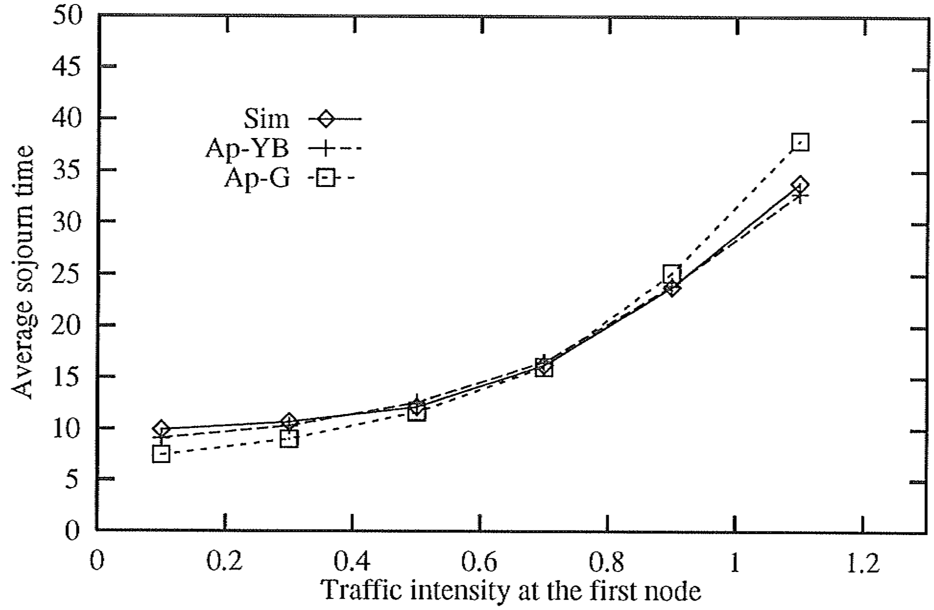

Figure 5.6: Results for a two-node tandem system

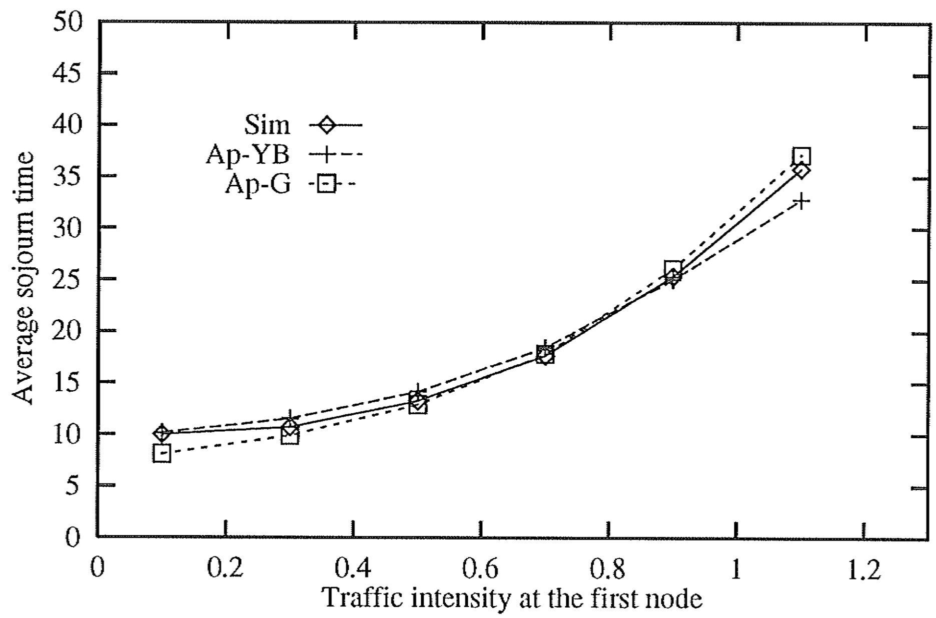

Figure 5.7: Results for a two-node tandem system

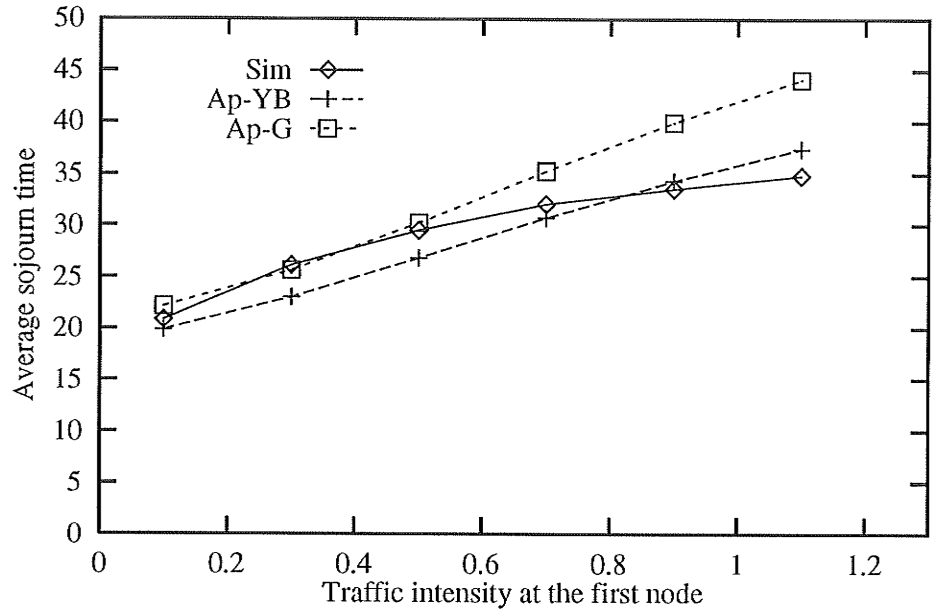

Figure 5.8: Results for a two-node tandem system

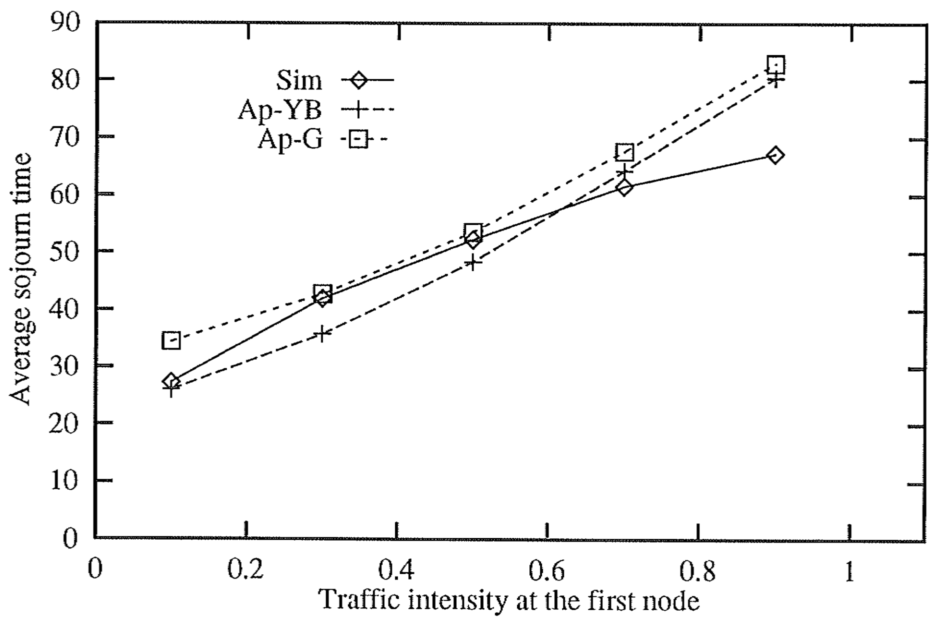

Figure 5.9: Results for a two-node tandem system

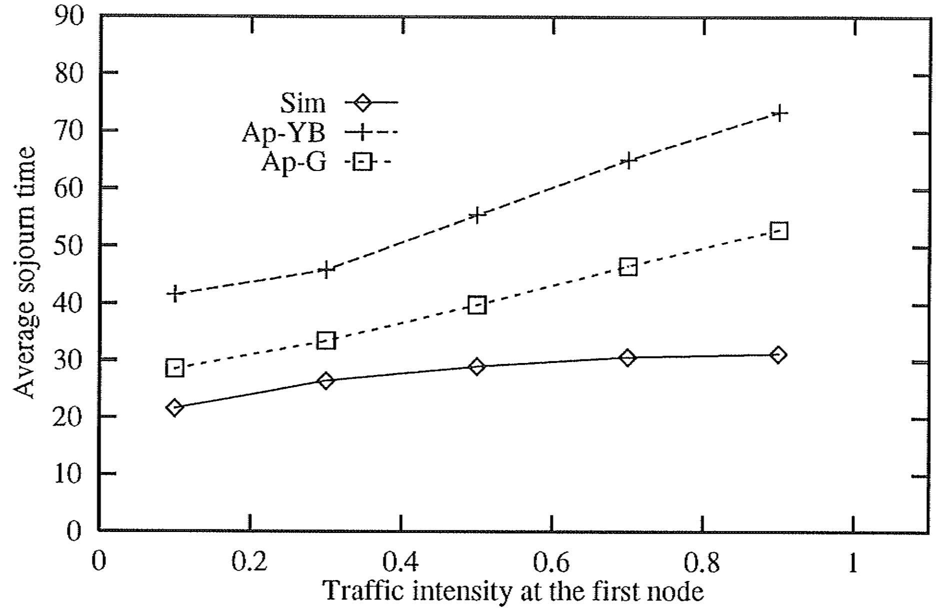

Figure 5.10: Results for a two-node tandem system

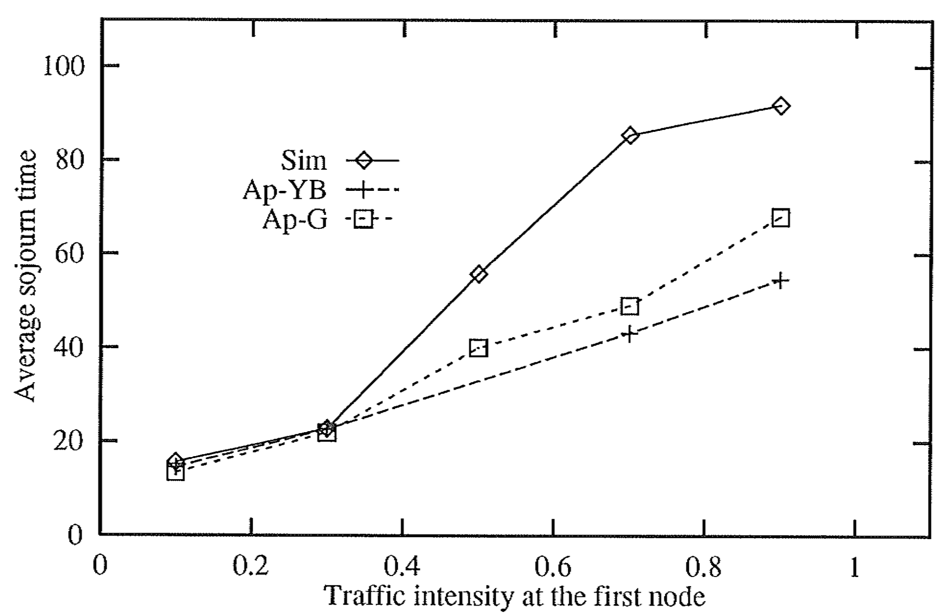

Figure 5.11: Results for a two-node tandem system

5.6.2 Split Configurations

We present seven Examples of split configurations. The descriptions of the seven systems and the average relative errors (ARE) are shown in Table 5.3. The results for Examples 11 to 17 are illustrated in Figures 5.12 to 5.18, respectively. We observe the algorithm gives good results for most of the Examples. As can be seen in Figures 5.12-5.18, the algorithm performs better for low and moderate traffic. It was observed that this method does not perform well in systems with high variability in their service and arrival processes (Example 17). This algorithm works for split systems that consist of second level nodes that are of the same type (which results in the equally splitting of the departure process from node 0 into n streams). Also, we have assumed that a blocked item at node 0 is forced to wait at node 0 until one of the second level nodes becomes not full. A more general approach would be to assign each item to a particular second level node. Thus, the blocked item would have to wait at node 0 until that particular second level node becomes not full. This is another limitation of this algorithm. Only one paper (Altiok and Perros [4]) has appeared dealing with split systems consisting of more than two second level nodes. Altiok and Perros assume that all service and external interarrival times have exponential distributions. They also assign a second level node to each item thus forcing a blocked item to wait at node 0 until its assigned second level node becomes not full. Thus their split system is slightly different than ours and thus a comparison between the two algorithms is not possible. More numerical results for split configurations are presented in Appendix D.

Table 5.3: Split configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
11	$\mu_0 = 0.25$ $\mu_1 = \mu_2 = 0.1$	$(E_2)_i$ $i = 0, 1, 2$	Exp.	$N_i = 3$ $i = 0, 1, 2$	7.7%	13.9%
12	$\mu_0 = 0.25$ $\mu_1 = \mu_2 = 0.1$	$(E_2)_i$ $i = 0, 1, 2$	Exp.	$N_i = 8$ $i = 0, 1, 2$	6.9%	9.8%
13	$\mu_0 = 0.5$ $\mu_i = 0.1, i = 1$ to 5	$(E_2)_i$ $i = 0$ to 5	Exp.	$N_i = 3$ $i = 0$ to 5	17.9%	27.9%
14	$\mu_i = 0.2$ $i = 0$ to 5	$(E_2)_i$ $i = 0$ to 5	Exp.	$N_i = 6$ $i = 0$ to 5	3.7%	10.6%
15	$\mu_0 = 0.13443751$ $\mu_i = 0.05, i = 1, 2$	$(C_2)_0(1.5)$ $(E_2)_i, i = 1, 2$	E_2	$N_i = 3$ $i = 0, 1, 2$	7.9%	6.9%
16	$\mu_0 = 0.13443751$ $\mu_i = 0.067218755,$ $i = 1, 2$	$(C_2)_i(1.5)$ $i = 0, 1, 2$	$C_2(1.5)$	$N_i = 4$ $i = 0, 1, 2$	16.5%	27.9%
17	$\mu_0 = 0.125$ $\mu_i = 0.0625, i = 1, 2$	$(C_2)_i(5)$ $i = 0, 1, 2$	$C_2(2)$	$N_i = 2$ $i = 0, 1, 2$	211%	101%

