Multiple Ring Networks in Clustered Traffic Environments

By Breton Green
University of Natal
1998

Submitted in fulfilment of the academic requirements for the degree of Master of Science in Engineering in the Department of Electronic Engineering, University of Natal, 1998

ABSTRACT

Ring networks are appropriate for the full range of network levels, including multiprocessor systems, local area computer networks and high speed backbones. The most well known and widely implemented examples are the IBM token ring and FDDI networks. Ring networks have the advantages of high channel utilisation and bounded delay if an n-limited service policy is used. The packet transfer delay, defined as the average time a packet spends in the network from the time it is generated until the time it is received at its destination node, improves with the number of rings on which a node is connected. However, many ring connections are not economically feasible since the cost of the ring interface increases with the number of rings.

There has been an abundance of previous work on single token ring networks. A number of papers on slotted rings, register insertion rings and more complex ring architectures have also been published. However, there is very little existing literature on multiple ring networks as well as ring networks in clustered traffic environments, i.e. where nodes from the same cluster tend to communicate more with each other than with other nodes in the network. This thesis focuses on two network topologies that make use of multiple rings and are well suited to clustered traffic environments: the two-connected multiple ring (2-MR) and the destination removal double ring (DRDR).

For the 2-MR network, three different practical token-based protocols are investigated in an attempt to optimise performance. It is further shown that significant performance improvements can be achieved by employing a slotted ring protocol rather than the token ring protocol. The DRDR network is also examined and its performance compared to the aforementioned architectures. For each of the six cases, both random and clustered traffic patterns are considered and compared. Analytical results are derived which are verified by results obtained from computer simulations.

Furthermore, we look at exact methods of analysing ring networks. A mean value analysis of a single token ring network with a 1-limited service discipline is performed, which clearly shows the complexity exact methods introduce. Finally, although it has been stated in the literature that an exact analysis of a multiple symmetrical token ring network is intractable, we present a novel Markov chain approach that gives exact results for near zero loads.

PREFACE

The research work in this thesis was performed by Breton Green, under the supervision of Professor Fambirai Takawira, at the University of Natal's Department of Electronic Engineering. The thesis was sponsored in its first two years by Telkom SA Ltd, as part of the Teletraffic Initiative Programme (TIP).

Parts of this thesis have been presented by the student at the Teletraffic '96 conference in Durban, South Africa, and the Teletraffic '97 conference in Grahamstown, South Africa.

The entire thesis, unless specifically indicated to the contrary in the text, is the student's own work and has not been submitted in whole or in part to any other university.

ACKNOWLEDGEMENTS

I wish to thank Professor Takawira for his help, guidance and patience over the last three years.

I would like to give a special thanks to my parents, especially for proof reading. I am forever indebted to them for their love, support, patience and encouragement during a period, which at times never seemed to have an end in sight.

Thanks are also owed to Telkom SA (Pty) Ltd and the University of Natal for their financial support, without which this degree would not have been possible.

Finally, I would like to thank my postgraduate colleagues and friends who provided hours of entertainment during the boring times and a source of distraction during the busy times. Without their "harassment", I might never have completed the work, particularly the simulations. They made the past three years a fun and memorable experience.

TABLE OF CONTENTS

TITLE PAGE i

ABSTRACT ii

PREFACE iii

ACKNOWLEDGEMENTS iv

TABLE OF CONTENTS v

LIST OF FIGURES AND TABLES viii

LIST OF GRAPHS x

LIST OF SYMBOLS xiii

CHAPTER 1

INTRODUCTION 1-1

1.1 WHAT ARE RING NETWORKS? 1-1
1.2 RING NETWORKS VS. CONTENTION NETWORKS 1-2
1.3 RING NETWORKS VS. SWITCHED NETWORKS 1-4
1.4 WIRING 1-6
1.5 NODE STRUCTURE 1-7
1.6 SPATIAL BANDWIDTH 1-7
1.7 BASIC PROTOCOLS 1-8
1.7.1 Packet removal from the ring 1-8
1.7.2 Token ring 1-8
1.7.2.1 Service policy 1-9
1.7.3 Slotted ring 1-9
1.7.4 Buffer insertion ring 1-9
1.7.5 Variable vs. fixed length packets 1-10
1.7.6 Packet overhead 1-11
1.8 MULTIPLE RING NETWORKS 1-11
1.8.1 Multiple ring protocol issues 1-11
1.8.2 Multiple ring performance 1-12
1.8.3 Robustness of multiple rings 1-13
1.9 COST ISSUES 1-14
1.10 TRAFFIC PATTERNS 1-14
1.11 DETERMINING THE PERFORMANCE OF RING NETWORKS 1-15
1.12 THESIS OVERVIEW 1-15
1.13 ORIGINAL CONTRIBUTIONS OF THIS THESIS 1-17

CHAPTER 2

LITERATURE SURVEY 2-1

2.1 TOKEN RING PROTOCOLS 2-1
2.1.1 IEEE 802.5 2-2

2.1.2 FDDI 2-2
2.2 TOKEN RING ANALYSES 2-4
2.2.1 The M/G/1 queuing system 2-4
2.2.2 Polling systems and cyclic servers 2-6
2.2.3 Infinite capacity systems 2-7
2.2.4 Finite capacity systems 2-7
2.2.5 Approximate token ring analysis 2-8
2.2.5.1 Outline of method 2-8
2.2.5.2 1-limited token ring analysis 2-10
2.2.5.3 Gated token ring analysis 2-12
2.3 SLOTTED RING 2-15
2.3.1 Approximate slotted ring analysis 2-16
2.4 BUFFER INSERTION RING 2-18
2.5 SYMMETRICAL MULTIPLE RING ANALYSIS 2-19
2.6 TWO-CONNECTED MULTIPLE RING 2-20
2.6.1 Topology 2-20
2.6.2 Analysis 2-21
2.6.3 Results 2-27
2.7 RING NETWORKS WITH SPATIAL BANDWIDTH REUSE 2-30
2.7.1 Parallelring 2-30
2.7.2 Multiple-token protocol 2-31
2.7.3 PLAYTHROUGH ring 2-32
2.7.4 MetaRing 2-33
2.8 FAIRNESS ALGORITHMS 2-34
2.8.1 Timed token protocol 2-34
2.8.2 Helical window token ring 2-35
2.8.3 MetaRing fairness algorithms 2-36
2.8.3.1 Global fairness algorithm 2-36
2.8.3.2 Local fairness algorithm 2-37
2.9 SUMMARY 2-38

CHAPTER 3

TWO-CONNECTED MULTIPLE RING 3-1

3.1 OPTIMISATION CRITERIA 3-2
3.2 NODE MODEL 3-2
3.3 1-LIMITED SERVICE, INTERCLUSTER TRAFFIC PRIORITY TO MAIN RING (*1IO*) 3-4
3.3.1 Analysis 3-4
3.1.2 Results 3-8
3.4 GATED SERVICE, INTERCLUSTER TRAFFIC PRIORITY TO MAIN RING (*GGO*) 3-13
3.4.1 Analysis 3-13
3.4.2 Results 3-17
3.5 GATED SERVICE, IDEALLY BALANCED LOAD *(IDEAL)* 3-20
3.5.1 Analysis 3-21
3.1.2 Results 3-23
3.6 GATED SERVICE, LOAD BALANCING ALGORITHM *(ALG)* 3-25
3.6.1 Results 3-27
3.7 SLOTTED RING PROTOCOL 3-29
3.7.1 Analysis 3-29
3.7.2 Results 3-32
3.8 SUMMARY 3-36

CHAPTER 4

DESTINATION REMOVAL DOUBLE RING 4-1

4.1 TOPOLOGY 4-1
4.2 POSSIBLE PROTOCOLS 4-2
4.3 SLOTTED RING ANALYSIS 4-2
4.3.1 Random Traffic 4-4
4.3.2 Clustered Traffic 4-6
4.4 RESULTS 4-8
4.5 IMMEDIATE FLOW CONTROL SCHEME 4-11
4.5.1 Results 4-13
4.6 SUMMARY 4-14

CHAPTER 5

EXACT RING NETWORK ANALYSES 5-1

5.1 EXACT MEAN VALUE ANALYSIS OF A 1-LIMITED TOKEN RING 5-1
5.1.1 Solving for ϕ_n 5-8
5.1.2 Solving for the mean queue length 5-8
5.1.3 Solving for the mean packet transfer delay 5-9
5.1.4 Solving for a symmetrical load 5-9
5.1.5 Results 5-10
5.2 APPLYING MARKOV CHAINS TO MULTIPLE RING NETWORKS AT NEAR ZERO LOADS ... 5-11
5.3 SUMMARY 5-21

CHAPTER 6

CONCLUSION 6-1

REFERENCES R-1

LIST OF FIGURES AND TABLES

CHAPTER 1

Figure 1-1: Ring network 1-1
Figure 1-2: Switched network (star topology) 1-4
Figure 1-3: Star wiring configuration 1-6
Figure 1-4: Node model 1-7
Figure 1-5: Spatial bandwidth 1-8
Figure 1-6: Recovery from node and link failures 1-13

Table 1-1: Comparison of contention, ring and switched networks 1-6

CHAPTER 2

Figure 2-1: FDDI Token-Frame 2-3
Figure 2-2: FDDI Data-Frame 2-3
Figure 2-3: Delay faced by an arriving packet 2-9
Figure 2-4: Queue length vs. time for gated token ring 2-12
Figure 2-5: Network topology of two-connected multiple-ring network 2-21
Figure 2-6: Node model for 1-limited 2-MR with intracluster traffic priority 2-22
Figure 2-7: Intracluster traffic breakdown 2-22
Figure 2-8: Waiting time as seen by intracluster packets 2-24
Figure 2-9: Waiting time as seen by intercluster packets 2-25

CHAPTER 3

Figure 3-1: 2-MR node model 3-3
Figure 3-2: Node model for 1-limited 2-MR with intercluster traffic priority 3-4
Figure 3-3: Intracluster traffic breakdown 3-5
Figure 3-4: Expected queue length for main and cluster ring tokens 3-14
Figure 3-5: Node model for ideally balanced load 3-21

CHAPTER 4

Figure 5-1: Network topology of destination removal double ring 4-1
Figure 5-2: Waiting time as seen by an arriving packet in the *ith* intracluster queue 4-3
Figure 5-3: Immediate flow control slot format 4-12

CHAPTER 5

Figure 5-1: Three node token ring network 5-3
Figure 5-2: Markov chain for k where $R_k = 6k$ 5-13
Figure 5-3: Markov chain for k where $R_k = \frac{8}{5}k$ 5-15
Figure 5-4: Markov chain for k where $R_k = \frac{7}{3}k$ 5-16

Figure 5-5: Markov chain for k where $R_k = 6k$, $state_0 = \frac{1}{2}$ 5-18

Figure 5-6: Markov chain for k where $R_k = 4k$, $state_0 = \frac{1}{3}$ 5-19

Table 5-1: Probability density of k 5-14
Table 5-2: Probability density of k 5-16
Table 5-3: Probability density of k 5-17
Table 5-4: Probability density of k 5-18
Table 5-5: Probability density of k 5-20

LIST OF GRAPHS

CHAPTER 1

Graph 1-1: Throughput vs. offered load for ring and contention networks 1-3

CHAPTER 2

Graph 2-1: Comparison of approximate 1-limited token ring analytical results to simulation results ($m = 30, x = 500, R = 3000$) 2-12
Graph 2-2: Comparison of approximate gated token ring analytical results to simulation results ($m = 30, x = 500, R = 3000$) 2-14
Graph 2-3: Comparison of approximate slotted ring analytical results to simulation results ($m = 30, x = 500, R = 3000$, 6 slots) 2-18
Graph 2-4: Mean packet transfer delays for all traffic classes under random traffic 2-28
Graph 2-5: Mean packet transfer delays for all traffic classes under clustered traffic 2-29
Graph 2-3: Comparison of 2-MR and double ring overall mean packet transfer delays 2-29

CHAPTER 3

Graph 3-1: Mean packet transfer delays for all traffic classes under random traffic 3-9
Graph 3-2: Comparison of *11o* and *11i* mean packet transfer delays for worst traffic class under random traffic 3-10
Graph 3-3: Comparison of *11o* and *11i* overall mean packet transfer delays under random traffic 3-11
Graph 3-4: Mean packet transfer delays for all traffic classes under clustered traffic 3-11
Graph 3-5: Comparison of *11o* and *11i* mean packet transfer delays for worst traffic class under clustered traffic 3-12
Graph 3-6: Comparison of *11o* and *11i* overall mean packet transfer delays of all traffic classes under clustered traffic 3-12
Graph 3-7: Mean packet transfer delays for all traffic classes under random traffic for *ggo* protocol 3-17
Graph 3-8: Comparison of *ggo* and *11o* mean packet transfer delays for worst traffic class under random traffic 3-18
Graph 3-9: Comparison of *ggo* and *11o* overall mean packet transfer delays of all traffic classes under random traffic 3-18
Graph 3-10: Mean packet transfer delays for all traffic classes under clustered traffic for *ggo* protocol 3-19
Graph 3-11: Comparison of *ggo* and *11o* mean packet transfer delays for worst traffic class under clustered traffic 3-19
Graph 3-12: Comparison of *ggo* and *11o* overall mean packet transfer delays under clustered traffic 3-20

Graph 3-13: Mean packet transfer delays for all traffic classes under random traffic for *Ideal* protocol 3-22
Graph 3-14: Comparison of *Ideal* and *ggo* mean packet transfer delays for worst traffic class under random traffic 3-23
Graph 3-15: Comparison of *Ideal* and *ggo* overall mean packet transfer delays under random traffic 3-24
Graph 3-16: Mean packet transfer delays for all traffic classes under clustered traffic for *Ideal* protocol 3-24
Graph 3-17: Comparison of *Ideal* and *ggo* mean packet transfer delays for worst traffic classes under clustered traffic 3-25
Graph 3-18: Comparison of *Ideal* and *ggo* overall mean packet transfer delays under clustered traffic 3-25
Graph 3-19: Mean packet transfer delays for all traffic classes under random traffic for *Alg* protocol 3-27
Graph 3-20: Mean packet transfer delays for all traffic classes under clustered traffic for *Alg* protocol 3-28
Graph 3-21: Comparison of *Alg* simulation and *Ideal* analysis of overall mean packet transfer delays for all traffic classes under both traffic patterns 3-28
Graph 3-22: Mean packet transfer delays for all traffic classes under random traffic for 2-MR slotted ring 3-32
Graph 3-23: Comparison of mean packet transfer delays for worst traffic classes of slotted ring, *11o* and *Alg* protocols under random traffic 3-33
Graph 3-24: Comparison of overall mean packet transfer delays of slotted ring, *11o* and *Alg* protocols under random traffic 3-34
Graph 3-25: Mean packet transfer delays for all traffic classes under clustered traffic for 2-MR slotted ring 3-34
Graph 3-26: Comparison of mean packet transfer delays for worst traffic classes of slotted ring, *11o* and *Alg* protocols under clustered traffic 3-35
Graph 3-27: Comparison of overall mean packet transfer delays of slotted ring, *11o* and *Alg* protocols under clustered traffic 3-35

CHAPTER 4

Graph 4-1: Intensity of intracluster traffic passing through each node of a ten node cluster and originating from each node 4-7
Graph 4-2: Comparison of 2-MR and DRDR mean packet transfer delays for all traffic classes under random and clustered traffic 4-9
Graph 4-3: Mean packet transfer delays for traffic originating from each of the nodes of clusters 1 and 3 under clustered traffic on one of the rings 4-10
Graph 4-4: Mean packet transfer delays for traffic originating from each of the nodes of clusters 1 and 3 under clustered traffic on both of the rings 4-11
Graph 4-5: Mean packet transfer delays for all traffic classes under random traffic 4-13
Graph 4-6: Mean packet transfer delays for all traffic classes under clustered traffic 4-13

CHAPTER 5

Graph 5-1: Comparison of exact single token ring analysis to simulation $m = 10, x = 500, R = 3000$ 5-11
Graph 5-2: Comparison of exact single token ring analysis to simulation $m = 30, x = 500, R = 3000$ 5-11

LIST OF SYMBOLS

Symbol	Chapter	Description
a	3	Number of segments separating the tokens after ring initialisation
a_i	2,3	Proportion of intracluster packets transmitted on the main ring
B	2	Number of rings
$b(x)$	5	Service time generating function
$B^*(s)$	5	Laplace transform of service time generating function
C	2-4	Number of clusters in the network
$\boldsymbol{D}$	4	Probability matrix describing traffic pattern
D_{avg}	2,3,5	Mean transfer delay of all packets
d_{ci}	3	Slot waiting time for the ith cluster ring
d_{ei}	3	Effective slot waiting time to the ith cluster intracluster queue
d_i	2,4	Slot waiting time for the ith node
D_i	2,3	Transfer delay of ith cluster intracluster packets
D_i	2,4	Transfer delay of a packet originating from the ith node
d_m	3	Slot waiting time for the main ring
d_{mi}	3	Waiting time for intracluster queue to receive a main ring free slot
D_n	5	Transfer delay of a packet originating from the nth node
D_o	2,3	Transfer delay of intercluster packets
$f^i_{a_j}$	2	Partial derivative of f^i with respect to a_j
$f_{n,k}$	5	Second "type" of term group on LHS of equation (5.16)
$F_{n,k}(z)$	5	Z-transform of $f_{n,k}$
$g_{n,k}$	5	First "type" of term group on RHS of equation (5.16)

$G_{n,k}(z)$	5	Z-transform of $g_{n,k}$
h	4	Number of hops packets have to travel to reach their destination
$\boldsymbol{I}$	5	Identity matrix
k	5	Number of packet lengths between leading and lagging tokens
l	5	Number of ring segments over which a packet extends
m	2-5	Number of nodes in the network
N_i	2-4	Number of nodes in cluster i
$\boldsymbol{p}$	5	State vector
q	5	Queue length
Q_i	4	Node i queue length
Q_{ii}	2	Node i intracluster queue length
Q_n	5	Mean number in the system for node n
q_n	2,5	Queue length immediately after the nth packet has departed or the queue length at the node at which the token arrives at time n
$Q_n(z)$	5	Z-transform of q_n
Q_{oi}	2,3	Intercluster queue length of ith cluster node
Q_{Tc}	3	Intracluster queue length at cluster ring token arrival instant
R	2-5	Ring delay (Ring walk time)
R_{ci}	2,3	Walk time of ith cluster ring
R_k	5	Ratio of ring delay to packet length
R_m	2,3	Walk time of main ring
s	5	Number of segments into which the ring is divided
T	2	Token interarrival time
t	2	Server interarrival time
$\boldsymbol{T}$	5	Transition matrix of Markov chain
T_{ci}	2,3	Token rotation time for the ith cluster ring
t_{ci}	3	Server interarrival time for the ith cluster ring

T_{Rci}	2,3	Residual token rotation time for the *i*th cluster ring
T_{ei}	2,3	Effective token interarrival time for *i*th cluster intracluster queues
t_{ei}	3	Effective server interarrival time to the *i*th cluster intracluster queue
T_{Rei}	2,3	Effective residual interarrival rotation time for *i*th cluster intracluster queue
T_i	4	Probability matrix describing routing of packets over the two rings
T_m	2,3	Token rotation time for the main ring
t_m	3	Server interarrival time for the main ring
T_{mi}	2,3	Main ring token interarrival time to the intracluster queue
T_{Rmi}	2,3	Residual main ring token interarrival time to the intracluster queue
t_{mi}	3	Residual main ring server interarrival time to the intracluster queue
T_{Rm}	2,3	Residual token rotation time for the main ring
T_R	2	Residual token interarrival time
t_R	2	Residual server interarrival time
T_{Ri}	2,3	Residual token walk time of the *i*th ring
T_{wi}	2	Token walk time of the *i*th ring
U	2	Nominal ring utilisation
U_{ci}	3	Nominal utilisation of the *i*th cluster ring
U_i	4	Utilisation of ring at the *i*th node (slotted ring)
U_m	3	Nominal utilisation of the main ring
U_T	4	Utilisation of link downstream of the node (slotted ring)
v	5	The number of customers that arrive at the queue whilst a customer is being served, at steady state
$v(\lambda_n, \Delta_{q_{n-i}})$	5	The number of packets that arrive at node *n* whilst packets are actively being transmitted by node *n - i*

$V_{n,k}(z)$	5	Z-transform of $v\left(\lambda_n, \Delta_{q_{n-k}}\right)$
v_R	5	The total number of packets that arrive at the node whilst the token is being passed between nodes
$V_R(z)$	5	Z-transform of v_R
W	2	Packet queuing delay
W_i	2,3	Queuing delay of node i intracluster packets
W_i	2,4	Queuing delay of packet originating from the ith node
W_o	2,3	Mean queuing delay of all intercluster packets
W_{oi}	2,3	Queuing delay of node i intercluster packets
x	2-5	Packet length (fixed, except in section 5.1)
X	3	Number of intracluster packet arrivals during cluster ring token rotation
X_i	2	Queue length at token arrival
Y	2	Total number of packets transmitted during a token rotation
Y_o	3	Total number of intercluster packets transmitted during a main ring token rotation
z_i	2,3	Variable used in extended Newton-Raphson method
α	2,4	Time remaining for current slot to pass by when packet arrives
α_i	3	Ratio of ith cluster to total intercluster packet arrival rates
$\alpha_{n,k}$	5	Coefficient of $Q_{n-k}'(1)$ in $F_{n,k}''(1)$
$\beta_{n,k}$	5	Constant expression in $F_{n,k}''(1)$
$\gamma_{n,k}$	5	Constant expression in $F_{n,k}'(1)$
ϕ	2	Actual utilisation of the ring
ϕ_i	2,3	Actual utilisation of the ith cluster ring
ϕ_m	2,3	Actual utilisation of the main ring
ϕ_n	5	Probability that $q_n = 0$

γ	4,5	Propagation delay of packets
γ_i	2-5	Propagation delay of packets originating from node i
γ_{ij}	2,4	Propagation delay of packets originating from node i, destined for node j
$\kappa_{n,k}$	5	Constant expression in $G_{n,k}'(1)$
λ	2-5	Total packet arrival rate to the network
λ_{cn}	4	Intracluster packet arrival rate to node n
λ_i	2,3	Intracluster packet arrival rate to cluster i (2-MR)
λ_i	2,4,5	Packet arrival rate to node i (ringle/double ring)
λ_m	2,3	Total packet arrival rate to the main ring
$\lambda_{n,k}$	5	Sum of the packet arrival rates to nodes $n-k+1$ to n
λ_o	2,3	Total intercluster packet arrival rate to the network
λ_{oi}	2,3	Intercluster packet arrival rate to cluster i
λ_{on}	4	Intercluster packet arrival rate to node n
λ_T	3,5	Total packet arrival rate to a ring
$\tilde{\lambda}_{n,k}$	5	Sum of the packet arrival rates to nodes $n-m+1$ to $n-m+k$
ρ	2	Node utilisation
ρ	5	Queue utilisation
ρ_n	5	Utilisation of node visited at time n
$\tau_{n,k}$	5	Coefficient of $Q_{n-k}'(1)$ in $G_{n,k}''(1)$
$\omega_{n,k}$	5	Constant expression in $G_{n,k}''(1)$

Note: All times are measured in bit times, unless otherwise specified.

CHAPTER 1

INTRODUCTION

There is an ever present need for better communications systems. Users and applications alike seem to always demand more than is available. Although greater bandwidths and a wider range of services are continually being implemented, the problem has no foreseeable short-term solution. With the boom in multimedia applications and the vast array of possibilities in the field, engineers are working hard to turn out faster, better networks at all levels.

This thesis examines ring networks. They are appropriate for the full range of network levels, including multiprocessor systems, local area computer networks and high speed backbones. There is obviously a lot of competition between different network protocols and topologies. Ring networks offer a number of advantages but also have their limitations. This chapter describes what ring networks are, compares them to other network types and takes a look at most of their fundamental features and issues.

1.1 What are ring networks?

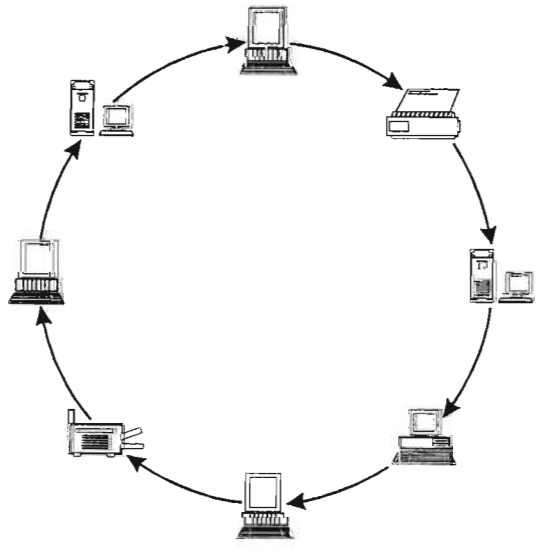


Figure 1-1: Ring network

A ring network is a network topology in which the nodes are connected in a closed loop or ring. Packets circulate around the ring in a pre-determined direction. As opposed to a *bus* or

broadcast network, there is not a direct connection between the receive and transmit transducers at each node. When a packet arrives at a node, it is received by the *receive transducer*, buffered by a shift register for a time period named the *node latency* and then transmitted to the next node on the ring by the *transmit transducer* (see section 1.5). In this way, after being transmitted from a source node, a packet is passed from node to node around the ring until it reaches its destination node. It usually continues around the ring back to the source node.

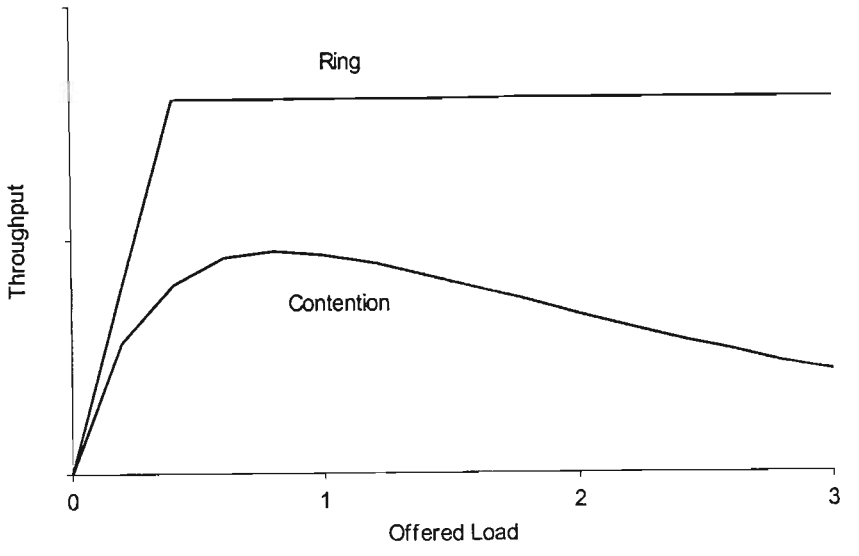
A ring network consists of a closed circuit of unidirectional point-to-point connections. This offers a couple of inherent advantages:

- High speed guided mediums, such as optical fibre, are only useful for point-to-point connections. This makes them suitable for ring networks.
- There is no wasted bandwidth as a direct result of propagation delays. In a bus based system, if a node transmits bits onto the medium, they will only be received by the other nodes at some later time. During this interval, these other nodes will obviously not be able to transmit without a collision occurring. Take, for example, a 100Mbps Ethernet network. Nodes A and B are separated by 100m of cable. Node B is waiting to transmit a packet but the channel is currently in use by A. Assuming, for convenience, that the signal propagates at $2x10^8 m.s^{-1}$, node B detects an empty channel 50 bit times after the packet has left A. This is a significant overhead and occurs to some degree every time a node has to wait for access. Unidirectional point-to-point connections incur no such delays. Data can be transmitted continuously on the link.

1.2 Ring networks vs. contention networks

In order to determine the usefulness of ring networks, it is necessary to evaluate them in terms of the major competition. The two main alternative network types are contention based broadcast networks and switched networks.

Ethernet is by far the most common LAN standard world-wide. It is in fact an implementation of CSMA/CD (Carrier Sense Multiple Access with Collision Detection) which is actually a broadcast network where users have to contend for access. When a user wishes to transmit data, it waits until the network is idle and immediately starts transmission. If a collision occurs, transmission is halted and retried at a later time determined by the *backoff* algorithm. These packet collisions degrade the performance of the network and can theoretically lead to infinite transmission delays and zero network throughput.



Graph 1-1: Throughput vs. offered load for ring and contention networks

In general, when the offered load to a broadcast network surpasses some critical value, the throughput of the network actually starts to decrease, tending asymptotically towards zero (as shown in graph 1-1). This curve shape is characteristic of all networks where users have to contend for access.

Access delays (the length of time a node has to wait before receiving access to the network) are unbounded because collisions can theoretically occur for an infinite length of time. System behaviour is thus somewhat unpredictable and no guarantees can be made concerning service times without additions to the protocol. Together with this goes the concept of fairness. A contention network is inherently *unfair*. Some nodes may receive preferential treatment at the expense of other nodes. Consider a UTP Ethernet network where the nodes are connected in a star configuration using a hub. The node closest to the hub will receive each transmission first and will thus always detect that the channel has become idle before the rest of the nodes. It receives preferential service because it has a greater chance of seizing the channel before the other nodes. On the other hand, the node furthest from the hub will be the last to detect when the channel becomes idle and hence will have the worst chance of seizing the channel. *Backoff* algorithms are designed to counter this problem but it can never be completely solved.

Ring networks, on the other hand, have no losses due to packet collisions. Users generally have equal access to the system in an ordered, *fair* manner. They do not have to contend for available bandwidth. This results in the throughput vs. offered load curve depicted in graph 1-1. Network throughput remains equal to the offered load until the ring becomes saturated. The throughput

reaches a maximum value, which is maintained for any load higher than the maximum throughput.

Combined with the advantages given in section 1.1, a ring network can achieve a far higher throughput than a contention network using the same medium. In their favour, however, contention networks have no inherent access delays at low loads. If a packet arrives at a node it can be transmitted immediately, so long as the channel is idle. Also, the propagation delay from the source node to the destination node will normally be slightly higher for ring networks than for contention networks. Furthermore, contention buses wired in a star configuration are extremely simple and robust. This is probably the major reason for the proliferation of Ethernet.

1.3 Ring networks vs. switched networks

Figure 1-2 shows a diagram of a switched network. All nodes are connected to one or more central switches. There are generally separate connections for received and transmitted traffic.

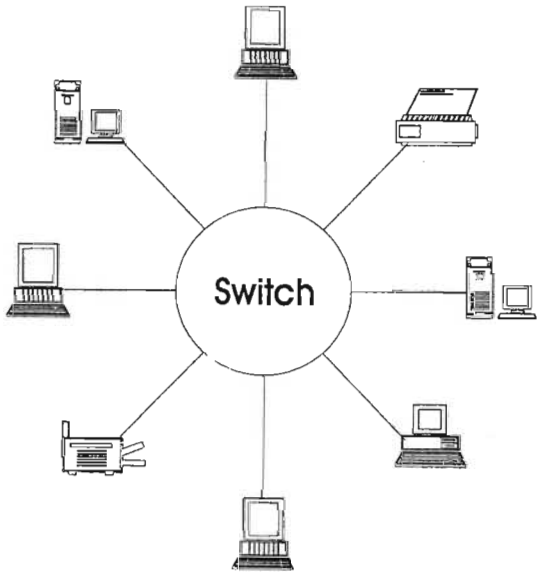


Figure 1-2: Switched network (star topology)

Switched networks sit at the opposite end of the scale to contention buses. They are inherently more complicated, less robust and generally more expensive. However, they offer the promise of far greater bandwidth. In a 100Mbps contention or ring network, all the users share the 100Mbps channel. In a 100Mbps switched network, there is a dedicated 100Mbps for each user. This translates into far greater maximum network throughputs.

Furthermore, in non-blocking switches, nodes do not have to wait for network access. Since each node has its own connection to the switch, it can transmit packets as soon as they arrive. This advantage is nullified to a certain extent because packets may have to be queued in the switch.

This is the main problem with switched networks. They are entirely dependent on the switch. This has a number of disadvantages:

- If the switch fails, the entire network goes down. This can be countered by adding redundancy in the switch, but this adds significantly to the expense of the switch.
- Since packets are queued in the switch, memory is required. This is in addition to the queues at all of the nodes.
- Since queues have finite lengths, packets may be lost. This problem is especially likely to happen when a great deal of the network traffic is destined for the one node. Consider a 100Mbps switched network. If the load destined for a node exceeds 100Mbps for a sufficiently long period of time, queue lengths will obviously grow until they exceed their limit and consequently packets will be lost. Packets will also be lost when the total offered load exceeds the internal bandwidth of the switch.
- Maximum network throughput is limited by the speed of the switch.
- The network is not as scalable as distributed networks (contention buses and rings). In distributed networks the network can be as small as two nodes or as large as the protocol allows. The cost is roughly proportional to the number of nodes. In a switched network, at least one switch is always required. A trade-off has to be made between cost and expandability. A switch can be purchased which just meets the network demand (in terms of node connections and internal bandwidth). If the network is expanded, however, a new switch will have to be purchased. Conversely, a switch with a capacity significantly higher than the current requirement can be purchased, but at significant extra expense. Switches themselves are designed to be expandable by simply plugging in extra cards with additional ports, but the base price of the switch will still be governed by its maximum capacity.
- Switches require routing. When a packet arrives at an input port of the switch, its destination address has to be examined and it has to be routed to the correct output port. This adds extra complexity as well as the requirement for a processor in the switch. This processor will have to be dimensioned for maximum load and will hence be under utilised most of the time. This obviously adds to the cost.

Table 1-1 briefly lists some of the advantages and disadvantages of the various networks.