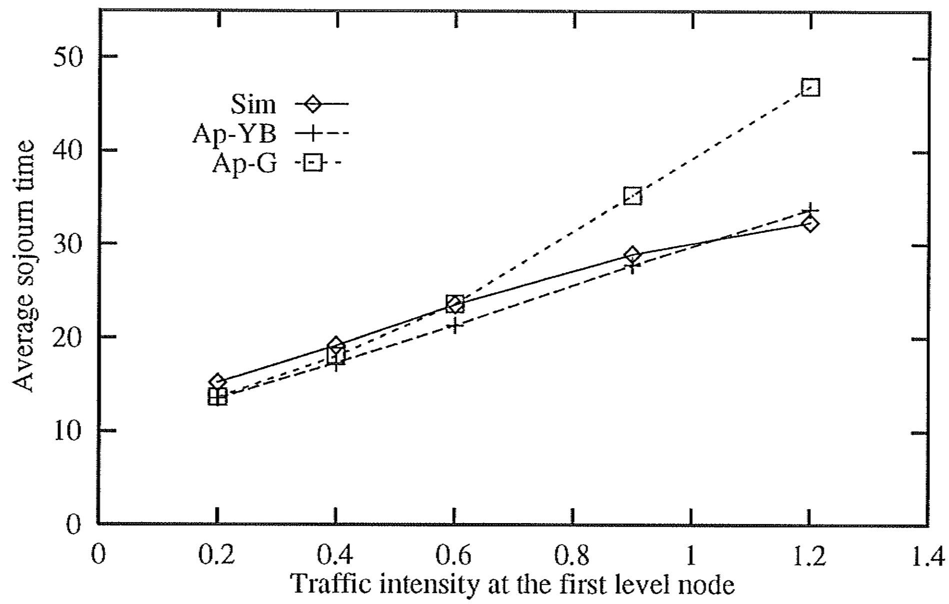

Figure 5.12: Results for a split system with two second level nodes

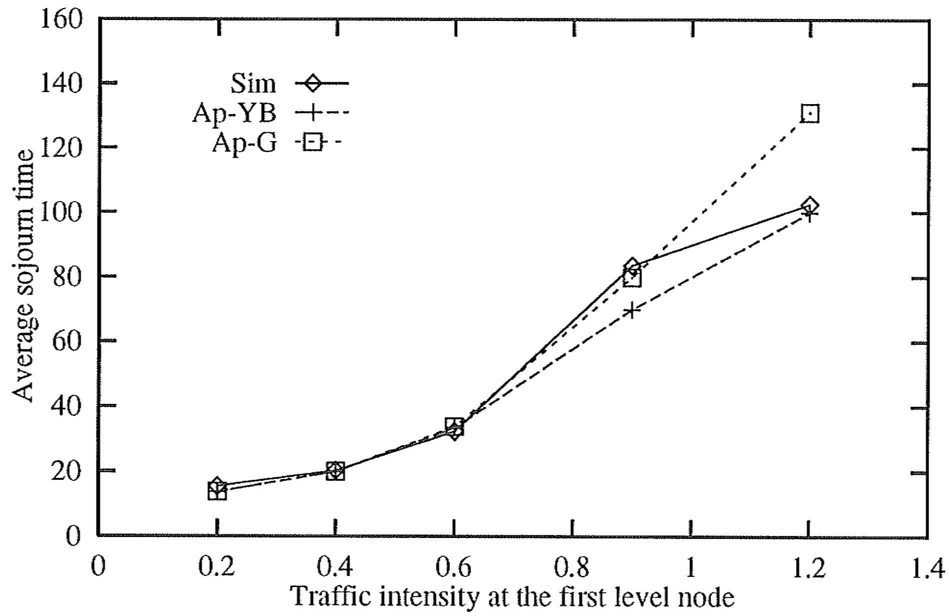

Figure 5.13: Results for a split system with two second level nodes

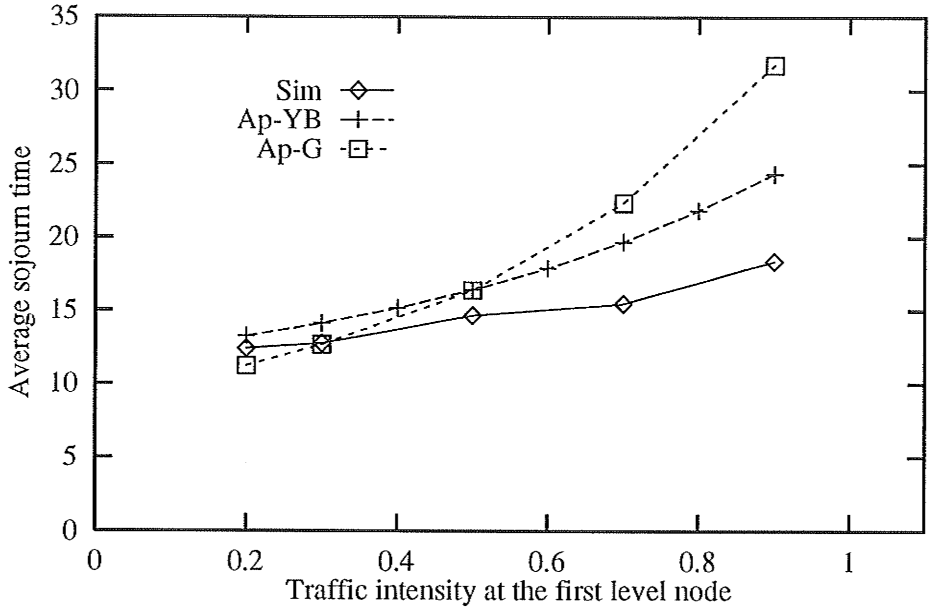

Figure 5.14: Results for a split system with five second level nodes

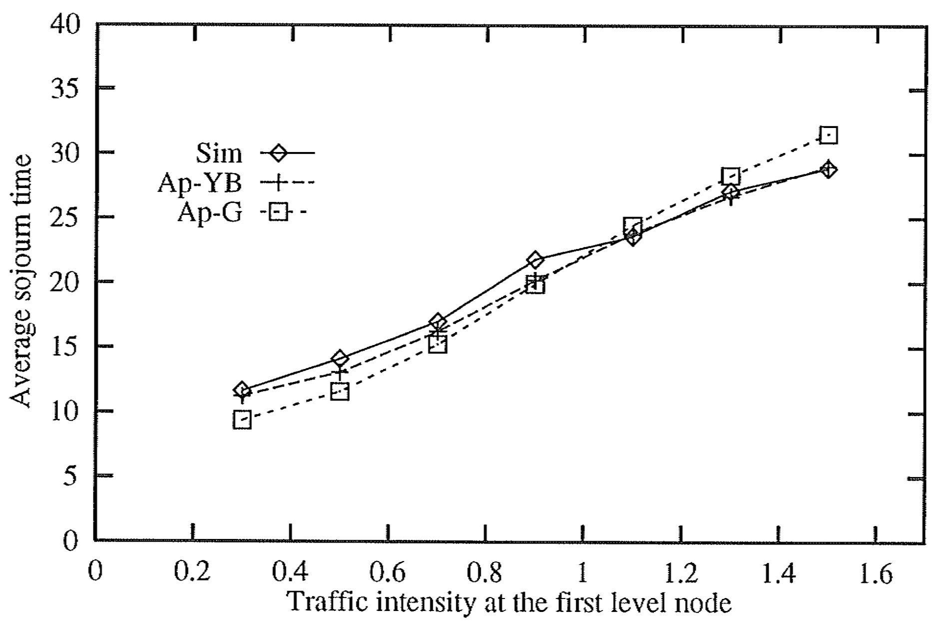

Figure 5.15: Results for a split system with five second level nodes

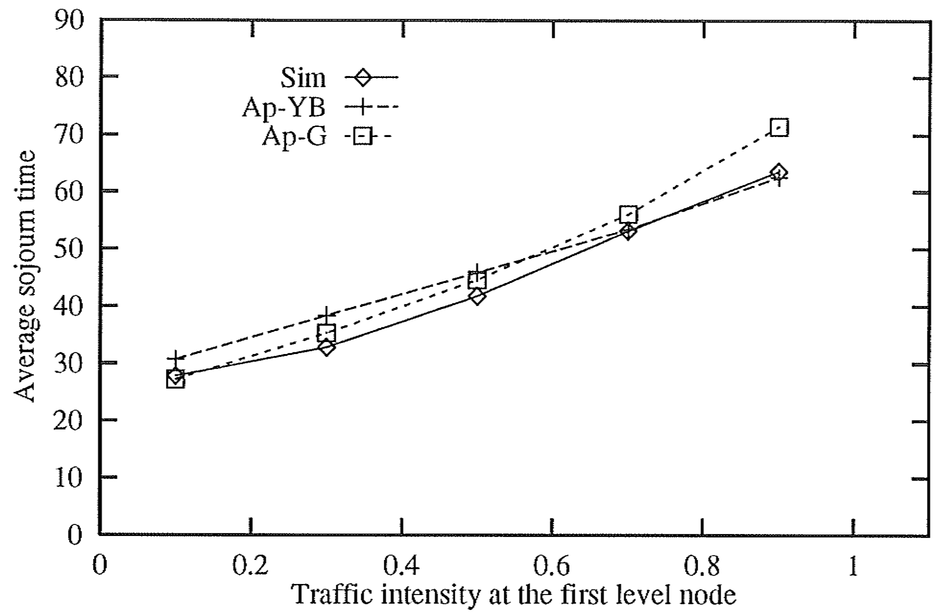

Figure 5.16: Results for a split system with two second level nodes

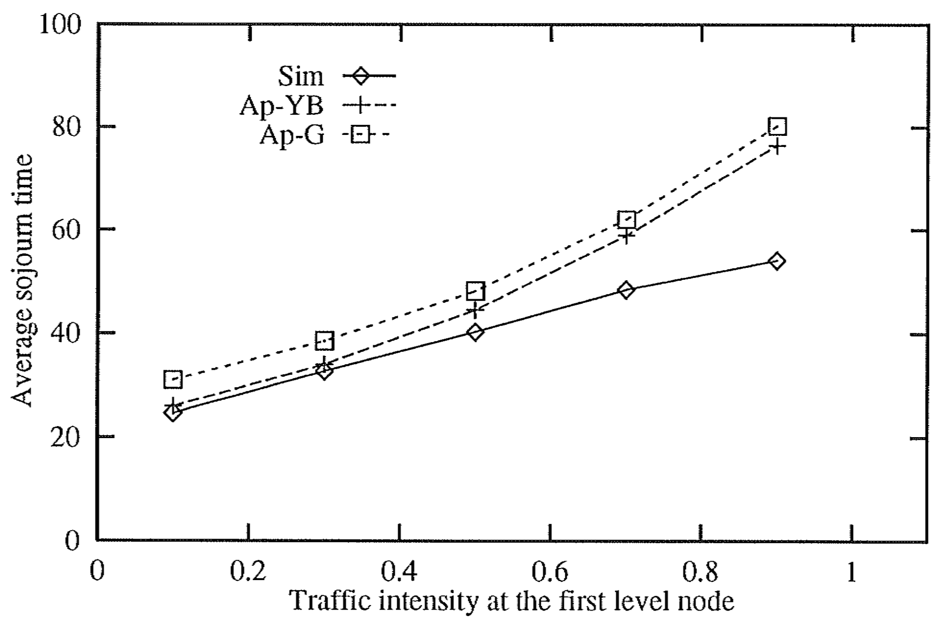

Figure 5.17: Results for a split system with two second level nodes

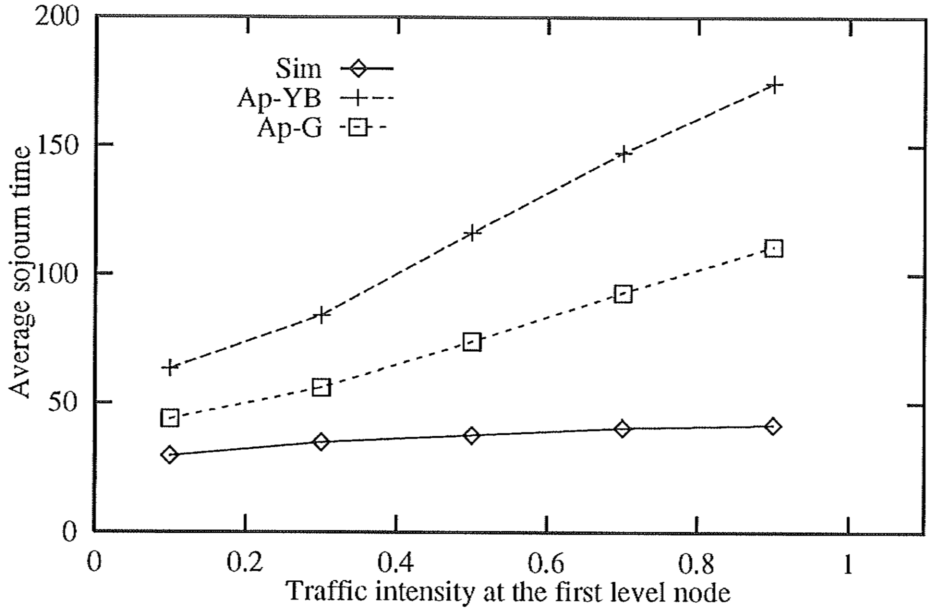

Figure 5.18: Results for a split system with two second level nodes

5.6.3 Merge Configurations

We first present seven Examples of merge configurations. The descriptions of the seven systems and the average relative errors (ARE) are shown in Table 5.4. The results for Examples 18 to 24 are illustrated in Figures 5.19 to 5.25, respectively. As can be seen, the algorithm gives good approximations of the average sojourn time for most Examples. In some cases (Examples 21, 23), the average relative errors are small for even heavy traffic. Once more, we observe that the method does not perform well in systems with high variability in their arrival and service processes (Example 24).

Only two papers (Altiok and Perros [4], and Lee and Pollock [34]) have appeared in the literature that deal with merge configurations that consist of more than two first level nodes. Both these papers assume exponential service and external interarrival times. Our goal is to introduce a simple, and quick approximation algorithm which can work with general distributions and give good results. Exponential networks is only a class of problems that our algorithm can handle. Nevertheless, the performance of our algorithm will be tested against these two other methods in order to see how close our results are to those obtained by other more complex methods. For this, three Examples taken from Altiok and Perros will be used.

First consider a merge system with two first level nodes. The service rates, arrival rates, and buffer sizes are $\mu_1 = 5, \mu_2 = 3, \mu_0 = 7$, $\lambda_1 = 4, \lambda_2 = 2$, and $N_1 = 4, N_2 = 2, N_0 = 4$, respectively. The estimates for the average sojourn time through each branch are shown in Table 5.5. Ap-Y, Ap-G, AP, and LP stand for our algorithm with Yao-Buzacott's method, our algorithm with Gelenbe's method, Altiok and Perros method, and Lee and Pollock method, respectively. We see that our algorithm gives results that are not too far from the ones obtained by the other two methods, even though our method is simpler and faster to implement and in addition able to handle more than exponential distributions.

Next we consider a merge system with four first level nodes. The service rates, arrival rates, and buffer sizes are $\mu_1 = 4, \mu_2 = 5, \mu_3 = 6, \mu_4 = 7, \mu_0 = 20$, $\lambda_1 = 2, \lambda_2 = 3, \lambda_3 = 4, \lambda_4 = 5$, and $N_1 = N_2 = N_3 = N_4 = 3, N_0 = 5$, respectively. The results are shown in Table 5.6. We see that our algorithm gives results that are very close to the results of the other two methods.

The third merge system consists of four first level nodes. The service rates, arrival rates, and buffer sizes are $\mu_1 = \mu_3 = 3, \mu_2 = \mu_4 = 2, \mu_0 = 8$, $\lambda_1 = \lambda_3 = 2, \lambda_2 = \lambda_4 = 1$, and $N_1 = N_2 = N_3 = N_4 = N_0 = 5$, respectively. The results are illustrated in Table 5.7. We observe that our algorithm gives better estimates for branches 2 and 4 than the other two methods.

Table 5.4: Merge configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
18	$\mu_0 = 0.3$ $\mu_i = 0.1, i = 1,2,3$	$(E_2)_i$ $i = 0$ to 3	Exp.	$N_i = 7$ $i = 0$ to 3	7.6%	12.3%
19	$\mu_0 = 0.5$ $\mu_i = 0.1, i = 1$ to 4	$(E_2)_i$ $i = 0$ to 4	Exp.	$N_i = 5$ $i = 0$ to 4	12.2%	10.6%
20	$\mu_0 = 0.4$ $\mu_i = 0.2, i = 1,2$	$(C_2)_0(1.5)$, $(E_2)_i, i = 1,2$	E_2	$N_i = 3$ $i = 0,1,2$	7.4%	6.2%
21	$\mu_0 = 0.22$ $\mu_i = 0.13443751$, $i = 1,2$	$(C_2)_i(1.5)$ $i = 0,1,2$	$C_2(1.5)$	$N_i = 6$ $i = 0,1,2$	3.2%	10%
22	$\mu_0 = 0.35$ $\mu_i = 0.13443751$, $i = 1,2,3$	$(C_2)_i(1.5)$ $i = 0$ to 3	$C_2(1.5)$	$N_i = 2$ $i = 0$ to 3	3.9%	4.1%
23	$\mu_0 = 0.25$, $\mu_i = 0.05$, $i = 1$ to 4	$(C_2)_i(1.5)$ $i = 0$ to 4	$C_2(1.5)$	$N_i = 2$ $i = 0$ to 4	2%	5.6%
24	$\mu_0 = 0.25$ $\mu_i = 0.125$ $i = 1,2$	$(C_2)_i(5)$ $i = 0,1,2$	$C_2(2)$	$N_i = 2$ $i = 0,1,2$	78.3%	28.9%

Table 5.5: Average sojourn time

	Average sojourn time				
Branch	Ap-YB	Ap-G	AP	LP	Exact
1	0.83	0.84	0.87	0.86	0.88
2	0.80	0.83	0.83	0.85	0.87

Table 5.6: Average sojourn time

	Average sojourn time				
Branch	Ap-YB	Ap-G	AP	LP	Simulation
1	0.44	0.51	0.51	0.52	0.52
2	0.46	0.46	0.46	0.46	0.47
3	0.41	0.41	0.41	0.41	0.41
2	0.38	0.38	0.38	0.38	0.38

Table 5.7: Average sojourn time

	Average sojourn time				
Branch	Ap-YB	Ap-G	AP	LP	Simulation
1	1.14	1.06	1.15	1.51	1.17
2	1.22	1.22	1.29	1.67	1.24
3	1.14	1.06	1.15	1.51	1.17
2	1.22	1.22	1.29	1.67	1.24

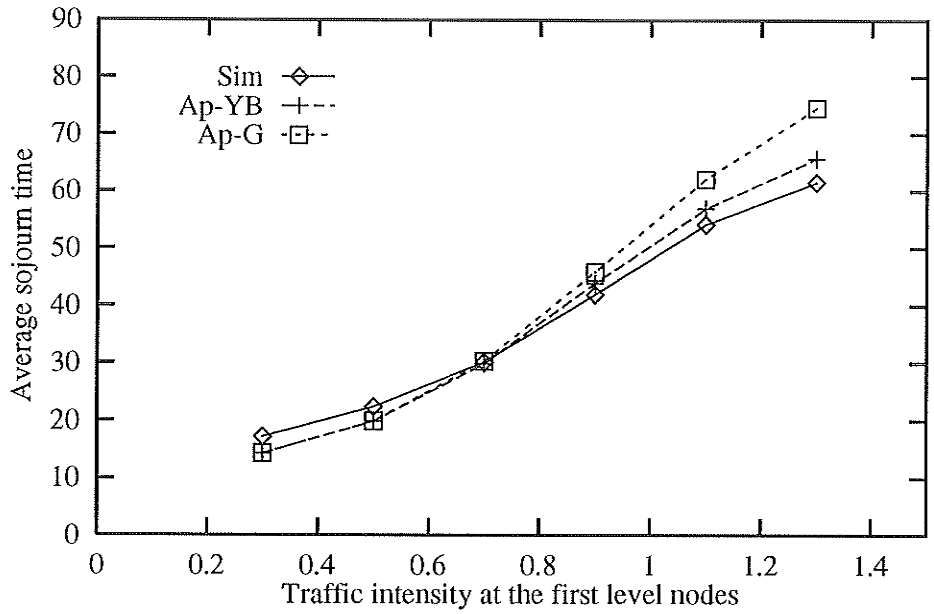

Figure 5.19: Results for a merge system with three first level nodes

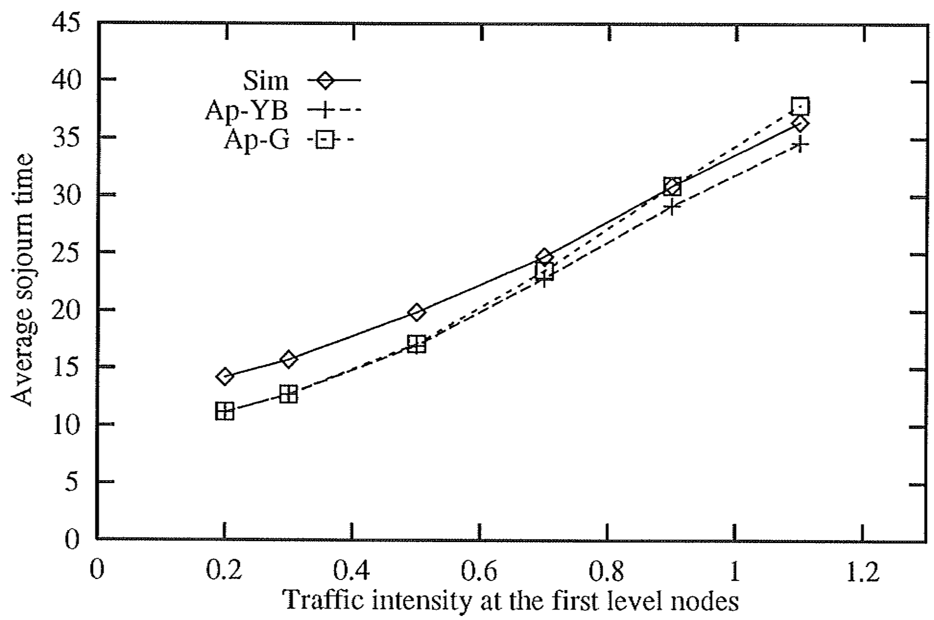

Figure 5.20: Results for a merge system with four first level nodes

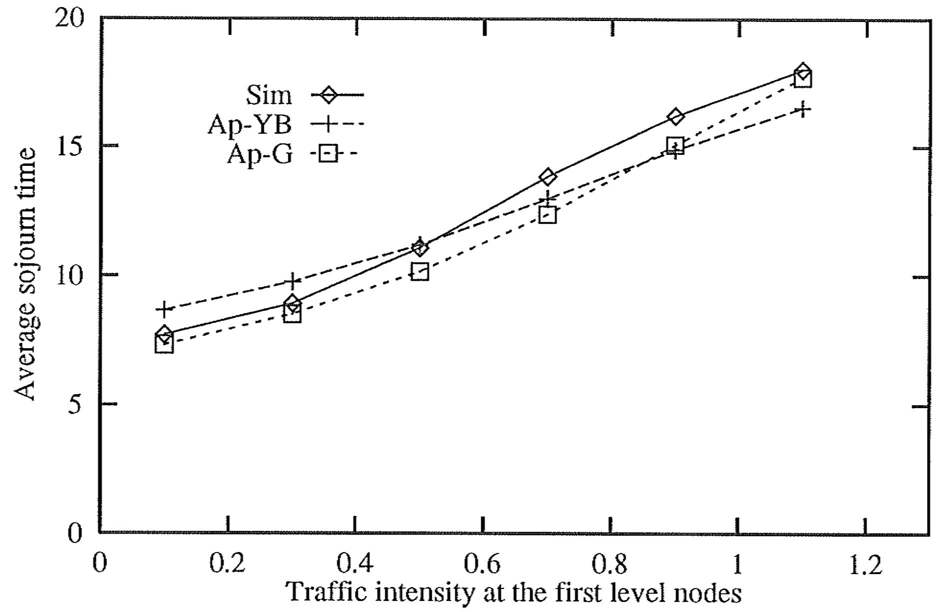

Figure 5.21: Results for a merge system with two first level nodes

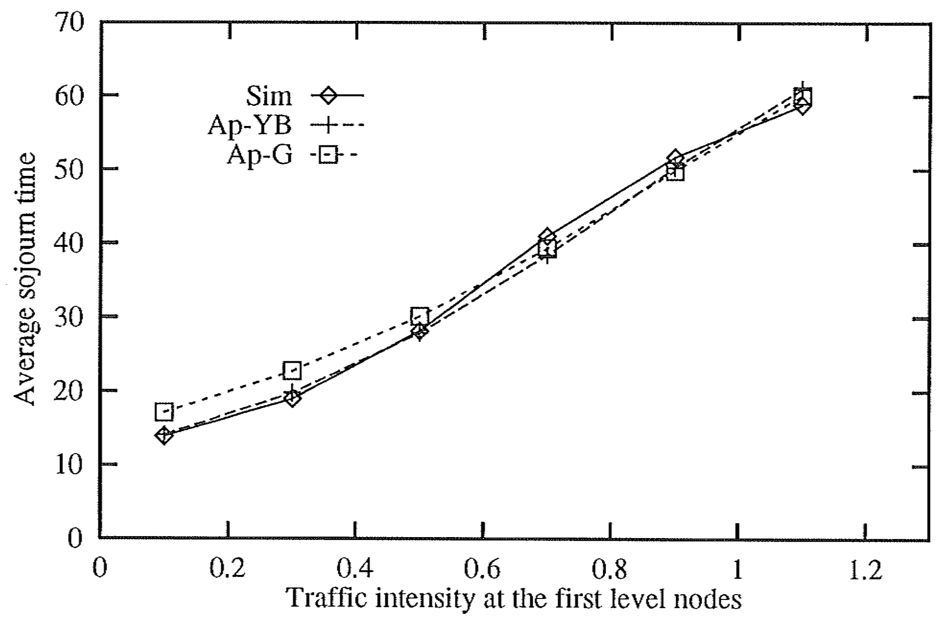

Figure 5.22: Results for a merge system with two first level nodes

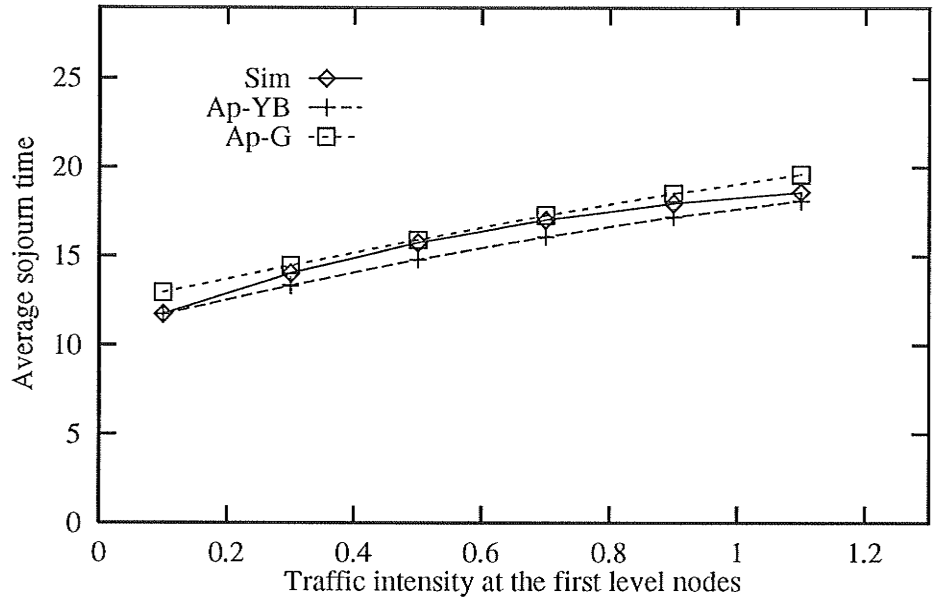

Figure 5.23: Results for a merge system with three first level nodes

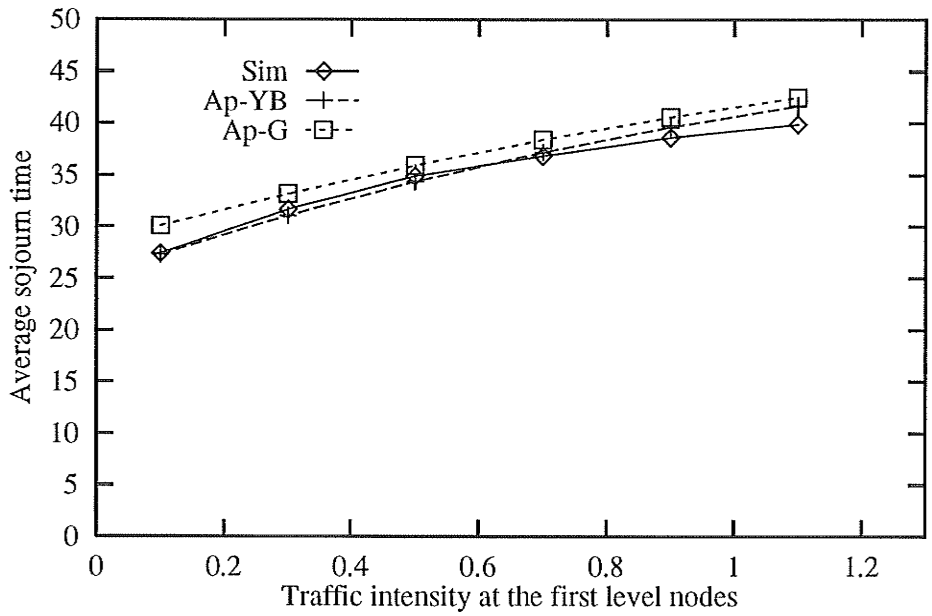

Figure 5.24: Results for a merge system with four first level nodes

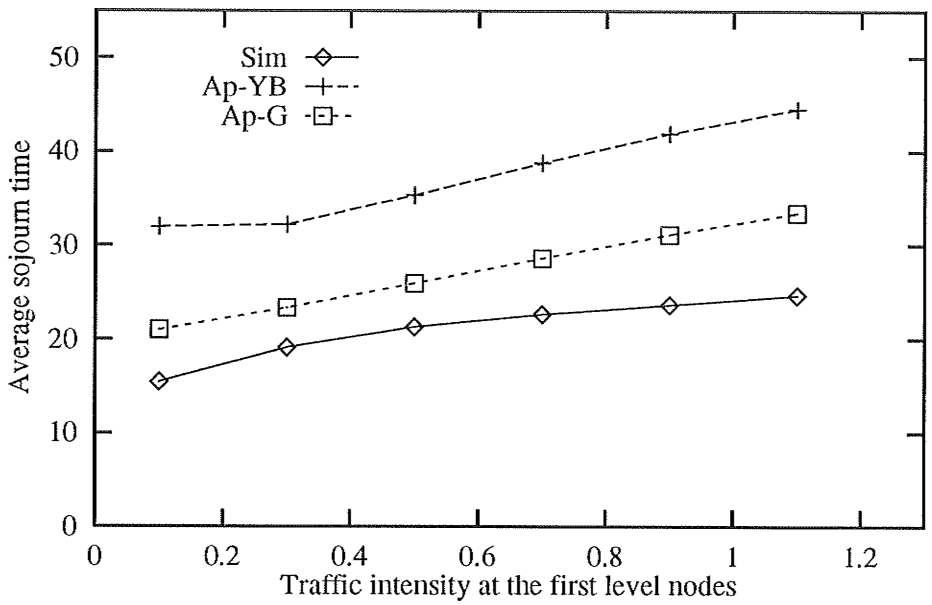

Figure 5.25: Results for a merge system with two first level nodes

5.6.4 The Effect of the Two Single Node Approximation Methods on the Performance of the Algorithm

As has been shown through many examples the performance of the proposed algorithm depends on the single node approximation method used. Let us focus on three cases: (i) All scvs of all processes are less than 1. It was seen that for low traffic ($\rho_1 < 0.5$) the results obtained by the algorithm using the two single node approximations are very close. However, as traffic increases the performance of the algorithm seems to depend on which single node approximation is used. More specifically, when Yao-Buzacott's method is used, the algorithm seems to give reasonably good results. On the other hand when Gelenbe's method is used the algorithm seems to overestimate the average sojourn time when compared to simulation results. Thus, we suggest the use of Yao-Buzacott's method when all scvs are less than 1. (ii) All scvs are ≤ 2. In this case it was observed that in some experiments Ap-YB performed better than Ap-G and in some experiments Ap-G gave better results. However, the results obtained by Ap-YB and Ap-G were close to each other for low and moderate traffic. (iii) Some or all scvs are greater than 2. In this case both Ap-YB and Ap-G do not give good results and overestimate simulation results. Yao and Buzacott in their paper stated that their method does not give good results in systems with high variability in their arrival and service processes. We observed that although Ap-G overestimated simulation results it gave better results than Ap-YB in these cases.

5.7 Summary

A simple, and quick approximation method for the analysis of tandem, split, and merge configurations with general processes was presented in this Chapter. As was stated earlier, the motivation for the development of this method is to help the analyst to deal with some of the weaknesses (i.e high computational time and computer memory requirements, complex structure) of the existing approximation methods. The proposed algorithm was tested for many combinations of probability distributions, buffer sizes, and queue configurations. The advantages of the algorithm can be summarized as follows:

- The proposed method yields good estimates of the average sojourn time through the network and the relative errors are within 10% (20%) of the simulation results for moderate (heavy) traffic.
- It is very fast. Usually the computational time required by other existing approximation methods increases with the increase in the number of nodes in the system. However, the size of the system does not seem to affect the speed of our algorithm. For all examples considered, the required CPU time was less than 0.1 seconds. The programs are written in FORTRAN 77 and were run on Sun workstations.
- It can be used for the analysis of tandem, split, and merge systems.
- There are no limitations on the types of probability distributions involved.
- It is the first method to present results for split, and merge systems consisting of more than three nodes with general stochastic processes.

The results of this Chapter have been submitted for publication to INFOR (Yannopoulos and Alfa [52]).

CHAPTER 6

Discussion and Future Research

6.1 Discussion and Conclusions

In this thesis, the research has focused on the analysis of open queueing networks. Open queueing networks are used to represent many industrial and service activities (for example production systems). Exact analysis is only possible for small, and simple networks. For large, and complicated systems, approximation methods are employed. Most of the existing approximations work only for simplified versions of the actual queueing systems. The reason is that finite buffers, general probability distributions, and blocking make the analysis very difficult. In this thesis three approximation methods were presented, and some of the concluding points are listed below:

- Real life systems are combinations of tandem, split, and merge networks with finite buffers and general probability distributions. One would expect to find an approximation method which could be used for the analysis of any type of real life systems. But this is not the case. Most of the existing approximations are designed to work only for specific types of networks (for example tandem). Furthermore, the types of systems that have been considered are simplified networks with convenient assumptions about the nature of the traffic and service processes. Real life problems have not been analysed yet. Thus, in

Chapter 3, an approximation method designed to work for the analysis of any real life problem was proposed. The proposed method transforms the actual system into an equivalent system with no buffer capacity constraints and with revised service times. This method was applied to the modelling of a part of a conveyor system installed in a large manufacturing company. It was shown that the results obtained by the approximation were good for light, and moderate traffic. However, for heavy traffic, the method gave poor estimates of the system performance measures.

- Numerous methods have been published dealing with queues in tandem. These methods differ in the types of tandem systems considered and in the accuracy of their results. In todays highly competitive world, a slight improvement in efficiency can result in significant reduction in operating costs. Thus, there is the need for the development of new methods which are more accurate and give more information for the system examined. The Brandwajn and Jow' (BJ) approximation seems to yield the most accurate results for exponential tandem networks. In Chapter 4, BJ' method is extended by considering solution cells that consist of triplets instead of pairs of adjacent nodes. This new method seems to give improved results when compared to the ones obtained by BJ's method. Another advantage of the proposed method is that it provides the joint steady state probability distributions of the number of items at each node for triplets of adjacent nodes, information that can not be obtained by other approximation methods.