Network Type	Contention	Ring	Switched
Packet collisions?	Yes	No	No
Routing required?	No	No	Yes
Packet losses?	Policy dependent	No	Yes
Maximum throughput	Low	Medium	High
Fairness	Backoff algorithm	Inherent	Switch dependent
Robustness	High with hub	High with multiple rings or hub	Redundancy required
Access delay	Unbounded	Bounded with n-limited service	Zero
Transport delay	Physical propagation delay	Node latency plus propagation delay	Queuing (in switch) plus propagation delay
Maximum number of nodes	No physical limit	No physical limit	Switch dependent

Table 1-1: Comparison of contention, ring and switched networks

1.4 Wiring

The basic wiring configuration for ring networks is, as the name implies, in a ring. The cabling goes directly from one node to the next in the ring. Ring networks can also be connected in a star configuration using a hub. This allows the network to recover from both node and link failures by switching out the connection to the offending node/link. It also allows inactive nodes to be switched out, which improves network performance. On the negative side, it requires more cabling and results in correspondingly longer propagation delays.

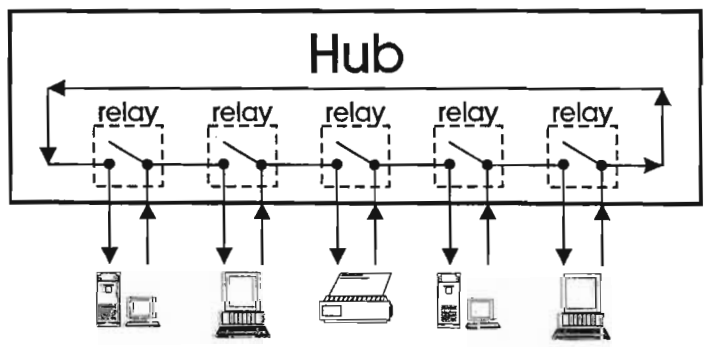


Figure 1-3: Star wiring configuration

1.5 Node structure

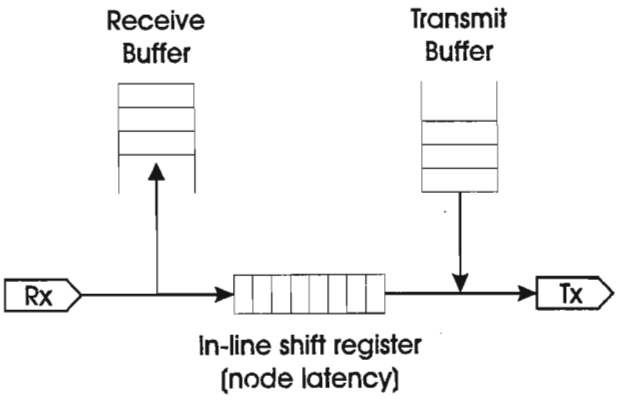


Figure 1-4: Node model

Figure 1-4 shows a simplified model of a generic ring network node. The ring connections are via receive and transmit transducers. There are two packet buffers, one for received packets and one for transmitted packets. Furthermore, there is a shift register in-line with the ring that produces what is known as the node latency, i.e. the delay experienced by data circulating the ring when passing a node. This shift register is required in order to give the node time to examine an arriving token or incoming packet header and decide what to do with it. The packet can either be removed from the ring or passed on to the next node. In order to make this decision, the node usually requires the source or destination address of the packet.

1.6 Spatial bandwidth

Spatial bandwidth refers to the bandwidth distributed around the ring. When one refers to a 100Mbps ring network, one is actually referring to the fact that each link in the network is 100Mbps. If, however, the network has 10 nodes and 10 links, the total spatial bandwidth, i.e. the sum of the bandwidths of the individual links, will be 1Gbps. In practice, the network can never have a 1Gbps throughput because packets have to traverse more than one link to reach their destination. In optimising the performance of ring networks, it is crucial to utilise the available bandwidth as fully as possible.

If the maximum throughput of a network is less than the nominal bandwidth, the spatial bandwidth is not fully utilised. If the maximum throughput is greater than the nominal bandwidth, the network is said to *reuse* the available spatial bandwidth. Instead of every packet traversing every remaining link around the ring after reaching its destination, other packets can *reuse* this remaining bandwidth.

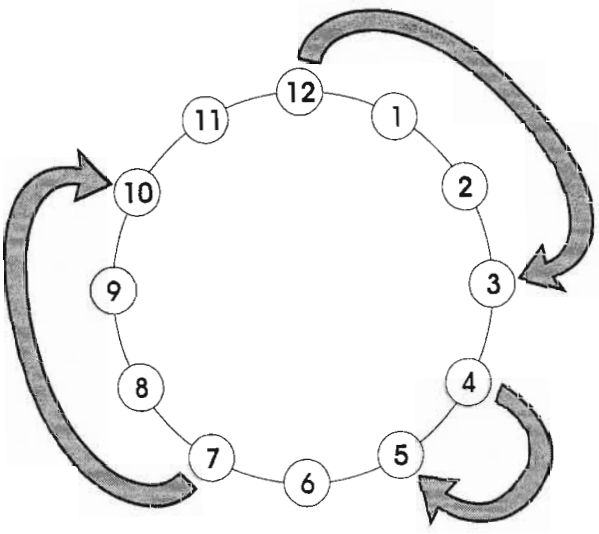


Figure 1-5: Spatial bandwidth

Figure 1-5 shows how the spatial bandwidth of a ring network can be reused. The transmission of three packets is shown, using the bandwidth that is usually used by a single packet when it traverses the entire ring.

1.7 Basic Protocols

There are three fundamental ring protocols: the *token* ring, the *slotted* ring and the *buffer insertion* ring. This section reviews these three protocols as well as a number of other aspects that are involved in ring protocols.

1.7.1 Packet removal from the ring

Packets can either be removed by the source node after a complete circulation of the ring (source removal) or by the destination node upon arrival (destination removal). Source removal offers the advantage of fast and simple flow control but prevents spatial bandwidth reuse.

1.7.2 Token ring

The token ring protocol is by far the most commonly implemented ring protocol. In token ring networks, a unique bit pattern called the *free* or *idle* token is used as a "permission to send." A node with data packets to send, waits until it receives the token. It then seizes the token and commences transmission of a *connector* followed by the data packet. If the token is passed as

soon as the packet has been transmitted, it is referred to as *early* token release. The other option is for the node to retain the token until the packet has completed circulating the entire ring.

1.7.2.1 Service policy

The service policy or discipline refers to the number of packets that can be transmitted (served) each time the token is received.

- *Exhaustive service:* Packets are transmitted until the queue is empty. This service policy has the disadvantage that the token may never leave the current node if there are continuous arrivals of packets to the node.
- *Gated:* All packets present in the queue when the token arrives are transmitted. Any new packets that arrive while the node holds the token have to wait for the next token arrival before they can be transmitted.
- *1-limited:* Only one packet is transmitted for each token arrival. This discipline is also called ordinary or non-exhaustive service. However, these terms, especially the latter, are somewhat ambiguous and hence will not be used in this thesis.
- *n-limited:* A maximum of *n* packets are transmitted for each token arrival.

1.7.3 Slotted ring

A slotted ring is formatted into a constant number of fixed length slots that continuously circulate around the ring. A flag within the slot header is used to indicate whether the slot is empty or full. Any station on the ring that has a packet to transmit waits until it observes an empty slot, sets the flag to indicate that the slot is full and places its data into the slot. When the slot passes the destination node, the packet is read as it passes by and an acknowledgement is usually inserted into the packet trailer. This ensures fast and effective flow control. The packet is removed by the source station upon its return and the slot is once again marked as empty. It is always passed on to the next station (rather than being reused) to prevent *hogging* of the slot by a particular node. Since the total number of slots is known during the initialisation operation, the source station can recognise its transmitted packet by simply counting the number of slots passing by.

1.7.4 Buffer insertion ring

The buffer insertion ring operates more as a random access protocol than token and slotted rings. Referring back to figure 1-4, we see that a node in every ring protocol has an in-line shift

register. For token and slotted ring protocols, this shift register is usually quite small, e.g. two bytes or symbols. For a buffer insertion ring, however, there is an additional shift register, called the *insertion buffer,* in series with the normal one which is large enough to buffer entire packets.

A node is allowed to transmit a packet when the receive transducer detects an idle channel and the *insertion buffer* is empty. If a packet arrives at the receive transducer of a node currently transmitting a data, it is buffered in the *insertion buffer* until the node has completed its transmission. If the node is not currently sending data it simply copies the incoming bits directly to the transmit transducer. A station is not able to send data if the ring is already busy or if its *insertion buffer* contains data. If the *insertion buffer* does contain data, this data must be transmitted before the node can transmit its own packets.

There are two types of buffer insertion protocols: static and dynamic. In a static buffer insertion ring, the *insertion buffer* is a fixed length and can either be switched in or out of the circuit. In a dynamic buffer insertion ring, the *insertion buffer* is variable and its length is dynamically adjusted to the number of bits that have been received since buffering commenced. The dynamic buffer is thus more efficient.

1.7.5 Variable vs. fixed length packets

Token ring networks are not inherently restricted to any packet length at all. In practice, a nominal maximum packet length will usually be imposed to prevent the token spending too long at one node. On the other hand, slots in slotted ring networks are fixed in length. Variable length packets do not fit exactly into an integral number of slots (unless slots are very small, in which case there is a high overhead). As a result bandwidth is wasted in the slots that are not entirely filled. Static buffer insertion rings are similar to slotted rings in that variable length packets result in wasted bandwidth, however dynamic buffer insertion rings allow variable packet lengths that are restricted to some maximum value determined by the buffer length.

The analyses in this thesis assume that all packet lengths are fixed. This biases the results in favour of slotted rings somewhat, because the token passing protocols are unnecessarily restricted. Fixed packets lengths are not unrealistic, however. A major justification for fixed packet sizes is the emergence of ATM. It is a network protocol gaining widespread usage. It has a fixed cell size of 53 bytes, 48 of which contain data. In this thesis, packet sizes are assumed to be 500 bits. This translates to an ATM cell size plus some additional system overhead.

1.7.6 Packet overhead

Every packet that is transmitted around the ring has to have a header and trailer. This results in wasted bandwidth. The overhead increases as packet length decreases. In this thesis, this overhead is ignored. When we refer to an offered load of 50Mbps, we are referring to the ring formatted load, i.e. including the overhead.

1.8 Multiple ring networks

A multiple ring network refers to any network that has two or more physical rings to which nodes are connected. Each node may be connected to any number of these rings. A *symmetrical multiple ring* network refers to a network in which each node is connected to every ring and where each ring has approximately the same ring delay. In such networks, the tokens may either be *synchronous* or *asynchronous*. Synchronous tokens travel together around the ring. The situation is equivalent to having only one token that is used to grant access to all the rings. The advantage of such a system arises from the possibility for a node to transmit more than one packet simultaneously onto different rings or, alternatively, a single packet can be transmitted in *parallel* on all the rings simultaneously. Asynchronous tokens, on the other hand, operate independently of each other. They will generally arrive at different times to a given node, which reduces the *access delay* (see below). From this point onwards, all references to multiple token rings assume that the tokens are asynchronous.

Two different multiple ring topologies are investigated in this thesis. Although symmetrical multiple ring networks are the most common, the major portion of the research in this thesis is devoted to a network known as the "Two-Connected Multiple Ring". This asymmetric network topology can include many rings, but each node is only connected to two of the rings:

- the main ring - to which all nodes are connected
- a cluster ring - to which only a subset of the nodes are connected

The second topology that is examined is a double ring network where the rings are counter-rotating and destination packet removal is employed. Further description of both these will be deferred until later in the thesis.

1.8.1 Multiple ring protocol issues

Multiple rings introduce a host of new protocol complexities. We will proceed to give two examples. The most obvious is whether or not nodes are allowed to transmit packets

simultaneously onto different rings. More complexity in the *medium access control* (MAC) entity is required, but higher performance results. The alternative is to immediately pass on any tokens that arrive at a node that is currently transmitting a packet on another ring.

The second example concerns the gated service policy. This policy is fairly straightforward for a single ring: all packets that are present in a queue when the token arrives are transmitted during that visit of the token. What happens in the multiple ring case when a token arrives at a node that is already being served by another token? There are a number of alternatives, including:

- The new token can create its own "gate", i.e. packets are transmitted until all the packets that were present on the token's arrival have been served. This option requires keeping track of what packets are in the queue when each new token arrives.
- The new token could be restricted to the packets that were gated by the first token to arrive.
- The new token could be simply passed on immediately.
- The original token could be passed on, leaving the new token to service the queue according to the standard gated discipline.

Whatever the chosen protocol, it needs to account for all such issues.

1.8.2 Multiple ring performance

Employing more than one ring in a ring network obviously increases the available bandwidth and hence the maximum throughput. A corresponding reduction in packet transfer delays also results. The performance improvement is not a linear function of the number of rings and is very dependent on the protocol employed.

The total transfer delay of a packet transmitted via a ring network (and in fact any network), can be subdivided into four parts:

- *Access delay*: This is the time interval from the instant the packet arrives at the source node until the node gains access to the network.
- *Queuing delay*: This is the time taken to transmit all packets ahead of the packet in the queue, measured from the instant the source node gains access to the network.
- *Packet transmission delay*: This is the time it takes to physically transmit the packet onto the ring. It is equal to the packet length (including overhead) when measured in bit times.
- *Propagation delay:* This is the time it takes a bit to physically propagate from the source node to the destination node. It includes all node latencies along the way (which can be fairly large in buffer insertion rings). If data transmission is in the same direction for all the rings, the mean propagation delay will remain the same as in the single ring case – half the

ring delay (the time taken for data to circulate around the entire ring). If, however, there are counter-rotating rings and data is always sent on the ring with the shortest distance to the destination, the mean propagation delay is reduced to a quarter of the ring delay.

The packet transmission delay remains unaffected by adding multiple rings. This also holds for the propagation delay, unless there are counter rotating rings. Even in this event, only two rings are required to receive maximum benefit.

Therefore, only the access and queuing delays can be improved by adding rings. The access delay is improved for two reasons:

- Increased number of *access points*: In a multiple ring token ring, for example, the more rings to which a node is connected, the more tokens will be available. This results in reduced token interarrival times. Consequently, the access delay is also reduced because it is equal to the residual token rotation time, i.e. the time interval from a random moment in time until the next token arrival. The relationship is not a linear function of the number of tokens, since they tend to coalesce rather than distribute evenly around the rings. In the slotted ring case, the number of access points is equal to the number of slots passing by, which is proportional to the number of rings.
- Increased *bandwidth*: The available bandwidth is proportional to the number of rings. The traffic on each ring is inversely proportional to the number of rings. The service rate decreases with increased loading of the ring. Thus, the service time is reduced when more rings are added, due to the lower traffic on each ring. This leads to a corresponding decrease in the residual service time, which is equal to the access delay.

The queuing delay is improved solely because of the increased bandwidth and service rate. The faster packets are served, the quicker the queue is emptied and hence the shorter the queuing delay.

1.8.3 Robustness of multiple rings

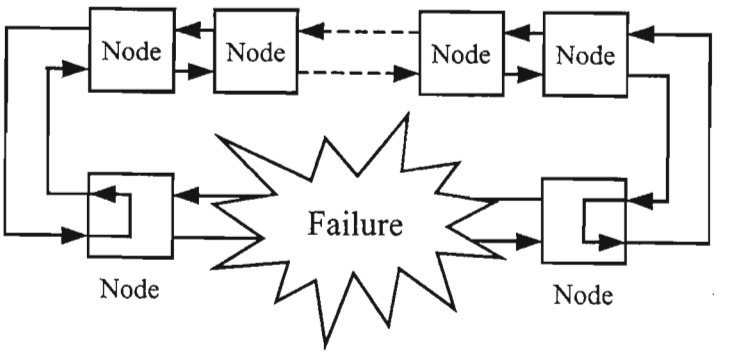


Figure 1-6: Recovery from node and link failures

Besides the performance improvements, multiple rings offer the advantage of increased robustness. If one ring fails, there are other rings to use. If a node or an entire link between two nodes (i.e. all rings) goes down, the network can still recover by pairing counter-rotating rings into new larger rings where each node has two connections to the ring, as shown in figure 1-6.

1.9 Cost issues

In any system and especially networks, one wants to obtain the greatest possible performance, for the lowest possible cost. There are a number of issues that affect the cost of a ring network:

- *Channel bit rate:* The cost of each adapter increases non-linearly with increases in channel bit rate. This cost increase affects most components, including the transceivers, medium access control (MAC) entities, processors, buffer memory and cable type and quality.
- *Number of rings:* The amount of cabling is approximately directly proportional to the number of rings.
- *Number of ring connections per node*: Each ring connection requires a separate transceiver and MAC entity per node. The cost of these items increases almost linearly with the number of ring connections.
- *Buffer space:* The greater the required buffer space, the greater the cost of memory chips. Buffer requirements are dictated by the acceptable probability of the buffer being full, which in turn is governed by a number of factors, including average and peak offered load, packet arrival statistics and the mean and variance of the service rate.
- *Processing requirements:* The amount and speed of processing required affects the cost of the processor. Generally, more complex protocols will require more processing.

The last two are becoming less important as the cost of memory and processors continues its rapid downward trend.

1.10 Traffic patterns

The traffic pattern of a network refers to the distribution of traffic intensity between each pair of nodes in the network. In this thesis we consider two such patterns:

- *Random* traffic: This is the most commonly assumed traffic pattern in the performance evaluation of networks. In a random traffic environment, the mean offered load to all nodes in the network is equal and packets arriving at a node are destined with equal probability to all other nodes in the network.
- *Clustered* traffic: Many networks have clustered traffic, i.e. a transmitted packet is destined with greater probability to nodes in the same cluster as the source node than to the other

nodes in the network. This is evident in most LAN installations, where users typically belong to different work groups and members of a work group tend to communicate with each other more frequently than with members of other work groups. This thesis examines ring protocols that take advantage of this factor to provide greatly improved performance.

We evaluate the performance of various multiple ring protocols. In each instance, both random and clustered traffic patterns are considered.

1.11 Determining the performance of ring networks

There is obviously no better way to determine the performance of a network than to get it up and running and take real world measurements. This is clearly not practical when one is designing and optimising a protocol, however. The performance needs to be determined by modelling the network and obtaining results either by analysis or simulation. Both methods are approximate to some degree. The more accurate the results, the more complex and computationally expensive both methods become.

In the past, computing systems were far slower and more expensive than today. Analytical methods were thus a more attractive option than simulations in most cases. As computers are becoming faster and cheaper, we have reached the point where simulations are perhaps the better alternative. They generally allow considerably more accurate results to be obtained because extra variables can be accommodated. Far fewer assumptions are required, if any at all.

Analytical methods still have an important place, however. They are useful for initial investigations and for system dimensioning. Furthermore, they often give a deeper insight into network performance, which can help in optimisation and design. Also, where the network being examined is extremely large, approximate analyses are often the only possible method to achieve results within reasonable time periods. In these contexts, it is usually more useful to have a simple approximate method than an extremely complex and cumbersome exact method. This approach is taken in this thesis. Simplicity is preferred to accuracy where acceptable. Simulations are used to provide accurate results and validate the analytical models.

1.12 Thesis overview

This thesis includes six chapters. This first chapter introduced the basic concepts of ring networks and some of the issues they involve. It described what they are and how they compare to other network types. Their advantages, as well as limitations, were discussed. The different basic protocols were outlined. The focus of this thesis is multiple ring networks and hence some

time was spent looking at the various issues they involve. Other aspects, including wiring, spatial bandwidth, costs, traffic patterns and performance evaluation, were also discussed.

Chapter two is a literature survey of the existing, relevant work that has been published on the subject of ring networks. Firstly, we take a look at single ring networks, including the token, slotted and buffer insertion ring protocols. The two most commonly implemented ring network protocols are discussed: the IEEE 802.5 standard and FDDI. We outline the history of single ring network analysis and briefly detail the approximate methods we use later in the thesis. We then move on to multiple ring networks. Descriptions and approximate analyses are presented for both the symmetrical multiple token ring and the Two-Connected Multiple Ring - a protocol that was proposed for use in clustered traffic environments. We also investigate four ring networks with spatial bandwidth reuse as well as four fairness algorithms.

Chapter three includes the major portion of new work. We investigate the Two-Connected Multiple Ring more closely and attempt to find a protocol that optimises the performance of the topology. The initial proposal utilised a l-limited token ring protocol. We consider four alternative protocols and evaluate their performance via both analysis and computer simulation. Both random and clustered traffic patterns are considered. Each protocol introduces some performance improvement over the previous proposal, until a near optimum is reached.

In chapter four, we consider an alternative network that can be used in clustered traffic environments: a destination removal double ring. By reusing the available spatial bandwidth and having counter-rotating rings, this network provides significant performance improvements. A general approximate analysis is presented that can accommodate any traffic pattern. We then specifically consider the performance of the network under random traffic and clustered traffic, where nodes in the same cluster are assumed adjacent. We show that the destination removal double ring outperforms the Two-Connected Multiple Ring by a wide margin. Furthermore, a scheme that still allows immediate flow control (a benefit of source removal ring networks) is proposed and its performance effects are determined.

Chapter five considers methods of obtaining exact analytical results for single and multiple ring networks. Firstly, we present an exact analysis of a single token ring with a 1-limited service policy and variable packet lengths, based on a two-moment method. We then look at a novel Markov chain approach that gives exact results for multiple symmetrical token rings with fixed packet lengths at near zero loads.

Finally, chapter six presents the conclusions drawn in this thesis.

1.13 Original contributions of this thesis

The original contributions of this thesis are listed below.

- Five new 2-MR protocols are formulated and their performance is investigated, with the aim of optimising the performance of the 2-MR network for both clustered and random traffic environments.
- Approximate analyses of all the new 2-MR protocols are presented.
- A simple approximate DRDR slotted ring analysis is presented. The performance advantages of the DRDR network in clustered traffic environments, where nodes belonging to the same cluster are spatially close together, are shown.
- An exact analysis of symmetrical multiple unidirectional token ring networks at near zero loads is performed, using a Markov chain approach.

CHAPTER 2

LITERATURE SURVEY

There has been an abundance of literature published on the subject of ring networks, the major portion of which concerns single token ring networks - particularly the analysis thereof. This chapter summarises the most significant and relevant portions of this body of work. Various existing and proposed protocols are examined.

We first take a look at single ring networks. Section 2.1 discusses the token ring protocol and presents the two most commonly implemented ring network protocols: the IEEE 802.5 standard and FDDI, both of which utilise token passing schemes. Section 2.2 then proceeds to outline the plethora of token ring analyses that have been performed. Sections 2.3 and 2.4 then discuss slotted and buffer insertion rings respectively. We also detail the approximate analyses we use later in the thesis.

We then move on to multiple ring networks. Section 2.5 presents an approximate symmetrical multiple token ring analysis. Following this, section 2.6 describes the topology and approximate analysis of the Two-Connected Multiple Ring. In section 2.7, we present four ring networks with spatial bandwidth reuse and finally, section 2.8 examines four different fairness algorithms.

2.1 Token ring protocols

The token ring is an evolution of the polled bus. Computer systems all have some sort of central bus through which the various system components communicate and transfer data. Polling is normally performed by a central controller, usually triggered by an interrupt. The controller itself polls each node in turn to determine whether it requires access to the bus.

A *token passing* bus is a polled system where control becomes *distributed* rather than *centralised*. Instead of a central controller performing the polling directly and granting nodes access in this manner, access is now granted whenever a node receives the token. When the node has completed transmission, it passes the token to the next node in the polling *cycle* according to a *polling order table*. It is important to note that whenever the token is passed, the

address of the next recipient must be explicitly given, since all nodes will receive the transmission.

The token ring simplifies this situation by creating an inherent polling order. The next node to be polled is simply the downstream node on the ring. Addresses are not needed when passing the token since there is only one possible recipient. Furthermore, bandwidth is not wasted as a result of propagation delays as in contention networks (see section 1.2). We now look at the two most common ring network standards, namely IEEE 802.5 and FDDI, both of which are token rings.

2.1.1 IEEE 802.5

The IEEE 802.5 standard specifies a transmission speed of 4Mbps using shielded twisted pair cabling. A maximum of 250 stations is allowed on the ring with a maximum cable length of 10km. As per usual, variable length packets are allowed. The token is only released when the packet has returned to the source node. This does not introduce a significant overhead, since a single packet usually occupies the entire ring. Wiring is normally via a hub.

Consider, as an example, a 50 station network, with a total cable length of 1km. This is equivalent to a mean distance of 10m from the hub. Each node has a latency of one byte, i.e. 8 bit times. For ease of calculation, we assume the signal propagates at two thirds the speed of light. The mean token passing time (the mean time the token takes to travel from one node to the next) is thus 100 nanoseconds, which translates to 0.4 bit times. The ring delay is thus 420 bit times, of which 400 bit times is from node latency. Throughout this thesis we assume packet lengths of 500bits, which is not particularly large. This packet length is longer than the ring delay in our example. Hence very little bandwidth will be lost waiting for the packet to return to its source before the token is passed.

2.1.2 FDDI

The fibre distributed data interface (FDDI) [Burr, 1986], [Ross, 1986] is probably the most common fibre-optic network standard for LAN's and until recently, site backbones as well. As has already been mentioned in the introduction, ring networks are very suitable for fibre because they consist of multiple point to point links and do not inherently waste bandwidth as a result of propagation delays. These two factors have historically made the token ring protocol the choice for optical networks rather than Ethernet, which is elsewhere by far the more common.

FDDI, as with the IEEE 802.5 standard, is based on the token ring protocol. However the physical implementation is somewhat different. FDDI allows maximum of 500 stations with a maximum total cable length of 100km. The transmission rate is 100Mbps. Since packets are normally significantly shorter than the ring delay, *early token release* is employed. A four out of five (4B/5B) group coding scheme is used. Each code group is referred to as a symbol.

Figures 2-1 and 2-2 show the FDDI token-frame (or just token) and data-frame respectively. As can be seen by comparing the two, the token is simply a special type of fixed length frame, i.e. without address or data. The Starting Delimiter (SD) consists of two uniquely recognisable symbols. The Frame Control field (FC) also consists of two symbols that contain format and control bits and indicate the frame type, e.g. token-frame. The minimum node latency in FDDI is two symbols (ten bit times).

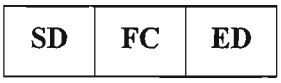


Figure 2-1: FDDI Token-Frame

Figure 2-2: FDDI Data-Frame

The DA and SA fields are the destination and source addresses respectively (either 4 or 16 symbols, as specified in the FC field). The DATA field is a variable length field containing the data packet. Packets are limited to 9000 symbols (4500 bytes). The Frame Check Sequence (FCS) contains a 32-bit cyclic redundancy check, based on the FC, DA, SA, and DATA fields. The Ending Delimiter (ED) consists of one (data frames) or two (token frames) non-data symbols to indicate the end of the frame. The Frame Status (FS) contains the *error detected*, *address recognised* and *frame copied* indicators, each of which is one symbol long.

FDDI has a timed token scheme which limits the time a node can hold the token and consequently how many packets can be transmitted. This scheme will be discussed in section 2.8.1.

Consider a 50 station network with a total cable length of 10km. Each node has a latency of 10 bit times. Once again, for ease of calculation, we assume the signal propagates at two thirds the speed of light. The mean token passing time is thus 1 microsecond, which translates to 100 bit

times. The ring delay is thus 5500 bit times, of which only 500 bit times are a result of node latency.

Without *early token release* and with a packet length of 500 bits and 1-limited service, the maximum throughput would only have been 7Mbps! Even with early token release and a 500 bit packet length, the maximum throughput is limited to 66Mbps for a 1-limited service policy and 80Mbps for a gated policy (remember that due to the 4B/5B group coding, there is a 20% overhead). Comparing this example to the one given for the IEEE 802.5 standard, we see an important trend that is occurring. The ratio of data transmission time to propagation delay is decreasing. This is an important factor to consider when attempting to improve network performance.

2.2 Token ring analyses

We now move on to an examination of the literature on token ring analyses. There has been an abundance of work on the subject. Both approximate and exact analyses have been performed for the various service disciplines and with varying simplifying assumptions. As we shall see, there is a trade-off between accuracy and complexity.

2.2.1 The M/G/1 queuing system

Queuing theory is a branch of mathematics that deals with any system which can be modelled in some way as *customers* that have to be serviced by a *server*. These customers wait for service in a *queue*. This is exactly what we have in computer networks. In our case, the customers are data packets and the server is the network. If packets arriving at a node are buffered in memory (in a queue) whilst waiting to be transmitted, there are at least as many queues present as there are nodes in the network. Furthermore, it may require more than one server to accurately model the network.

The Kendall notation [Schwartz, 1987] is used to describe a queuing system. It is given in the form *A/B/C*, where *A* represents the arrival process, *B* represents the service process and *C* is the number of servers. Each item is given in an abbreviated form. The arrival and service *processes* refer to the statistical distributions of the customer and server interarrival times. Some examples of such processes include:

M: The *M* is short for *Markov*. The interarrival times have a negative exponential distribution and the number of arrivals per unit time is Poisson. The memoryless nature

of this process simplifies the analysis of the queuing system. It is the most commonly assumed arrival process in the performance evaluation of data networks.

D: The *D* is short for *deterministic.* A deterministic process usually refers to one with constant interarrival times, but the term is applicable to any process where the interarrival times are generated from a deterministic function rather than a statistical distribution.

G: The *G* is short for *general.* In this case, the interarrival times have an arbitrary statistical distribution.

There are a variety of other statistical distributions that can be used, but they are beyond the scope of this thesis. The simplest queuing system is the M/M/1 system, which has Poisson packet arrivals, exponential server interarrival times and a single server.

It is difficult to accurately model the packet arrival process of a data network because it is application dependent. As a result, a Poisson (Markov) arrival process is normally assumed. It has shortcomings, however. It ignores the usually bursty nature of data traffic and assumes packets are independent of each other. Nevertheless, it is simple and it does provide a fairly good indication of the network performance. Consequently, we assume that the packet arrival process is Poisson throughout this thesis. The service process, on the other hand, is dependent upon the nature of the network and a more complicated model is required.

The M/G/1 queuing system is defined as one that has Poisson (Markov) customer arrivals, general service statistics and a single server. This type of system has been fully analysed in works such as [Kleinrock, 1975a] and [Gross and Harris, 1985]. It is considered *standard* queuing theory.

The label M/G/1 is appropriate for token ring networks with Poisson arrivals. However, standard queuing theory results cannot be directly applied because the derivation of the results assumes that the service statistics are independent from other statistics, including queue lengths. This is not the case with any polled system. Long polling cycles, i.e. the time between successive polls of a station, and long queue lengths tend to occur together. The one tends to lead to the other.

The general M/G/1 model consists of a single queue with a Poisson arrival rate λ being served by a single server with a service time generating function of $b(x)$, which is an independent variable. Kleinrock's analysis considers the imbedded Markov chain at packet departure instants. The variable q_n describes the queue length immediately after the nth packet has departed. It is described by the following equation

$$q_{n+1} = q_n - \Delta_{q_n} + v_n \tag{2.1}$$

where

$$\Delta_{q_n} = \begin{cases} 0 & if \quad q_n = 0 \\ 1 & if \quad q_n > 0 \end{cases} \tag{2.2}$$

and v_n is the number of customers that arrive between departure instants n and $n + 1$. Up to this point, the equations still hold true for each queue in a token ring network. Kleinrock's next step, however, assumes that v_n is independent of q_n., when he takes the following z-transforms.

$$z^{q_{n+1}} = z^{q_n - \Delta_{q_n}} z^{v_n} \tag{2.3}$$

This step clearly does not hold for the token ring case, since the time between packet departures is dependent on the length of the packets that are departing. Consequently, the number of arrivals (v_n) between departures is not an independent variable. This correlation has to be taken into account if exact results are desired. Therefore, polled systems are a great deal more complicated to analyse.

2.2.2 Polling systems and cyclic servers

The polled system has been around for a while and initially had nothing to do with data networks. The standard version of the initial problem is as follows. A machine repairman is responsible for maintaining the operation of the machines in a factory. He repeatedly checks each of the machines in turn. When he comes across a machine that is not working, he fixes it. He then continues on to the next machine. There is obviously a *walk time* between successive machines.

In queuing theory language, what we have here is a system of N queues (machines) each with a capacity of one customer (machine failure). There is a single server that cycles through the queues, serving each one in turn. This is referred to as a cyclic server. There is a switchover time (walk time) between each of the queues. The service time is equivalent to the length of time required to fix a machine.

A general polling system consists of N queues of capacity K_i, where K_i is often infinite. Customer service times have some known general distribution. Solutions date back to [Mack *et al*, 1957] and [Mack, 1957] which consider fixed and variable service times respectively.

2.2.3 *Infinite capacity systems*

An infinite capacity system refers to an idealised system where the queue lengths are infinite in length. Carsten *et al* [1977] derived exact results for asymmetric token ring systems with exhaustive service. Their approach was based on the queue occupancies at the server visit times. The solution was complex and required solving N^2 equations with N^3 intermediate results.

Ferguson and Aminetzah [1985], derived a simpler solution requiring the solution of N^3 equations for both exhaustive and gated service. Their work was based on earlier efforts by Aminetzah [1975] and Humblet [1978] which introduced the key idea of defining *terminal service times* for each node. Baker and Rubin [1987] extended the results of Ferguson and Aminetzah [1985] to polling systems with general-service order tables.

Many approximate analyses have also been performed, including [Bux and Truong, 1983], [Sethi and Saydam, 1985] and [Coffman and Gilbert, 1986]. [Takagi, 1986] provides an excellent review of the work in this field. Takagi derives exact delay expressions for exhaustive, gated and 1-limited service in his book.