- Most of the existing approximation methods are time consuming, have complicated structures, and require considerable computer memory space. These disadvantages become obvious when these methods are involved in optimization procedures for which execution time is very important. Also, as was noted earlier, most of the existing methods deal only with specific networks configurations, and probability distributions. Furthermore, results for split, and merge systems consisting of more than three nodes with general processes have not been reported yet. Thus in Chapter 5, an approximation algorithm is proposed, which is simple, quick, and can be used for the analysis of tandem, split, and merge configurations with general processes. As was shown in Chapter 5, the results for most of the experiments are reasonably good. This is the first time that results for split, and merge configurations with more than three nodes and with general processes are reported.

6.2 Future Research

The three approximation methods presented in this thesis can be extended to give improved results and or to include more types of networks. More specifically:

- As was discussed in Chapter 3, the performance of the approximation method developed for real life systems, depends on the performance of sub-approximations (i.e GEM, single-node approximations) that were used as components of the method. It is expected that by including improved sub-approximations as components of the method, the whole approximation method will give better results. Another possible future development, would be the modification of the method to give good results for heavy traffic.

- Further improvement of the approximation method of Chapter 4 is difficult. This method could be possibly improved by considering cells that consist of four adjacent nodes instead of three. This could lead to improved results, but it would also lead to excessive computational time requirements. This is because, at each step of each iteration, a system of four nodes could have to be solved. However, a possible extension of this method, could be the development of an approximation for split, and merge systems. A three node solution approach could be easily used (for example for a merge system, a three node cell could consist of two first level nodes and the second level node).

- The approximation method presented in Chapter 5 could be modified: i) to give improved results for cases with probability distributions having large values of squared coefficients of variation (scv > 2) ii) to give improved results for cases with heavy traffic iii) to give improved results in cases where there is a big difference in the service rates of successive nodes iv) in order to be used in the analysis of a wider variety of queueing networks (for example arbitrary queue configurations) v) to be used in the analysis of multi-server systems and closed networks.

APPENDICES

APPENDIX A

Erlang-k, Hyperexponential-k, and Coxian-k Distributions

The Erlang-k (E_k) probability distribution can be viewed as the sum of k exponential distributions. Figure A.1 illustrates an Erlang distribution with k stages, with each stage being an exponential distribution with parameter $k\mu$. The density function of the Erlang-k distribution is given by the formula:

$$b(x) = \frac{k\mu(k\mu x)^{k-1} e^{-k\mu x}}{(k-1)!}$$

The squared coefficients of variation of all Erlang distributions are less than 1. The squared coefficient of variation of the Exponential distribution is equal to 1.

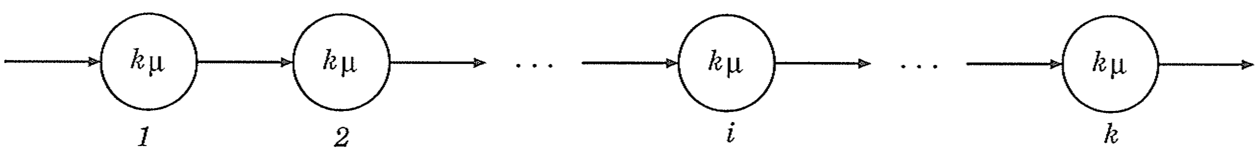

Figure A.1: A k stage Erlang distribution

A family of distributions with squared coefficients of variation greater than 1 is the Hyperexponential-k (H_k) family. The Hyperexponential-k distribution can be viewed as a parallel arrangement of k stages each having an exponential distribution (Figure A.2). The density function of the H_k distribution is given by the formula:

$$b(x) = a_1\mu_1 e^{-\mu_1 x} + ... + a_i\mu_i e^{-\mu_i x} + ... + a_k\mu_k e^{\mu_k x}$$

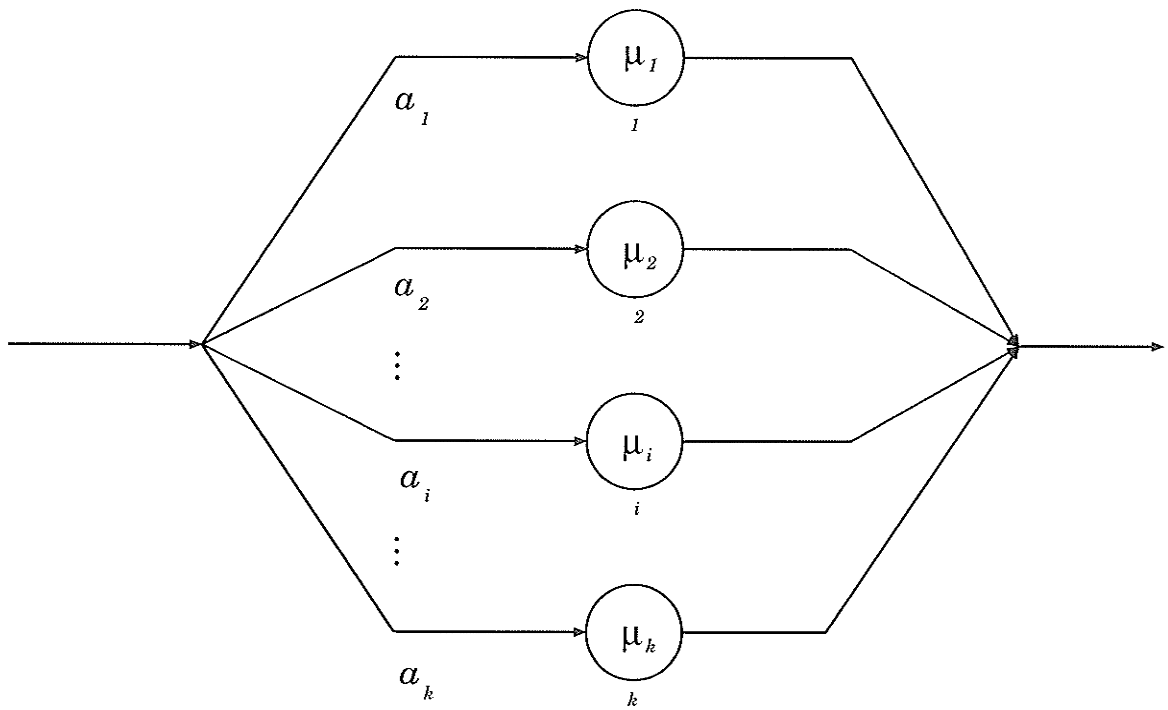

Figure A.2: A k stage Hyperexponential distribution

The Coxian-k distribution (C_k) consists of k exponential stages with parameters $\mu_1, ..., \mu_k$. An item receives service at the ith stage and then with probability a_i receives a new service at the $(i+1)$th stage. The Coxian-2 distribution is illustrated in Figure A.3. An item first receives service at the first stage and then with probability p receives service at the second stage.

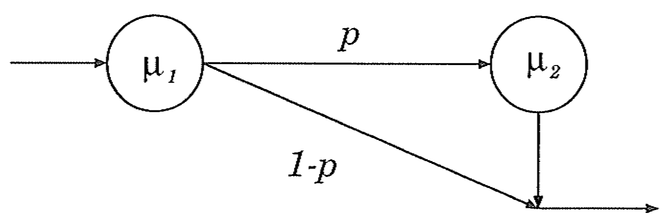

Figure A.3: The Coxian-2 distribution

The first three moments m_1, m_2, m_3 of the C_2 distribution are as follows (Yao and Buzacott [55]):

$$m_1 = \frac{1}{\mu_1} + \frac{p}{\mu_2}$$

$$m_2 = \frac{2}{\mu_1^2} + \frac{2p(\mu_1 + \mu_2)}{\mu_1 \mu_2^2}$$

and

$$m_3 = 6/\mu_1^3 + 6p(\mu_1^2 + \mu_1\mu_2 + \mu_2^2)/(\mu_1^2\mu_2^3)$$

The mean (μ) and variance (V) of the C_2 distribution are given as:

$$\mu = \frac{1}{\mu_1} + \frac{p}{\mu_2}$$

and $V = m_2 - m_1^2$ which can be written as:

$$V = \frac{1}{\mu_1^2} + p(2-p)\frac{1}{\mu_2^2}$$

The squared coefficient of variation c^2 of the C_2 is given below:

$$c^2 = \frac{\mu_2^2 + p(2-p)\mu_1^2}{(\mu_2 + p\mu_1)^2}$$

and

$$c^2 = \frac{1 + p(2-p)r^2}{(1+pr)^2}$$

where

$$r = \frac{\mu_1}{\mu_2}$$

Marie's approximation

To approximate a general distribution (when only the mean and scv are known) by a Coxian-2 distribution, Marie [36] proposed the following formulas:

$$\mu_1 = 2\mu, \quad p = 0.5/c_s^2, \quad \mu_2 = \mu_1 p$$

where μ and c_s^2 are the mean and scv of the general distribution.

APPENDIX B

The Multivariable Newton Method

Let us suppose that we want to solve the following system of simultaneous nonlinear equations (see Yakowitz and Szidarovszky [49]):

$$f_i(x_1, ..., x_n) = 0, \quad (i = 1, 2, ..., n)$$

Let us introduce the Jacobian matrix $\mathbf{J}(\mathrm{x})$, the $(i,j)th$ element of which is defined to be

$$\mathbf{J}_{ij}(\mathrm{x}) = \frac{\partial}{\partial x_j} f_i(x_1, x_2, ..., x_n)$$

Define $\mathbf{J}^k = \mathbf{J}(\mathrm{x}^{(k)})$ to be the Jacobian matrix of $(f_1(\mathrm{x}), ..., f_n(\mathrm{x}))$, evaluated at the kth iteration estimate $\mathrm{x}^{(k)}$, and introduce the vector

$$\mathrm{f}^{(k)} = \begin{bmatrix} f_1(\mathrm{x}^{(k)}) \\ \vdots \\ f_n(\mathrm{x}^{(k)}) \end{bmatrix}$$

Then according to the multivariable Newton method we have:

$$\mathrm{x}^{(k+1)} = \mathrm{x}^{(k)} - (\mathbf{J}^{(k)})^{-1}\mathrm{f}^{(k)}$$

and

$$\mathbf{J}^{(k)}(\mathrm{x}^{(k+1)} - \mathrm{x}^{(k)}) = -\mathrm{f}^{(k)}$$

which is solved by Gauss elimination methods.

APPENDIX C

The Single Node Approximation methods of Yao-Buzacott and Gelenbe

The two single node approximation methods of Yao-Buzacott and Gelenbe are presented. The formulas are taken from Springer and Makens [46]. Denote by k, λ, c_a^2, μ and c_s^2 the buffer size (including the one in service), arrival rate, scv of the arrival process, service rate, and scv of the service process respectively. Define

$$\rho = \lambda/\mu$$

$$\beta = \lambda - \mu$$

$$\alpha = \lambda c_a^2 + \mu c_s^2$$

$$\gamma = 2\beta/\alpha$$

The method of Yao and Buzacott

The approximation of the probability p_0 of the system being empty is given as:

$$p_0 = \begin{cases} \frac{(1-\rho)(c_a^2+1)}{\rho(1-c_a^2)+(c_a^2+1)-\rho e^{\gamma(k-1)}(\rho(c_s^2+1)+(1-c_s^2))} & (\rho \neq 1), \\ \\ \frac{1}{1+\frac{c_s^2+1}{c_a^2+1}+\frac{4(k-1)}{(c_a^2+1)(c_a^2+c_s^2)}} & (\rho = 1) \end{cases}$$

The approximation of the probability p_k the system is full is given as:

$$p_k = \frac{\rho p_0 e^{\gamma(k-1)}(c_s^2+1)}{c_a^2+1}$$

The approximation of the mean number L_s of customers in the system is given as:

$$L_s = \begin{cases} \frac{2\rho p_0(\frac{1}{\gamma}-\frac{1}{2})}{(\rho-1)(c_a^2+1)} + \frac{2p_k(k-\frac{1}{\gamma}-\frac{1}{2})}{(\rho-1)(c_s^2+1)} + p_k k & (\rho \neq 1), \\ \\ \frac{2k(k-1)+k(c_a^2+c_s^2)(c_s^2+1)}{4(k-1)+(c_a^2+c_s^2)(c_a^2+c_s^2+2)} & (\rho = 1) \end{cases}$$

The method of Gelenbe

The approximation of the probability p_0 is given below:

$$p_0 = \begin{cases} \frac{1-\rho}{1-\rho^2 e^{\gamma(k-1)}} & (\rho \neq 1), \\ \\ \frac{1}{2+2(k-1)/(c_a^2+c_s^2)} & (\rho = 1) \end{cases}$$

The approximation of p_k is given as:

$$p_k = \rho p_0 e^{\gamma(k-1)}$$

The approximation of L_s is given as:

$$L_s = \begin{cases} \frac{p_0\rho(\frac{1}{\gamma}-\frac{1}{2})+p_k(k-\frac{1}{\gamma}-\frac{1}{2})}{\rho-1} + p_k k & (\rho \neq 1) \\ \\ \frac{1}{2}k & (\rho = 1) \end{cases}$$

APPENDIX D

Additional Numerical Results

Additional numerical results for the simple and quick approximation algorithm, presented in Chapter 5, are displayed in this appendix.

D.1 Tandem Configurations

We present eleven more Examples for tandem configurations. The descriptions of these systems are shown in Tables D.1 and D.2. The results for Examples 1 to 11 are illustrated in Figures D.1 to D.11.

Table D.1: Tandem configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
1	$\mu_1 = \mu_2 = 0.3$	$(E_2)_i$ $i = 1, 2$	Exp.	$N_i = 3$ $i = 1, 2$	12.8%	12.7%
2	$\mu_i = 0.2$ $i = 1$ to 4	$(E_2)_i$ $i = 1$ to 4	Exp.	$N_i = 5$ $i = 1$ to 4	9.1%	13.1%
3	$\mu_i = 0.2$ $i = 1, 2$	$(E_2)_1$, $(C_2)_2(1.5)$	Exp.	$N_i = 4$ $i = 1, 2$	6.3%	11.1%
4	$\mu_i = 0.2$ $i = 1, 2$	$(E_2)_1$, $(C_2)_2(1.5)$	E_2	$N_i = 4$ $i = 1, 2$	9.6%	10%
5	$\mu_i = 0.13443751$ $i = 1, 2$	$(C_2)_1(1.5)$, $(E_2)_2$	Exp.	$N_i = 4$ $i = 1, 2$	4%	5.6%
6	$\mu_1 = 0.13443751$ $\mu_2 = 0.1$	$(C_2)_i(1.5)$ $i = 1, 2$	$C_2(1.5)$	$N_i = 7$ $i = 1, 2$	10.6%	11.4%

Table D.2: Tandem configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
7	$\mu_i = 0.13443751$ $i = 1, 3, \mu_2 = 0.1$	$(C_2)_i(1.5)$ $i = 1, 2, 3$	$C_2(1.5)$	$N_i = 2$ $i = 1, 3, N_2 = 4$	8.4%	24.3%
8	$\mu_i = 0.13443751$ $i = 1$ to 6	$(C_2)_i(1.5)$ $i = 1$ to 6	$C_2(1.5)$	$N_i = 2$ $i = 1$ to 6	10%	12.1%
9	$\mu_i = 0.13443751$ $i = 1, 2$	$(C_2)_1(1.5)$ $(E_2)_2$	$C_2(1.5)$	$N_i = 4$ $i = 1, 2$	10.5%	8.5%
10	$\mu_i = 0.2$ $i = 1, 2$	$(E_2)_1$, $(C_2)_2(1.5)$	$C_2(1.5)$	$N_i = 4$ $i = 1, 2$	11.7%	13.6%
11	$\mu_i = 0.125$ $i = 1, 2, 3$	$(C_2)_i(5)$ $i = 1, 2, 3$	$C_2(2)$	$N_i = 2$ $i = 1, 2, 3$	80%	47.1%

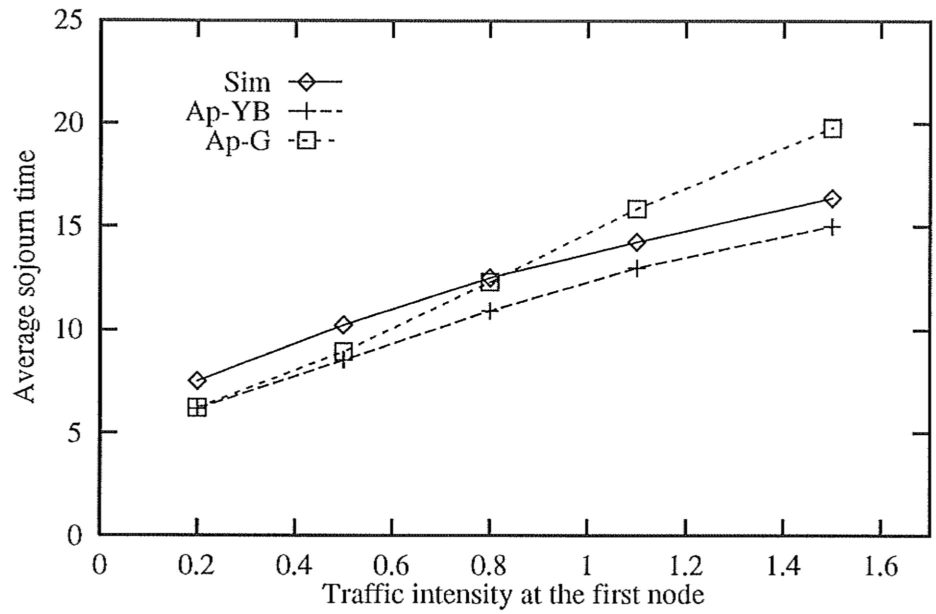

Figure D.1: Results for a two-node tandem system

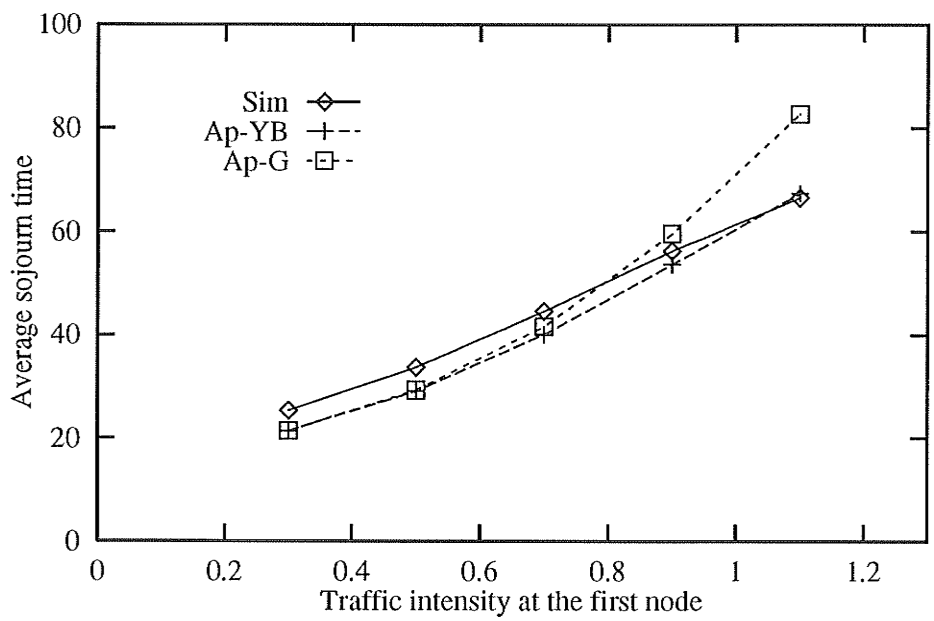

Figure D.2: Results for a four-node tandem system

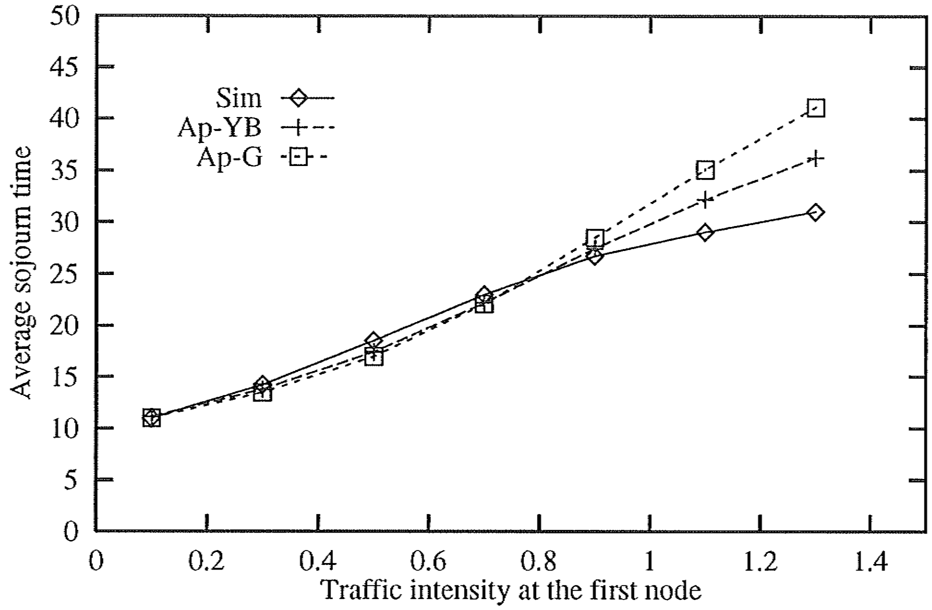

Figure D.3: Results for a two-node tandem system

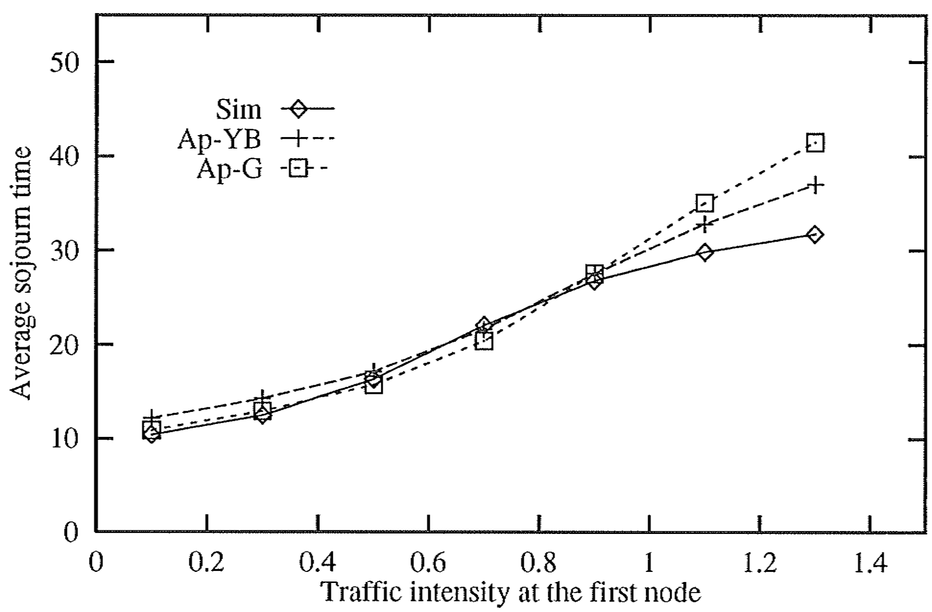

Figure D.4: Results for a two-node tandem system

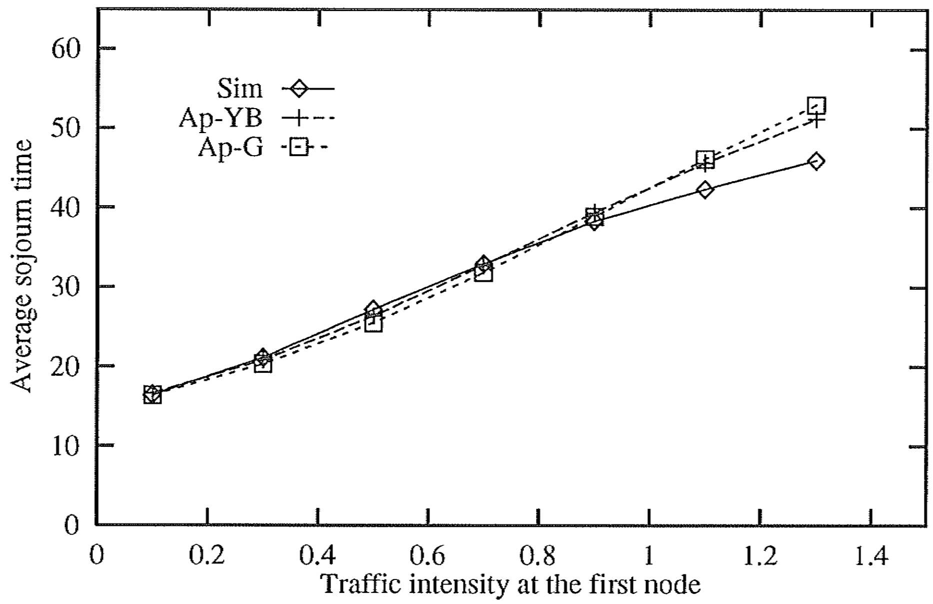

Figure D.5: Results for a two-node tandem system

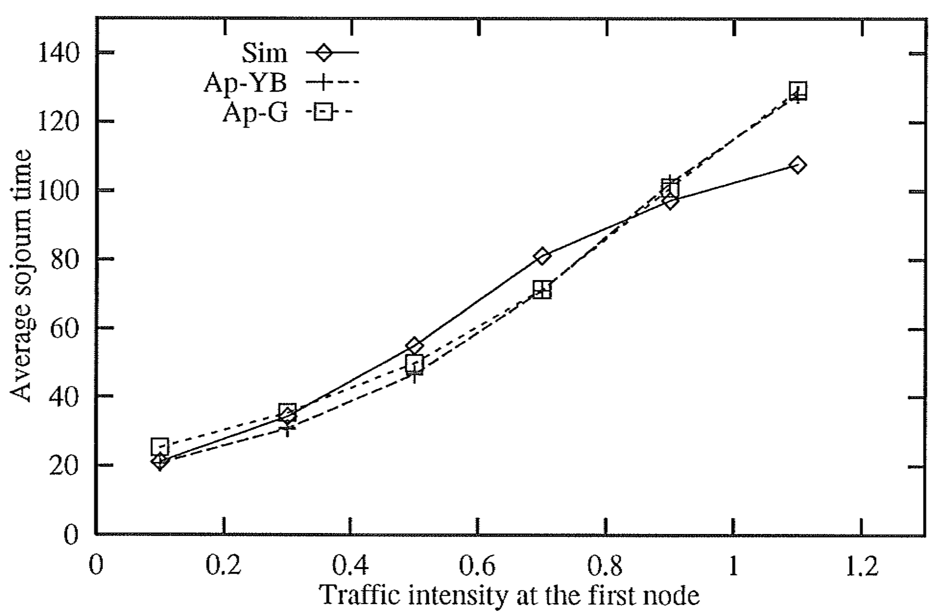

Figure D.6: Results for a two-node tandem system

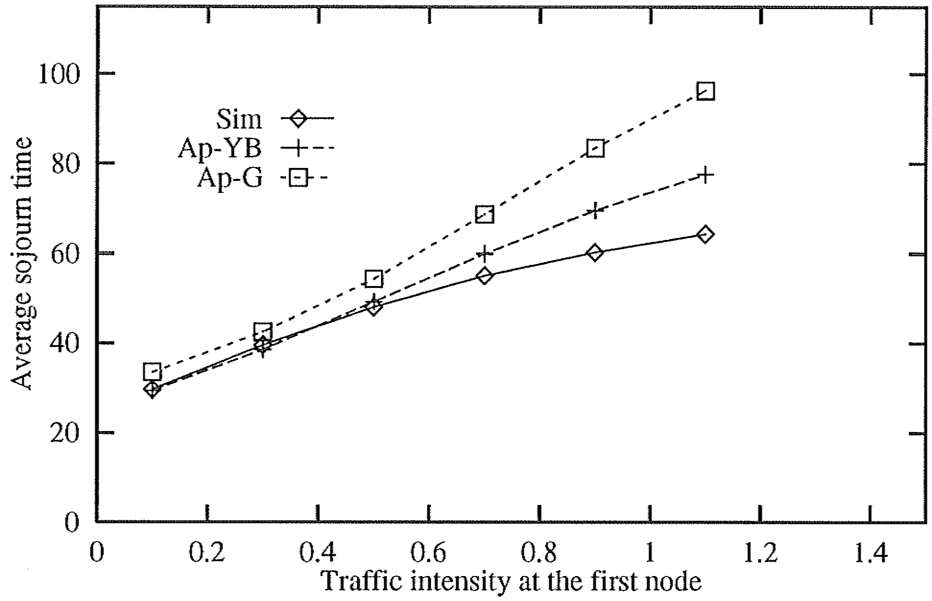

Figure D.7: Results for a three-node tandem system

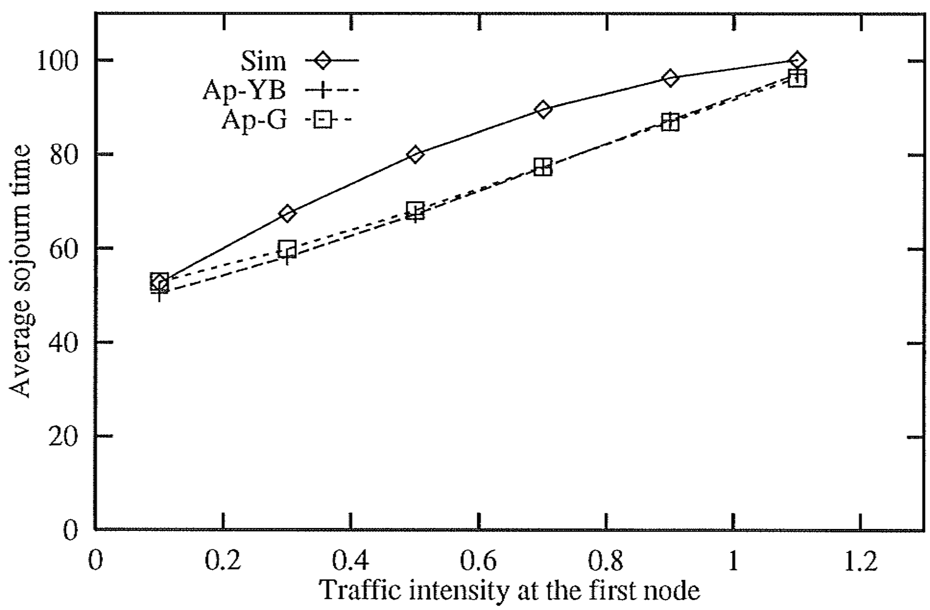

Figure D.8: Results for a six-node tandem system

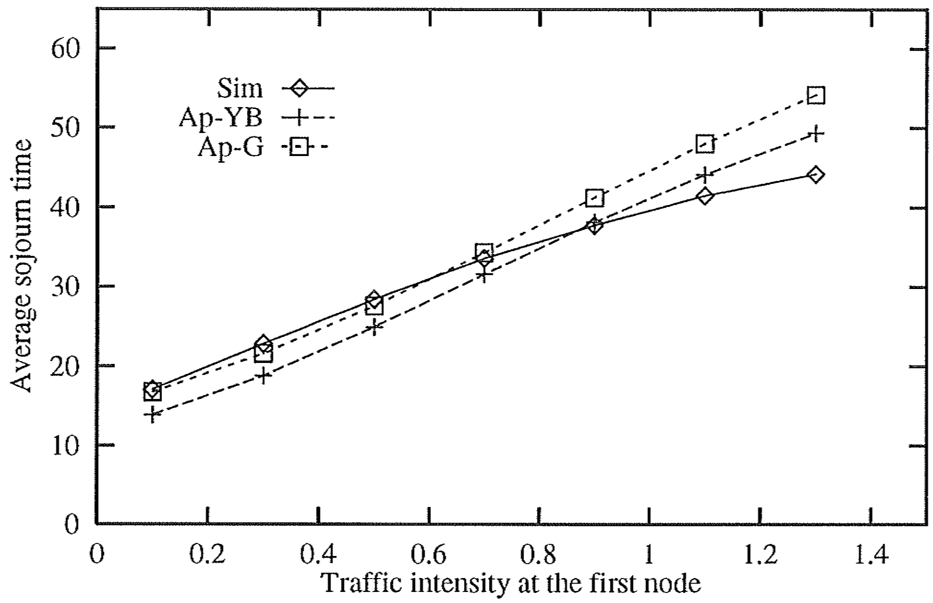

Figure D.9: Results for a two-node tandem system

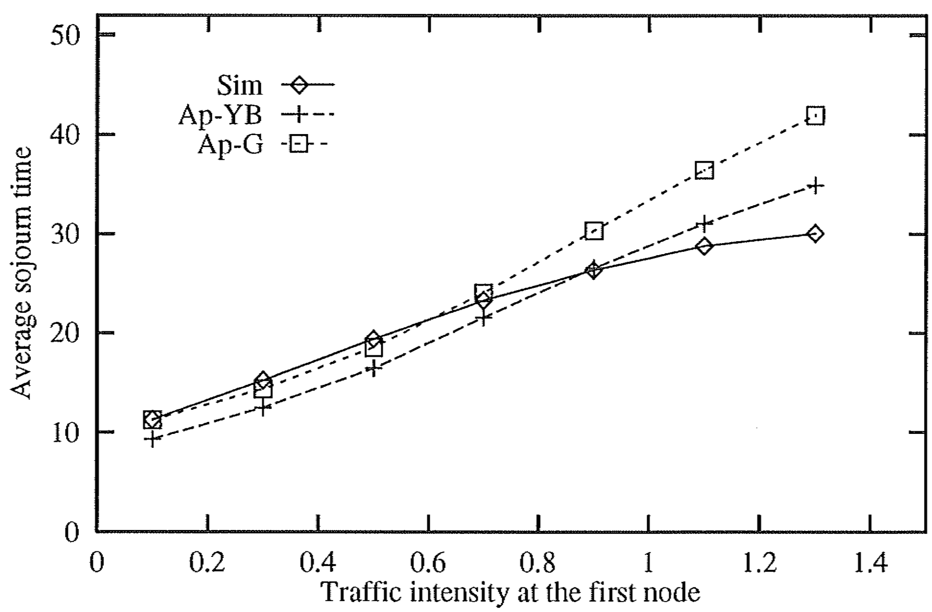

Figure D.10: Results for a two-node tandem system

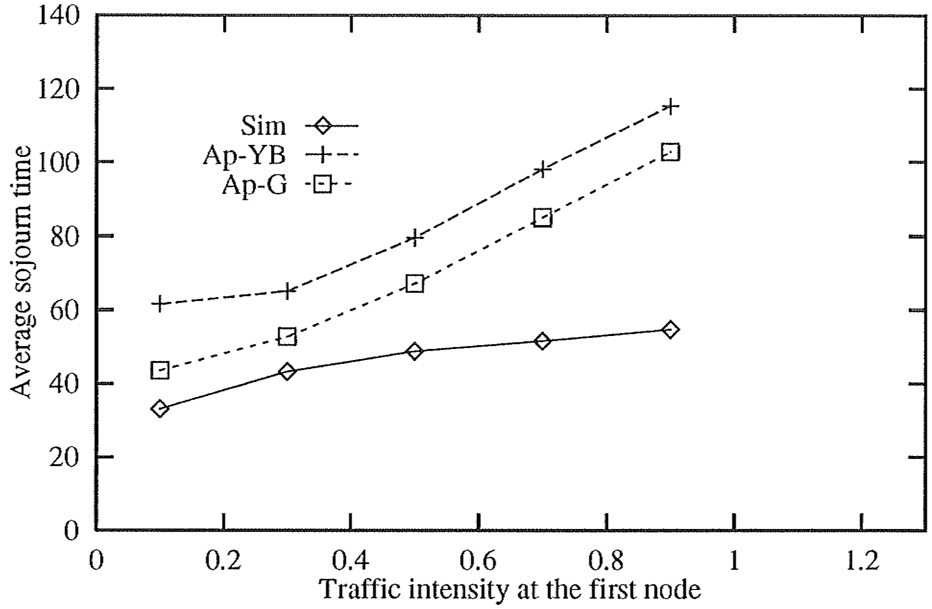

Figure D.11: Results for a three-node tandem system

Table D.3: Split configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
12	$\mu_0 = 0.2$ $\mu_1 = \mu_2 = 0.1$	$(E_2)_i$ $i = 0, 1, 2$	Exp.	$N_i = 3$ $i = 0, 1, 2$	7.5%	24.1%
13	$\mu_0 = 0.13443751$ $\mu_i = 0.05, i = 1, 2$	$(C_2)_0(1.5)$ $(E_2)_i, i = 1, 2$	Exp.	$N_i = 3$ $i = 0, 1, 2$	3%	9.1%
14	$\mu_0 = 0.3$ $\mu_i = 0.125, i = 1, 2$	$(E_2)_0$ $(C_2)_i(1.5), i = 1, 2$	Exp.	$N_i = 3$ $i = 0, 1, 2$	21.1%	20%
15	$\mu_0 = 0.3$ $\mu_i = 0.125, i = 1, 2$	$(E_2)_0$ $(C_2)_i(1.5), i = 1, 2$	E_2	$N_i = 3$ $i = 0, 1, 2$	24.9%	18.4%

D.2 Split Configurations

We present eight more Examples for split configurations. The descriptions of these systems are shown in Tables D.3 and D.4 . The results for Examples 12 to 19 are illustrated in Figures D.12 to D.19.

Table D.4: Split configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
16	$\mu_0 = 0.45$ $\mu_i = 0.1, i = 1$ to 4	$(E_2)_i$ $i = 0$ to 4	Exp.	$N_i = 3$ $i = 0$ to 4	10.6%	27%
17	$\mu_0 = 0.13443751$ $\mu_i = 0.055, i = 1, 2$	$(C_2)_i(1.5)$ $i = 0, 1, 2$	$C_2(1.5)$	$N_i = 2$ $i = 0, 1, 2$	18.6%	35.6%
18	$\mu_0 = 0.13443751$ $\mu_i = 0.05, i = 1, 2$	$(C_2)_0(1.5)$ $(E_2)_i, i = 1, 2$	$C_2(1.5)$	$N_i = 3$ $i = 0, 1, 2$	8.1%	14.5%
19	$\mu_0 = 0.3$ $\mu_i = 0.125, i = 1, 2$	$(E_2)_0$ $(C_2)_i(1.5), i = 1, 2$	$C_2(1.5)$	$N_i = 3$ $i = 0, 1, 2$	13.6%	27.1%