2.2.4 *Finite capacity systems*

In real world systems, queues are limited to some finite capacity K_i. A number of analyses have been performed for single customer ($K_i = 1$) systems, notably [Takagi, 1985] and [Takine *et al*, 1988]. [Robillard, 1974] studied the case of $K_i = 2$. Besides this single exception, up until 1990, the finite capacity system with $1 \leq K_i < \infty$ had only been solved approximately, in such works as [Ganz and Chlamtac, 1988] and [Tran-Gian and Raith, 1988].

Ibe and Trivedi [1990] applied a technique called *stochastic Petri nets* that can numerically handle exponentially distributed service and switchover times. Based on this work, Takagi [1991] derived the exact distribution of waiting time for finite capacity systems with general service and switchover distributions. The result requires the solution of N^3 simultaneous linear equations. Jung and Un [1994] made the comment that the approach taken in [Takagi, 1991] "is too complicated for numerical computation." They performed an exact analysis of an exhaustive service system, using a virtual buffer technique, which provides simpler results.

2.2.5 Approximate token ring analysis

There has been a fairly continuous succession of approximate methods given in the literature. [Blanc, 1992] and [Karvelas and Leon-Garcia, 1993] are examples of fairly recent techniques. However, most of the approximate analyses presented in this thesis are based on the same basic procedure, originally used by Sethi and Saydam [1985] in their single token ring analysis. A general outline of this procedure, which is actually based on standard queuing theory, is first presented to eliminate redundancy in later sections. This is then followed by brief synopses of the single ring analysis performed in [Sethi and Saydam, 1985], as given by Bhuyan *et al* [1989].

2.2.5.1 Outline of method

A data transmission network consists of a group of interconnected nodes, sequentially numbered from 1 to m. Its purpose is to transfer data packets from one node to another across a physical medium. Data packets are generated by a user at the source node. Since network access is generally not instantaneous, it is necessary to buffer arriving packets in a queue at the source node whilst they wait to be transmitted. When a packet is transmitted, it experiences a propagation delay across the physical medium. The packet can only be considered to be received when the last data bit has arrived at the destination node. There will thus be a further delay that is proportional to the length of the packet.

Consequently, a general expression for the transfer delay of packets originating from the ith node is given by,

$$\overline{D_i} = \overline{W_i} + \bar{x} + \overline{\gamma_i} \tag{2.4}$$

where $\overline{W_i}$ is the mean time packets spend in the queue (mean queue waiting time or queuing delay), $\bar{x}$ is the mean packet length (which is assumed to be a fixed length) and $\overline{\gamma_i}$ is the mean propagation delay of packets from the source node to the destination node. Expressions for $\overline{W_i}$ and $\overline{\gamma_i}$ are thus required.

Figure 2-3 shows the delay faced by a packet when it arrives at an infinite queue with a server interarrival time of t. First, it has to wait for the packet at the head of the queue to be served. This time interval is equal to the residual server interarrival time t_R. It then has to wait for all

other packets ahead of it to be served. Finally, the packet itself is served after a further interval of t. Hence an ith node packet has to wait a total time of

$$\overline{W_i} = \overline{t_R} + \overline{Q_i}\bar{t} \tag{2.5}$$

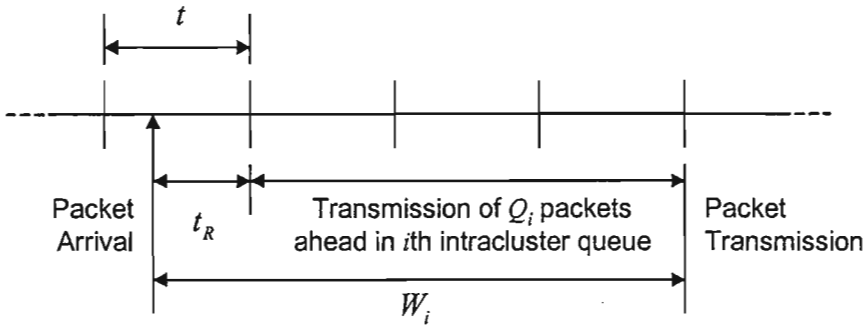


Figure 2-3: Delay faced by an arriving packet

Little's theorem states

$$\overline{Q_i} = \lambda_i \overline{W_i} \tag{2.6}$$

Assuming that $\overline{Q_i}$ and $\bar{t}$ are independent and applying Little's result leads to the standard queuing theory formula

$$\overline{W_i} = \frac{\overline{t_R}}{1 - \lambda_i \bar{t}} \tag{2.7}$$

We know from renewal theory (e.g. [Kleinrock, 1975a]) that the residual life of the random variable t is given by

$$\overline{t_R} = \frac{\overline{t^2}}{2\bar{t}} \tag{2.8}$$

which can also be written as

$$\overline{t_R} = \left(1 + \frac{\sigma_t^{\,2}}{\bar{t}^2}\right)\frac{\bar{t}}{2} \tag{2.9}$$

In general, there is no exact explicit formula that can be obtained for the service statistics of the queue. The approach taken in [Sethi and Saydam, 1985] was to approximate t as a function of the node utilisation

$$t = f(\rho) \tag{2.10}$$

where the utilisation is defined as follows

$$\rho = \lambda_i \bar{t} \tag{2.11}$$

Then the only remaining variable to be solved is the mean packet propagation delay. For *i*th node packets, the mean propagation delay is given by

$$\overline{\gamma_i} = \sum_{j=1}^{m} \overline{\gamma_{ij}} p_{ij} \tag{2.12}$$

where $\overline{\gamma_{ij}}$ is the mean propagation delay from node i to node j and p_{ij} is the probability that a packet that originates from node i is destined for node j. If the destination node is uniformly random, this expression simplifies to

$$\overline{\gamma_i} = \frac{\sum_{j=1}^{m} \gamma_{ij}}{m} \tag{2.13}$$

For a ring network where the nodes are spaced evenly around the ring, this further simplifies to

$$\bar{\gamma} = \frac{R}{2} \tag{2.14}$$

where R is the ring delay (ring walk time), which is the time it takes a bit to physically propagate around the entire ring. It consists of time it takes to physically traverse each link as well as the node latency of each node on the ring.

2.2.5.2 1-limited token ring analysis

The single token ring analysis given in [Bhuyan *et al*, 1989] assumes a 1-limited service discipline and symmetrical load, i.e. the packet arrival rates to the various nodes in the network are equal. We consider an m node network, with a packet arrival rate to each node of λ_i, governed by an independent Poisson process. Packets are assumed to have a fixed length x and R is the ring delay, including node latencies. Any overhead associated with the transmission of *connectors* can be assumed to be included in the packet headers.

The server interarrival time for token ring networks is equal to the token rotation time. A general expression for the token rotation time is

$$T = Yx + R \tag{2.15}$$

where Y is the total number of packets transmitted during the token rotation. For 1-limited service, the probability that a node has a packet to transmit is equal to the utilisation of that node. Hence, the number of packets transmitted during a token rotation can be approximated by the binomial expression

$$P(Y = k) = \binom{m}{k} \rho^k (1-\rho)^{m-k} \tag{2.16}$$

which has a mean of $m\rho$ and a variance of $m\rho(1-\rho)$. By definition, we know that

$$\rho = \lambda_i \overline{T} \tag{2.17}$$

Taking the expectations of (2.15) and substituting, we obtain

$$\overline{T} = \frac{R}{1 - m\lambda_i x} \tag{2.18}$$

Furthermore, the variance of T is given by

$$\sigma_T{}^2 = m\rho(1-\rho)x^2 \tag{2.19}$$

By substituting (2.18) and (2.19) into (2.9) (with $t = T$) we can obtain $\overline{T_R}$, which allows us to determine the mean packet queuing and transfer delays.

Although not pointed out in [Bhuyan *et al*, 1989], it is clear from equation (2.16) that Y is limited to a maximum value of m. Consequently, the token rotation time is also limited to a maximum value, as follows.

$$T_{\max} = mx + R \tag{2.20}$$

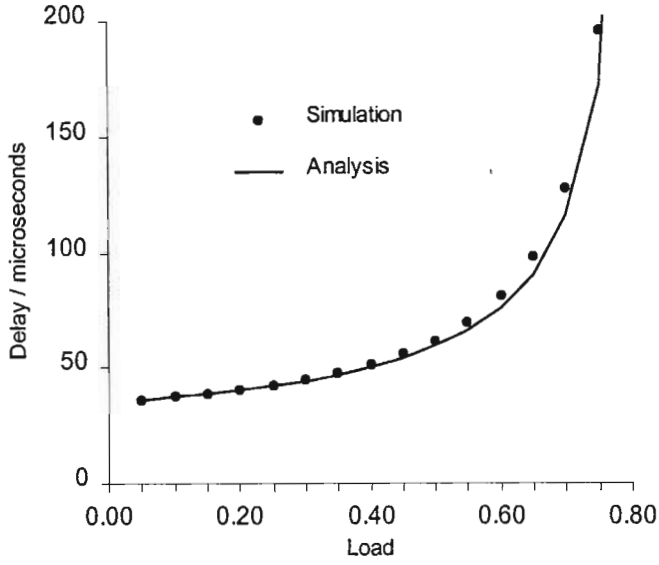
This occurs when every node transmits a packet during the token rotation. When the network load becomes high enough, every node will always have a packet to transmit. The token rotation time is constantly $T_{\max}$ and the mean queuing and transfer delays become infinite for every node. This situation arises whenever the ring utilisation is greater than or equal to one. Notice that we have used the term *ring utilisation*, rather than *node utilisation*. The ring utilisation is equal to the ratio of the total offered load to the maximum useable bandwidth, as given by

$$\phi = \lambda\left(x + \frac{R}{m}\right) \tag{2.21}$$

If the offered load to each node in the network is equal, the utilisation of each node will be equal to the ring utilisation. If, however, the offered load to each node is different, some nodes will have a higher utilisation than others will. Consequently, the node with the highest utilisation will be the first to become congested, i.e. the packet arrival rate will become higher than the service rate and the mean queuing delay will become infinite. However, the ring will not yet be fully utilised, i.e. it will not be transporting as many packets per unit time as it is able to. It would be able to accommodate a greater offered load if it was not restricted by the 1-limited service policy.

Graph 2-1 compares results obtained from the above approximate analysis to simulation results. The network parameters assumed include 30 nodes, a 500 bit packet length and a ring delay of 3000 bit times. If this analysis is compared to the exact analysis presented in section 5.1, it is

clear that it is a great deal simpler. Given that it is sufficiently accurate for most purposes, it is generally a far more attractive alternative.



Graph 2-1: Comparison of approximate 1-limited token ring analytical results to simulation results ($m = 30$, $x = 500$, $R = 3000$)

2.2.5.3 *Gated token ring analysis*

Although we have not come across an instance in the literature, a similar analysis to the analysis in [Sethi and Saydam, 1974] and [Bhuyan *et al*, 1989], as described above, can also be performed for a token ring with a gated service policy. We now proceed to perform such an analysis. We again have an m node network with a Poisson packet arrival rate to the ith node of λ_i. The ring delay is R and the packet length is fixed at x.

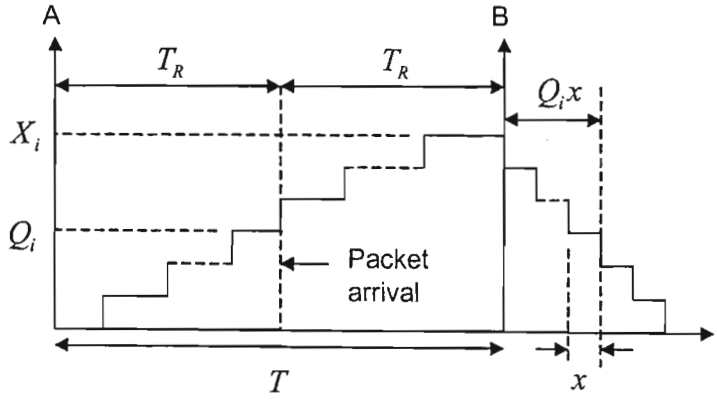


Figure 2-4: Queue length vs. time for gated token ring

Figure 2-4 shows the queue length vs. time for a node in the network. *A* and *B* represent successive token arrival instants. According to the gated service policy, every time the token arrives, all the packets currently in the queue are served. From this point onward, we no longer considered them to be in the queue. Any packets that arrive during service are only served at the next token arrival. As a result, time can be segmented into periods beginning with token arrival *A* and ending with *B*. During this time the queue length increases from a value of zero to some maximum value we will call X_i.

Now an arriving packet will find a number of packets in the queue. The expected number of these packets is equal to the mean queue length $\overline{Q_i}$, since this is in fact the definition of the mean (or expected) queue length. Furthermore, the time period from the last token arrival to the packet arrival instant is, on average, equal to the mean residual token rotation time $\overline{T_R}$. Applying Little's Theorem to this time period we arrive at the equation,

$$\overline{Q_i} = \lambda_i \overline{T_R} \tag{2.22}$$

The describing equation for the queue waiting time of a gated service system differs from that of a 1-limited service system in that service time of each of the packets ahead of the tagged packet in the queue is simply equal to the packet length rather than an entire token rotation time, as follows

$$W = T_R + Q_i x \tag{2.23}$$

Taking expectations of both sides gives

$$\overline{W} = \overline{T_R} + \overline{Q_i} x \tag{2.24}$$

which, unlike the 1-limited analysis, is not an approximation. Using equation (2.22), equation (2.24) can be simplified to

$$\overline{W} = \overline{T_R}(1 + \lambda_i x) \tag{2.25}$$

As we have previously seen in equation (2.8),

$$\overline{T_R} = \frac{\overline{T^2}}{2\overline{T}} \tag{2.26}$$

Hence, we only need to find the statistics of the token rotation time. It is at this point that we make our first and only assumption. We assume that the last token rotation time was equal to the average token rotation time. Since the packet arrival process is Poisson, the probability density function for the total number of packets transmitted during the token rotation time ending at *A* is given by

$$P(Y = k) = \frac{(\lambda \overline{T})^k e^{-\lambda \overline{T}}}{k!} \tag{2.27}$$

which has a mean and variance of $\lambda\overline{T}$, where λ is the total packet arrival rate to the entire network, given by

$$\lambda = \sum_{i=1}^{m} \lambda_i \tag{2.28}$$

The token rotation time is dependent on the total number of packets transmitted during the rotation plus the ring walk time, as given by

$$T = Yx + R \tag{2.29}$$

Taking expectations leads to

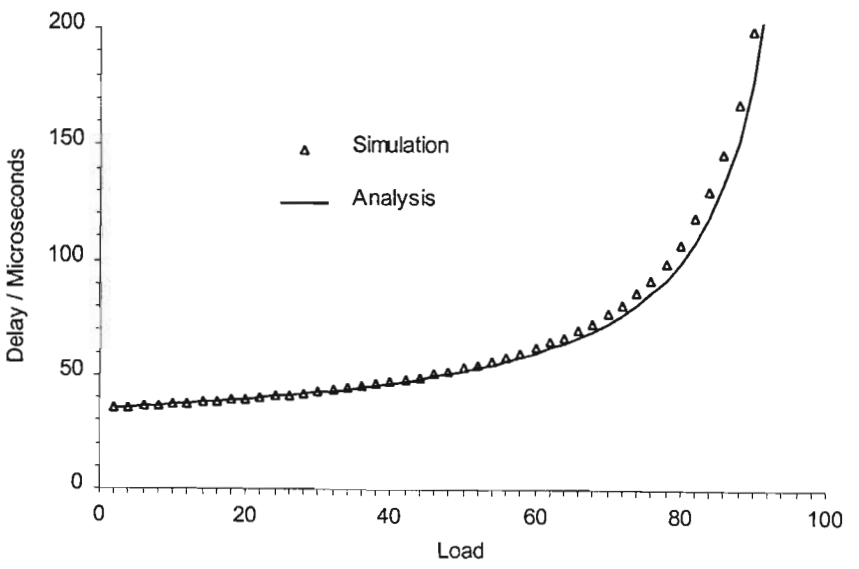
$$\overline{T} = \overline{Y}x + R \tag{2.30}$$

Substituting for $\overline{Y}$ and solving gives

$$\overline{T} = \frac{R}{1 - \lambda_T x} \tag{2.31}$$

Notice that this is exactly the same formula as for the 1-limited protocol. In the gated case, however, the token rotation time is not restricted to a maximum value. It tends towards infinity as the offered load tends towards the nominal bandwidth of the ring. To find $\overline{T^2}$ we square (2.29) and take the expectations of both sides, as follows

$$\overline{T^2} = \overline{Y^2}x^2 + 2\overline{Y}xR + R^2 \tag{2.32}$$



Graph 2-2: Comparison of approximate gated token ring analytical results to simulation results ($m = 30, x = 500, R = 3000$)

Since the variance equals the mean for a Poisson distribution, $\overline{Y^2}$ can be found using the simple formula

$$\overline{Y^2} = \overline{Y}^2 + \sigma_Y{}^2 \tag{2.33}$$

Finally, substituting for $\overline{Y}$ we get

$$\overline{T^2} = \left(\lambda^2 \overline{T}^2 + \lambda \overline{T}\right)x^2 + 2\lambda \overline{T} x R + R^2 \tag{2.34}$$

We now consider a 30 node network with a ring delay of 3000 bit times and a 500 bit packet length. The bit rate is 100Mbps. Graph 2-2 clearly shows that the analysis is accurate, due to its close approximation to the simulation results. It is only at high loads that there is a noticeable deviation, but its magnitude is more than acceptable for most purposes.

2.3 Slotted ring

Perhaps the most famous implementation of a slotted ring network is the Cambridge digital communications network [Wilkes and Wheeler, 1979], which was later renamed the Cambridge fast ring [Hopper and Needham, 1988]. The slotted ring protocol is far less common than the token ring protocol, but has significant advantages over its counterpart. Because there is normally more than one slot circulating around the ring, multiple access points are provided. This leads to lower access delays at low loads because nodes do not have to wait as long for medium access. In a token ring network, nodes have to wait half the ring on average to gain access. In slotted rings on the other hand, nodes have to wait only half a slot length. The access delay is reduced by a factor equal to the number of slots, if the ring delay is the same.

An S slot slotted ring is actually equivalent to an S ring multiple token ring with the following parameters:

- the channel bit rate a factor of S less than the slotted ring
- the ring delays of all the token rings are equal to the ring delay of the slotted ring when measured in seconds (in bit times they are reduced by a factor of S)
- the token ring packet lengths are equal to the slot length (in bits)
- during network initialisation, the tokens are staggered uniformly around the ring, i.e. a packet length apart

Therefore, although the slotted protocol is a *multiple access* protocol, it is not *random access*. It is actually a polled system. Furthermore, it has been stated on numerous occasions in the literature, that an exact analysis of a multiple token ring is intractable. It thus stands to reason the same holds for the slotted ring, which has indeed also been reported.

The slotted ring protocol is described in [Bux, 1981], [Hammond and O'Reilly, 1986] and [Kamal and Hamacher, 1990]. Simple approximate analyses have been performed by both Loucks *et al* [1985] and Bhuyan *et al* [1989]. More accurate models have been presented in [Kamal and Hamacher, 1989], [King and Mitrani, 1987], [Van Arem and Van Doorn, 1990] and [Zafirovic-Vukotic *et al*, 1988]. We will proceed with a brief illustration of the approximate analysis by Bhuyan *et al* [1989].

2.3.1 Approximate slotted ring analysis

Let m be the total number of stations in the network. It is assumed that all m stations are independent of each other and all stations contribute equally to the network traffic. The arrival rate of packets to each station is according to a Poisson process with a mean of λ_i. Buffer lengths are assumed to be infinite. The analysis is once again based on that given in section 2.1.5.1. Thus all we need to derive are the server interarrival statistics, namely $\bar{t}$ and $\overline{t_R}$.

At this juncture, we introduce a second ring utilisation measure that is of crucial importance. This definition of the ring utilisation differs slightly from the first definition given in section 2.1.5.2, in which it was defined as the ratio of the offered load to the maximum useable bandwidth. To differentiate between the two in further slotted ring analyses, we shall refer to this value as the *actual ring utilisation*. The new measure, on the other hand, is the ratio of the offered load to the nominal bandwidth, which we shall refer to as the *nominal ring utilisation*. We have never come across both of them in the same analysis in past literature. Therefore, there has never been a differentiation between the two. Nevertheless, the symbols are different and we will thus continue to use them as they have been defined. The nominal ring utilisation U is defined by

$$U = \lim_{t\to\infty}(C/t) \tag{2.35}$$

where C is the total number of ring formatted bits transmitted from all stations during time period t. If the packet/slot length is x, then the ring utilisation can be expressed as

$$U = m\lambda_i x \tag{2.36}$$

The probability that a passing slot is full is equal to the ring utilisation. We assume that this is a memoryless process. Hence, the probability that a node has to wait for i slots to pass by before it obtains an empty one, is given by the geometric distribution

$$U^i(1-U)$$

The mean waiting time to obtain an empty slot, measured from a slot boundary, is thus given by

$$\bar{d} = \sum_{i=0}^{\infty} ixU^i(1-U) \tag{2.37}$$

which results in

$$\bar{d} = \frac{xU}{1-U} \tag{2.38}$$

Now consider an empty slot arriving at a node that has packets queued. This slot will be filled in a time interval x. The node then has to wait for another empty slot before it can transmit another packet. Thus, the mean server interarrival time is given by

$$\bar{t} = x + \bar{d} \tag{2.39}$$

To determine the mean residual server interarrival time, consider a packet arriving at a node. There will be a slot passing by as it arrives. Since the slot length is constant, the mean time for this slot to completely pass by and a new slot to arrive, is half the slot length. Thereafter, the additional time before an empty slot arrives and the packet at the head of the queue is served, is given by $\bar{d}$. Thus, the mean residual server interarrival time is given by

$$\overline{t_R} = \tfrac{1}{2}x + \bar{d} \tag{2.40}$$

The mean queuing and packet transfer delays can then be calculated as in section 2.1.5.1.

As in the 1-limited token ring case, the mean slot waiting time is also limited to a maximum value. When a slot is used by a transmitting node, it circulates around the ring, past its destination and back to its source. It is then passed on to the neighbouring downstream node on the ring. If the slot is used by every successive node around the ring, it will eventually be available to the original transmitting node after a time interval given by $(m + 1)R$. Dividing this value by the number of slots on the ring, we obtain the maximum mean slot interarrival time, as follows

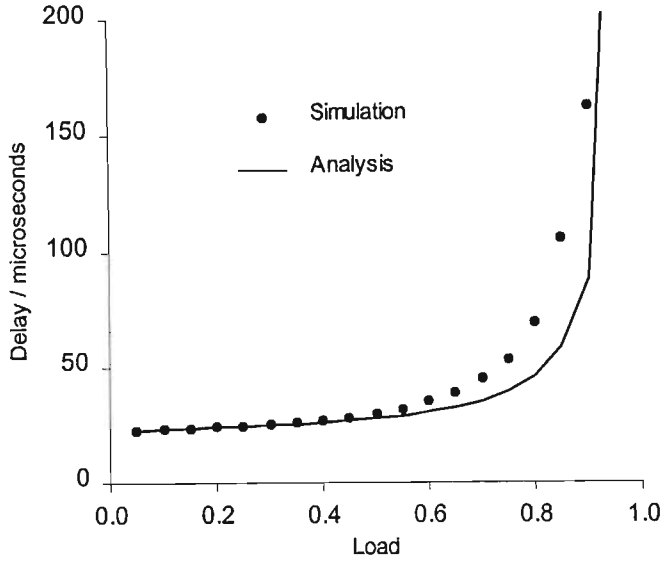
$$\bar{t}_{\max} = (m+1)x \tag{2.41}$$

This situation arises whenever the *actual ring utilisation* is greater than or equal to one. The actual ring utilisation is given by

$$\phi = \lambda_i x(m+1) = U_i\left(\frac{m+1}{m}\right) \tag{2.42}$$

Graph 2-3 compares results obtained from the above approximate analysis to simulation results for a six slot ring. The network parameters assumed, include 30 nodes, a 500 bit packet length and a ring delay of 3000 bit times. The analysis is accurate for low loads and correctly predicts the maximum throughput, but it is inaccurate around the "corner" of the curve. It is somewhat unfortunate that the analysis errs on the low side. Generally, one would prefer to obtain

overestimated delays for network design purposes. Nevertheless, the simplicity of the analysis and its similarity with the approximate token ring analysis have resulted in it being chosen for use in this thesis.



Graph 2-3: Comparison of approximate slotted ring analytical results to simulation results ($m = 30$, $x = 500$, $R = 3000$, 6 slots)

2.4 Buffer insertion ring

Buffer insertion rings [Reames and Liu, 1976], like slotted rings, have a *multiple access* protocol. Whereas slotted rings have S access points, where S is the number of slots on the ring, buffer insertion rings have m access points, where m is the number of nodes. As a result, buffer insertion rings have the nice feature of zero access delay at low loads. However, this comes at the expense of large transmission delays at high load. When the load is high, most of the nodes will have their registers inserted and the ring delay will thus be increased. This overhead becomes less severe as the channel bit rate is increased because the ratio of packet length, and hence the delay added by each buffer, to the propagation delay around the ring, reduces substantially. Destination packet removal is usually employed in buffer insertion rings, which allows spatial bandwidth reuse. The downside of this is that a fairness algorithm is required to ensure that all nodes are treated equally.

Examples of buffer insertion ring implementations are SILK [Hubber *et al*, 1983] and Orwell [Falconer and Adams, 1985]. Approximate analyses have been performed in [Bux and Schlatter, 1983] and [Bhuyan *et al*, 1989].

2.5 Symmetrical multiple ring analysis

Bhuyan *et al* [1989] presented a uniform method of analysing single and multiple ring networks based on token, slotted and buffer insertion protocols. All the methods are based on the single token ring analysis presented in section 2.2. Their analysis of a symmetric multiple token ring network with a 1-limited service discipline, in which all the rings transmit in the same direction, is quite accurate. We proceed to outline results from the analysis that will be used in later sections.

We consider a network with m nodes, each of which has connections to all of the B rings. The B tokens in the B rings are independent and operate asynchronously. Once again, the packet arrival rate to each of the nodes is λ_i, the ring delay of each of the rings is R and the packet length is x (fixed).

The key aspect of the analysis lies in finding the token interarrival statistics. We assume that a particular station has just received a token and proceed to calculate the probability distribution of the time interval before the next token arrives. There are obviously $B - 1$ tokens that could possibly be received next. We arbitrarily label these tokens 1 to i. The token interarrival time is thus given by

$$T = \min\left(T_{Rw1}, T_{Rw2}, \ldots, T_{Rw(B-1)}\right) \tag{2.43}$$

where T_{Rwi} is the residual life of the ith token rotation time. As before, the token rotation time (walk time) of each individual token is given by

$$T_{wi} = N_i x + R \tag{2.44}$$

To find the distribution function of T, we assume that each T_{Rwi} is an independent and identically distributed random variable. Furthermore, it is assumed that the $B - 1$ tokens are uniformly distributed around the $m - 1$ stations. Bhuyan *et al* [1989] point out that this is a major assumption, but it simplifies an otherwise intractable analysis. The tokens tend to coalesce rather than being uniformly distributed around the rings. This phenomenon will be discussed in depth in chapter five. As a result of the assumptions, we get

$$P\left(T_{Ri} \le t\right) = \frac{t}{m\rho x + R} \tag{2.45}$$

for $0 \le t \le m\rho x + R$. Combining this result with equation (2.41), the probability distribution for T can be calculated to be

$$P(T \le t) = 1 - \left(1 - \frac{t}{m\rho x + R}\right)^{B-1} \tag{2.46}$$

which has a mean of

$$\overline{T} = \frac{m\rho x + R}{B} \tag{2.47}$$

Furthermore, the mean residual life of T is given by

$$\overline{T_R} = \frac{m\rho x + R}{B+1} = \frac{B}{B+1}\overline{T} \tag{2.48}$$

The version of this last expression applicable to a double ring network crops up regularly in the remainder of this thesis, namely $\overline{T_R} = \frac{2}{3}\overline{T}$.

2.6 Two-Connected Multiple Ring

The Two-Connected Multiple Ring (2-MR) network was proposed by Lye *et al* [1995] in an effort to optimise network performance under clustered traffic. It only requires two ring connections per node and hence the ring interface is no more expensive than for a double ring. Under clustered traffic it offers significant performance advantages. However, this comes at the expense of poorer performance under random traffic.

This section describes the 2-MR network and details the analysis present in [Lye *et al*, 1995]. Results are verified by computer simulation and compared to the performance of a double ring network.

2.6.1 Topology

Figure 2-5 shows a general 2-MR topology. The ordering of nodes is for illustrative convenience only. C physical clusters are shown with N_i nodes in each cluster i (i = 1, 2, ... C) and a total of m nodes in the network. Each node has two transceivers that connect it to two physical rings. The main ring connects the node to *all* other nodes in the network. The cluster ring, ring i, connects the node to the other nodes in the ith cluster.

Each ith cluster node has two queues: a queue for intercluster traffic and a queue for intracluster traffic. Both queues are assumed to be of infinite length. The intercluster queue buffers traffic destined for nodes outside of the ith cluster and is connected only to the main ring. The intracluster queue buffers traffic destined for other nodes in the ith cluster and is connected to

both the main and cluster rings. Priority is given to the packet at the head of the intracluster queue over that in the intercluster queue when access to the main ring is obtained.

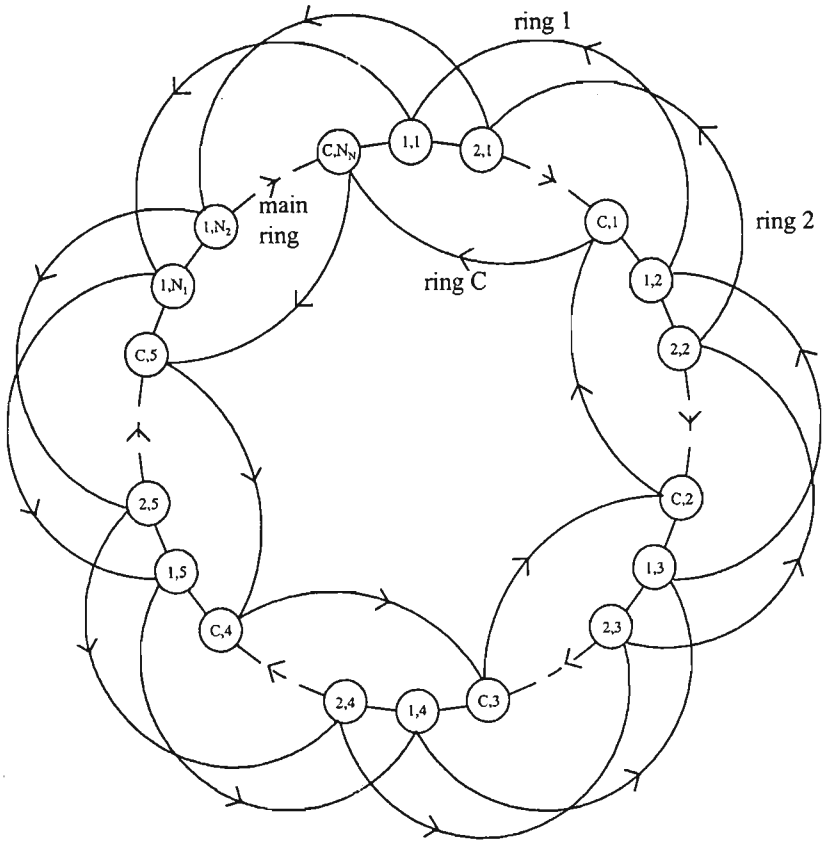


Figure 2-5: Network topology of two-connected multiple-ring network
N_i = Number of nodes in cluster i
C = Number of clusters

2.6.2 Analysis

Lye *et al* [1995] adapted the single token ring analysis in [Sethi and Saydam, 1985] to the 2-MR network. They introduced a technique to "detach" the rings from each other so that each one could be analysed separately. An expression was also found to accommodate the effect of main ring access contention on intercluster packets. They derived an expression for the mean packet transfer delays of the intracluster packets of each cluster, as well as the intercluster packets of the entire network. The overall average packet transfer delay was also found. We now proceed to outline their analysis.

A model of an *i*th cluster node is shown in figure 2-6. As has already been stated, there are two separate queues of infinite length: the intracluster queue and the intercluster queue. Each queue

has packets arriving according to a Poisson process. The packet arrival rates for all the nodes in each cluster are assumed to be equal. The total arrival rate of intracluster packets to cluster i nodes is λ_i. Hence the arrival rate to the intracluster queue of an ith cluster node is λ_i/N_i. The total arrival rate of intercluster packets to all the nodes in the network is λ_o. Furthermore, the arrival rate of intercluster packets to the ith cluster is λ_{oi} and hence the arrival rate to the intercluster queue of an ith cluster node is λ_{oi}/N_i.