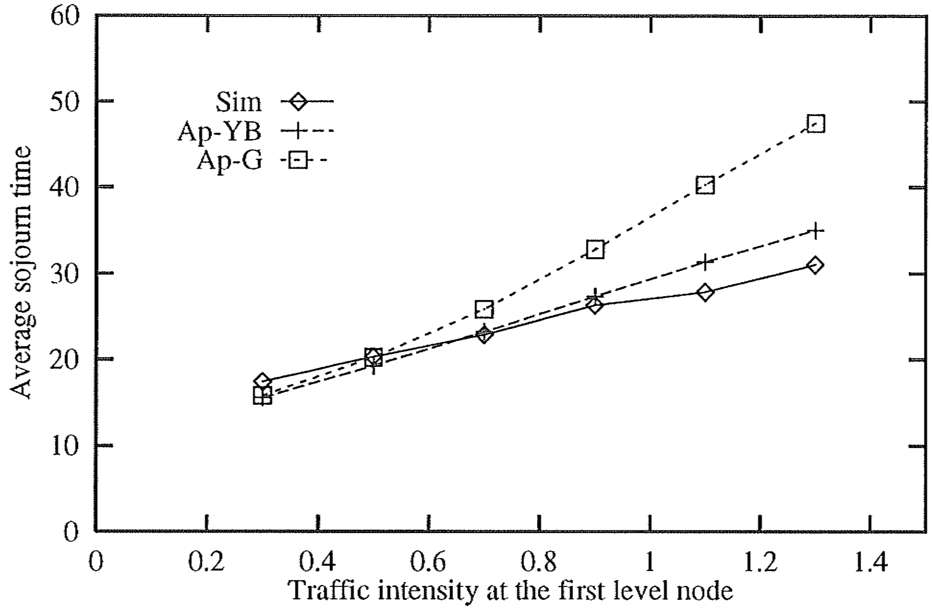

Figure D.12: Results for a split system with two second level nodes

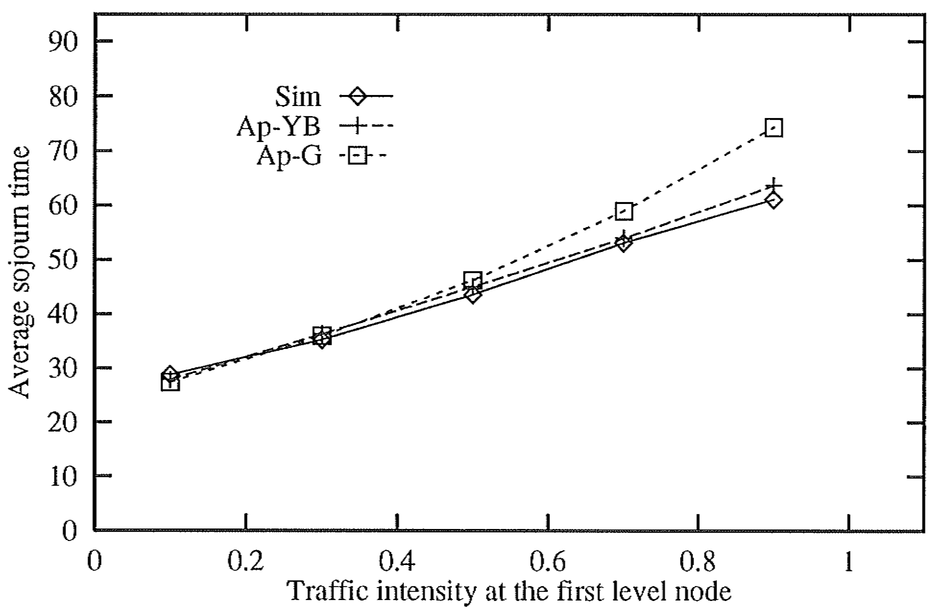

Figure D.13: Results for a split system with two second level nodes

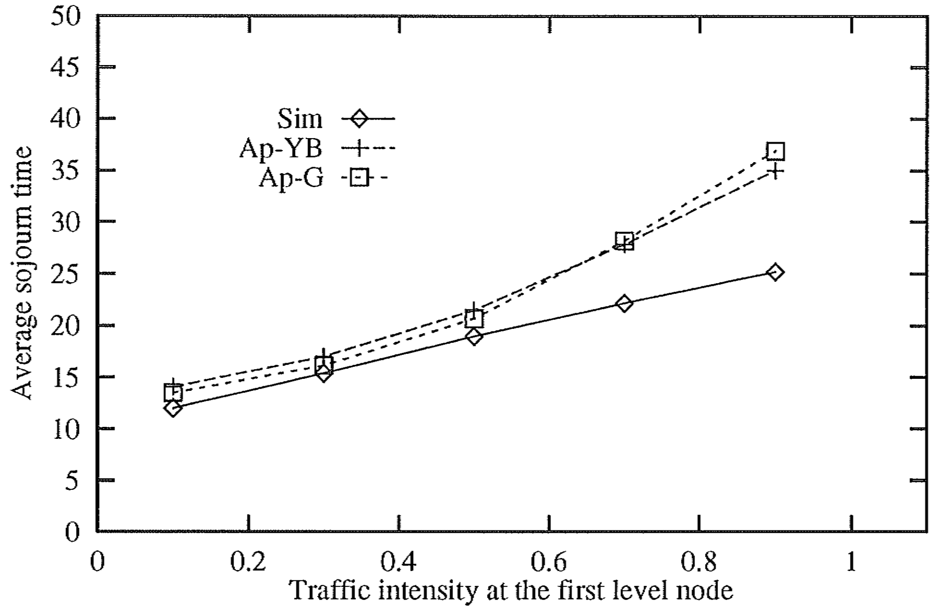

Figure D.14: Results for a split system with two second level nodes

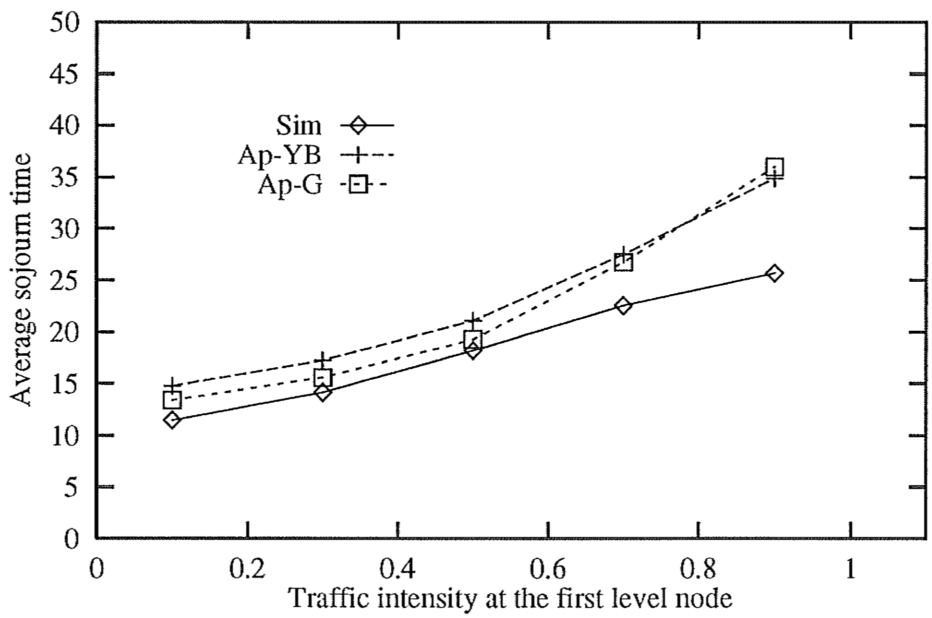

Figure D.15: Results for a split system with two second level nodes

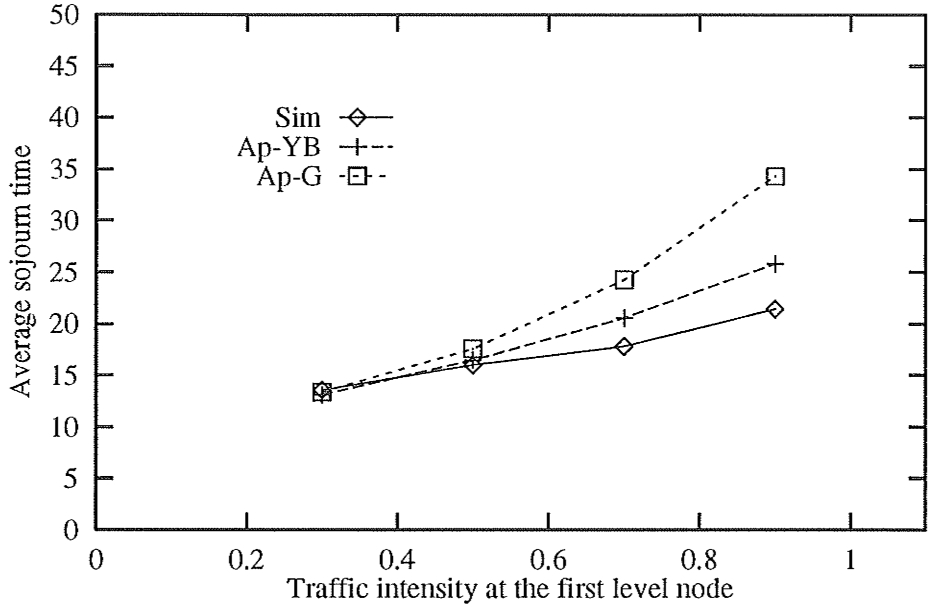

Figure D.16: Results for a split system with four second level nodes

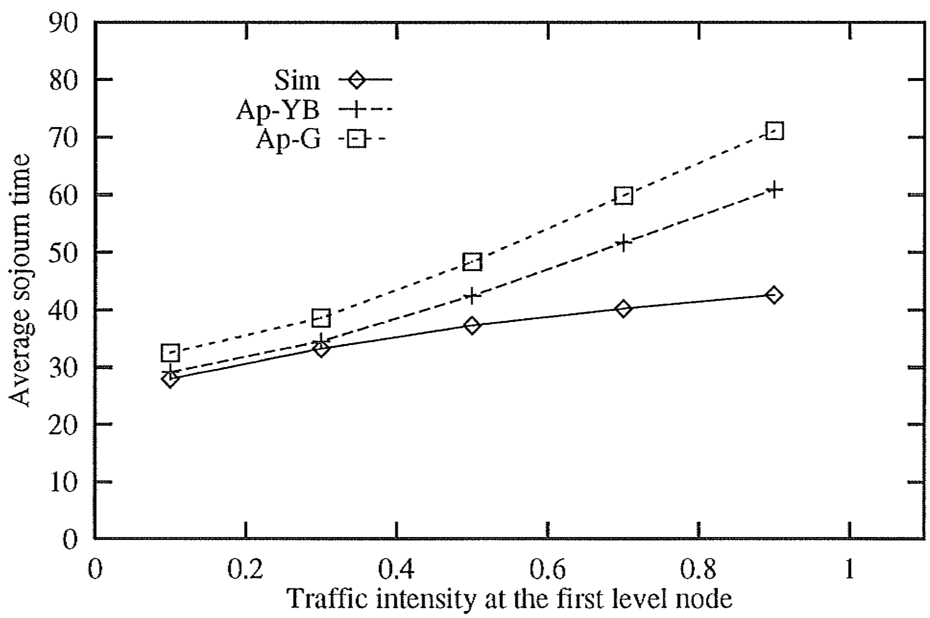

Figure D.17: Results for a split system with two second level nodes

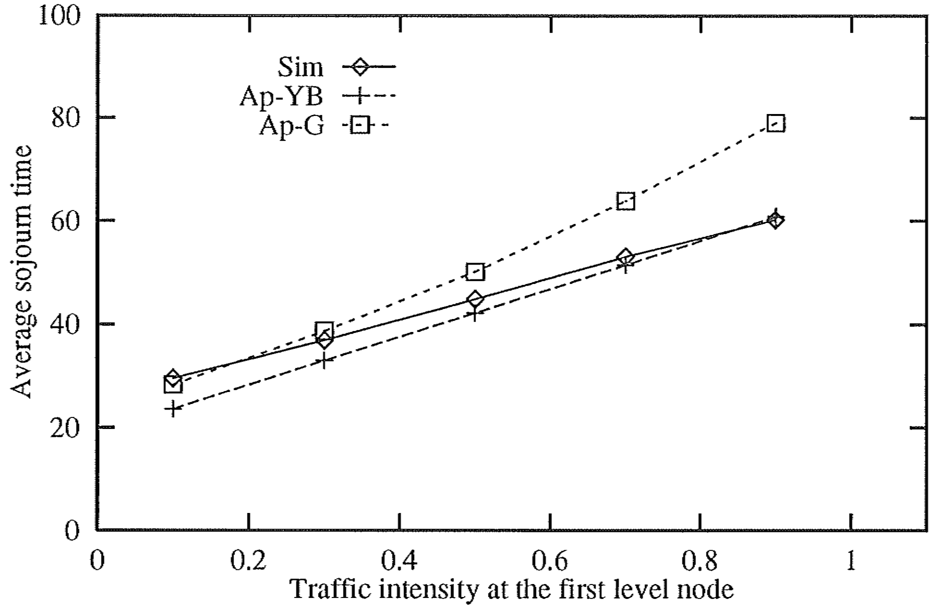

Figure D.18: Results for a split system with two second level nodes

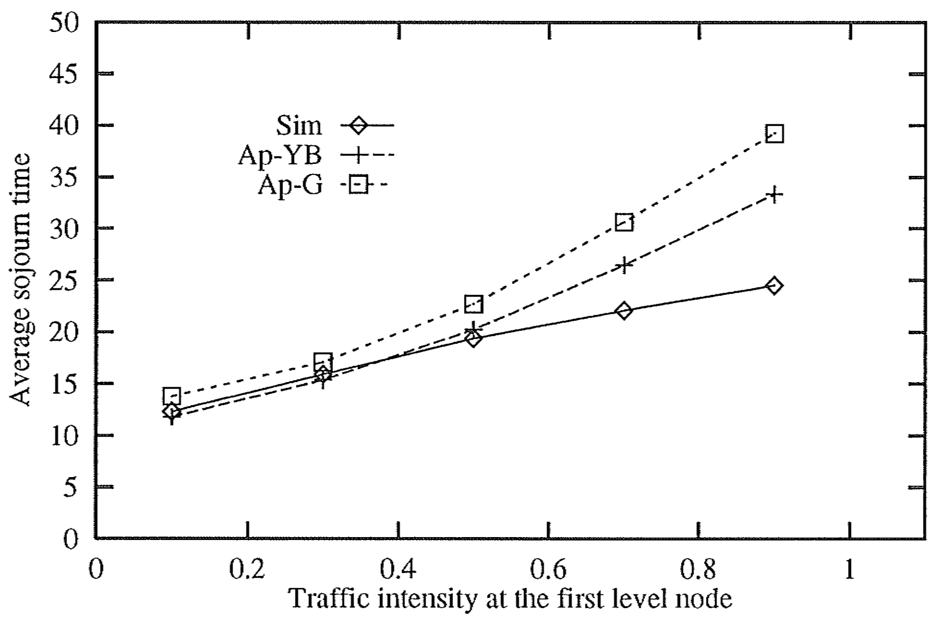

Figure D.19: Results for a split system with two second level nodes

D.3 Merge Configurations

We present ten more Examples for merge configurations. The descriptions of these systems are shown in Tables D.5 and D.6 . The results for Examples 20 to 29 are illustrated in Figures D.20 to D.29.

Table D.5: Merge configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
20	$\mu_0 = 0.3$ $\mu_1 = 0.12$ $\mu_2 = 0.1$	$(E_2)_i$ $i = 0, 1, 2$	Exp.	$N_i = 3$ $i = 0, 1, 2$	10.1%	9.5%
21	$\mu_0 = 0.6$ $\mu_i = 0.3, i = 1, 2$	$(E_2)_i$ $i = 0, 1, 2$	Exp.	$N_i = 5$ $i = 0, 1, 2$	9.7%	8.5%
22	$\mu_0 = 0.5$ $\mu_i = 0.2, i = 1, 2, 3$	$(E_2)_i$ $i = 0$ to 3	Exp.	$N_0 = 7$ $N_i = 3$, $i = 1, 2, 3$	13.9%	23.6%
23	$\mu_0 = 0.4, \mu_1 = 0.1$ $\mu_2 = 0.15, \mu_3 = 0.2$	$(E_2)_i$ $i = 0$ to 3	Exp.	$N_0 = 5$ $N_i = 2$, $i = 1$ to 3	9.1%	21.9%
24	$\mu_0 = 0.4$ $\mu_i = 0.2, i = 1, 2$	$(C_2)_0(1.5)$, $(E_2)_i, i = 1, 2$	Exp.	$N_i = 3$ $i = 0, 1, 2$	6.9%	5%

Table D.6: Merge configurations

Ex.	Service times		Ex. In.	Buffers	ARE	
	Rates	Distribution	Times		Ap-YB	Ap-G
25	$\mu_0 = 0.26887502$ $\mu_i = 0.13443751$ $i = 1, 2$	$(E_2)_0$ $(C_2)_i(1.5)$, $i = 1, 2$	Exp.	$N_i = 3$ $i = 0, 1, 2$	7.8%	2.7%
26	$\mu_0 = 0.26887502$ $\mu_i = 0.13443751$ $i = 1, 2$	$(E_2)_0$ $(C_2)_i(1.5)$, $i = 1, 2$	E_2	$N_i = 3$ $i = 0, 1, 2$	21.1%	5%
27	$\mu_0 = 0.16$ $\mu_i = 0.05$, $i = 1$ to 4	$(C_2)_i(1.5)$ $i = 0$ to 4	$C_2(1.5)$	$N_i = 2$ $i = 0$ to 4	9.1%	4.9%
28	$\mu_0 = 0.4$ $\mu_i = 0.2$ $i = 1, 2$	$(C_2)_0(1.5)$ $(E_2)_i$, $i = 1, 2$	$C_2(1.5)$	$N_i = 3$ $i = 0, 1, 2$	15.4%	5%
29	$\mu_0 = 0.26887502$ $\mu_i = 0.13443751$ $i = 1, 2$	$(E_2)_0$ $(C_2)_i(1.5)$, $i = 1, 2$	$C_2(1.5)$	$N_i = 3$ $i = 0, 1, 2$	4.1%	5.4%

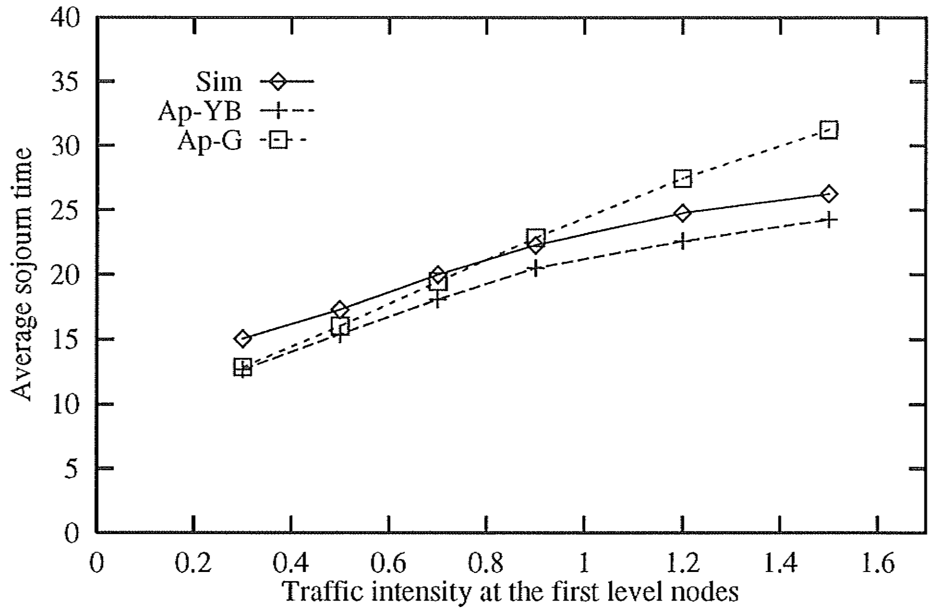

Figure D.20: Results for a merge system with two first level nodes

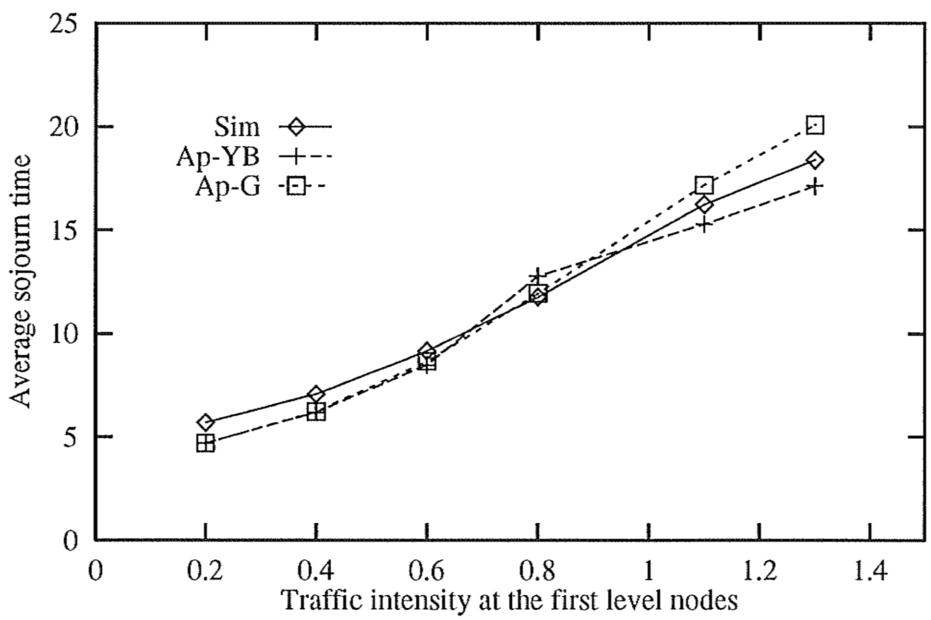

Figure D.21: Results for a merge system with two first level nodes

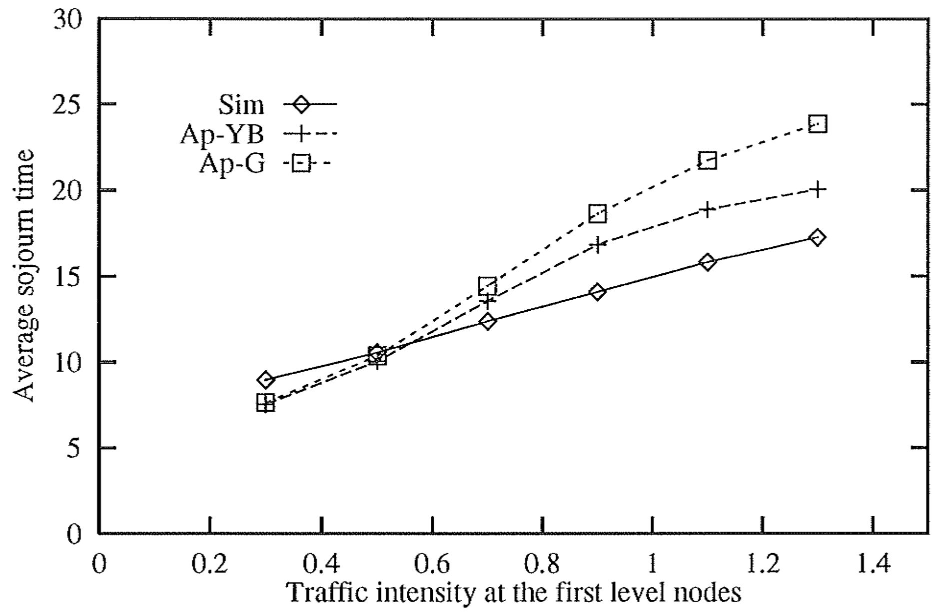

Figure D.22: Results for a merge system with three first level nodes

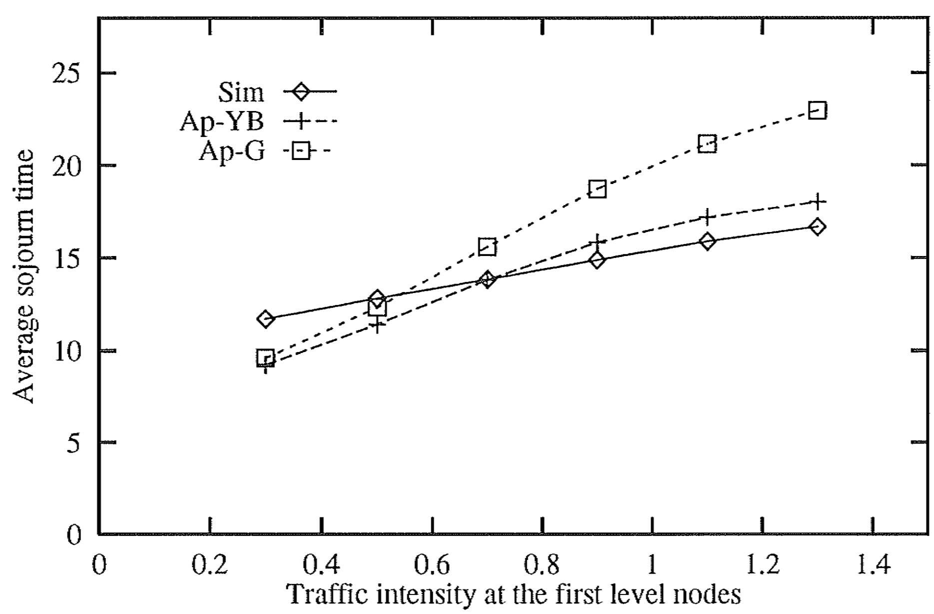

Figure D.23: Results for a merge system with three first level nodes

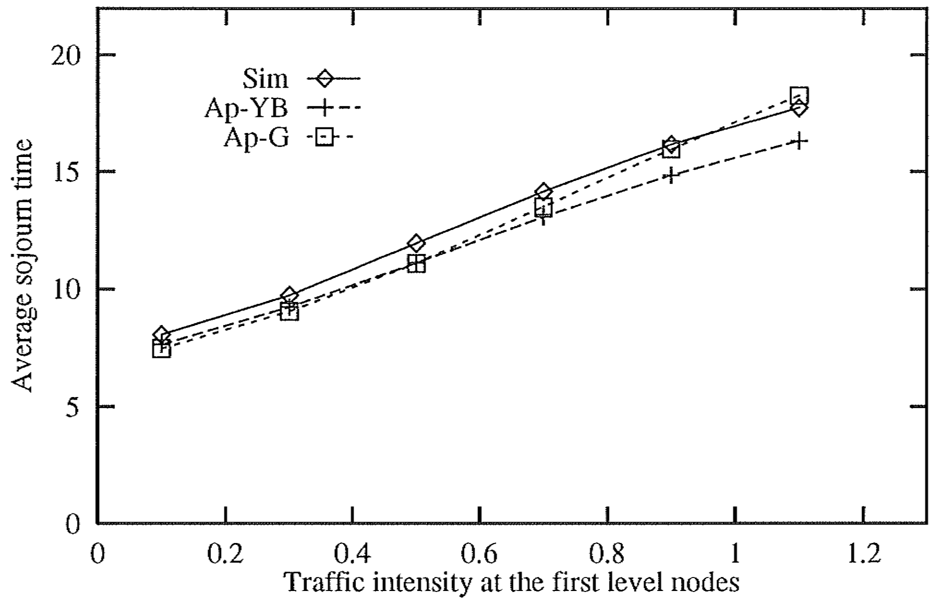

Figure D.24: Results for a merge system with two first level nodes

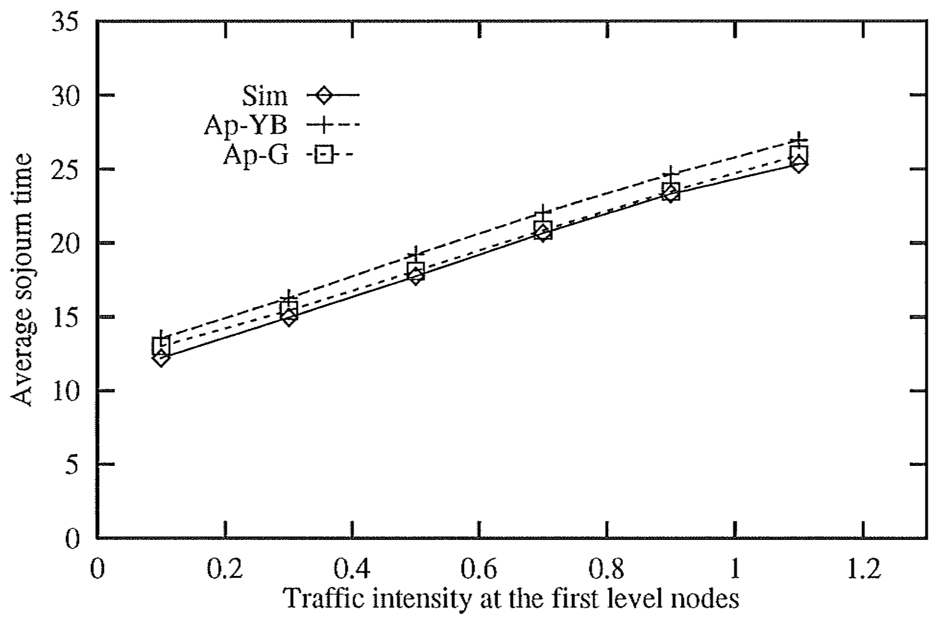

Figure D.25: Results for a merge system with two first level nodes

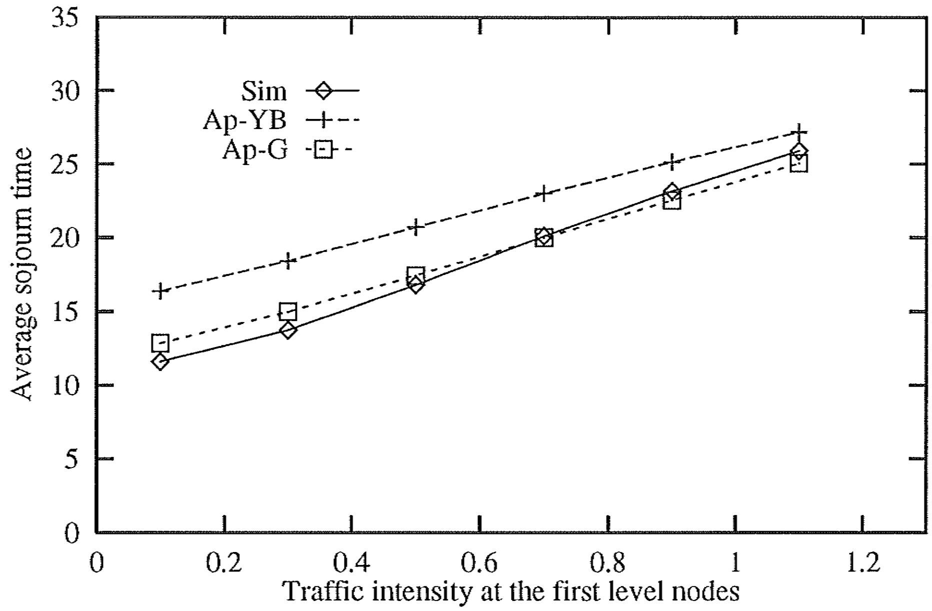

Figure D.26: Results for a merge system with two first level nodes

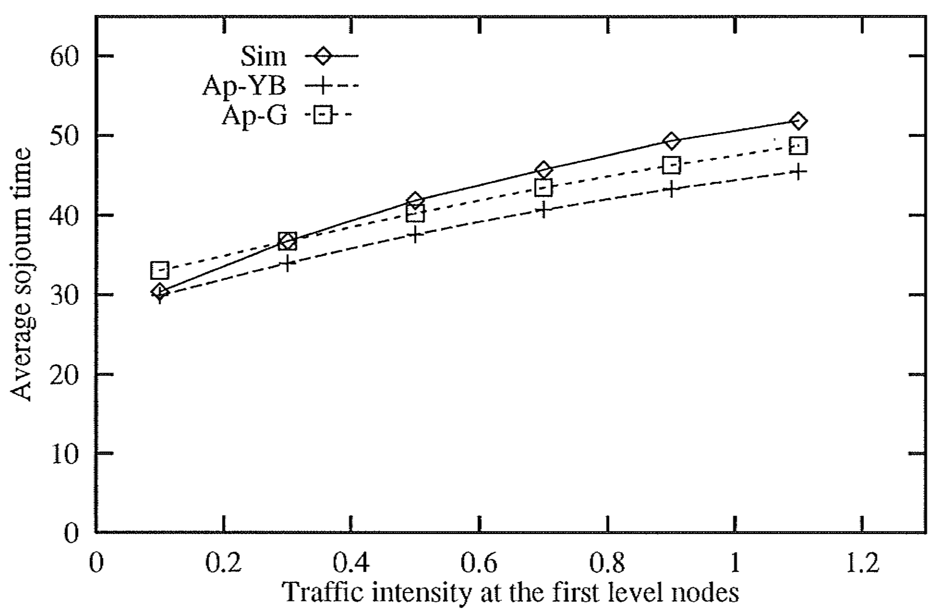

Figure D.27: Results for a merge system with four first level nodes

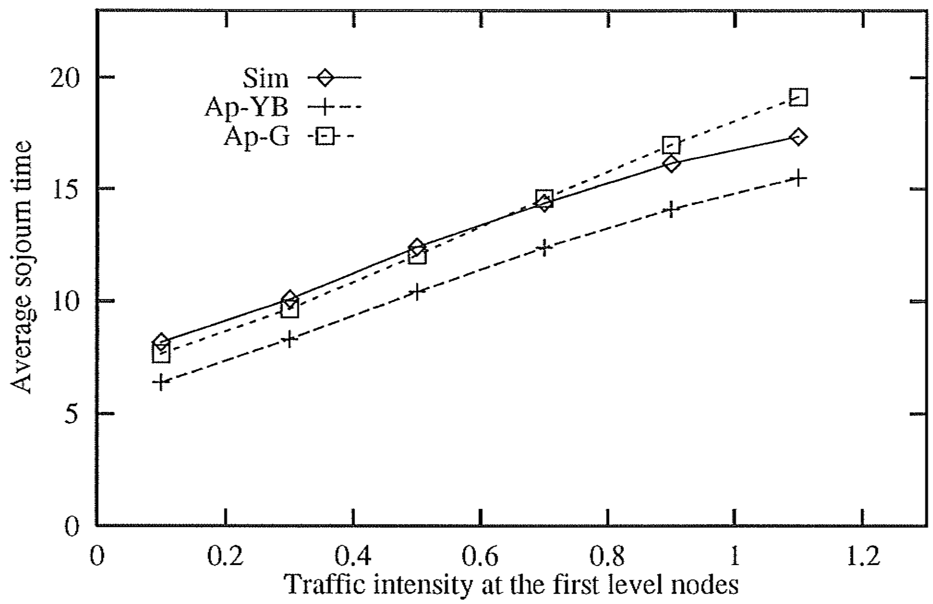

Figure D.28: Results for a merge system with two first level nodes

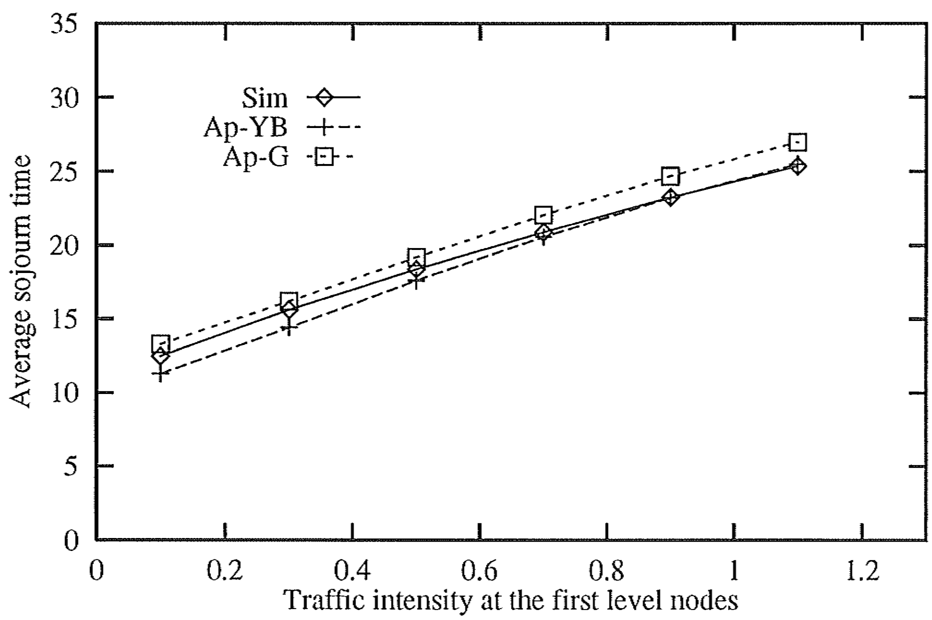

Figure D.29: Results for a merge system with two first level nodes

REFERENCES

REFERENCES

[1] Altiok,T., "Approximate analysis of exponential tandem queues with blocking" EJOR 11 (1982) 390-398.

[2] Altiok,T., "Production lines with phase-type operation and repair times and finite buffers" IJPR 23(3) (1985) 489-498.

[3] Altiok,T., "Approximate analysis of queues in series with phase-type service times and blocking" Operations Research 37(4) (1989) 601-610.

[4] Altiok,T., and Perros,H.G., "Open networks of queues with blocking: Split and merge configurations" IIE Transactions Sept (1986) 251-261.

[5] Altiok,T., and Perros,H.G., "Approximate analysis of arbitrary configurations of open queueing networks with blocking" Annals of Operations Research 9 (1987) 481-509.

[6] Altiok,T., and Ranjan,R., "Analysis of production lines with general service times and finite buffers: A two node decomposition approach" Engineering Costs and Production Economics 17 (1989) 155-165.

[7] Altiok,T., and Stidham,S.JR., "The allocation of interstage buffer capacities in production lines" IIE Transactions 15(4) (1983) 292-299.

[8] Baskett,F., Chandy,K.M., Muntz,R.R., and Palacios,F.G., "Open, closed, and mixed networks of queues with different classes of customers" JACM 22(2) (1975) 248-260.

[9] Boxma,O.J., and Konheim,A.G., "Approximate analysis of exponential queueing systems with blocking" Acta Informatica 15 (1981) 19-66.

[10] Brandwajn,A., "Equivalence and decomposition in queueing systems - A unified approach" Performance Evaluation 5 (1985) 175-186.

[11] Brandwajn,A., and Jow,Y.L., "An approximation method for tandem queues with blocking" Operations Research 36 (1988) 73-83.

[12] Buzen,J.P., "Computational algorithms for closed queueing networks with exponential servers" Communications of the ACM 16 (1973) 527-531.

[13] Choong,Y.F., and Gershwin,S.B., "A decomposition method for the approximate evaluation of capacitated transfer lines with unreliable machines and random processing times" IIE Transactions 19(2) (1987) 150-159.

[14] Dallery,Y., David,R., and Xie,X., "An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers" IIE Transactions 20(3) (1988) 280-283.

[15] Gelenbe,E., "On approximate computer system models" JACM 22(2) (1975) 261-269.

[16] Gershwin,S.B., "An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking" Operations Research 35(2) (1987) 291-305.

[17] Gershwin,S.B., "Assembly-Disassembly systems: An efficient decomposition algorithm for tree-structured networks" MIT Laboratory for Manufacturing and Productivity Report LIDS-P-1579, LMP-88-007 (1988).

[18] Gershwin,S.B., "An efficient decomposition algorithm for unreliable tandem queueing systems with finite buffers" Queueing Networks with Blocking H.G.Perros and T.Altiok (eds) (1989) 127-146.

[19] Gershwin,S.B., and Berman,O., "Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers" AIIE Transactions 13(1) (1981) 2-11.

[20] Gershwin,S.B., and Schick,I.C., "Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers" Operations Research 31(2) (1983) 354-380.

[21] Gordon,W.J., and Newell,G.F., "Closed queueing systems with exponential servers" Operations Research 15(2) (1967) 254-265.

[22] Gross,D., and Harris,C.M., Fundamentals of Queueing Theory (1985) Wiley New York.

[23] Gross,D., Miller,D.R., and Soland,R.M., "A closed queueing network model for multi-echelon repairable item provisioning" IIE Transactions 15(4) (1983) 344-352.

[24] Gun,L., and Makowski,A.M., "An approximation method for general tandem queueing systems subject to blocking" Queueing Networks with Blocking H.G.Perros and T.Altiok (eds) (1989) 147-171.