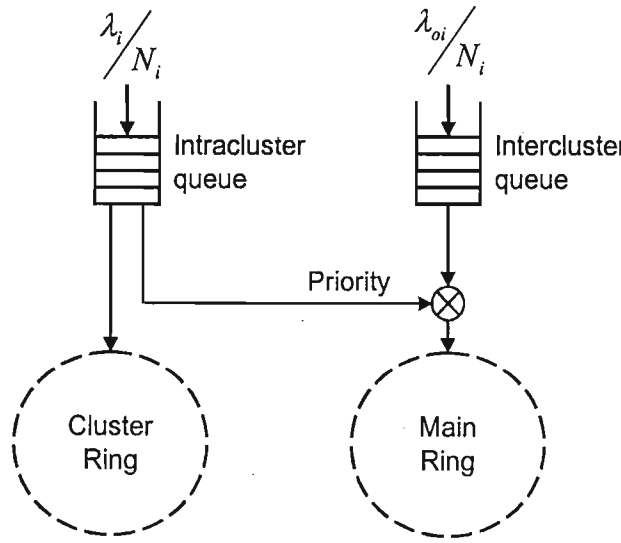


Figure 2-6: Node model for 1-limited 2-MR with intracluster traffic priority

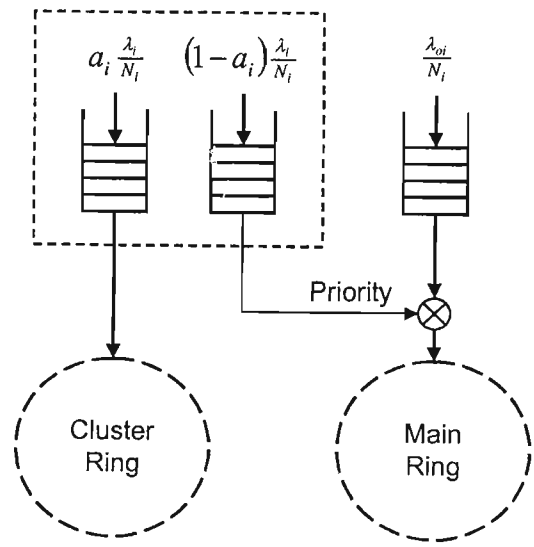


Figure 2-7: Intracluster traffic breakdown

Packets from the intracluster queue can be transmitted on either of the two rings, depending on which token arrives first. Lye *et al* [1995] made the assumption that a fixed, time-invariant portion of the intracluster packets is transmitted onto each of the rings. They effectively split the

intracluster queue into two separate queues, with packets arriving to each queue according to separate independent Poisson arrival processes. This is shown in figure 2-7.

The variable a_i is defined as the proportion of ith cluster intracluster packets that are transmitted on the ith cluster ring. Hence the proportion of ith cluster intracluster packets that are transmitted on the main ring, will be $(1 - a_i)$. For a given load, a_i is a constant time-invariant value. At this stage, we have no expression for obtaining a_i, but we are now able to express the total packet arrival rate to each ring in terms of a_i. This then allows us to determine the token rotation time of each ring, which accomplishes a major part of the analysis.

We have already seen in equation (2.18) that the mean token rotation time is given by the general formula

$$\overline{T} = \frac{R}{1-\lambda_T x} \tag{2.49}$$

where R is the ring delay, λ_T is the total packet arrival rate to the ring and x is the packet length. In this case we have two rings to consider (main and cluster) each with their own R's, λ_T's and x's. The walk time of the main ring is R_m, whereas each cluster ring has a walk time of R_{ci}. The total packet arrival rate to the main ring is given by

$$\lambda_m = \sum_{i=1}^{C}(1-a_i)\lambda_i + \lambda_o \tag{2.50}$$

The total packet arrival rate to the ith cluster ring is $a_i\lambda_i$. All packets have a fixed length of x. According to equations (2.18) and (2.20), the mean token rotation times for the main ($\overline{T_m}$) and cluster ($\overline{T_{ci}}$) rings are thus given by,

$\phi_m < 1$: $$\overline{T_m} = \frac{R_m}{1-\lambda_m x} \tag{2.51}$$

$\phi_i < 1$: $$\overline{T_{ci}} = \frac{R_{ci}}{1-a_i\lambda_i x} \tag{2.52}$$

$\phi_m \geq 1$: $$\overline{T_m} = mx + R_m \tag{2.53}$$

$\phi_i \geq 1$: $$\overline{T_{ci}} = N_i x + R_{ci} \tag{2.54}$$

where ϕ_m and ϕ_i are the utilisations of the main and ith cluster rings respectively and, according to equation (2.21), are given by

$$\phi_m = \lambda_m\left(x + \frac{R_m}{m}\right) \tag{2.55}$$

$$\phi_i = a_i \lambda_i \left(x + \frac{R_{ci}}{N_i} \right) \tag{2.56}$$

When the utilisations reach a value of one, the ring has become saturated, i.e. the load has reached the maximum useable throughput. At this point, the queuing delays will increase without bound. Every node on the ring always has a packet to transmit for each token arrival. Since the service discipline is 1-limited, the token rotation times are limited to the expressions given in (2.53) and (2.54). Both ϕ_m and ϕ_i are dependent upon a_i and hence cannot be calculated independently of the other equations.

The intracluster queue receives tokens from both the cluster ring and the main ring. We need to combine the token rotation statistics for the main and cluster rings to obtain the mean server interarrival statistics for the intracluster queue. The combined arrival rate of tokens to the intracluster queue from both the main and cluster rings is equal to the sum of the rates at which tokens arrive on each separate ring. The mean token rotation times of the main and cluster rings can thus be combined to give an effective mean token interarrival time to the intracluster queue as follows

$$\overline{T_{ei}} = \left(\frac{1}{\overline{T_{ci}}} + \frac{1}{\overline{T_m}} \right)^{-1} \tag{2.57}$$

We still need to derive an expression for a_i. Lye *et al* [1995] assumed that the number of packets transmitted by a node onto each ring is proportional to the token arrival rates from each ring. Therefore, a_i is equal to the ratio of the cluster ring token arrival rate to the total token arrival rate from both rings, as follows

$$a_i = \frac{1/\overline{T_{ci}}}{1/\overline{T_m} + 1/\overline{T_{ci}}} = \frac{\overline{T_m}}{\overline{T_m} + \overline{T_{ci}}} \tag{2.58}$$

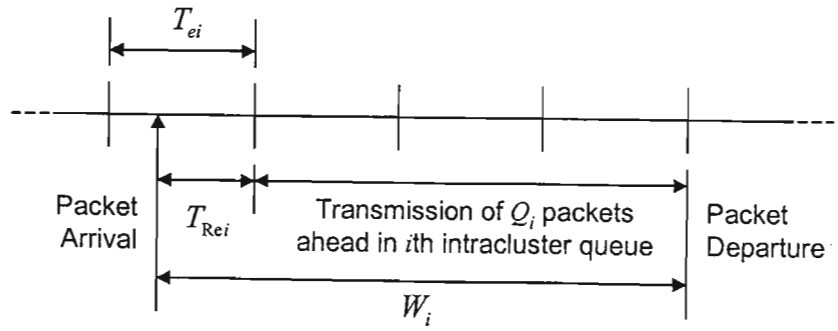


Figure 2-8: Waiting time as seen by intracluster packets

Once this has been found and referring to figure 2-8, we are able to obtain the mean queue waiting time for intracluster packets from equation (2.7), as follows

$$\overline{W_i} = \frac{\overline{T_{Rei}}}{1 - \frac{\lambda_i}{N_i}\overline{T_{ei}}} \tag{2.59}$$

We approximate $\overline{T_{Rei}} = \frac{2}{3}\overline{T_{ei}}$, according to equation (2.48). Finally, we get the mean packet transfer delays

$$\overline{D_i} = \overline{W_i} + x + \tfrac{1}{2}\left[a_i R_{ci} + (1 - a_i)R_m\right] \tag{2.60}$$

Note that the mean propagation delay is different for intracluster and intercluster packets. Equation (2.60) takes this into account by weighting the contributions of each packet type according to the number of packets transmitted on each ring.

We now consider the delay performance of intercluster packets. Since intracluster packets have transmission priority over intercluster packets, an arriving intercluster packet must wait until the intracluster queue is empty before it can be transmitted. When an intercluster packet arrives, the intracluster and intercluster queues have mean lengths of $\overline{Q_{ii}}$ and $\overline{Q_{oi}}$ respectively.

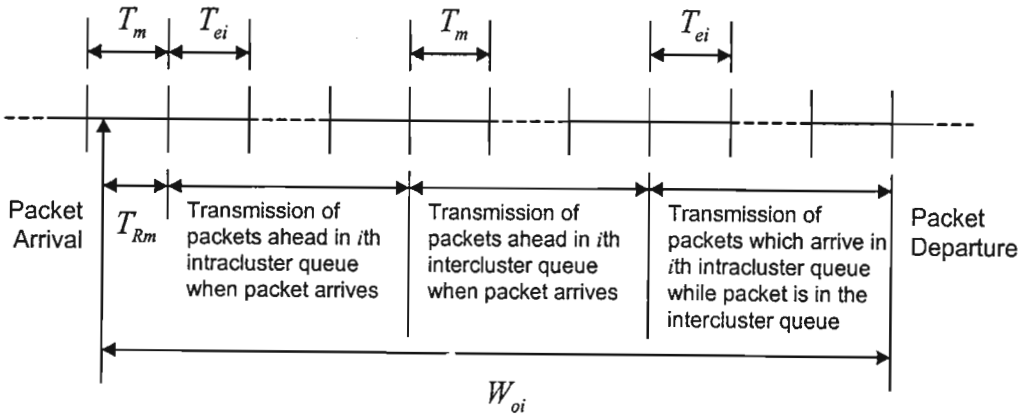


Figure 2-9: Waiting time as seen by intercluster packets

Referring to figure 2-9, we can see that the total waiting time can be broken into three components. Firstly, the packet has to wait for all the packets present in the intracluster queue when it arrives to be transmitted. Secondly, it has to wait for all the packets ahead of it in the intercluster queue to be transmitted. Thirdly, it has to wait for all the intracluster packets that arrive while the packets ahead of it in the intercluster queue are being transmitted. This leads to the following expression for the queuing delay of intercluster packets originating from the *i*th cluster.

$$\overline{W_{oi}} = \overline{T_{Rm}} + \overline{Q_{ii}}\,\overline{T_{ei}} + \overline{Q_{oi}}\,\overline{T_m} + \frac{\lambda_i}{N_i}\overline{T_{ei}}\,\overline{W_{oi}} \tag{2.61}$$

Applying Little's Theorem and rearranging leads to

$$\overline{W_{oi}} = \frac{\overline{T_{Rm}} + \frac{\lambda_i}{N_i}\overline{W_i}\,\overline{T_{ei}}}{1 - \frac{\lambda_{oi}}{N_i}\overline{T_m} - \frac{\lambda_i}{N_i}\overline{T_{ei}}} \tag{2.62}$$

We approximate $\overline{T_{Rm}} = \frac{1}{2}\overline{T_m}$, according to (2.48). We now need to combine the mean queuing delays of the intercluster packets in each of the clusters into a single average. The number of intercluster packets transmitted by each cluster is proportional to the intercluster packet arrival rate to each cluster. Hence, the mean intercluster packet transfer delay is given by

$$\overline{W_o} = \sum_{i=1}^{C} \frac{\lambda_{oi}}{\lambda_o}\overline{W_{oi}} \tag{2.63}$$

where the sum is weighted by the proportion of main ring traffic contributed by each cluster. The mean intercluster packet transfer delay is then given by

$$\overline{D_o} = \overline{W_o} + x + \tfrac{1}{2}R_m \tag{2.64}$$

Finally, the overall packet transfer delay of the entire network is given by

$$\overline{D_{ave}} = \sum_{i=1}^{C}\left(\frac{\lambda_i}{\lambda}\overline{D_i}\right) + \frac{\lambda_o}{\lambda}\overline{D_o} \tag{2.65}$$

where the mean packet transfer delay for each cluster (as well as for intercluster packets) is weighted by the proportion of packets transmitted by that cluster.

In general, it is not possible to obtain explicit algebraic solutions for the above analysis. In order to obtain actual numerical solutions, it is necessary to numerically solve the set of simultaneous equations given in equations (2.50) to (2.56) and (2.58). Lye *et al* [1995] used the Extended Newton-Raphson method. Since it is used extensively in this thesis, it will be briefly outlined before proceeding.

Given a set of equations $f^i(z_1, z_2 \ldots, z_N) = 0$ we find all

$$f^i_{z_j}(z_1, z_2 \ldots, z_j \ldots, z_N) = \frac{df^i_{z_j}(z_1, z_2 \ldots, z_j \ldots, z_N)}{dz_j} \tag{2.66}$$

and then solve the matrix equation

$$\begin{bmatrix} f^1_{z_1} & f^1_{z_2} & \cdots & f^1_{z_N} \\ f^2_{z_1} & f^2_{z_2} & \cdots & f^2_{z_N} \\ \vdots & \vdots & \ddots & \vdots \\ f^N_{z_1} & f^N_{z_2} & \cdots & f^N_{z_N} \end{bmatrix} \begin{bmatrix} \Delta z_{1,n} \\ \Delta z_{2,n} \\ \vdots \\ \Delta z_{N,n} \end{bmatrix} = \begin{bmatrix} -f^1(z_1, z_2 \ldots, z_N) \\ -f^2(z_1, z_2 \ldots, z_N) \\ \vdots \\ -f^N(z_1, z_2 \ldots, z_N) \end{bmatrix} \tag{2.67}$$

The equations can then be solved iteratively by applying the formula

$$z_{i,n+1} = z_{i,n} + \Delta z_{i,n} \tag{2.68}$$

Returning to the 1-limited 2-MR network with intercluster traffic priority to the main ring, we can reduce our set of equations when $\phi_m < 1$ and $\phi_i < 1$ to the following C equations:

$$f^i = A_i a_i^{\,2} + B_i a_i + C_i a_i \sum_{\substack{j=1 \\ j \neq i}}^{C} a_j \lambda_j + D \tag{2.69}$$

where

$$A_i = \lambda_i x (R_m - R_{ci}) \tag{2.70}$$

$$B_i = -\left[R_{ci}(1 - \lambda x) + R_m (1 - \lambda_i x)\right] \tag{2.71}$$

$$C_i = R_{ci} x \tag{2.72}$$

$$D = R_m \tag{2.73}$$

and we are using $z_i = a_i$. The partial derivatives are then given by

$$f^i_{a_j} = \begin{cases} 2A_i a_i + B_i + C_i \sum_{\substack{k=1 \\ k \neq i}}^{C} a_k \lambda_k & for \quad i = j \\ C_i a_i \lambda_j & for \quad i \neq j \end{cases} \tag{2.74}$$

Solutions for all the other combinations of ϕ_m and ϕ_i can also be found [Lye *et al*, 1995]. They are somewhat simpler and will not be reproduced here.

2.6.3 Results

Lye *et al* [1995] examined the performance of the 2-MR network under both random and clustered traffic. They used the following network parameters

- total number of nodes in network, $m = 30$
- number of nodes in clusters 1 and 2, $N_1 = N_2 = 10$
- number of nodes in clusters 3 and 4, $N_3 = N_4 = 5$
- channel bit rate = 100Mbps
- ring delays, $R_m = R_{ci} = 3000$ bit times
- packet length, $x = 500$bits
- *Random traffic packet arrival rates*

  By decomposition of independent Poisson processes, the packet arrival rates are given by

$$\lambda_i = P_{ii}\lambda = \frac{N_i}{m}\frac{N_i - 1}{m - 1}\lambda,$$

$$\lambda_{oi} = P_{oi}\lambda = \frac{N_i}{m}\frac{m - N_i}{m - 1}\lambda$$

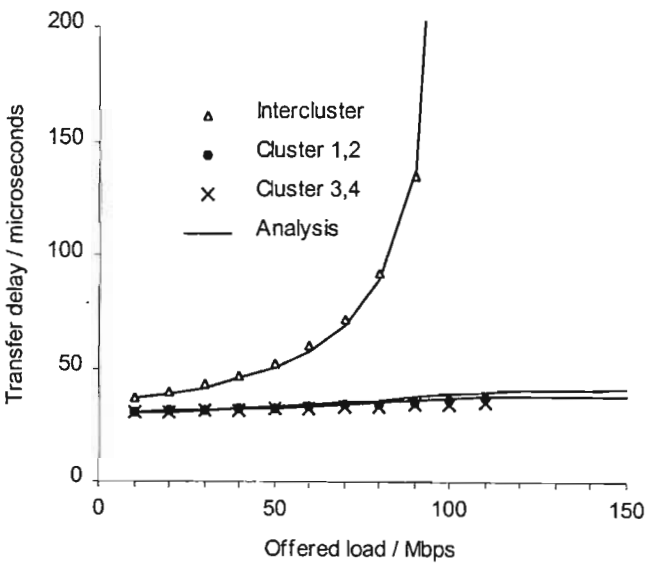
$$\text{and } \lambda_o = \sum_{i=1}^{C} P_{io}\lambda = \left(1 - \sum_{i=1}^{C} P_{ii}\right)\lambda$$

where P_{ii} is the probability that a packet originates from an ith cluster node and is destined for another ith cluster node. P_{io} is the probability that a packet originates from an ith cluster node and is destined for a node in a cluster other than i. These formulae give numerical values of $\lambda_1 = \lambda_2 = 0.023\lambda$, $\lambda_3 = \lambda_4 = 0.103\lambda$ and $\lambda_o = 0.747\lambda$, i.e. 75% of the network traffic is due to intercluster packets.

- *Clustered traffic arrival rates*

  As in Lye *et al* [1995], the packet arrival rates used for the clustered traffic results are $\lambda_1 = \lambda_2 = 0.19\lambda$, $\lambda_3 = \lambda_4 = 0.26\lambda$, $\lambda_o = 0.1\lambda$ and $\lambda_{oi} = \lambda_o N_i / m$, i.e. 90% of the network traffic is due to intracluster packets.

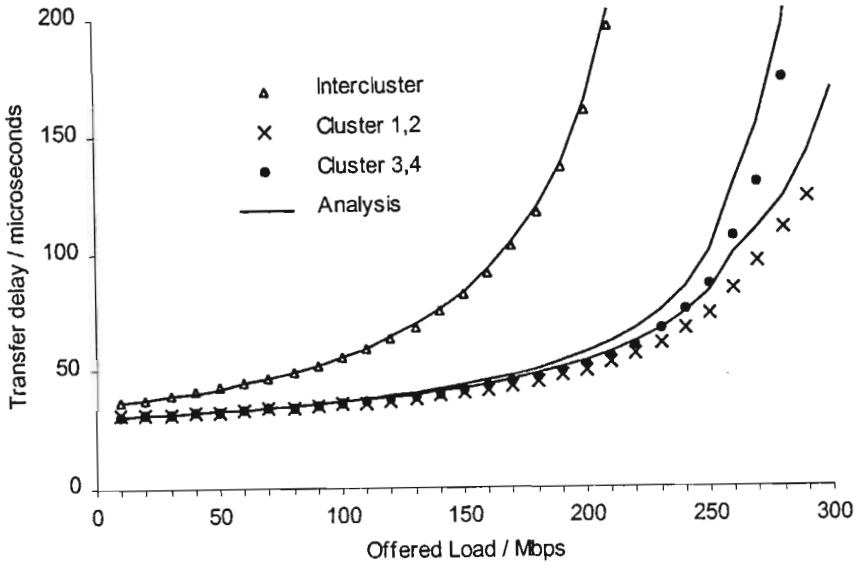
Graph 2-4 shows the performance of the 2-MR under random traffic. It is clear that the intracluster packet transfer delay performance is very good. The delay remains low up to very high loads. However, the intercluster packet delay performance is poor. It is, in fact, considerably worse than a standard double token ring network. This is because the main ring carries over 75% of the total network traffic and thus becomes saturated fairly easily. Since all the intercluster packets have to be transmitted on this ring, their performance suffers accordingly.



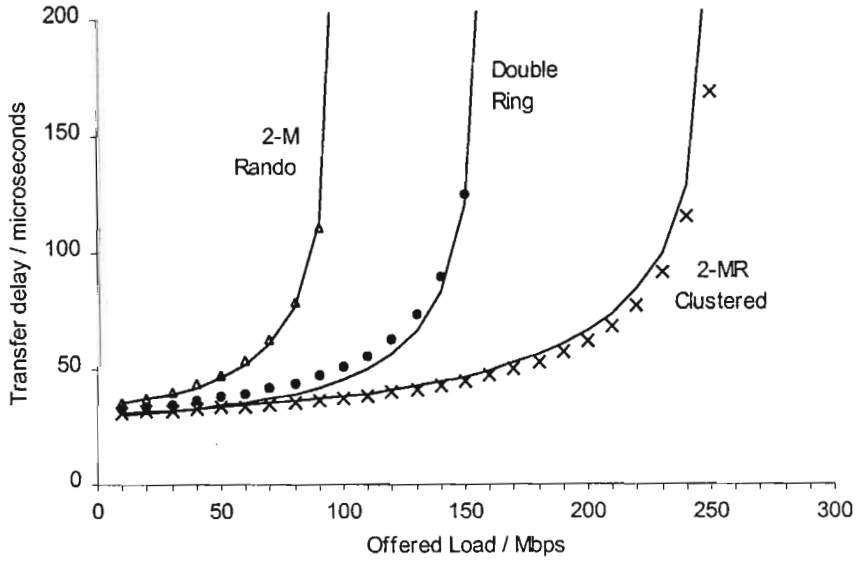
Graph 2-4: Mean packet transfer delays for all traffic classes under random traffic

Graph 2-5 shows the performance of the 2-MR network under clustered traffic. In this scenario, the load is balanced far more evenly over the rings. The intracluster packet performance is again

good although, since the cluster rings are more loaded, not quite as good as the random case. There is a vast improvement in the intercluster packet delay performance.



Graph 2-5: Mean packet transfer delays for all traffic classes under clustered traffic



Graph 2-6: Comparison of 2-MR and double ring overall mean packet transfer delays

The consequences of this improvement are evident when the network performance is compared to a double token ring network, where the tokens operate asynchronously and rotate in the same direction. These results can be seen in graph 2-6, which compares the overall average packet transfer delays. The performance of the double ring is unaffected by the traffic pattern so long

as the propagation delays from the source to destination nodes remains the same. The 2-MR network outperforms the double ring by a significant margin under clustered traffic, as expected. However, this improvement comes at the expense of degraded performance under random traffic.

Lye *et al* [1995] also compared the performance of the 2-MR token ring to a double ring token ring network with priority for intracluster traffic. This network has slightly better intracluster traffic performance than a standard double ring, at the expense of poorer intercluster traffic performance. However, the overall average transfer delays are the same as for the normal double ring.

2.7 Ring networks with spatial bandwidth reuse

A plethora of ring network protocols have been proposed over the last ten years. Each of them offers improved performance, additional services or both. These include names such as the Orwell Ring [Falconer and Adams, 1985], ATM Ring [Ohnishi *et al*, 1989], Playthrough Ring, Parallel Ring and MetaRing to name a few. We will proceed to examine four protocols that offer improved performance, primarily by taking advantage of spatial bandwidth reuse, as well as some additional features. The parallelring and multiple-token protocol are extensions of the token ring protocol, the Playthrough Ring is a circuit switched protocol and the MetaRing allows both slotted and buffer insertion modes.

As reported in [Cidon and Ofek, 1993], there are a number of problems associated with ring networks that implement spatial bandwidth reuse. The most important is that advantage is given to upstream nodes (nodes which are transmitting via the node in question), which can lead to starvation or lack of access to the ring. For example, consider a network where node nine is constantly receiving packets from node seven. As a result, node eight will never receive access to the ring. Spatial bandwidth reuse thus leads to inherently unfair protocols. Measures need to be taken to counter this flaw. Furthermore, it is difficult to implement bandwidth reservation schemes and support multiple access priority levels in ring networks with spatial bandwidth reuse.

2.7.1 Parallelring

The parallelring, proposed by Qu and Landweber [1992], is a token ring network with a single token and destination packet removal that allows multiple concurrent transmissions. A node is allowed to initiate transmission not only upon the receipt of the token but also upon the receipt

of a data frame that is addressed to it. If the node has no queued frames, it either forwards the incoming token or generates a new token while removing the incoming packet. When a station is transmitting its own frame and another frame arrives, the station either removes the frame if it is destined for the station or otherwise delays it in an *insertion buffer*. After the transmission is complete, the station forwards the delayed frame.

Each station, the token and every data frame are assigned a priority. A station with a higher priority than that of the token or of a removed frame is allowed to transmit its own frame. Stations with different applications and time constraints are given different priorities. Qu and Landweber [1992] claim that this property enables the parallelring to support media access fairness and real time services. They showed that the parallelring outperforms the PRONET [Proteon Associates, 1982], a protocol similar to the IEEE 802.5 protocol but with early token release.

The major problem with this protocol is that it no longer maintains the polling order of a normal token ring. Usually, the token is passed from each node to its immediate downstream neighbour and thus each node in the network is polled in turn. In the parallelring, however, the token is only passed to a neighbouring station if there are no frames to transmit. If the station does transmit a frame, the token skips all the nodes between the source and destination stations. This can lead to *starvation*. The performance gains of the parallelring are probably insufficient to justify the extra complexity and overhead required for fairness algorithms.

2.7.2 Multiple-token protocol

Cohen and Segall [1994] proposed a new media access protocol, called "the multiple-token protocol", that increases the throughput and decreases the access delay of the one token protocol. They note that there is a trend in ring networks of decreasing packet to ring delays that is accompanying the increases in channel bit rates. This leads to an increasing proportion of wasted bandwidth. To remedy the situation, their protocol allows multiple tokens to simultaneously circulate around the ring, creating a number of logical rings on the same physical medium. The number of tokens can dynamically be adjusted to optimise performance. Furthermore, the protocol uses timers to maintain spatial separation of the tokens. This counters the previously mentioned phenomenon of token coalescing.

Unlike the parallelring, the multiple-token protocol maintains the polling order of the one token protocol. This results in inherent fairness and hence fairness algorithms are only required if guaranteed deadlines are needed.

The multiple-token protocol is a trade-off between the one token protocol and the buffer insertion protocol. The one token protocol provides for only one access point to the ring, but introduces very little delay at intermediate stations. The buffer insertion protocol, on the other hand, provides up to m access points, where m is the number of stations on the ring. This results in significantly lower access delays at low loads ($1/m$ of the one token protocol), but it increases the ring delay by up to $(m - 1)F$, where F is the maximum frame length. The multiple-token protocol provides N access points and increases the ring delay by up to $(N - 1)F$, where N is the number of tokens. In particular, when $N = 1$, the multiple-token protocol acts like the one token protocol and when $N = m$, it acts like the buffer insertion protocol. Performance can be optimised by correctly choosing N.

The multiple-token protocol holds some promise if an efficient algorithm for determining N can be found. A lot more work is required in this regard, as well as in determining the performance of the protocol.

2.7.3 PLAYTHROUGH ring

The PLAYTHROUGH ring was first proposed by Wilson and Silio [1979]. Further work has been presented in [Silio *et al*, 1992] and [Ghafir and Silio, 1993] amongst a multitude of others. As mentioned previously, the PLAYTHROUGH protocol is a circuit-switched ring protocol. In other words, connections are established and later released between nodes. These connections are used to transmit data. Since the connections do not encompass the entire ring, circuit-switched protocols allow for spatial bandwidth reuse.

The two other principal circuit-switched rings are the Jafari "new loop" and the Leventis "new loop." Jafari *et al* [1980] proposed sending messages on one ring to a central controller that establishes settings for a separate second ring on which data transfer takes place. Leventis *et al* [1982] proposed combining data and control messages on the same ring using a modified form of free and busy tokens.

The PLAYTHROUGH protocol also passes data and control messages on the same ring. It uses a perpetually circulating control frame to establish and release connections. This control frame comprises a leading FLAG character and a trailing GO character. A node can insert a fixed length control message between FLAG and GO, after any other control messages and immediately before GO. Only one control message may be inserted each time. These control messages can either *start* or *stop* a transfer and are removed by the station that inserted them. A new connection is made for each variable length data packet transfer. The PLAYTHROUGH

ring thus has destination data packet removal and source control frame removal. If the control frame arrives at a node currently transmitting data, it is allowed to "play through" in golfing terms, i.e. data transmission is delayed while the control frame is immediately forwarded to the next node.

It is evident that the control frame has a ring circulation time equal to the ring delay plus the transmission time of the control messages that were added in that particular circulation. This time period governs the access time to the network. Furthermore, each connection can only be released when the control frame passes. As a result, if the channel bit rate is relatively high and a long ring delay results, the performance of the PLAYTHROUGH ring will be reduced. Thus, although it provides some definite advantages, it is a less than optimal choice for high speed networks.

2.7.4 MetaRing

The MetaRing, proposed by Cidon and Ofek [1993] and again described in [Ofek, 1994] , is a full-duplex dual ring topology with fairness and spatial bandwidth reuse. The rings transmit data in opposite directions, one clockwise and the other counter clockwise. They are referred to as the inner and outer rings. With the use of an efficient routing algorithm, the furthest that packets have to travel is half way around the ring. Control signals generally travel in the opposite direction to the packets they are controlling, as can be seen in the implementation of the fairness algorithms (discussed in section 2.8.3).

The MetaRing is able to process both synchronous and asynchronous data. It also supports two modes of operation: slotted and buffer insertion. The slotted mode is used when the packet length is fixed, whereas the buffer insertion mode supports variable length packets with an upper limit on size.

Several addressing modes are also supported. The MetaRing not only supports the traditional point-to-point addressing and multicasting, but it also supports a "neighbour mode", "copy mode" and a "point-to-list" mode. The neighbour mode sends from one machine to its nearest neighbour and the copy mode sends to all machines between two stations on the ring. Point-to-list mode includes a list of the addresses of all the packet destinations. As the stations receive the packet, they remove their address from the list.

Cidon and Ofek [1993] present simulation results of the MetaRing operating in slotted mode. They show that the maximum throughput of the network is eight times the nominal bandwidth

of each ring. If a global fairness algorithm is used (see section 2.8.3.1) this value is reduced to between 6.32 and 8 times the nominal bandwidth of each ring, depending on the parameters chosen. The results are reported to be similar for buffer insertion mode [Chen *et al*, 1991].

These results are to be expected when one considers the following argument. Due to shortest path routing, each packet has to travel a maximum of half way around the ring. This results in an average transmission distance of a quarter of the ring. Consequently, an average of four concurrent transmissions is possible. The throughput of each ring is thus four times the nominal bandwidth and the combined throughput is therefore eight times the nominal bandwidth of each ring. This performance gain is theoretically available to any ring network that efficiently reuses the available spatial bandwidth.

Since 1989, there has been a running prototype of the MetaRing at IBM's Watson Research Laboratory, but to date there do not appear to be any commercial products available. However, the MetaRing seems to have served as a stepping stone to new technologies, such as ORBIT and MetaNet [Yung and Ofek, 1995].

2.8 Fairness algorithms

Fairness algorithms are required to ensure that all nodes get sufficient access to the ring. It has previously been pointed out that in ring networks that reuse spatial bandwidth, it is possible for a node to become *starved* of access to the ring because there are continuous transmissions passing by. Fairness algorithms attempt to ensure that packets that have been queued for a long time get transmitted as soon as possible. There are various methods of trying to achieve this. For example, fairness algorithms for unidirectional single slotted rings with spatial bandwidth reuse were introduced in Orwell [Falconer and Adams, 1985], ATMR [Ohnishi *et al*, 1989], MAGNET [Lazar et al, 1990] and BCMA [Heinzmann et al, 1991]. In this section we discuss the fairness algorithms used by FDDI, the helical window token ring and the MetaRing.

2.8.1 Timed token protocol

The timed token protocol has been incorporated into a number of network standards, including the IEEE 802.4 protocol, the High-Speed Data Bus (HSDB), the High-Speed Ring Bus (HSRB), the Survivable Adaptable Fiber Optic Embedded Network (SAFENET) and most notably FDDI.

In this scheme, messages are divided into two separate classes: synchronous and asynchronous. Synchronous messages arrive at regular intervals and may be associated with deadline

constraints. The protocol attempts to control the token rotation time. During the network initialisation, a parameter called the Target Token Rotation Timer (TTRT) is determined, which indicates the expected token rotation time. Each station is assigned a fraction of the TTRT, called the synchronous bandwidth, which specifies a maximum time for which the station is allowed to transmit synchronous packets every time it receives the token. When a node receives the token, it transmits its synchronous packets for a time no longer than its allocated synchronous bandwidth. It is then allowed to transmit asynchronous packets only while the time elapsed since the previous token departure is less than the TTRT, i.e. only if the token has arrived earlier than expected.

It has been formally proved in [Johnson, 1987] and [Johnson and Sevcik, 1987] that under normal operation, the upper bound on the token rotation time is twice the TTRT. Furthermore, Agrawal *et al* [1994] illustrate a synchronous bandwidth allocation scheme that can guarantee synchronous message deadlines for synchronous traffic of up to 33 percent of the available bandwidth. The timed token protocol thus provides "fair" access for synchronous packets. Asynchronous packets, however, may suffer as a result.

2.8.2 Helical window token ring

The helical window token ring [Kschischang and Molle, 1989] attempts to serve packets on a network wide first come first served basis, with a time granularity of one *window* interval.

The operation of a token ring network through time may be represented by the surface of an infinitely long cylinder, with the circumference of the cylinder representing the spatial extent of the ring and the longitudinal axis representing time. As the token circulates around the ring, it plots a continuous curve on the cylinder surface. If the token moves at a constant rate, this curve will be a helix of constant pitch in space-time. Such a helix partitions the time axis, as viewed by a station, into time segments of equal length called *windows*.

The service policy allows each station seeing the idle token for the *k*th time to transmit only those packets that arrived during the *k*th *window*. In order to prevent the token from overtaking the leading edge of the window it is delayed by a *rest period* at each station if need be. The token thus has two different rotation speeds, the window speed and the *backlogged* speed. The network is said to be *backlogged* whenever the token lags behind the leading edge of the *window*. The *backlogged* speed is equal to the normal token rotation speed without *rest periods*.

Kschischang and Molle [1989] point out that the performance of the protocol can be improved if the rest periods are scrapped altogether, by taking the *window* size to be the minimum of some constant w and the current *backlog* of the algorithm at this point. The protocol then becomes similar to the MetaRing global fairness algorithm, which will be discussed shortly.

2.8.3 MetaRing fairness algorithms

The discussion of fairness algorithms for the MetaRing makes up a very large portion of the published information on the protocol because of the problem of starvation that spatial reuse creates. There are two main approaches to fairness: local and global algorithms. Global fairness uses the entire ring as its measure and distributes the bandwidth among all of the nodes in the ring. This ignores the reality that different parts of the ring have very different usage patterns and needs. Local fairness algorithms, on the other hand, take this into account.

2.8.3.1 Global fairness algorithm

The global fairness algorithm, described in [Cidon and Ofek, 1993] and [Ofek, 1994], hinges on a single control packet called SAT, which is short for satisfied. SAT usually rotates around the ring in the opposite direction to the data.

A node is not allowed to send more than a predefined number of packets k between SAT visits. It keeps track of the number of packets it has transmitted since the departure of the last SAT signal in a variable called COUNT. A node is only allowed to transmit a packet if its insertion buffer is empty and its COUNT variable is smaller than k. When the SAT signal arrives, it will be held until either the insertion buffer is empty or COUNT reaches a predetermined length l, where $l \leq k$. In both cases, SAT will be forwarded to the next node and COUNT reset to zero.

The SAT signal cannot be blocked by data flowing in the opposite direction because it can be transmitted in the middle of a data packet without causing problems. This is referred to as "pre-emptive resume priority". In addition, a timeout procedure is implemented that will generate a new SAT if it does not arrive within a maximum period of time. If the timeout occurs prematurely, it is possible to have two or more SAT signals rotating at a time. This situation is also handled. If a node currently holding a SAT signal receives a second one, it will simply discard the latter.

The SAT algorithm can also operate under failure conditions. When a node in the system goes down, the system can be configured as a series of dual bus networks. Under these conditions the SAT signal is transformed into two signals SAT and SAT'. The SAT signal is generated at the head of one of the dual buses and passed down to the other end. The other end removes SAT and sends back SAT' in the opposite direction. When the ring is fixed, the SAT' and any extra SAT signals are removed in a similar fashion to before.

2.8.3.2 Local fairness algorithm

The other approach to eliminating starvation and to ensure fairness is the use of local fairness algorithms. Using this approach, each node decides for itself if it is starved and initiates the algorithm. The simplest local fairness method, described in [Simha and Offek, 1991], only requires one control signal called REQ, or request. In this method, a starved node will send a REQ signal to the upstream nodes. This signal requests that an empty packet be transmitted downstream that only the requesting node can use. Any node that receives a REQ signal, has an empty insertion buffer and has no other REQs to process will send an empty packet. Using this method, the node is guaranteed that an empty packet for its use will arrive in less than m hops, where m is the number of nodes in the system. Unfortunately, this algorithm will only work for fixed packet sizes and therefore a slightly more complex method was devised.

This new method, described in [Chen *et al*, 1993], requires two control signals to control the state of the nodes in the starved area. The first control signal is the REQ signal as discussed before. When a node is starved it creates a REQ signal which is forwarded upstream. This REQ is forwarded until it finds an idle node that can fill the request. Then nodes between the requesting node (the tail of the restricted area) and the idle node (the head of the restricted area) enter a restricted transmission state which limits the amount of packets they can send. These intermediate nodes are called the body of the restricted area.

When the tail of the restricted area reaches its sending limit, it returns to an unrestricted state and creates a GNT signal. The GNT is forwarded to the neighbouring upstream node, which now becomes the tail of the restricted region. When the new tail reaches its limit, it forwards the GNT to the next node until the entire restricted region is gone. There can be several restricted regions in the ring, therefore the algorithm has rules on combining the heads and tails of separate areas into one big restricted area. Sometimes the starvation can encompass the entire ring, in which case the algorithm performs very similarly to the SAT method used before.

Fairness algorithms are necessary to ensure that the delays of the ring are bounded, however they can limit the bandwidth of the system under certain conditions. It has been shown in [Chen *et al*, 1993] that the local method is more efficient than the global method under many traffic configurations. There is thus a trade-off between complexity and efficiency.

2.9 Summary

Chapter two was a review of the relevant existing literature on ring networks. Firstly, we examined the single token, slotted and buffer insertion ring protocols, including the two most common implementations and the many analyses that have been performed. We outlined an approximate method for analysing ring networks in general and proceeded to give the specifics of the approximate token and slotted ring analyses we will use in chapters three and four.

We then looked at two multiple token ring networks. The first of these was the symmetrical multiple token ring, for which we gave some approximate analytical results. Importantly, we then reviewed the 2-MR network. This token ring network was proposed to provide improved performance in clustered traffic environments. The original conclusions of the proposers, that it significantly outperforms a double ring network, were verified.

Next, we examined four protocols that implement spatial bandwidth reuse: the parallelring, the multiple-token protocol, the PLAYTHROUGH ring and the MetaRing. These destination removal networks provide improved performance over their source removal counterparts by allowing multiple concurrent transmissions over the various segments of the ring.

Finally, we discussed four fairness algorithms: the timed token protocol, the helical window token ring and the MetaRing fairness algorithms. Fairness algorithms are especially important in ring networks that reuse the available spatial bandwidth because *starvation* of nodes can occur. Before we undertake an investigation into ring networks that are optimised for clustered traffic environments, we proceed with some necessary preliminary work in chapter three.

CHAPTER 3

TWO-CONNECTED MULTIPLE RING

The literature abounds with analyses of single and multiple ring networks. However, there has been very little work involving clustered traffic. The focus of this thesis, is the use of multiple ring networks in clustered traffic environments. Symmetrical multiple ring networks with source packet removal are not optimal for these environments. We attempt to find more suitable network solutions that offer greater performance at a reasonable cost.

The 2-MR network is one proposed solution. It clearly provides significant performance gains over a double ring network in clustered traffic environments, yet it still requires only two transceivers and MAC entities per node. However the protocol adopted in [Lye *et al*, 1995], as has already been discussed in section 2-6, is not optimum. It seems to have been designed for low intracluster packet transfer delays rather than overall network performance. Furthermore, the 1-limited service discipline assumed, although offering some advantages, is rarely used as such in practice due to its inherent performance penalty. This chapter takes a more in depth look at the 2-MR network and considers various protocols with different performance properties, with the aim of finding an optimum.

As discussed in the introduction, there are three main ring protocols: token, slotted and buffer insertion. The token ring protocol is the most widely implemented, probably because it is the simplest and most flexible. It also offers the most scope for optimisation, as there are many possible variations of token ring protocols suitable for the 2-MR network. Three different practical token ring protocols as well as a slotted ring protocol will be examined – each one introducing a performance improvement over the previous one. A short abbreviated name is given to each protocol to facilitate referencing.

In order to simplify the following text, the term *class* is introduced to refer to the intracluster traffic from a single cluster or the total intercluster traffic in the network. Consequently, there will be a total of $c + 1$ traffic classes in the network (c intracluster + 1 intercluster).

3.1 Optimisation Criteria

There are two performance criteria that can be considered when attempting to find an optimum protocol:

- *Average packet transfer delay:* Fortunately, low transfer delays usually coincide with high maximum throughputs in ring networks, however this is not a strict rule. The matter is further complicated in the 2-MR network by the fact that there are $c + 1$ traffic classes each with their own average transfer delays. A direct measure is needed to compare the performance of various protocols. The two most appropriate measures are:
    1. Overall mean packet transfer delay (mean of the average transfer delays for all the $c + 1$ classes). This gives a good indication of the overall performance of the network.
    2. Largest mean packet transfer delay (this could be any one of the $c + 1$ classes). The class that has the largest average transfer delay can vary with load. The results always give the largest value at a particular load. The curve may consist of the combined results of more than one class.
- *Maximum useable network throughput:* This is the maximum throughput of the entire network before packets from any particular class cease to get serviced. Because some nodes become congested before others, the maximum useable throughput is, in general, slightly lower than the maximum possible throughput, which is the load at which all nodes in the network become congested. For any symmetrically loaded ring network, because there is no contention, average network throughput is equal to the offered load, until the threshold of the maximum throughput is reached.

We thus have two load dependent values (transfer delays) which will be graphed when comparing the performance and a single value for the maximum useable network throughput. It is important to note that maximum useable network throughput will be the value at which the average transfer delay curve tends to infinity with a vertical asymptote.

3.2 Node Model

Figure 3-1 shows a model of a node in a 2-MR network and illustrates the areas where the protocol can be varied, namely:

- *Servers*: A, B and C can be viewed as servers in the queuing theory sense of the word. A and B serve the intracluster queue while C serves the intercluster queue. A, B and C can each have one of various service policies, including 1-limited, gated and exhaustive. The service policy of each may be affected by the *IQSS* and *MRSS*.

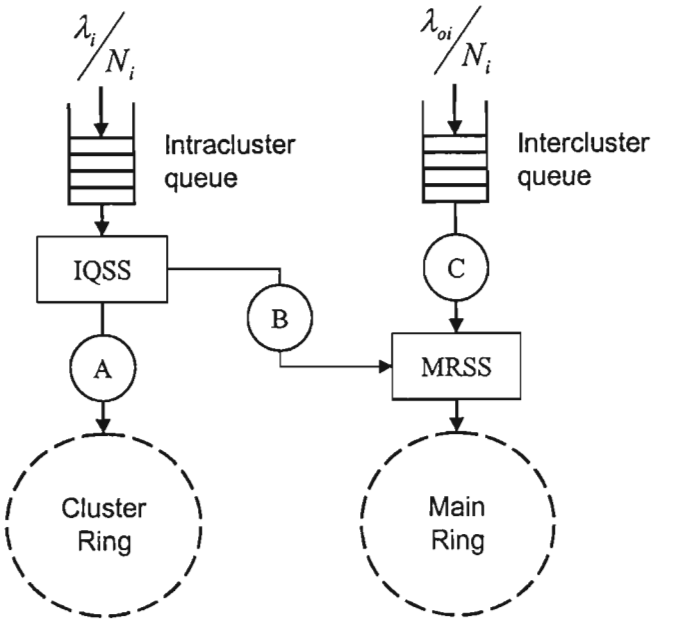


Figure 3-1: 2-MR node model

- *IQSS (Intracluster Queue Server Selector):* This selector (switch) determines how servers A and B contend for access to the intracluster queue. Some examples would be:
  1. A and B can concurrently service the queue.
  2. Either server is denied access to the queue if the other is already serving it.
  3. B is denied access to the queue if A currently has access but A can concurrently service the queue if B has already access.
  4. An arriving server has priority over the current server, which finishes serving the packet it is busy with and then moves on to the next node.

- *MRSS (Main Ring Server Selector):* This selector determines which of servers B and C will be connected to the main ring at any given time. Some examples would be:
  1. Intercluster queue has priority over intracluster queue.
  2. Intracluster queue has priority over intercluster queue.
  3. Packets are served on a FCFS basis from both queues.

There is no distinct division between the three variables. They are interdependent and affect each other. Nevertheless, this model is useful for illustrating where and how the protocol can be varied.

3.3 1-Limited Service, Intercluster Traffic Priority To Main Ring (*11o*)

This section makes only one minor change to the protocol used by Lye *et al* [1991]. The worst aspect of that protocol was the poor performance of intercluster traffic. To counter this, a 1-limited service protocol with intercluster traffic (rather than intracluster traffic) having priority access to the main ring is considered. The abbreviation is short for 1-limited/1-limited/ intercluster, i.e. 1-limited service policy for the intracluster queue, 1-limited service policy for the intercluster queue and intercluster queue access priority to the main ring. The originally proposed protocol will be referred to as the *11i* protocol, where the "*i*" refers the fact that the intracluster queue had access priority to the main ring

3.3.1 Analysis

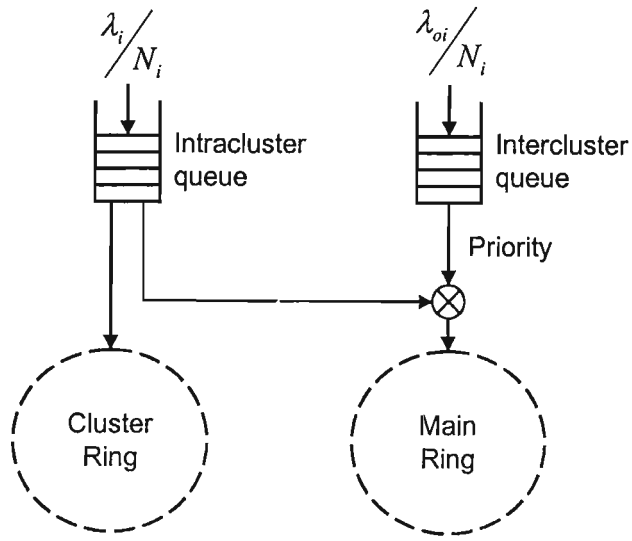


Figure 3-2: Node model for 1-limited 2-MR with intercluster traffic priority

We have an m node network with C clusters. The ith cluster has N_i nodes. The node model shown in figure 3-2 is the same as that presented by Lye *et al* [1995] except for the change that priority access to main ring is given to the intercluster queue rather than the intracluster queue.

Again, the total packet arrival rate to the network is λ. The arrival rate of intercluster packets to the network is λ_o. Furthermore, the arrival rates of intracluster and intercluster packets to the ith cluster (all N_i nodes) are λ_i and λ_{oi} respectively. The variable a_i is defined as the proportion of ith cluster intracluster packets that are transmitted on the ith cluster ring. Hence the proportion of cluster i intracluster packets that are transmitted on the main ring will be $(1 - a_i)$. As in [Lye

et al, 1995], the assumption is made that, for a given load, a_i is a constant time-invariant value. This assumption allows the rings to be detached as shown in figure 3-3.

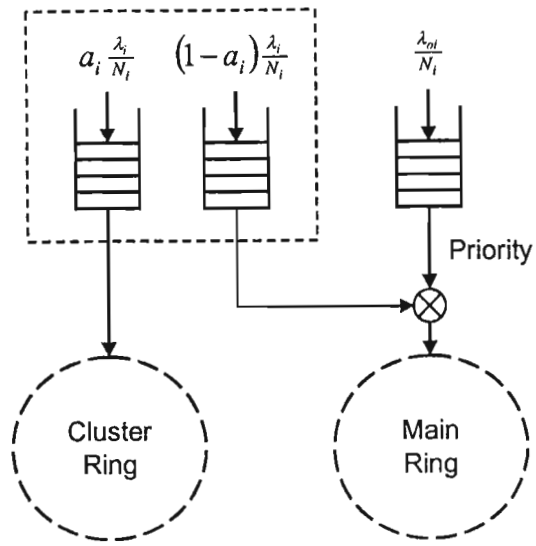


Figure 3-3: Intracluster traffic breakdown

We know from equation (2.18) that the mean token rotation time is given by the general formula

$$\overline{T} = \frac{R}{1-\lambda_T x} \tag{3.1}$$

where R is the ring delay, λ_T is the total packet arrival rate to the ring and x is the packet length. In this case we have two rings to consider each with their own R's, λ_T's and x's. As in equations (2.51) to (2.54), the mean token rotation times for the main and cluster rings are thus given by,

$$\phi_m < 1: \qquad \overline{T_m} = \frac{R_m}{1-\lambda_m x} \tag{3.2}$$

$$\phi_i < 1: \qquad \overline{T_{ci}} = \frac{R_{ci}}{1-a_i\lambda_i x} \tag{3.3}$$

$$\phi_m \geq 1: \qquad \overline{T_m} = mx + R_m \tag{3.4}$$

$$\phi_i \geq 1: \qquad \overline{T_{ci}} = N_i x + R_{ci} \tag{3.5}$$

where, according to (2.55), (2.56) and (2.50),

$$\phi_m = \lambda_m\left(x + \frac{R_m}{m}\right) \tag{3.6}$$

$$\phi_i = a_i\lambda_i\left(x + \frac{R_{ci}}{N_i}\right) \tag{3.7}$$

$$\lambda_m = \sum_{i=1}^{C} (1 - a_i)\lambda_i + \lambda_o \tag{3.8}$$

Both ϕ_m and ϕ_i are dependent upon a_i and hence cannot be calculated independently of the other equations. In practice, the appropriate curves can be tabulated and the correct values chosen.

Since the intracluster queue only receives main ring tokens when the intercluster queue is empty, the mean interarrival time of main ring tokens to the intracluster queue is given by

$$\overline{T_{mi}} = \frac{\overline{T_m}}{1 - \frac{\lambda_{oi}}{N_i}\overline{T_m}} \tag{3.9}$$

This is because the utilisation of the intercluster queue is $\frac{\lambda_{oi}}{N_i}\overline{T_m}$ and hence the probability that it is empty is given by

$$1 - \frac{\lambda_{oi}\overline{T_m}}{N_i}$$

As a result, the formula for a_i, as given in equation (2.58), must be adjusted to

$$a_i = \frac{\overline{T_{mi}}}{\overline{T_{mi}} + \overline{T_{ci}}} = \frac{\overline{T_m}}{\overline{T_m} + \overline{T_{ci}}\left(1 - \frac{\lambda_{oi}}{N_i}\overline{T_m}\right)} \tag{3.10}$$

Now the intracluster queue receives tokens from both the cluster ring and the main ring. The mean token rotation times of these rings can be combined to give an effective mean token interarrival time to the intracluster queue as follows

$$\overline{T_{ei}} = \left(\frac{1}{\overline{T_{ci}}} + \frac{1}{\overline{T_{mi}}}\right)^{-1} \tag{3.11}$$

Once this has been found, we are able to obtain the mean queue waiting times for both intracluster and intercluster packets from cluster i, according to equation (2.7), as follows

$$\overline{W_i} = \frac{\overline{T_{Rei}}}{1 - \frac{\lambda_i}{N_i}\overline{T_{ei}}} \tag{3.12}$$

$$\overline{W_{oi}} = \frac{\overline{T_{Rm}}}{1 - \frac{\lambda_{oi}}{N_i}\overline{T_m}} \tag{3.13}$$

As in equation (2.63), the overall mean queuing delay for intercluster packets is given by

$$\overline{W_o} = \sum_{i=1}^{C} \frac{\lambda_{oi}}{\lambda_m}\overline{W_{oi}} \tag{3.14}$$

Again, we approximate $\overline{T_{Rm}} = \frac{1}{2}\overline{T_m}$ and $\overline{T_{Rei}} = \frac{2}{3}\overline{T_{ei}}$, according to equation (2.48). Finally, as in equations (2.60) and (2.64), we get the mean packet transfer delays

$$\overline{D_o} = \overline{W_o} + x + \tfrac{1}{2}R_m \tag{3.15}$$

$$\overline{D_i} = \overline{W_i} + x + \tfrac{1}{2}a_i R_{ci} + \tfrac{1}{2}(1-a_i)R_{mi} \tag{3.16}$$

From equation (2.65), the overall packet transfer delay of the entire network is then given by

$$\overline{D_{ave}} = \sum_{i=1}^{C}\left(\frac{\lambda_i}{\lambda}\overline{D_i} + \frac{\lambda_{oi}}{\lambda}\overline{D_o}\right) \tag{3.17}$$

Which is the result we require. In order to obtain actual numerical solutions to the above equations, it is necessary to numerically solve the set of simultaneous equations given in equations (3.2) to (3.8) and (3.10). Lye *et al* [1995] used the Extended Newton-Raphson method. In general, it is not possible to obtain explicit algebraic solutions. Since the Extended Newton-Raphson method seems the best available option in terms of speed and simplicity, we use it here again.

We can reduce our set of equations when $\phi_m < 1$ and $\phi_i < 1$ to the following $C + 1$ equations:

$$f^i = (a_i - 1)(1 - a_i\lambda_i x)\overline{T_m} + a_i\left(1 - \tfrac{\lambda_{oi}}{N_i}\overline{T_m}\right)R_{ci} \quad for \quad 1 \le i \le C \tag{3.18}$$

$$f^{C+1} = \overline{T_m}(1 - \lambda_m x) - R_m \tag{3.19}$$

together with the substitutions,

$$z_i = a_i \quad \text{for} \quad 1 \le i \le C \tag{3.20}$$

$$z_{C+1} = \overline{T_m} \tag{3.21}$$

For $1 \le i \le C$:

$$f^i_{a_j} = \begin{cases} (-2\lambda_i x a_i + 1 + \lambda_i x)\overline{T_m} + \left(1 - \tfrac{\lambda_{oi}}{N_i}\overline{T_m}\right)R_{ci} & for \quad i = j \\ 0 & for \quad i \ne j \end{cases} \tag{3.22}$$

$$f^i_{\overline{T_m}} = (a_i - 1)(1 - a_i\lambda_i x) - a_i\tfrac{\lambda_{oi}}{N_i}R_{ci} \tag{3.23}$$

and finally

$$f^{C+1}_{a_j} = \lambda_j x\overline{T_m} \tag{3.24}$$

$$f^{C+1}_{\overline{T_m}} = 1 - \lambda_m x \tag{3.25}$$

When $\phi_m > 1$ and $\phi_i < 1$, we simply need to set $a_i = 1$ for all the clusters, so that the main ring traffic is as low as possible. Obviously the intercluster traffic will now have an infinite transfer delay time, but the intracluster packet transfer delay times can still be found using the same equations as before. Solutions for other combinations of ϕ_m and ϕ_i can be found following

similar lines to the solutions given in [Lye *et al*, 1995]. We will not proceed further since the remaining combinations do not crop up in the examples used in this thesis.

3.3.2 Results

Results are given for both random and clustered traffic environments. In order to make direct comparison possible, we use the same network parameters as those chosen in [Lye *et al*, 1995]:

- total number of nodes in network, $m = 30$
- number of nodes in clusters 1 and 2, $N_1 = N_2 = 10$
- number of nodes in clusters 3 and 4, $N_3 = N_4 = 5$
- channel bit rate = 100Mbps
- ring delays, $R_m = R_{ci} = 3000$ bit times
- packet length, $x = 500$bits
- *Random traffic packet arrival rates*

  By decomposition of independent Poisson processes, the packet arrival rates are given by

$$\lambda_i = P_{ii}\lambda = \frac{N_i}{m}\frac{N_i - 1}{m-1}\lambda,$$

$$\lambda_{oi} = P_{oi}\lambda = \frac{N_i}{m}\frac{m - N_i}{m-1}\lambda$$

$$\text{and } \lambda_o = \sum_{i=1}^{C} P_{io}\lambda = \left(1 - \sum_{i=1}^{C} P_{ii}\right)\lambda$$

  where P_{ii} is the probability that a packet originates from an ith cluster node and is destined for another ith cluster node. P_{io} is the probability that a packet originates from an ith cluster node and is destined for a node in a cluster other than i. These formulae give numerical values of $\lambda_1 = \lambda_2 = 0.023\lambda$, $\lambda_3 = \lambda_4 = 0.103\lambda$ and $\lambda_o = 0.747\lambda$, i.e. 75% of the network traffic is due to intracluster packets.

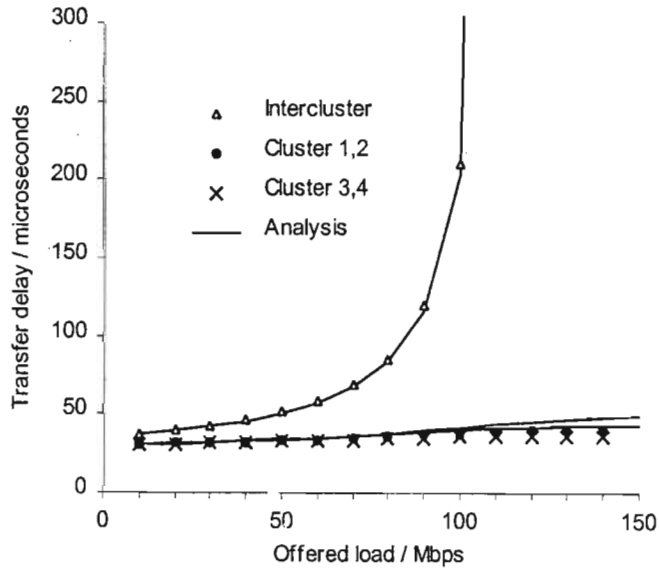
- *Clustered traffic arrival rates*

  As in Lye *et al* [1995], the packet arrival rates used for the clustered traffic results are $\lambda_1 = \lambda_2 = 0.19\lambda$, $\lambda_3 = \lambda_4 = 0.26\lambda$ and $\lambda_o = 0.1\lambda$, i.e. 90% of the network traffic is due to intracluster packets. Furthermore,

$$\lambda_{oi} = \frac{N_i}{m}\lambda_0$$

Graph 3-1 shows the mean packet transfer delays for the different traffic classes under random traffic. The various markers (triangles, dots or crosses) represent simulation results and the solid lines represent the corresponding results from the analytical model. Approximately 75% of the

total network traffic is intercluster traffic and has to be transmitted on the main ring. The other 25% of the traffic is spread over the other four rings. As a result, intercluster traffic performs far worse than the other traffic classes and intercluster packet transfer delays tend to infinity at offered loads of just over 100Mbps.



Graph 3-1: Mean packet transfer delays for all traffic classes under random traffic

Examining the analytical model, we see that the mean intercluster packet queuing delay is given by equation (3.13):

$$\overline{W_{oi}} = \frac{\overline{T_{Rm}}}{1 - \frac{\lambda_{oi}}{N_i}\overline{T_m}}$$

This equation tends to infinity as $\frac{\lambda_{oi}}{N_i}\overline{T_m}$ tends to 1. Setting

$$\frac{\lambda_{oi}}{N_i}\overline{T_m} = 1$$

we get from equation (3.2) that

$$\frac{\lambda_{oi}}{N_i} - \frac{R_m}{1 - \lambda_m x} = 1 \tag{3.26}$$

We have already seen that λ_{oi} is proportional to λ_o. We now introduce the constant α_i to simplify matters, such that

$$\lambda_{oi} = \alpha_i \lambda_o \tag{3.27}$$

where

$$\alpha_i = \frac{N_i}{m}\frac{m - N_i}{m - 1}\frac{\lambda}{\lambda_o} \tag{3.28}$$

Because intercluster packets have access priority to the main ring, for very high intercluster loads there will be no intracluster packets transmitted on the main ring since there will always be intercluster packets waiting for transmission. As a result, $\lambda_m = \lambda_o$ and hence the intercluster packet arrival rate at which the first nodes in the network become congested and hence the mean intercluster packet transfer delay becomes unbounded, is given by

$$\lambda_{o(\text{congestion})} = \frac{N_i}{N_i x + \alpha_{\max} R_m} \tag{3.29}$$

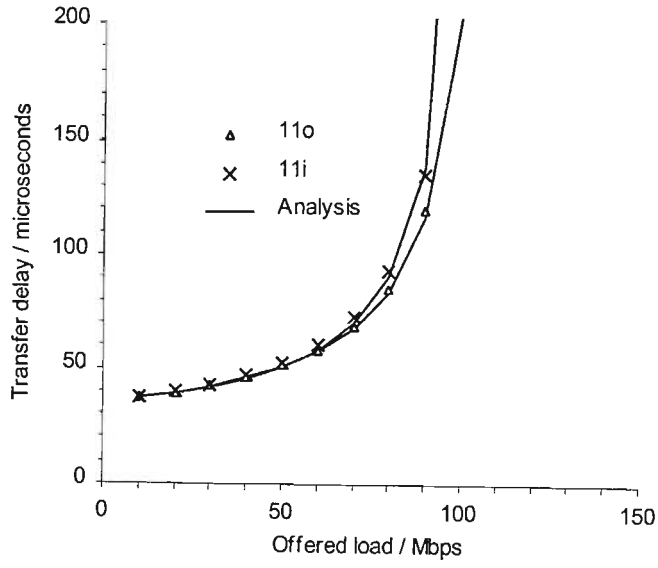
where $\alpha_{\max}$ refers to the largest α_i. In our example, $\alpha_1 = \alpha_2 = 0.144$ and $\alpha_3 = \alpha_4 = 0.230$. Plugging the higher of these two values into equation (3.29) gives

$$\lambda_{o(\text{congestion})} = 0.001625 \text{ packets/bit time}$$

This corresponds to a total offered load of 108,75Mbps. The maximum amount of intercluster traffic that the network can carry is slightly higher than this value. It is given by

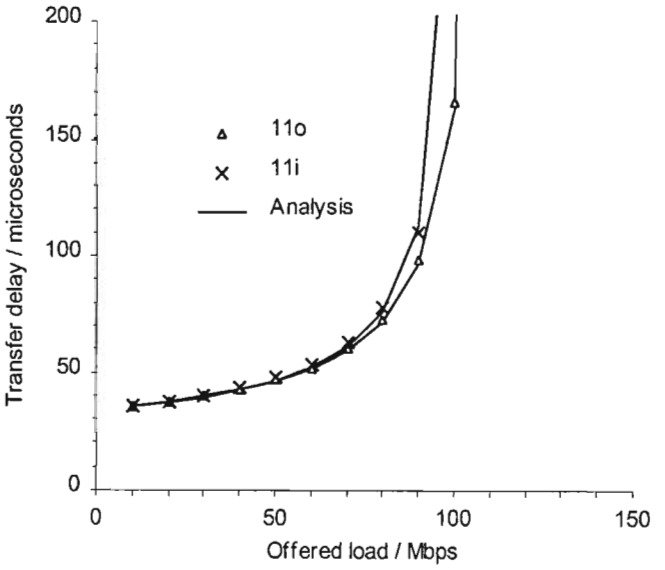
$$\lambda_{o(\max)} = \frac{m}{mx + R_m} = 0.001667 \text{ packets/bit time} \tag{3.30}$$

The other point of interest lies in the fact that intracluster packets experience lower transfer delays than intercluster packets at loads close to zero. This is because the residual token rotation time (which is the length of time a packet arriving at an empty queue has to wait before service) of the intracluster queue is lowered by the parallel operation of the two rings serving the queue. The increased rate of token arrivals leads to corresponding reductions in the token interarrival times and residual token interarrival times.

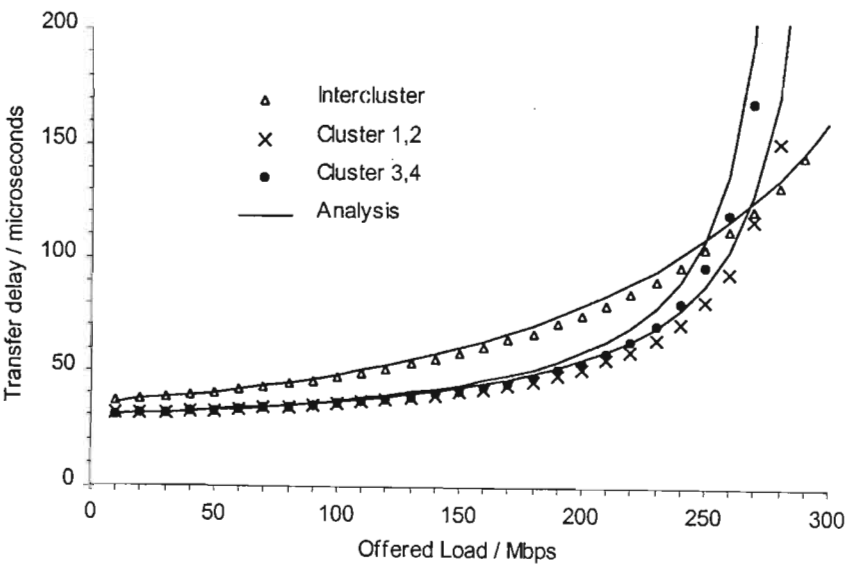


Graph 3-2: Comparison of *11o* and *11i* mean packet transfer delays for worst traffic class under random traffic

Graph 3-2 compares the performance of the traffic class with largest mean packet transfer delay in the *11i* network to the worst performing traffic class in the *11o* network. In both these cases, intercluster packets are obviously the culprits. The *11o* network performs slightly better in this regard because fewer intracluster packets are transmitted on the main ring leaving more available bandwidth for the intercluster packets. Since most of the network traffic consists of intercluster packets, we see a similar situation when looking at the overall mean packet transfer delays of all the traffic classes, as shown in graph 3-3.

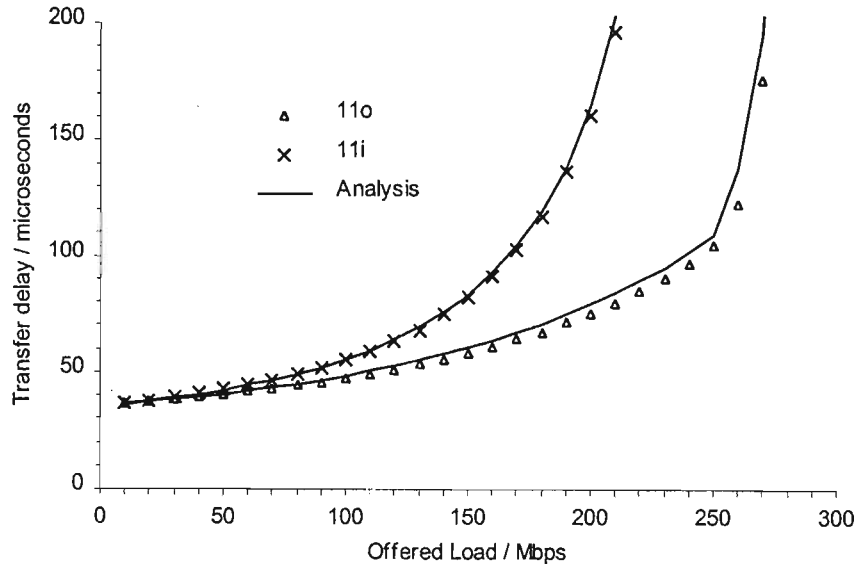


Graph 3-3: Comparison of *11o* and *11i* overall mean packet transfer delays under random traffic

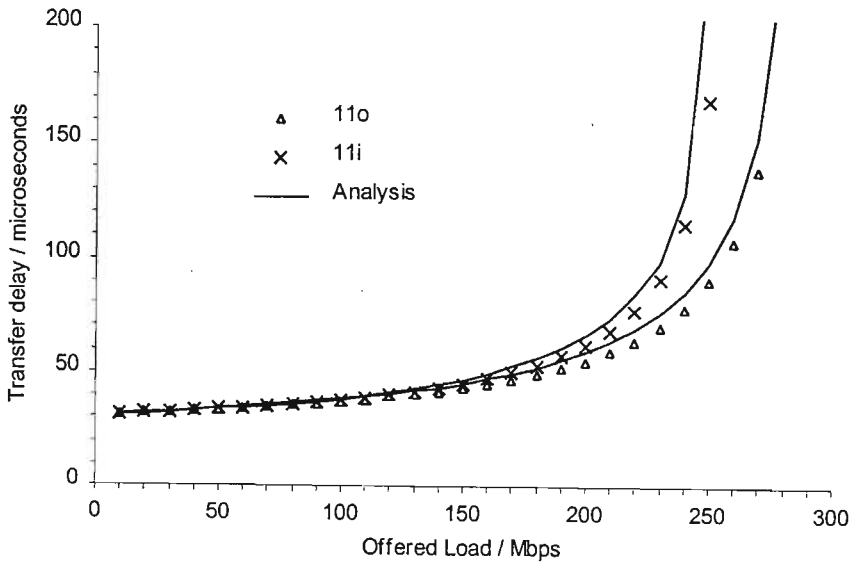


Graph 3-4: Mean packet transfer delays for all traffic classes under clustered traffic

We now turn our attention to clustered traffic. Examining graph 3-4, we see that the *11o* network performs significantly differently to the *11i* network in a clustered traffic environment. Although intercluster packets have higher transfer delays at low and medium loads than the other traffic classes, at high loads they perform substantially better. This is contrary to the case with the *11i* network.