[25] Gun,L., and Makowski,A.M., "Matrix-geometric solution for two node tandem queueing systems with phase-type servers subject to blocking and failures" Stochastic Models 5(3) (1989) 431-457.

[26] Hill,J.M., "The changing Profile of Material Handling Systems and Controls " Industrial Engineering 18(11) (1986) 68-73.

[27] Hunt,G.C., "Sequential arrays of waiting lines" Operations Research 5 (1956) 674-683.

[28] Jun,K.P., and Perros,H.G., "Approximate analysis of arbitrary configurations of queueing networks with blocking and deadlock" Queueing Networks with Blocking H.G.Perros and T.Altiok (eds) (1989) 259-279.

[29] Jun,K.P., and Perros,H.G., "An approximate analysis of open tandem queueing networks with blocking and general service times" EJOR 46 (1990) 123-135.

[30] Kerbache,L., and Smith,J.M., "The generalized expansion method for open finite queueing networks" EJOR 32 (1987) 448-461.

[31] Kleinrock,L., Queueing Systems Vol I: Theory (1975) Wiley New York.

[32] Kuehn,P.J., "Approximate analysis of general queueing networks by decomposition" IEEE Transactions on Communications Vol Com-27 No 1 January (1979) 113-126.

[33] Labetoulle,J., and Pujolle,G., "Isolation method in a network of queues" IEEE Transactions on Software Engineering Vol-SE-6(4) (1980) 373-381.

[34] Lee,H.S., and Pollock,S.M., "Approximate analysis for the merge configuration of an open queueing network with blocking" IIE Transactions 21(2) (1989) 122-129.

[35] Madu,C.N., "Determination of maintenance floats using Buzen's algorithm" IJPR 26(8) (1988) 1385-1394.

[36] Marie,R.A., "Calculating equilibrium probabilities for $\lambda(n)/C_k/1/N$ queue" In Proceedings Performance ' 80 Toronto Canada (May 28-30 1980) 117-125.

[37] Marshall,K.T., "Some inequalities in queueing" Operations Research 16 (1968) 651-665.

[38] Neuts,M.F., Matrix-Geometric Solutions in Stochastic Models (1981) The Johns Hopkins University Press Baltimore.

[39] Onvural,R.O., and Perros,H.G., "On equivalencies of blocking mechanisms in queueing networks with blocking" Operations Research Letters 5(6) (1986) 293-297.

[40] Patuwo,B.E., McNickle,D.C., and Disney,R.L., " Queues with non-renewal arrivals: Does it matter ?" IIE Conference Proceedings (1990) San Francisco 366-372.

[41] Perros,H.G., and Altiok,T., "Approximate analysis of open networks of queues with blocking: Tandem configurations" IEEE Transactions on Software Engineering Vol-SE-12(3) (1986) 450-461.

[42] Perros,H.G., and Snyder,P.M., "A computationally efficient approximation algorithm for feed-forward open queueing networks with blocking" Performance Evaluation 9 (1989) 217-224.

[43] Pourbabai,B., "Performance modeling of a closed loop material handling system" EJOR 32 (1987) 340-351.

[44] Pourbabai,B., "Analysis and design of an integrated material handling system" IJPR 26(7) (1988) 1225-1236.

[45] Pourbabai,B., and Sonderman,D., "A stochastic recirculation system with random access" EJOR 21 (1985) 367-378.

[46] Springer,M.C., and Makens,P.K., "Queueing models for performance analysis: Selection of single station models" EJOR 58 (1991) 123-145.

[47] Takahashi,Y., Miyahara,H., and Hasegawa,T., "An approximation method for open restricted queueing networks" Operations Research 28(3) (1980) 594-602.

[48] Whitt,W., "The queueing network analyzer" The Bell System Technical Journal 62(9) (1983) 2779-2815.

[49] Yakowitz,S., and Szidarovszky,F., An Introduction to Numerical Computations (1989) Second Edition Macmillan New York.

[50] Yannopoulos,E., and Alfa.A.S., "Analysing real life queueing network problems: From theory to practice" submitted for publication to Journal of Manufacturing Systems (1993) (also presented at the ORSA/TIMS Conference Anaheim California (1991)).

[51] Yannopoulos,E., and Alfa.A.S., "An approximation method for queues in series with blocking" (1993) Performance Evaluation (accepted).

[52] Yannopoulos,E., and Alfa.A., "A simple and quick approximation algorithm for tandem, split, and merge queueing networks with general arrival and service processes and blocking" (1992) submitted for publication to INFOR.

[53] Yannopoulos,E., Jenness,J.D., and Hawaleshka,O., "Animated simulation of an automatic conveyor system" Proceedings of the Thirteenth Canadian Congress of Applied Mechanics (1991) Winnipeg Manitoba Canada.

[54] Yao,D.W., and Buzacott,J.A., "Queueing models for a flexible machining station Part I: The diffusion approximation" EJOR 19 (1985) 233-240.

[55] Yao,D.W., and Buzacott,J,A., "Queueing models for a flexible machining station Part II: The method of Coxian phases" EJOR 19 (1985) 241-252.