Graph 3-5: Comparison of *11o* and *11i* mean packet transfer delays for worst traffic class under clustered traffic



Graph 3-6: Comparison of *11o* and *11i* overall mean packet transfer delays of all traffic classes under clustered traffic

Section 2.6 showed that, using the *11i* protocol, the main ring receives a higher proportion of the load and consequently it congests earlier than the other rings. Due to the change in access priority, the main ring of the *11o* network congests at far higher offered loads. In fact, the cluster rings congest before the main ring does. Graph 3-5 shows the performance comparison of the traffic classes with the largest mean packet transfer delays. The *11o* network is a significant improvement over the *11i* network in this regard, especially in terms of maximum throughput. The discontinuity in the curvature of the *11o* curve at an offered load of approximately 250Mbps occurs because, below this load, intercluster traffic has the largest delay whilst, above this load, the poorest performer is cluster three and four traffic.

Finally, graph 3-6 compares the overall mean packet transfer delays of the *11i* and *11o* networks under clustered traffic. Transfer delays are similar for low and medium loads, but the *11o* network gives a higher maximum throughput.

3.4 Gated Service, Intercluster Traffic Priority To Main Ring (*ggo*)

This protocol is a direct conversion of the one given in section 3.3 from a 1-limited to a gated service policy. As before, intercluster traffic is given priority on the main ring. The abbreviation *ggo* is short for gated/gated/intercluster, i.e. a gated service policy for the intracluster queue, a gated service policy for the intercluster queue and the intercluster queue has access priority to the main ring.

When the main ring token is received, the intercluster queue is served followed by the intracluster queue. The cluster and main ring can concurrently serve the intracluster queue, with independent gating mechanisms. In other words, when the intracluster queue receives a token, the cluster ring will serve any packets that were in the queue when the token arrived. If the main ring token arrives during this time period, the main ring will concurrently serve the queue until all the packets present on this token's arrival are served.

3.4.1 Analysis

We use the same general method for analysing this protocol as the last. The node model is identical to that given in figure 3-2 except that more than one packet can be served each time a token is seized. We once again assume that the proportion of intracluster packets that are transmitted on the cluster ring is a constant, time-invariant value a_i. However, there is extra complexity in calculating a_i, since we can no longer assume that it is simply the ratio of the rate

of cluster ring token arrivals to the total effective token arrival rate to the intracluster queue. This would only be the case if the number of packets served for each main ring token arrival were the same as the number served while each cluster ring token was held. On average, arriving main ring tokens will find less packets in the intracluster queue than cluster ring tokens will find.

The main ring token rotation time will generally be longer than the cluster ring token rotation times. We make a couple of assumptions at this point:

1. The cluster ring token rotation times are constant.
2. There are never two consecutive main ring token arrivals to a node, they are always separated by at least one cluster ring token arrival.

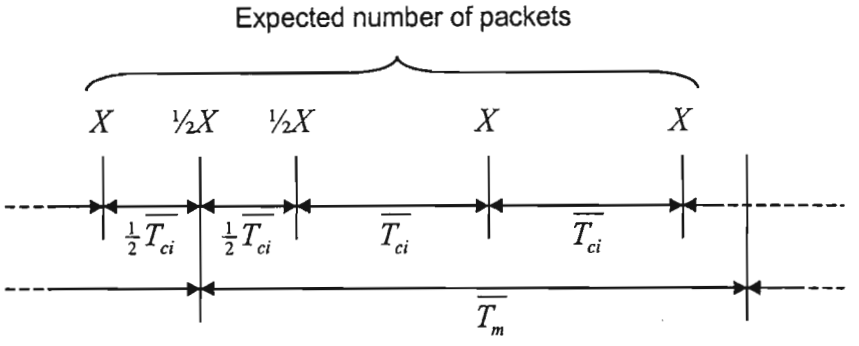


Figure 3-4: Expected queue length for main and cluster ring tokens

Figure 3-4 shows a timing diagram of token arrivals from both the main and cluster rings, showing the mean number of packets served at each arrival. The mean queue length a cluster ring token finds, given that the previous token was also from the cluster ring rather than from the main ring, is denoted by X. The mean queue length at a main ring token arrival will then be $\frac{1}{2}X$ since the arrival instant will have a uniform distribution between the previous cluster ring token arrival instant and the following one. Similarly, the mean queue length at the cluster ring arrival following a main ring token ring arrival will also be $\frac{1}{2}X$.

Clearly, an expected number of X packets arrive each cluster ring token rotation time. Hence

$$X = \frac{\lambda_i}{N_i}\overline{T_{ci}} \tag{3.31}$$

Consider one main ring token rotation time $\overline{T_m}$. The mean number of cluster ring tokens that arrive during this time is given by $\overline{T_m}/\overline{T_{ci}}$. At all of these cluster ring token arrivals, an expected X packets are served, except for the one immediately following the main ring token

arrival that started the time period under consideration - when an expected ½X packets are served. This gives an overall mean queue length at cluster ring token arrival instants of

$$\overline{Q_{Tc}} = \frac{\left(\overline{T_m}/\overline{T_{ci}}\right)X - \frac{1}{2}X}{\left(\overline{T_m}/\overline{T_{ci}}\right)} = \left(1 - \frac{\overline{T_{ci}}}{2\overline{T_m}}\right)X \tag{3.32}$$

In every time period $\overline{T_{ci}}$ an average of X packets arrive at the intracluster queue, $\overline{Q_{Tc}}$ of which are transmitted on the cluster ring. This leads directly to a_i, as follows.

$$a_i = \frac{\overline{Q_{Tc}}}{X} = \left(1 - \frac{\overline{T_{ci}}}{2\overline{T_m}}\right) \tag{3.33}$$

It was shown in equation (2.31) that the formula for the mean token rotation time of a token ring network with a gated service policy is identical to that for a 1-limited service policy. Hence, from equations (2.51) and (2.52), we have once again,

$$\overline{T_m} = \frac{R_m}{1 - \lambda_m x} \tag{3.34}$$

$$\overline{T_{ci}} = \frac{R_{ci}}{1 - a_i \lambda_i x} \tag{3.35}$$

The difference lies in the fact that these token rotation times are not limited to a maximum token rotation time at high loads, but rather tend to infinity as the offered load tends to infinity. As in equation (3.11), the effective mean token interarrival time to the intracluster queue of cluster i nodes is given by

$$\overline{T_{ei}} = \left(\frac{1}{\overline{T_{ci}}} + \frac{1}{\overline{T_m}}\right)^{-1} \tag{3.36}$$

According to equation (2.25), the mean queue waiting time for ith cluster intracluster packets is given by

$$\overline{W_i} = \overline{T_{\mathrm{Re}i}}\left(1 + \frac{\lambda_i}{N_i}x\right) \tag{3.37}$$

where, as before, we use the approximation $\overline{T_{\mathrm{Re}i}} = \frac{2}{3}\overline{T_{ei}}$. Similarly, the mean queue waiting time for intercluster packets originating from cluster i is given by

$$\overline{W_{oi}} = \overline{T_{Rm}}\left(1 + \frac{\lambda_{oi}}{N_i}x\right) \tag{3.38}$$

Hence, as in equation (2.63), the mean queue waiting time for intercluster packets originating from all clusters is given by

$$\overline{W_o} = \sum_{i=1}^{C} \frac{\lambda_{oi}}{\lambda_m} \overline{W_{oi}} \tag{3.39}$$

To calculate $\overline{T_{Rm}}$, we proceed as in section 3.1. According to the general residual life formula from renewal theory,

$$\overline{T_{Rm}} = \frac{\overline{T_m^{\,2}}}{2\overline{T_m}} \tag{3.40}$$

$\overline{T_m^{\,2}}$ can be found using equation (2.34), as follows

$$\overline{T_m^{\,2}} = \left(\overline{Y_o}^2 + \overline{Y_o}\right)x^2 + 2\overline{Y_o}xR_m + R_m^{\,2} \tag{3.41}$$

where $\overline{Y_o}$ is the mean number of packets that are transmitted onto the main ring during each token rotation.

$$\overline{Y_o} = \lambda_m \overline{T_m} \tag{3.42}$$

Finally, from (2.60) and (2.64), we can obtain the mean packet transfer delays

$$\overline{D_o} = \overline{W_o} + x + \tfrac{1}{2}R_m \tag{3.43}$$

$$\overline{D_i} = \overline{W_i} + x + \tfrac{1}{2}a_i R_{ci} + \tfrac{1}{2}(1-a_i)R_{mi} \tag{3.44}$$

As in (2.65), the overall packet transfer delay of the entire network is then given by

$$\overline{D_{ave}} = \sum_{i=1}^{C}\left(\frac{\lambda_i}{\lambda}\overline{D_i} + \frac{\lambda_{oi}}{\lambda}\overline{D_o}\right) \tag{3.45}$$

We need to solve the set of simultaneous equations (3.33) to (3.35) and (3.8) using the extended Newton-Raphson method. We can reduce the set of *2C + 2* equations to the following *C + 1* equations:

$$f^i = 2\overline{T_m}(a_i - 1)(1 - a_i\lambda_i x) + R_{ci} \quad for \quad 1 \le i \le C \tag{3.46}$$

$$f^{C+1} = \overline{T_m}(1 - \lambda_m x) - R_m \tag{3.47}$$

together with the substitutions,

$$z_i = a_i \quad \text{for} \quad 1 \le i \le C \tag{3.48}$$

$$z_{C+1} = \overline{T_m} \tag{3.49}$$

For $1 \le i \le C$:

$$f^i_{a_j} = \begin{cases} 2\overline{T_m}(-2\lambda_i x a_i + 1 + \lambda_i x) & for \quad i = j \\ 0 & for \quad i \ne j \end{cases} \tag{3.50}$$

$$f^i_{\overline{T_m}} = 2(a_i - 1)(1 - a_i\lambda_i x) \tag{3.51}$$

and finally

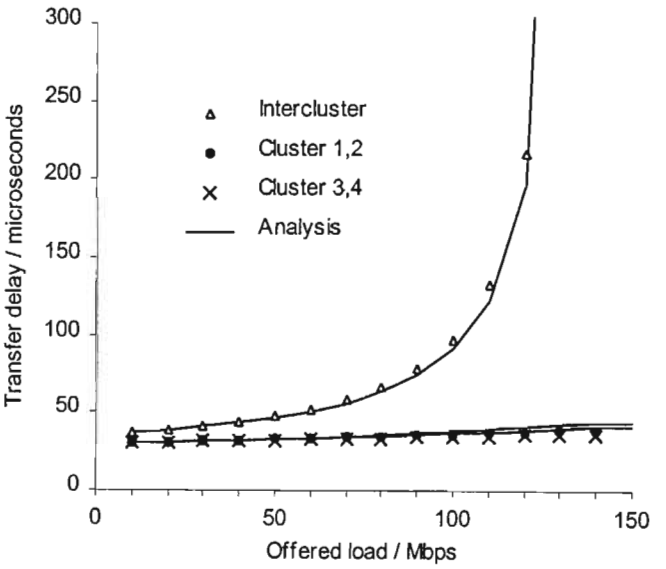
$$f^{C+1}_{a_j} = \lambda_j x \overline{T_m} \tag{3.52}$$

$$f^{C+1}_{\overline{T_m}} = 1 - \lambda_m x \tag{3.53}$$

This solution is valid when all the ring utilisations are less then one. If any of the utilisations go above one, negative token rotation times are generated. For valid intracluster packet results at high loads it is necessary to set $a_i = 1$ for $1 \le i \le C$ when $\lambda_o x \ge 1$.

3.4.2 Results

The same network parameters as for the *11o* results are used for the remainder of the thesis. Furthermore, the same series of graphs will be given in each section, where appropriate. Graph 3-7 shows the mean packet transfer delays for each of the traffic classes of the 2-MR network using the *ggo* protocol. It looks almost identical to graph 3-1 except for the slightly greater maximum throughput. This is shown better in graph 3-8 and graph 3-9, which compare the performance of the *ggo* protocol to the *11o* protocol.

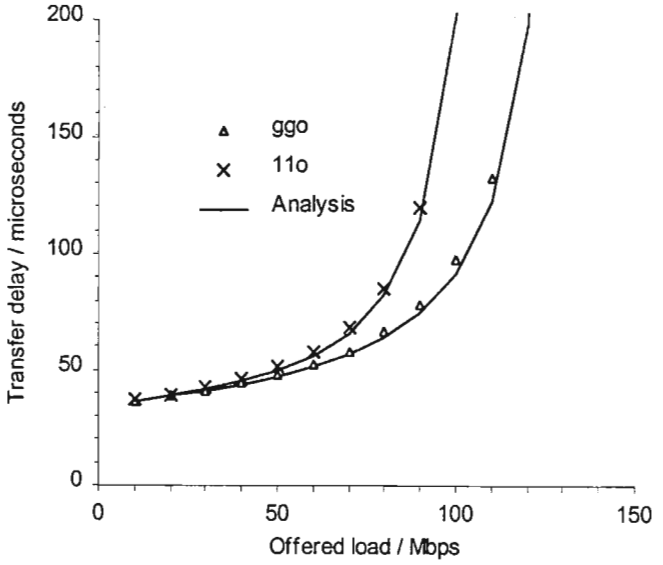


Graph 3-7: Mean packet transfer delays for all traffic classes under random traffic for *ggo* protocol

The increased maximum useable throughput is due to the elimination of an upper limit on the number of packets that can be transmitted during each token rotation. A 1-limited service discipline restricts the maximum possible throughput of a token ring network to a value less than the total available bandwidth. This is because m nodes can only be transmitted during each token rotation, during which there is a token passing overhead equal to the ring delay R. The following formula describes this:

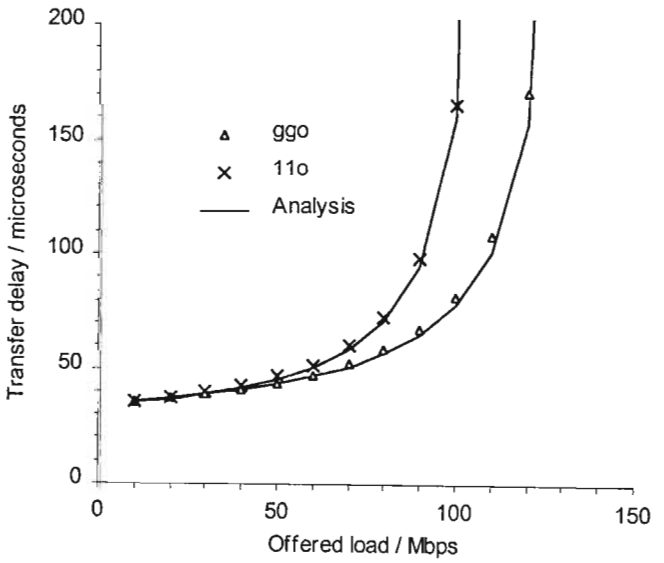
$$Max\ throughput = \frac{mx}{mx + R} \times bandwidth$$

With a gated service policy, on the other hand, the maximum throughput is equal to the total available bandwidth. Hence, when *mx* is relatively small compared to *R*, a gated service policy allows a far greater maximum throughput than a 1-limited service policy. As the ratio of packet length to ring delay increases, however, the performance gains are somewhat nullified.

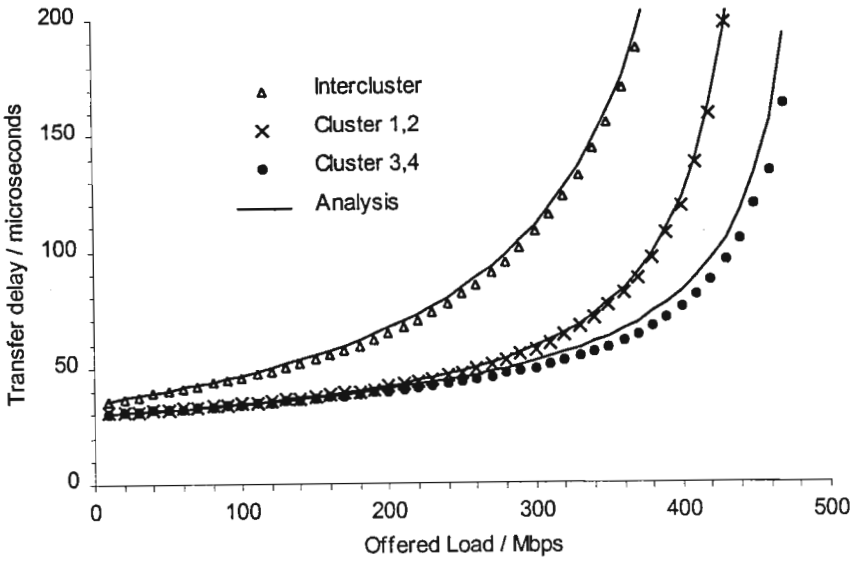


Graph 3-8: Comparison of *ggo* and *11o* mean packet transfer delays for worst traffic class under random traffic

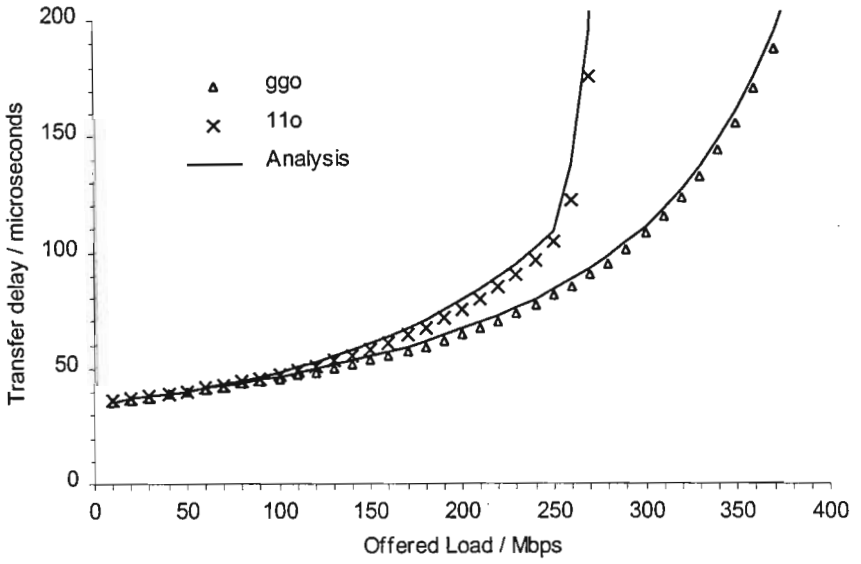
The difference between graph 3-8 and graph 3-9 is once again minor, because most of the network traffic is travelling on the main ring.



Graph 3-9: Comparison of *ggo* and *11o* overall mean packet transfer delays of all traffic classes under random traffic



Graph 3-10: Mean packet transfer delays for all traffic classes under clustered traffic for *ggo* protocol

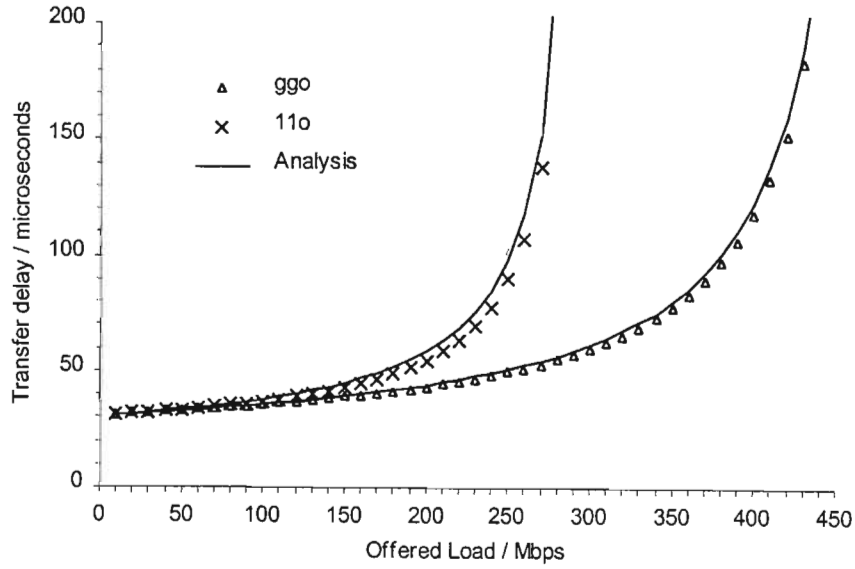


Graph 3-11: Comparison of *ggo* and *11o* mean packet transfer delays for worst traffic class under clustered traffic

We turn once again to clustered traffic. Graph 3-10 shows the mean packet transfer delays of the various traffic classes using the *ggo* protocol. Note the new scale on the x-axis. The important aspect of this graph, besides the obvious performance improvement that will be more apparent in the following two graphs, is the variation in the performance characteristics of the various traffic classes. Intercluster packets have significantly higher mean transfer delays than the other classes, even though the curves tend to infinity at similar loads. The minor

discrepancy between the simulation and analytical results validates the assumptions made in the analysis.

The similarity between the *ggo* and *11o* protocols is apparent in graph 3-11. The assignment of main ring access priority to the intercluster queue in both cases, leads to similar curves for intercluster traffic. The similarity also indicates that the *ggo* intercluster queues do not often have many packets present. This is a actually fairly obvious phenomenon when one takes into account the low intercluster load and the large number of nodes over which arriving intercluster traffic is distributed. The *11o* protocol suffers when the cluster rings become congested at about 250Mbps.



Graph 3-12: Comparison of *ggo* and *11o* overall mean packet transfer delays under clustered traffic

Graph 3-12 clearly indicates the vast performance improvement gained by implementing a gated service policy rather than a 1-limited service policy. The *11o* protocol provides a maximum useable throughput of less than 300Mbps while the *ggo* protocol has a maximum useable throughput of over 450Mbps.

3.5 Gated Service, Ideally Balanced Load *(Ideal)*

Although the protocol examined in section 3.4 provides vast performance improvements over the 1-limited protocols, it has a problem in that it does not balance the load over all the available rings in the network as evenly as possible. This leads to sub-optimal performance especially for intercluster packets. This section examines a theoretical protocol in which the

load is ideally balanced between the rings according to the arrival rates. It is not in itself a practical protocol since in a real environment there is no way of knowing exactly what average load each traffic class is contributing. In section 3.6, we will present a practical protocol that implements a simple algorithm to balance the load amongst the rings and hence performs very similarly to the *ideal* protocol presented in this section.

The protocol adopted here is extremely simple. Packets arriving at the intracluster queue are tagged for transmission on each of the two possible rings according to a fixed Markov probability function which is chosen in such a way as to balance the load on the rings as evenly as possible. This is equivalent to the node model shown in figure 3-5. The protocol effectively detaches the two queues from each other and the rings operate independently. This makes the analysis essentially equivalent to two separate gated single token ring analyses.

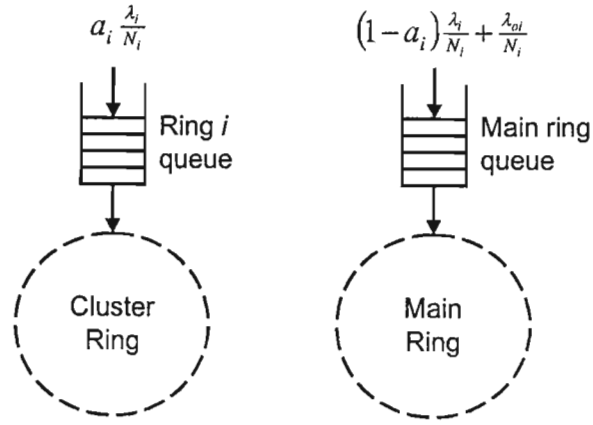


Figure 3-5: Node model for ideally balanced load

3.5.1 Analysis

As shown in figure 3-5, the variable a_i will again be used to represent the proportion of cluster i intracluster traffic that will be transmitted on ring i. In this case, however, a_i is not an approximation but an exact predetermined value. It is chosen to balance the load over all the rings as evenly as possible. We have a total packet arrival rate of $a_i\lambda_i$ to cluster ring i and the total packet arrival rate to the main ring is given by

$$\lambda_m = \sum_{i=1}^{C}(1-a_i)\lambda_i + \lambda_o \tag{3.54}$$

where λ_i and λ_o are defined as before. Furthermore, the arrival rate of intercluster packets to the main ring from cluster i is given by λ_{oi}. The token rotation times for the cluster and main rings are given by the usual formulae:

$$\overline{T_{ci}} = \frac{R_{ci}}{1 - a_i \lambda_i x} \tag{3.55}$$

$$\overline{T_m} = \frac{R_m}{1 - \lambda_m x} \tag{3.56}$$

From equation (2.25), we can immediately obtain the mean queue waiting times as follows

$$\overline{W_{ii}} = \overline{T_{Rci}}\left(1 + \frac{a_i \lambda_i x}{N_i}\right) \tag{3.57}$$

$$\overline{W_{oi}} = \overline{T_{km}}\left(1 + \frac{\lambda_{oi} x}{N_i}\right) \tag{3.58}$$

$$\overline{W_o} = \sum_{i=1}^{C} \frac{\lambda_{oi}}{\lambda_o} \overline{W_{oi}} \tag{3.59}$$

where the residual token rotation times can be derived as in section 2.2.5.3 and 3.4. These then lead to the mean packet transfer delays as required.

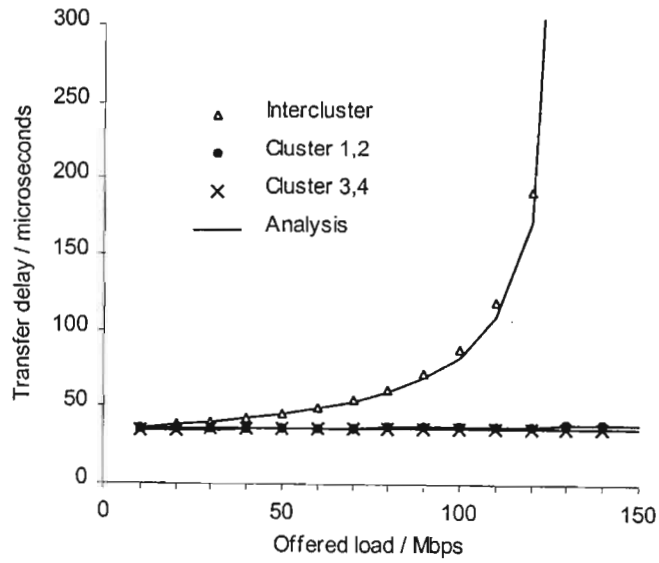
$$\overline{D_i} = a_i\left(\overline{W_{ii}} + x + \tfrac{1}{2}R_{ci}\right) + (1 - a_i)\left(\overline{W_{oi}} + x + \tfrac{1}{2}R_m\right) \tag{3.60}$$

$$\overline{D_o} = \overline{W_o} + x + \tfrac{1}{2}R_m \tag{3.61}$$

The overall packet transfer delay of the entire network is then given by

$$\overline{D_{ave}} = \sum_{i=1}^{C}\left(\frac{\lambda_i}{\lambda}\overline{D_i} + \frac{\lambda_{oi}}{\lambda}\overline{D_o}\right) \tag{3.62}$$

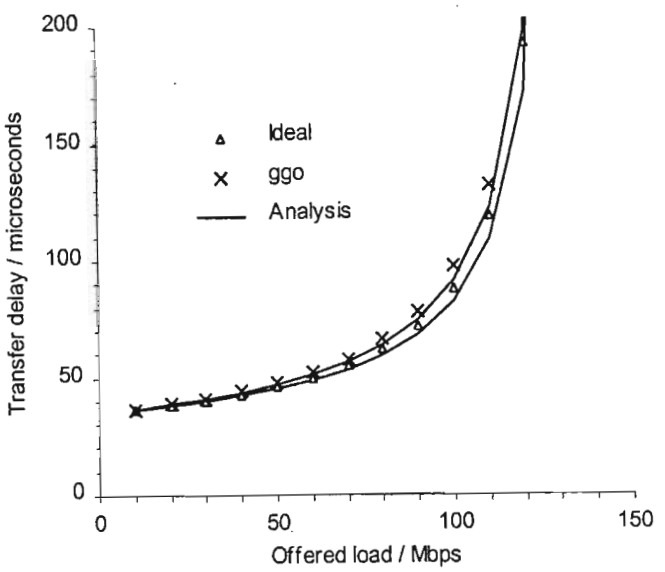
The results can be explicitly obtained using the above formulae. No numerical methods are needed to solve these equations.



Graph 3-13: Mean packet transfer delays for all traffic classes under random traffic for *Ideal* protocol

3.5.2 Results

The network we consider has five rings: four cluster rings and the main ring. The ideally balanced situation would be to have 20% of the total offered load carried on each ring. For random traffic, however, 75% of the total offered load has to be transmitted on the main ring. As a result, the most balanced loading we can obtain is to have no intracluster packets at all transmitted on the main ring. Therefore, we need to set $a_1 = a_2 = a_3 = a_4 = 1$.



Graph 3-14: Comparison of *Ideal* and *ggo* mean packet transfer delays for worst traffic class under random traffic

Since intercluster traffic comprises 75% of the load, it is impossible to achieve any semblance of even balance amongst the rings. Consequently, graph 3-13 is almost identical to graph 3-7. This is further emphasised by the close agreement of the curves in graph 3-14 and graph 3-15.

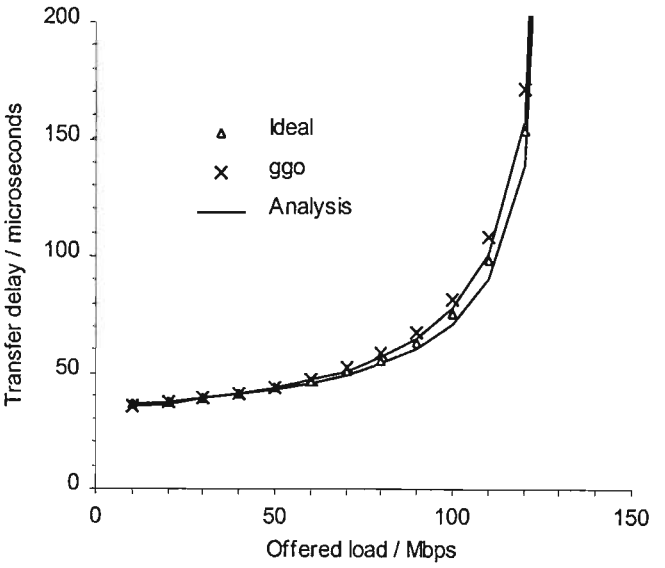
Recall that the packet arrival rates used for the clustered traffic results are $\lambda_1 = \lambda_2 = 0.19\lambda$, $\lambda_3 = \lambda_4 = 0.26\lambda$ and $\lambda_o = 0.1\lambda$. Cluster rings one and two cannot possibly receive 20% of the total offered load because the intracluster packet arrival rates of clusters one and two are too low. Hence, we need to set: $a_1 = a_2 = 1.00$, in order to load these two rings as much possible.

The intracluster packet arrival rates of clusters three and four, on the other hand, are greater than intercluster packet arrival rate. Hence, some of this traffic can be transmitted on the main ring so that cluster rings three and four as well as the main ring all carry exactly the same amount of traffic. The sum of the packet arrival rates of class zero, three and four traffic is

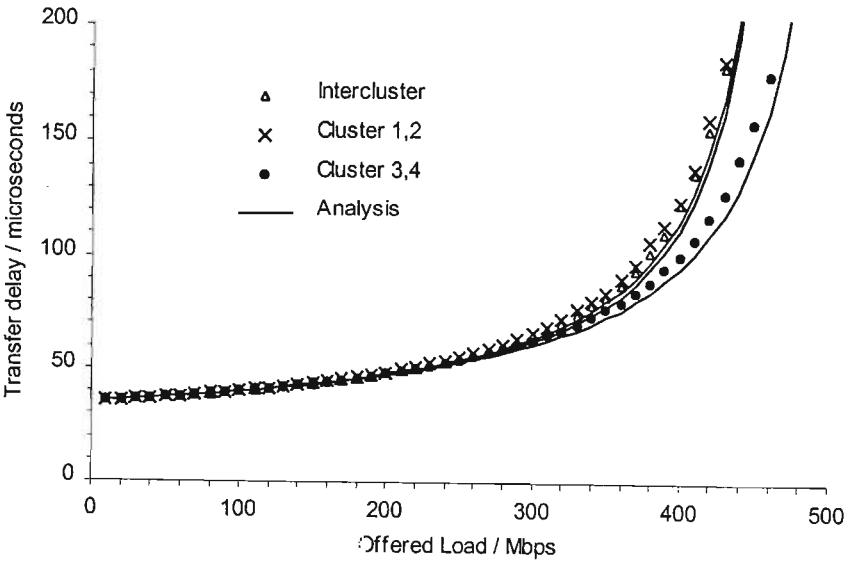
0.62λ. Dividing this by three, we see that we need to transmit 0.2067λ on each ring, i.e. 20,67% of the total offered load. Consequently, we need to set

$$a_3 = a_4 = \frac{0.2067\lambda}{0.26\lambda} = 0.79$$

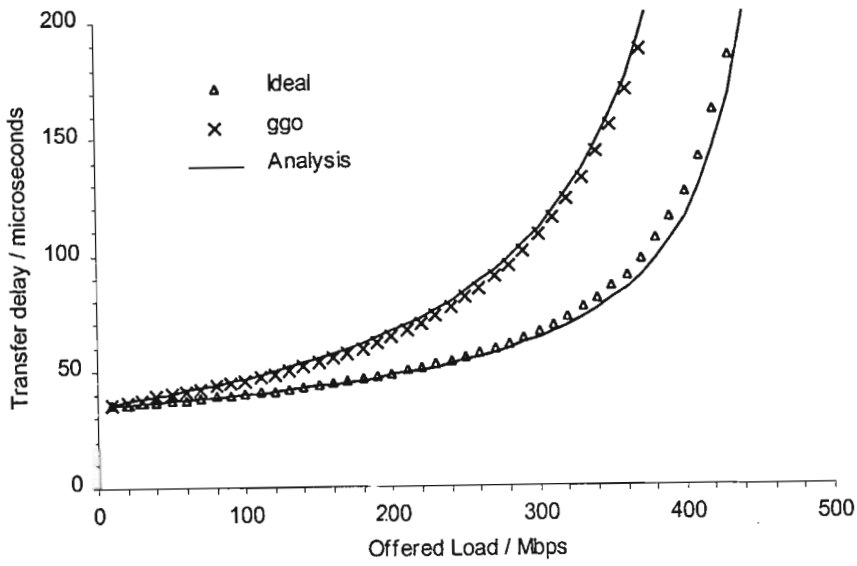
The effect of balancing the load only now becomes evident. Graph 3-16 is totally different to graph 3-10. The curves are now closely bunched.



Graph 3-15: Comparison of *Ideal* and *ggo* overall mean packet transfer delays under random traffic

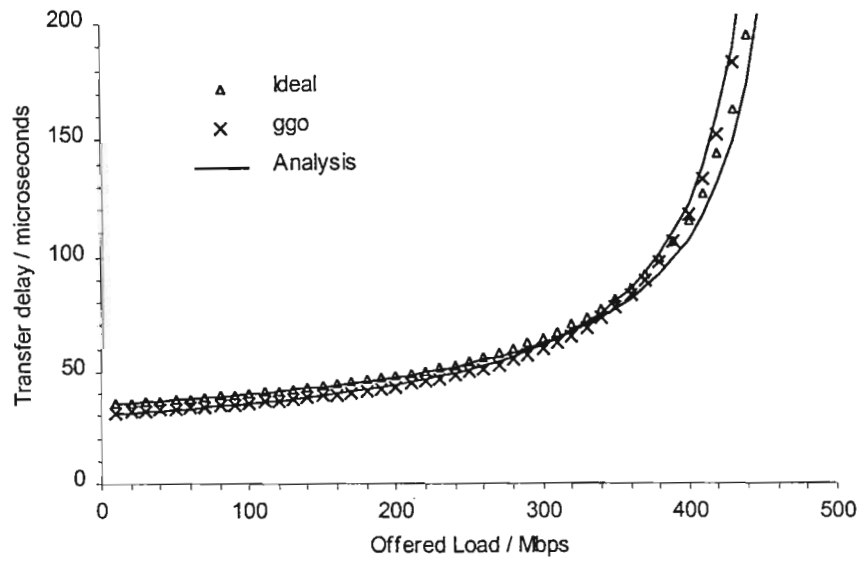


Graph 3-16: Mean packet transfer delays for all traffic classes under clustered traffic for *Ideal* protocol



Graph 3-17: Comparison of *Ideal* and *ggo* mean packet transfer delays for worst traffic classes under clustered traffic

The advantage of load balancing is evident in graph 3-17. The *ideal* protocol results in significantly lower mean intercluster packet transfer delays. It could thus be considered to be a *fairer* protocol.



Graph 3-18: Comparison of *Ideal* and *ggo* overall mean packet transfer delays under clustered traffic

Graph 3-18 shows that there is not a great deal of difference between the overall performance of the *ideal* and *ggo* protocols. The *ideal* protocol actually performs slightly worse at low loads because the separation of the intracluster traffic on arrival results in only one ring serving each

queue. This results in lower token arrival rates and hence higher residual token interarrival times. On the other hand, the *ideal* protocol provides a slightly higher maximum throughput.

3.6 Gated Service, Load Balancing Algorithm *(Alg)*

It has already been mentioned that the *ideal* protocol is not practical since there is no way of knowing what the average offered load to the network is. An algorithm could be developed that measures this load, but this is complicated and computationally expensive. Not only would it be difficult to take accurate measurements, but also additional problems such as stability have to be considered. There could well emerge a scenario where the greater portion of the load oscillates between the main and cluster rings. Of course, the protocol could be designed to compensate for such effects, but it introduces needless complications.

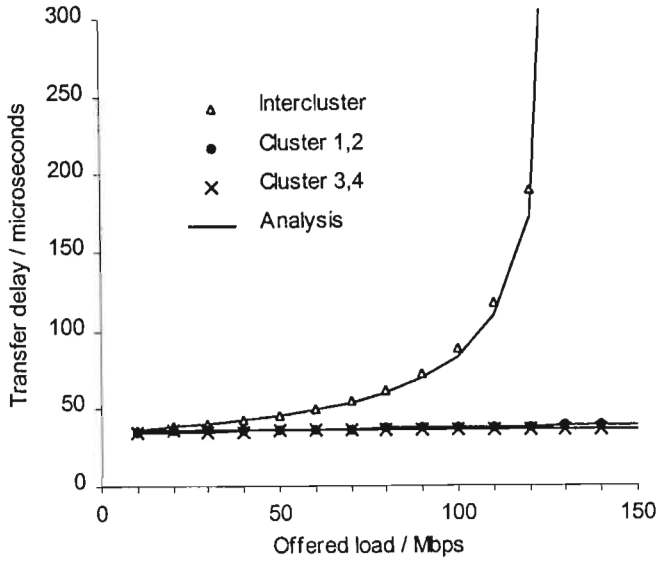
The protocol suggested here implements an extremely simple algorithm to practically realise load balancing. It assumes that all the rings transmit in the same direction and only requires keeping track of which of the main ring and cluster ring tokens was the last to be received. As with the *ggo* protocol, upon arrival of a main ring token, the intercluster queue is first served according to the gated service policy. The main ring only proceeds to serve intracluster packets if there has been no cluster ring token received since the last main ring token, i.e. if this is the second main ring token in a row. In this case, all packets present in the intracluster queue at this time are served. The intracluster queue is only gated after the intercluster queue has been served.

The algorithm attempts to keep the token rotation times of the various rings equal to each other. As we have seen in the results, the main ring generally receives more traffic than the cluster rings, so in order to achieve balance, main ring traffic has to be reduced when necessary. The only way this can be done is by reducing the amount of intracluster traffic travelling on the main ring.

If main and cluster ring tokens are travelling at the same rate, each node will receive them in an alternating fashion. If a cluster ring token is being received less often than main ring tokens by a node (i.e. two main ring tokens in a row), there is too little intracluster traffic from that cluster being transmitted on the main ring. Consequently, an arriving main ring token must serve the intracluster queue. If, on the other hand, the tokens are being received at an equal rate (i.e. alternating), the main ring does not need to serve intracluster packets. In this way, a cluster with low intracluster load will rarely have packets transmitted on the main ring, while a cluster with excess traffic will get the correct proportion transmitted on the main ring.

3.6.1 Results

Since this algorithm is intended to realise the *ideal* protocol as closely as possible, the analytical results from the *ideal* protocol are used.

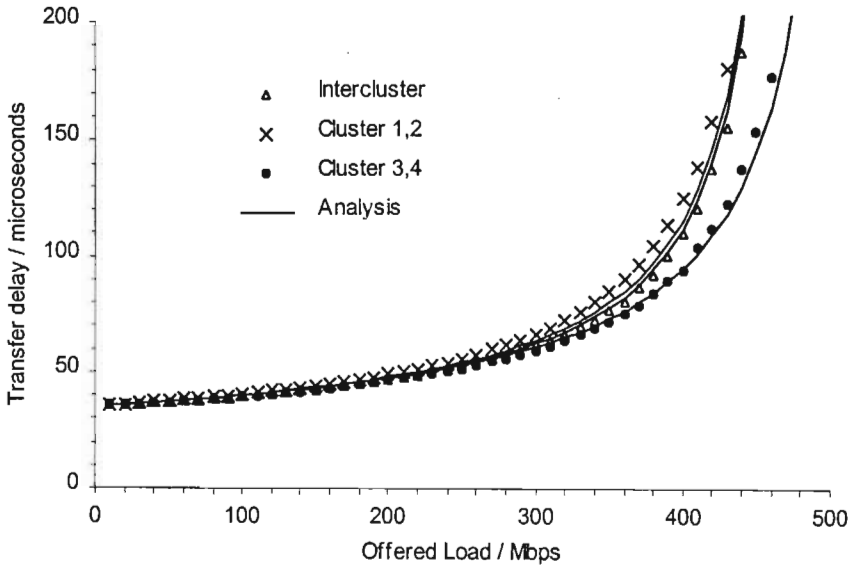


Graph 3-19: Mean packet transfer delays for all traffic classes under random traffic for *Alg* protocol

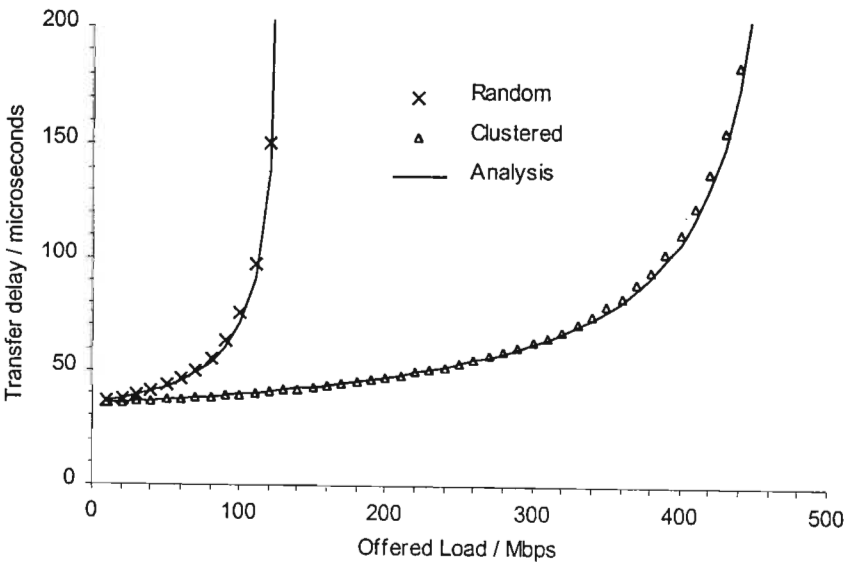
Graph 3-19 shows the performance of the protocol under random traffic. The analytical results are from the *Ideal* protocol and it is evident that they accurately describe the *Alg* protocol. Conversely, the algorithm approximates the ideally balanced load situation well under random traffic. In the random traffic case, the best balance that can be achieved is to have no intracluster packets transmitted on the main ring at all. Recall that to balance the load using the *ideal* protocol we set $a_i = 1$ for all i. The fact that the *Alg* results correlate well with the *Ideal* results, indicates that there is low intracluster packet "leakage" onto the main ring. Such "leakage" will obviously only occur in instances where a cluster ring token rotation time is longer than the main ring token rotation time. This in turn is only likely to occur to a significant degree when the mean cluster ring token rotation time is fairly close to the mean main ring token rotation time. In a random traffic environment, the main ring to cluster ring token rotation time ratio rapidly increases with increasing load. Hence, the low "leakage" is to be expected.

Graph 3-20 shows that even in the clustered traffic case, intracluster traffic "leakage" onto the main ring is negligible. In fact, if the results are closely compared to graph 3-16, it is apparent that the *Alg* protocol leads to very slightly lower mean packet transfer delay than the *Ideal* protocol. Once again, the solid lines represent the *Ideal* analysis. The only real discrepancy

between the performance of the *Alg* protocol and the *Ideal* protocol is the larger difference between the intercluster and cluster 1,2 delays. The simulation results for the mean intercluster packet transfer delay are noticeably lower than those in graph 3-16. Graph 3-21 is included to show that the overall mean packet transfer delays under both random and clustered traffic also agree well with the Ideal analysis.



Graph 3-20: Mean packet transfer delays for all traffic classes under clustered traffic for *Alg* protocol



Graph 3-21: Comparison of *Alg* simulation and *Ideal* analysis of overall mean packet transfer delays for all traffic classes under both traffic patterns

3.7 Slotted ring protocol

It was shown in [Bhuyan *et al*, 1989] that a slotted ring protocol outperforms a token ring protocol when the network parameters are the same and packet lengths are fixed. This section investigates whether the same gains can be obtained by using a slotted ring protocol in the 2-MR network. The protocol is based on the *11o* protocol. Intercluster packets once again have priority access to the main ring.

3.7.1 Analysis

The following 2-MR slotted ring analysis is based on that given in [Bhuyan *et al*, 1989], as was illustrated in section 2.3. The packet arrival rate to the main ring is, once again, given by equation (2.50).

$$\lambda_m = \sum_{i=1}^{C} (1 - a_i)\lambda_i + \lambda_o \tag{3.63}$$

From equation (2.36), the nominal ring utilisation of the main ring is then given by

$$\overline{U_m} = \lambda_m x \tag{3.64}$$

This is also the probability that a passing main ring slot will be full. The probability that the node has to wait j more slots before it finds an empty one can thus be approximated by the geometric distribution

$$\overline{U_m}^{\,j}\left(1 - \overline{U_m}\right)$$

Therefore, the mean waiting time $\overline{d_m}$ until an empty slot arrives on the main ring is given by

$$\phi_m < 1: \qquad \overline{d_m} = \sum_{j=1}^{\infty} jx\overline{U_m}^{\,j}\left(1 - \overline{U_m}\right) = \frac{x\overline{U_m}}{1 - \overline{U_m}} \tag{3.65}$$

if the main ring is not congested. Similarly, the nominal ring utilisation of the ith cluster ring is given by

$$\overline{U_{ci}} = a_i \lambda_i x \tag{3.66}$$

and hence, if the cluster ring is not congested, the mean waiting time for a cluster ring free slot is given by

$$\phi_i < 1: \qquad \overline{d_{ci}} = \frac{x\overline{U_{ci}}}{1 - U_{ci}} \tag{3.67}$$

where, according to (2.42), the actual ring utilisation of the main and cluster rings is given by

$$\phi_m = \lambda_m x\left(\frac{m+1}{m}\right) \tag{3.68}$$

$$\phi_i = a_i \lambda_i x \left(\frac{N_i + 1}{N_i} \right) \tag{3.69}$$

Once the packet at the head of the queue has been served, the time before the next packet is served is made up of the time for the current slot to pass by plus the time until another slot arrives. We use the random variables t_m and t_{ci} to denote the service times of the main and cluster rings respectively. As in equation (2.39), their means are given by,

$$\phi_m < 1: \qquad \overline{t_m} = \overline{d_m} + x = \frac{x}{1 - \lambda_m x} \tag{3.70}$$

$$\phi_i < 1: \qquad \overline{t_{ci}} = \overline{d_{ci}} + x = \frac{x}{1 - a_i \lambda_i x} \tag{3.71}$$

We have already seen in equation (2.41) that these are limited to maximum values, as follows

$$\phi_m \geq 1: \qquad \overline{t_m} = (m+1)x \tag{3.72}$$

$$\phi_i \geq 1: \qquad \overline{t_{ci}} = (N_i + 1)x \tag{3.73}$$

This, of course, implies:

$$\phi_m \geq 1: \qquad \overline{d_m} = mx \tag{3.74}$$

$$\phi_i \geq 1: \qquad \overline{d_{ci}} = N_i x \tag{3.75}$$

Both ϕ_m and ϕ_i are dependent upon a_i and hence cannot be calculated independently of the other equations. The intracluster queue only receives main ring access when the intercluster queue is empty. The utilisation of the intercluster queue is $\frac{\lambda_{oi}}{N_i}\overline{t_m}$ and hence the probability that it is empty is given by

$$1 - \frac{\lambda_{oi}\overline{t_m}}{N_i}$$

As a result, the mean waiting time for a main ring free slot to arrive at the intracluster queue is given by

$$\overline{d_{mi}} = \frac{\overline{d_m}}{1 - \frac{\lambda_{oi}}{N_i}\overline{t_m}} \tag{3.76}$$

Similarly, the mean time it takes the main ring to serve the intracluster queue is

$$\overline{t_{mi}} = \frac{\overline{t_m}}{1 - \frac{\lambda_{oi}}{N_i}\overline{t_m}} \tag{3.77}$$

The formula for a_i is consequently given by

$$a_i = \frac{\overline{t_{mi}}}{\overline{t_{mi}} + \overline{t_{ci}}} = \frac{\overline{t_m}}{\overline{t_m} + \overline{t_{ci}}\left(1 - \frac{\lambda_{oi}}{N_i}\overline{t_m}\right)} \tag{3.78}$$

Both the cluster ring and the main ring service the intracluster queue. Combining the statistics of the two servers in a similar fashion to equation (2.57), we arrive at the following effective mean slot waiting time for intracluster packets.

$$\overline{d_{ei}} = \left(\frac{1}{\overline{d_{ci}}} + \frac{1}{\overline{d_{mi}}} \right)^{-1} \tag{3.79}$$

Also, the mean service time of the intracluster queue is

$$\overline{t_{ei}} = \left(\frac{1}{\overline{t_{ci}}} + \frac{1}{\overline{t_{mi}}} \right)^{-1} \tag{3.80}$$

Once this has been found, we are able to obtain the mean queue waiting times for both intracluster and intercluster packets, similarly to equations (3.12) to (3.14), as follows

$$\overline{W_i} = \frac{\frac{1}{2}x + \overline{d_{ei}}}{1 - \frac{\lambda_i}{N_i}\overline{t_{ei}}} \tag{3.81}$$

$$\overline{W_{oi}} = \frac{\frac{1}{2}x + \overline{d_m}}{1 - \frac{\lambda_{oi}}{N_i}\overline{t_m}} \tag{3.82}$$

$$\overline{W_o} = \sum_{i=1}^{C} \frac{\lambda_{oi}}{\lambda_o} \overline{W_{oi}} \tag{3.83}$$

As in equations (2.60) and (2.64), we can then get the mean packet transfer delays

$$\overline{D_o} = \overline{W_o} + x + \tfrac{1}{2}R_m \tag{3.84}$$

$$\overline{D_i} = \overline{W_i} + x + \tfrac{1}{2}a_i R_{ci} + \tfrac{1}{2}(1 - a_i)R_m \tag{3.85}$$

From equation (2.65), the overall mean packet transfer delay of the entire network is then given by

$$\overline{D_{ave}} = \sum_{i=1}^{C} \left(\frac{\lambda_i}{\lambda} \overline{D_i} + \frac{\lambda_{oi}}{\lambda} \overline{D_o} \right) \tag{3.86}$$

We once again need to use the extended Newton-Raphson method to solve our set of simultaneous equations. We can reduce the set of equations when $\phi_m < 1$ and $\phi_i < 1$ to the following $C + 1$ equations:

$$f^i = (a_i - 1)(1 - a_i\lambda_i x)\overline{t_m} + a_i\left(1 - \frac{\lambda_{oi}}{N_i}\overline{t_m}\right)x \quad for \quad 1 \le i \le C \tag{3.87}$$

$$f^{C+1} = (1 - \lambda_m x)\overline{t_m} - x \tag{3.88}$$

together with the substitutions,

$$z_i = a_i \quad \text{for} \quad 1 \le i \le C \tag{3.89}$$

$$z_{C+1} = \overline{t_m} \tag{3.90}$$

We then calculate the partial derivatives. For $1 \le i \le C$:

$$f^{i}_{a_j} = \begin{cases} (-2a_i\lambda_i x + 1 + \lambda_i x)\overline{t_m} + \left(1 - \frac{\lambda_{oi}}{N_i}\overline{t_m}\right)x & for \quad i = j \\ 0 & for \quad i \neq j \end{cases} \tag{3.91}$$

$$f^{i}_{\overline{t_m}} = (a_i - 1)(1 - a_i\lambda_i x) - a_i \frac{\lambda_{oi}}{N_i} x \tag{3.92}$$

and finally

$$f^{C+1}_{a_j} = \lambda_j x \overline{t_m} \tag{3.93}$$

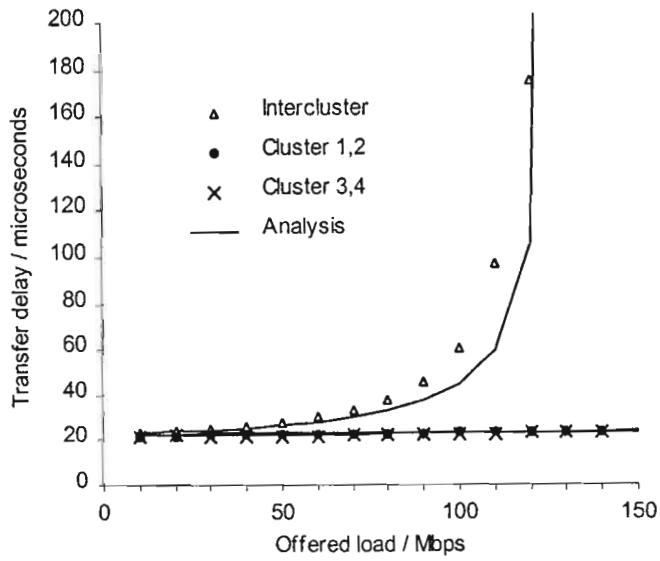
$$f^{C+1}_{\overline{t_m}} = 1 - \lambda_m x \tag{3.94}$$

3.7.2 Results

To keep consistency with the 2-MR token ring results, the identical network parameters are assumed, with the following additions:

- Slot length, x = 500bits
- Number of slots per ring = 6

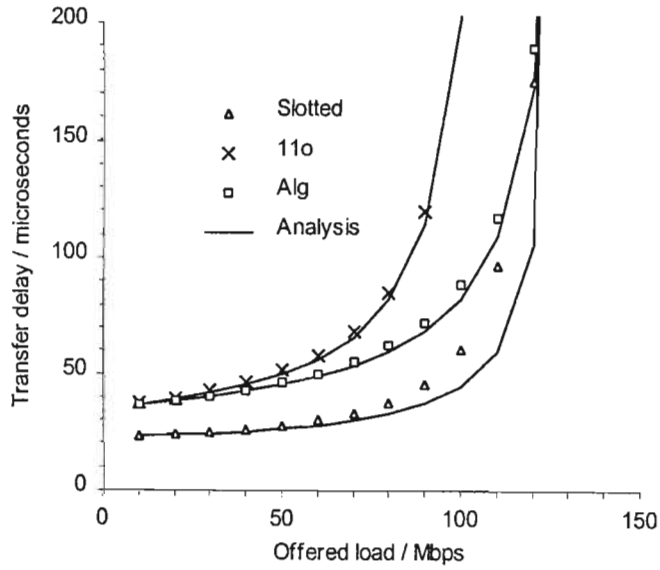
In practice, a slotted ring protocol would probably have a slightly higher overhead than a corresponding token ring protocol, but we do not take this factor into account.



Graph 3-22: Mean packet transfer delays for all traffic classes under random traffic for 2-MR slotted ring

Graph 3-22 shows a discrepancy between the analytical and simulation results around the "corner" of the graph. This phenomena was discussed in section 2.3 and was thus to be

expected. It is evident in all the following slotted ring results. The most striking feature of graph 3-22 is the significantly lower mean packet transfer delays. When the load is near zero, the mean packet transfer delays approximately equal 22.5μs. This compares to a value of between 30μs and 35μs for the token ring case. The improvement arises because nodes in the slotted ring network only have to wait half a slot length on average to receive access, whereas nodes in the token ring network have to wait half a token rotation time. The queuing delays are thus reduced by a factor equal to the number of slots (assuming the ring walk times are equal).



Graph 3-23: Comparison of mean packet transfer delays for worst traffic classes of slotted ring, *11o* and *Alg* protocols under random traffic

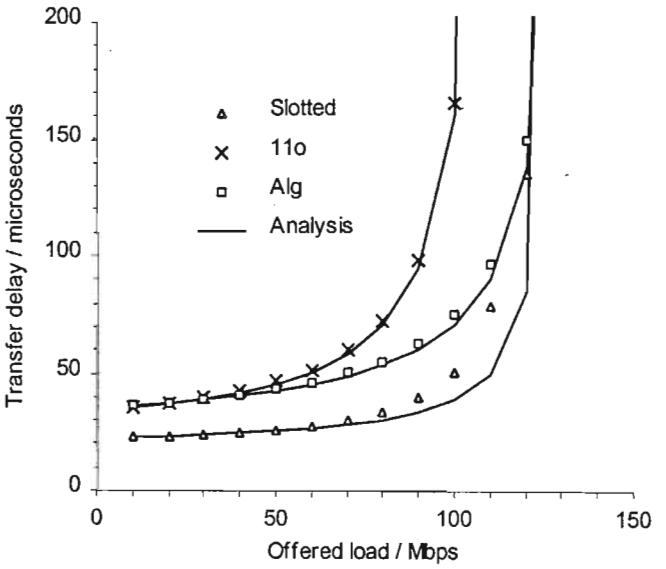
The improvement in the mean packet transfer delays is easily seen in graph 3-23 and graph 3-24. It is important to note that the delay times remain low right up to high loads. Furthermore, the maximum throughput is of a similar magnitude to the *Alg* protocol, which has been the best performer up to now. The slotted ring protocol is, in fact, a l-limited protocol since only one packet can be serviced for each arrival of an empty slot. Although the priority assignment is the same as the *11o* protocol, there is a vast performance improvement. Recall from equation (3.30) that for the l-limited 2-MR token ring protocol,

$$\lambda_{o(\max)} = \frac{m}{mx + R_m} \tag{3.95}$$

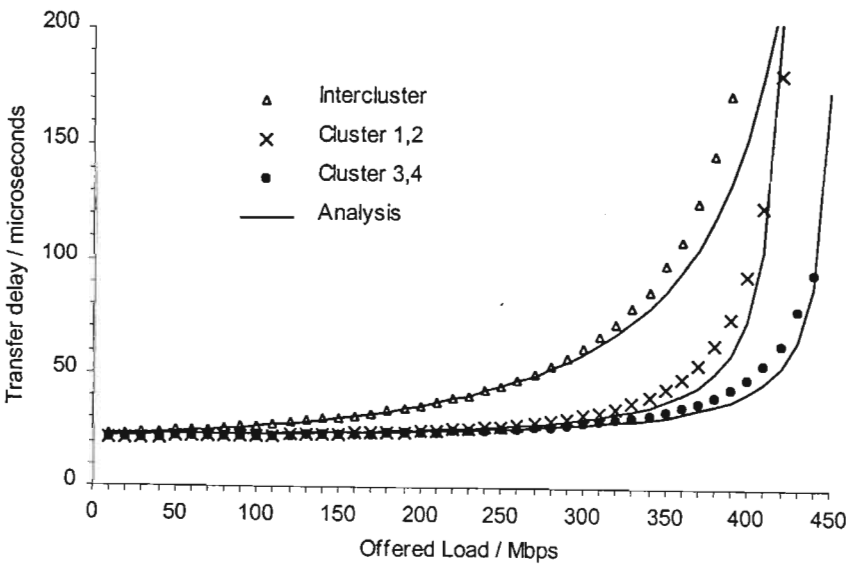
This results from the token passing overhead. During every token rotation, only m packets can be transmitted, while the time taken for the token to travel around the ring (R_m) is wasted. In the slotted ring case, the only loss comes from the measure introduced to prevent a node "hogging" a slot. A node is not allowed to reuse a slot. Consequently, the period of time it takes for the slot

to move to the next node is wasted. This time period is, on average, equal to the ring delay divided by the number of nodes. The slot is used for m such intervals and unused for one. As a result, the packet arrival rate the main ring can accommodate is given by,

$$\lambda_{o(\max)} = \frac{m}{(m+1)x} \tag{3.96}$$

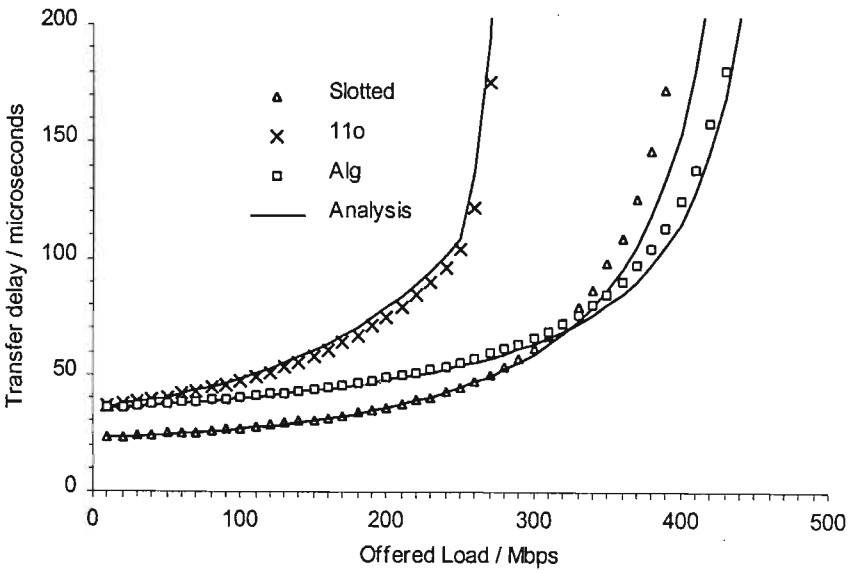


Graph 3-24: Comparison of overall mean packet transfer delays of slotted ring, *11o* and *Alg* protocols under random traffic

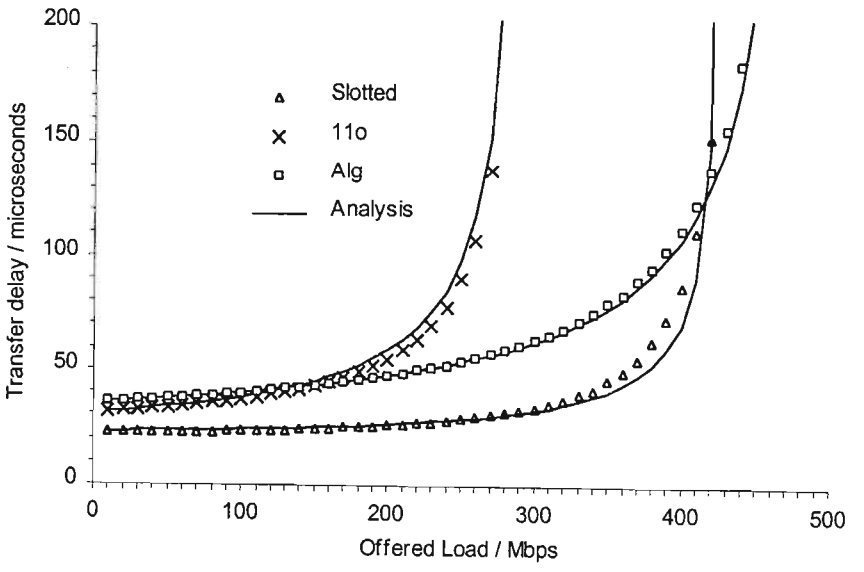


Graph 3-25: Mean packet transfer delays for all traffic classes under clustered traffic for 2-MR slotted ring

If we look at the performance of the 2-MR slotted ring under clustered traffic, we once again see the improved mean packet transfer delays. The curves in graph 3-25 are markedly "flat", i.e. the mean packet transfer delays increase very gradually with increases in offered load, over most of the domain. Like the *11o* and *ggo* protocols, the slotted protocol could be considered slightly unfair because intercluster packets have poorer performance than intracluster packets. Even so, graph 3-26 shows that their performance is still better than the *Alg* protocol (which was designed for fairness) for an offered load of up to about 330Mbps.



Graph 3-26: Comparison of mean packet transfer delays for worst traffic classes of slotted ring, *11o* and *Alg* protocols under clustered traffic



Graph 3-27: Comparison of overall mean packet transfer delays of slotted ring, *11o* and *Alg* protocols under clustered traffic

The performance gains achieved by using the slotted ring protocol are clearly evident in graph 3-27 which compares the overall performance of the *11o*, *ggo* and slotted protocols. Packet transfer delays are significantly lower at all but very high loads and the maximum throughput is only marginally less than the *Alg* protocol.

One must keep in mind, however, that practical systems rarely have fixed packet sizes. When packet sizes are variable, the slotted ring protocol incurs an additional performance penalty, because many slots are only partially filled. The token ring protocol has no inherent requirement for fixed packet sizes and thus does not incur the same penalties.

Furthermore, the slotted ring protocol does introduce extra complexities, such as the need to format the ring correctly into an integral number of slots. The ring delay may have to be increased somewhat to make this possible by introducing a variable length in-line shift register. Whenever nodes are added to or removed from the ring, the slot formatting will have to be adjusted. Nevertheless, the performance gains offered by a slotted system when the packet sizes are fixed are undeniable.

3.8 Summary

Chapter three was an investigation into the Two-Connected Multiple Ring. The original proposal by [Lye *et al*, 1995] utilised a l-limited token ring protocol where intracluster packets have access priority over intercluster packets to the main ring. This protocol is obviously non-optimal. Intercluster packets suffer an unfair performance penalty, especially under clustered traffic. This not only leads to higher intercluster packet transfer delays but also a higher overall average.

We considered four alternative protocols in an attempt to find a protocol that optimises the performance of the topology and evaluated the performance of each via both analysis and computer simulation. Both random and clustered traffic patterns were considered. Each protocol introduced some performance improvement over the previous proposal, until a near optimum was reached.

The first improvement (*11o*) simply involved changing the access priority to the main ring from the intracluster queue to the intercluster queue. The random traffic results improved marginally, but there was a vast improvement in intercluster and hence overall average performance under clustered traffic. In the second protocol (*ggo*), the service policies were changed from 1-limited

to gated. As expected, this led to significantly higher throughputs on all the rings, with a corresponding reduction in packet transfer delays at high loads for all classes.

It was then pointed out that the ideal scenario (*ideal*) would be to have the load balanced as equally as possible over all the rings. An analysis was performed assuming a known traffic pattern and optimally chosen a_i's. Random traffic performance again improved marginally. Under clustered traffic, the main improvement was evident in the performance of the class with the highest delay. The delay performance of all the classes proved very similar. A practical protocol that included a load-balancing algorithm (*Alg*) was then proposed. It was shown that this algorithm produced very similar results to the ideally balanced case and, in fact, even resulted in slightly lower packet transfer delays.

Finally, we considered a 2-MR network based on a slotted ring protocol. As is the case for single ring networks, the slotted protocol reduced the packet transfer delays considerably in all cases. It did, however, allow a slightly lower maximum useable throughput than the *ggo* and *Alg* protocols.

CHAPTER 4

DESTINATION REMOVAL DOUBLE RING

In a clustered traffic environment, nodes belonging to the same cluster are often likely to be geographically close together. This chapter proposes an alternative network for such environments that takes advantage of this fact and, like the 2-MR network, also requires only two transceivers and MAC entities per node: a destination removal double ring (DRDR) using a slotted protocol. This network has only two rings to which all nodes are connected. It provides a major performance improvement by allowing multiple simultaneous transmissions on the same ring and thus reusing the available spatial bandwidth.

An approximate analysis of the DRDR network is performed and the analytical results are verified by computer simulation. Results are compared to the 2-MR slotted ring network. They are given for both random and clustered traffic environments.

Furthermore, a novel immediate flow control scheme is proposed that allows the protocol to retain the benefits of immediate responses from destination nodes. Its performance is evaluated via computer simulation.

4.1 Topology

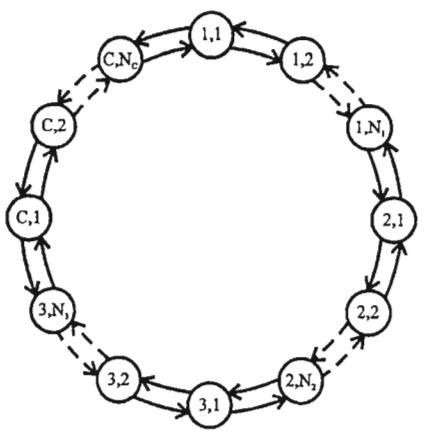


Figure 4-1: Network topology of destination removal double ring
N_i = Number of nodes in cluster i
C = Number of clusters

Figure 4-1 shows a general DRDR topology. C physical clusters are shown with N_i nodes in each cluster i (i = 1, 2,.... C) and a total of m nodes in the network. Nodes in the same cluster are adjacent. Each node has two transceivers that connect it to two physical rings. Data bits on each of these two rings travel in opposite directions. Packets are always transmitted on the ring with the shortest number of hops to the destination. Each node has a separate queue for each of the two rings. Both queues are assumed to be of infinite length.

4.2 Possible Protocols

To take full advantage of the available spatial bandwidth, a multiple access protocol needs to be used. A standard token ring protocol only allows access by a single node at a time: the node that currently holds the token. A slotted protocol allows a node to transmit as soon as an empty slot passes by. Thus, if there are s slots, s nodes can transmit simultaneously. A buffer insertion protocol allows a node to transmit whenever the ring is idle, i.e. whenever there is no packet currently passing by. Thus, it is possible for all nodes on the ring to transmit simultaneously.

A multiple token protocol has been proposed in [Cohen and Segall, 1994] which allows multiple access and this would be a possibility to investigate in future work. However, it has not been tried and tested and the performance characteristics are not well known.

This leaves us with the choice between slotted and buffer insertion protocols. It has been illustrated in [Bhuyan *et al*, 1989] that for fixed packet sizes the slotted ring protocol outperforms its counterpart, except at low loads. Since we are dealing with fixed packet sizes, we will choose the slotted protocol for our investigation. A buffer insertion protocol is an attractive alternative, however, when packet sizes are variable.

The MetaRing architecture (see section 2.7.4) is essentially a destination removal double with a few added extras, such as fairness controls. It has the option of using either a slotted or a buffer insertion protocol, depending on the application. The buffer insertion ring seems to be the more popular choice, but this is probably because variable packet sizes are usually involved.

4.3 Slotted ring analysis

The analysis of the DRDR follows similar lines to the analysis presented in [Bhuyan *et al, 1989*], which was used in section 3.7. In this case however, we can no longer use a single

utilisation value for the entire ring, since different sections of the ring have different loads. A separate utilisation value is required for each node.

Once again, we assume a Poisson packet arrival process and use an M/G/1 queuing model. Since the two rings operate independently of each other, the analysis examines only one ring. The results are applicable to both. The slotted ring protocol operates as before, except that now the destination node empties the slot and it can be used more than once during each circulation of the ring.

As for any network, the mean packet transfer delay for packets transmitted by node i is given by,

$$\overline{D_i} = \overline{W_i} + x + \overline{\gamma_i} \tag{4.1}$$

where $\overline{W_i}$ is the mean waiting time for packets in the node i queue, x is the slot (or packet) length and γ_i is the propagation delay from the source node (node i) to the destination. The slot length x is known, so we just need expressions for $\overline{W_i}$ and $\overline{\gamma_i}$.

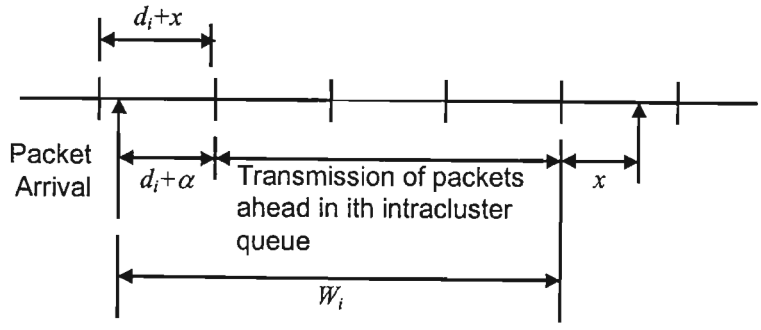


Figure 4-2: Waiting time as seen by an arriving packet in the *ith* intracluster queue

From figure 4-2, it can be seen that $\overline{W_i}$ is given by,

$$\overline{W_i} = \overline{d_i} + \alpha + \overline{Q_i}(\overline{d_i} + x) \tag{4.2}$$

where α is the remaining time for the current slot to pass by, d_i is the time until an empty slot arrives at the ith node and Q_i is the queue length at node i.

Assuming that consecutive slots are independent, the probability that node i has to wait j more slots before it finds an empty one is the geometric distribution

$$\overline{U_i}^{j}(1-\overline{U_i})$$

Hence the mean waiting time $\overline{d_i}$ to find an empty slot is given by

$$\overline{d_i} = \sum_{j=1}^{\infty} jx\overline{U_i}^{j}\left(1-\overline{U_i}\right) = \frac{\overline{U_i}x}{\left(1-\overline{U_i}\right)} \tag{4.3}$$

where $\overline{U_i}$ is the nominal utilisation of the ring at node *i*, given by

$$\overline{U_i} = \sum_{j=1}^{m}\sum_{k=1}^{m}\lambda_j[D]_{jk}[T_i]_{jk}x \tag{4.4}$$

where λ_i = Packet arrival rate to node *i*

$[D]_{jk}$ = Probability that a packet originating from node *j* is destined for node *k*

$[T_i]_{jk}$ = Probability that a packet originating from node *j* and destined for node *k* passes through node *i*

By applying Little's theorem ($\overline{Q_i} = \lambda\overline{W_i}$) and assuming a uniform distribution for α, equations (4.2) and (4.3) can be simplified to:

$$\overline{W_i} = \frac{\frac{1}{2}x(1+U_i)}{1-U_i-\lambda_i x} \tag{4.5}$$

The only other variable that remains unsolved is the mean propagation delay:

$$\overline{\gamma_i} = \sum_{j=1}^{m}\gamma_{ij}[D]_{ij} \tag{4.6}$$

where γ_{ij} is the propagation delay from node *i* to node *j*.

The overall packet transfer delay $\overline{D}$ can then be obtained by

$$\overline{D} = \sum_{i=1}^{m}\frac{\lambda_i}{\lambda}\overline{D_i} \tag{4.7}$$

In order to solve for a given packet arrival pattern, it is thus necessary to set up the matrices $\boldsymbol{D}$ and $\boldsymbol{T_i}$.

4.3.1 Random Traffic

Although the above analysis is all that is required to obtain the results for any traffic pattern, it can be simplified for random traffic environments. For the random traffic case, the analysis is symmetrical, i.e. the same for each node in the network.

In order to simplify the following discussion, we introduce the concept of a traffic unit, which we denote by the symbol u. We define it as the amount of traffic that flows between two specific nodes. For the random traffic case, it will obviously be a constant value that is directly proportional to the total offered load to the network. Since there are m nodes in the network and each node has $m-1$ other nodes for which its transmitted packets are destined, u is given by

$$u = \frac{\lambda x}{m(m-1)} \tag{4.8}$$

where λ is the total packet arrival rate to the network (both rings) and x is the mean slot length.

We now proceed to determine an expression for the mean number of hops packets travel from their source to destination nodes. The situation differs slightly depending on whether there are an even or odd number of nodes in the networks. If there are an even number of nodes, each source node will have a destination node exactly opposite it on the ring. In other words, the number of hops to this node will be the same on both rings. As a result, half the packets destined for this node will be transmitted on each of the two rings. There will thus be one traffic unit transmitted from the source node to each downstream node less than half the ring away and half a unit to the opposite node. Therefore, the mean number of hops packets have to travel is given by

$$\overline{h_{even}} = \frac{\frac{1}{2}m + \sum_{i=1}^{\frac{1}{2}m-1} i}{\frac{1}{2}m - \frac{1}{2}} = \frac{m^2}{4(m-1)} \tag{4.9}$$

If there are an odd number of nodes, on the other hand, the mean number of hops packets have to travel is given by

$$\overline{h_{odd}} = \frac{\sum_{i=1}^{\frac{1}{2}(m-1)} i}{\frac{1}{2}m - \frac{1}{2}} = \frac{m^2 - 1}{4(m-1)} \tag{4.10}$$

These results can both be accommodated by the following formula

$$\bar{h} = \frac{\left\lfloor \frac{1}{2}m \right\rfloor \cdot \left\lceil \frac{1}{2}m \right\rceil}{m-1} \tag{4.11}$$

If the ring delay for each of the two rings is R, this immediately leads to an expression for the mean propagation delay

$$\bar{\gamma} = \frac{\left\lfloor \frac{1}{2}m \right\rfloor \cdot \left\lceil \frac{1}{2}m \right\rceil}{m-1} \cdot \frac{R}{m} \tag{4.12}$$

Each packet, and therefore each traffic unit, travels $\bar{h}$ hops on average. This corresponds to a mean distance of $\bar{h}$ / m, when expressed as a fraction of the ring. Since there are m nodes each transmitting ½(m – 1) traffic units onto each ring, the total traffic intensity on a given link is given by

$$U_T = \tfrac{1}{2}(m-1)\bar{h}u = \frac{\bar{h}\lambda x}{2m} \tag{4.13}$$

Now the total traffic intensity on a link that connects a node to its downstream neighbour is equal to the sum of the intensity of traffic "passing through" that node and the traffic originating from that node. This can be written as

$$U_T = U_i + \tfrac{1}{2}(m-1)u \tag{4.14}$$

We can now solve for U_i using (4.13) and (4.14), which gives

$$U_i = \tfrac{1}{2}(m-1)(\bar{h}-1)u = \frac{(\bar{h}-1)\lambda x}{2m} \tag{4.15}$$

which can also be written as

$$U_i = \tfrac{1}{2}\left\lfloor \tfrac{1}{2}m-1 \right\rfloor \cdot \left\lceil \tfrac{1}{2}m-1 \right\rceil u \tag{4.16}$$

Equations (4.12) and (4.16) can be substituted into equation (4.5), noting that

$$\lambda_i = \frac{\lambda}{2m} \tag{4.17}$$

This solution removes the need for setting up all the matrices and computing the sums required earlier.

4.3.2 Clustered Traffic

It is easiest to simply apply the basic general analysis to the clustered traffic case since, in general, the utilisation values for each node are different. All that is required are expressions for λ_j, $[D]_{jk}$ and $[T_i]_{jk}$, which will be given. In equations (4.18) to (4.20), n refers to the cluster containing node j, λ_{cn} and λ_{on} refer to the intracluster and intercluster packet arrival rates to cluster n nodes and N_n refers to the number of nodes in cluster n.

$$\lambda_j = \frac{\lambda_{cn} + \lambda_{on}}{N_n} \tag{4.18}$$

If j and k are in the same cluster:

$$[D]_{jk} = \frac{\lambda_{cn}}{(\lambda_{cn} + \lambda_{on})N_n} \tag{4.19}$$

If j and k are in different clusters:

$$[D]_{jk} = \frac{\lambda_{on}}{(\lambda_{cn} + \lambda_{on})(m - N_n)} \tag{4.20}$$

The T_i matrices must be constructed as follows

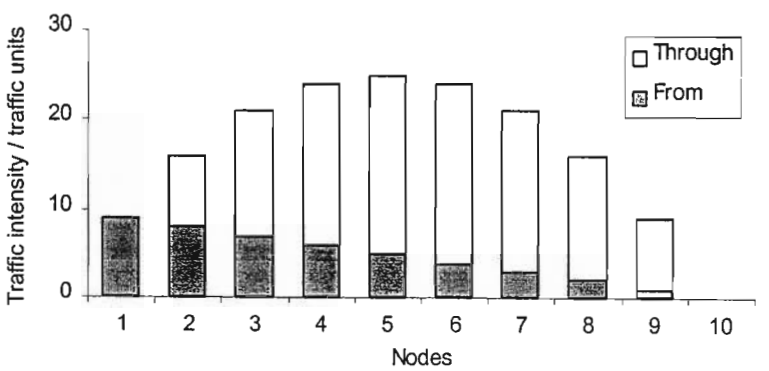
$[T_i]_{jk} = 1$ if j and k are less than half the ring apart and i lies between them ($i \neq j,k$)

$[T_i]_{jk} = ½$ if j and k are exactly half the ring apart and i lies between them ($i \neq j,k$)

$[T_i]_{jk} = 0$ otherwise

We now proceed to consider, a little more closely, the phenomenon of the ring utilisation being different at each node, since it is an inherent property of the protocol that has important consequences in terms of performance.

Once again, we use the concept of a traffic unit. This time, however, we alter the definition slightly to include only the amount of intracluster traffic that flows between two given nodes of the same cluster. As an example, we consider a ten node cluster. We only look at one of the two rings. Graph 4-1 shows the amount of intracluster traffic originating *from* each node in the cluster as well as the amount of intracluster traffic passing *through* each node.



Graph 4-1: Intensity of intracluster traffic passing through each node of a ten node cluster and originating from each node

The number of intracluster traffic units originating from each node is simply equal to the number of downstream nodes in the cluster. The number of intracluster traffic units passing through each node is equal to the sum of all traffic units originating from upstream nodes and destined for downstream nodes, which is simply given by the product of the number of

upstream nodes and the number of downstream nodes, i.e. for node n, the number of traffic units is $(n-1)(10-n)$. The amount of background intercluster traffic will also in general be different for each node and will tend to be lower further downstream the cluster.

The variation in traffic intensity around the ring has a few obvious consequences:

- The delay performance of each node will in general be different.
- There will be bottlenecks at areas of the ring with high utilisation.
- Certain nodes will be more likely to receive access than others, i.e. the network is not inherently fair.

These properties will become apparent in the results.

4.4 Results

The performance of the DRDR slotted ring network, as analysed above, is now quantitatively compared to the performance of the 2-MR slotted ring network (as presented in section 3.7) for both random traffic and 90% clustered traffic environments. Simulation results are also provided.

The network parameters used for the comparison are the same as those in chapter 4:

- total number of nodes in network, $m = 30$
- number of nodes in clusters 1 and 2, $N_1 = N_3 = 5$
- number of nodes in clusters 3 and 4, $N_2 = N_4 = 10$
- channel bit rate = 100Mbps
- slot/packet length, $x = 500$bits
- ring delay, $R = 3000$ bit times
- number of slots per ring = 6
- *Random traffic packet arrival rates*

  By decomposition of independent Poisson processes, the packet arrival rates are given by

$$\lambda_i = P_{ii}\lambda = \frac{N_i}{m}\frac{N_i - 1}{m-1}\lambda,$$

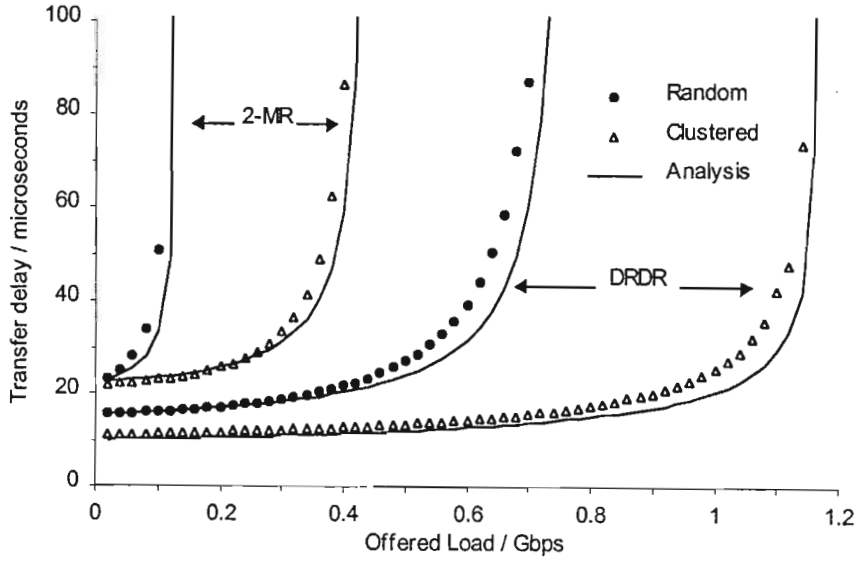
$$\lambda_{oi} = P_{oi}\lambda = \frac{N_i}{m}\frac{m - N_i}{m-1}\lambda$$

$$\text{and } \lambda_o = \sum_{i=1}^{C} P_{io}\lambda = \left(1 - \sum_{i=1}^{C} P_{ii}\right)\lambda$$

where P_{ii} is the probability that a packet originates from an ith cluster node and is destined for another ith cluster node. P_{iO} is the probability that a packet originates from an ith cluster node and is destined for a node in a cluster other than i. These formulae give numerical values of $\lambda_{c1} = \lambda_{c3} = 0.023\lambda$, $\lambda_{c2} = \lambda_{c4} = 0.103\lambda$ and $\lambda_o = 0.747\lambda$, i.e. 75% of the network traffic is due to intercluster packets.

- *Clustered traffic arrival rates*

  As in Lye *et al* [1995], the packet arrival rates used for the clustered traffic results are $\lambda_{c1} = \lambda_{c3} = 0.19\lambda$, $\lambda_{c2} = \lambda_{c4} = 0.26\lambda$, $\lambda_o = 0.1\lambda$ and $\lambda_{oi} = \lambda_o N_i / m$, i.e. 90% of the network traffic is due to intracluster packets.



Graph 4-2: Comparison of 2-MR and DRDR mean packet transfer delays for all traffic classes under random and clustered traffic

Graph 4-2 shows the performance of the 2-MR and DRDR slotted ring networks for both random and clustered traffic patterns. The analysis is accurate for low and medium loads and predicts the correct maximum throughput, but there is deviation from the simulation results for high loads, as is to be expected from this analysis technique.

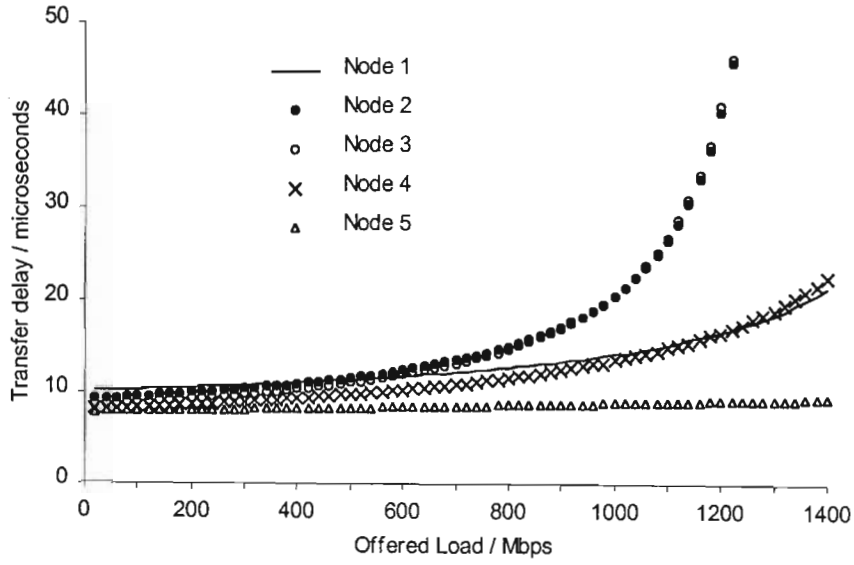
The 2-MR network is very inefficient with random traffic since over 75% of the traffic has to be transmitted on the main ring, which is only one of the five available rings. As a result, the main ring gets saturated far sooner than the cluster rings. The cluster rings perform well due to the low load offered to them (as shown in graph 3-22) but this has little effect on the overall network performance, which is similar to the intercluster performance. The 2-MR network

performs far better with the clustered traffic pattern because the load is now balanced over all five rings.

The DRDR network significantly outperforms the 2-MR network both in terms of average packet transfer delay and maximum throughput. In fact, the DRDR with random traffic performs far better than the 2-MR with clustered traffic, because it reuses the spatial bandwidth.

Due to source removal of packets, the 2-MR network is limited to a throughput of less than 100Mbps per ring and hence, under ideal conditions, less than 500Mbps for the entire network. The DRDR network has no such limitation because it reuses the available spatial bandwidth and can thus support throughputs far greater than 100Mbps per ring.

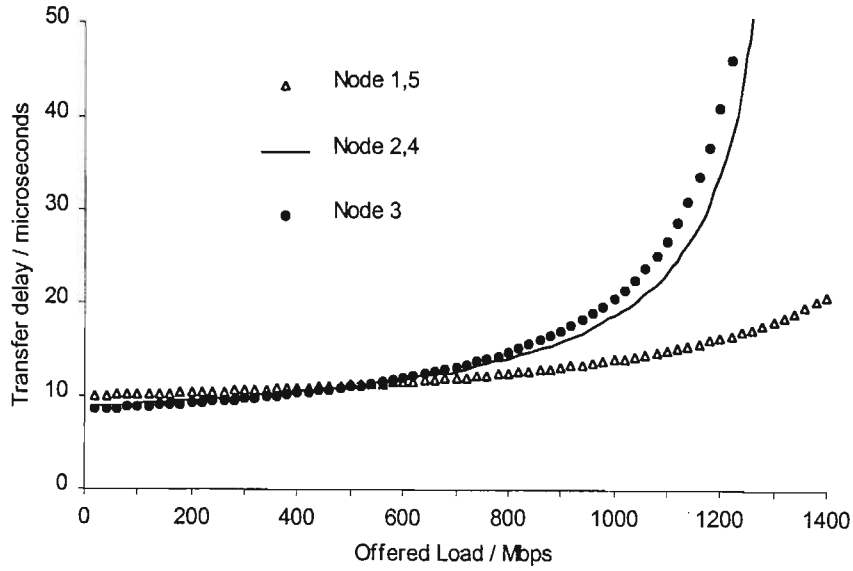
With the highly clustered arrangement of nodes assumed in this example, the DRDR performs extremely well under clustered traffic. However, if nodes from the same cluster were not adjacent, the results would be similar to those obtained for the random traffic case. In certain circumstances they could even be worse and hence it is important that nodes having a large volume of traffic between them, be spatially close together.



Graph 4-3: Mean packet transfer delays for traffic originating from each of the nodes of clusters 1 and 3 under clustered traffic on one of the rings

The DRDR performs impressively but, as mentioned earlier, it does have a few disadvantages. It is not "fair", i.e. it does not give equal access to all nodes in the network. The result of this

can be seen in graph 4-3 and graph 4-4, which show the packet transfer delays for the individual nodes of clusters 1 and 3 that were obtained from the analysis. The results for both clusters are identical due to the symmetry of the network. Graph 4-3 shows the results for the traffic transmitted on one of the two rings. Graph 4-4 shows the average delay for the traffic transmitted on both rings. Due to symmetry once again, nodes 1 and 5 as well as 2 and 3, have reversed performance on the two rings and hence the average delay for both nodes in each pair is identical.



Graph 4-4: Mean packet transfer delays for traffic originating from each of the nodes of clusters 1 and 3 under clustered traffic on both of the rings

The performance of the various nodes varies quite widely and there is nothing to stop a few of the nodes "hogging" the network. This problem can be overcome using "fairness" algorithms such as those specified in the MetaRing architecture (see section 2.8.3). Such algorithms do, however, introduce a significant overhead.

4.5 Immediate flow control scheme

This section proposes a novel flow control scheme that retains the advantage of immediate flow control that is often employed in source removal slotted ring protocols. The existing scheme uses a response field in the slot trailer that is set by the destination node immediately after reading a packet off the ring. This scheme ensures that a response is obtained from the destination node one ring delay after transmission of the packet. No additional overhead is required for low level flow control.

In the case of the DRDR, a single response field in the slot trailer can no longer be used. In order to maintain fast, low overhead flow control, a slot format similar to that depicted in figure 4-3 is proposed.

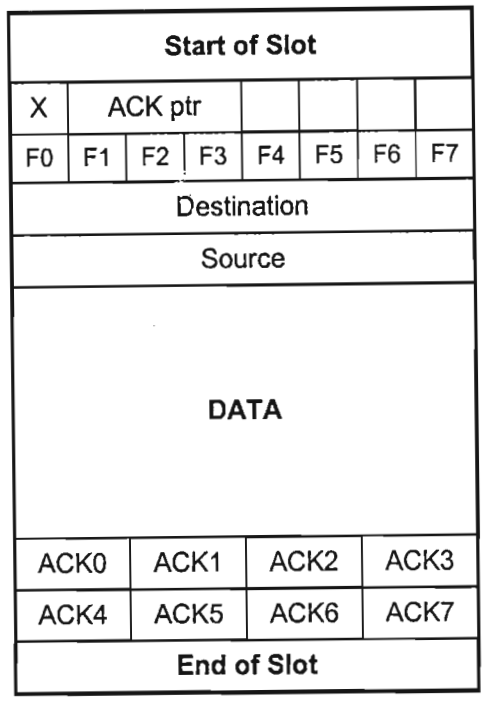


Figure 4-3: Immediate flow control slot format

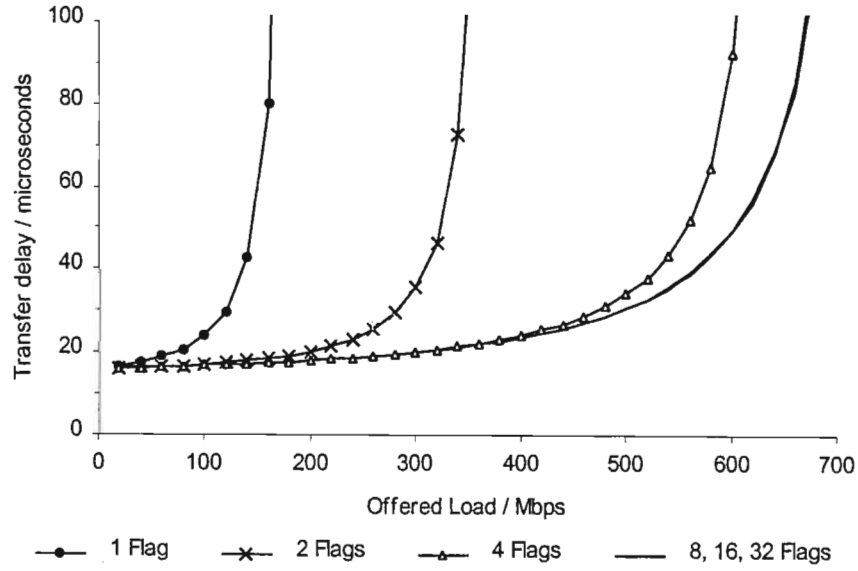
The slot now has eight 2 bit acknowledgement fields (ACK0 to ACK7) in the slot trailer. The header contains a 3 bit pointer (ACK ptr) indicating to the destination node which ACK field to use and eight 1 bit flags (F0 to F7) to indicate whether or not the corresponding acknowledgement field is in use.

When the slot arrives at the source node and it is determined to be empty (X flag), the node finds an empty ACK field by examining bits F0 to F7, finding one that is set to zero and setting it to one. It also sets "ACK ptr" to indicate to the destination node which field it has chosen. The destination node will then leave the header fields untouched except for the full/empty flag and place an acknowledgement in the correct ACK field. When the slot has circulated the entire ring and returns to the source node the appropriate F0 to F7 will be reset.

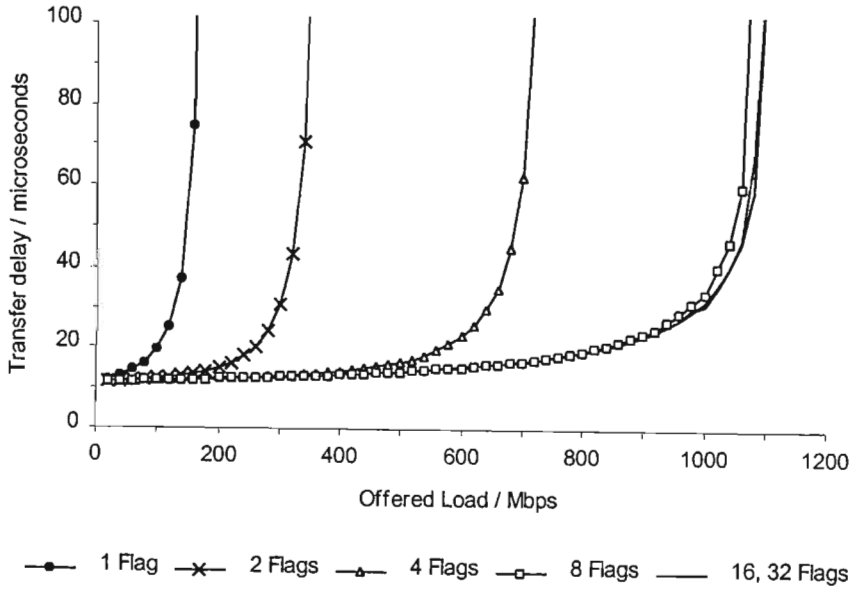
A side effect of this scheme is that each slot can be used at most eight times per ring circulation if only eight acknowledgement fields are used. Furthermore, if an error were to occur in the packet header, it would have an effect on more than one packet. Redundancy and error detection and recovery algorithms would be needed to deal with this situation.

4.5.1 Results

We now examine results of the scheme. The network parameters are the same as before except that the slot length has been lengthened to 525 bits to account for the overhead introduced by the scheme and the ring delay has been correspondingly increased to 3150 bit times.



Graph 4-5: Mean packet transfer delays for all traffic classes under random traffic



Graph 4-6: Mean packet transfer delays for all traffic classes under clustered traffic

Graph 4-5 and graph 4-6 show the effect of varying the number of acknowledgement fields in the slot header. The results were obtained from computer simulations. Having only one flag is equivalent to source removal of packets. Since there are only 30 nodes in the network, each slot can be used a maximum of 30 times per rotation (each node transmitting to its adjacent downstream neighbour), hence having 32 fields means an acknowledgement field is always available.

There is little difference between the performance with 8 acknowledgement fields and with 16 or 32, with only 30 nodes and 4 clusters. The extra overhead associated with the extra fields has not been considered, i.e. the more acknowledgement fields, the greater the overhead. For networks of this size, 8 fields are probably optimal. For larger networks, 16 or more fields would likely be required. The amount of clustering would obviously be a vital factor to consider.

4.6 Summary

In chapter four, a destination removal double ring using a slotted protocol was proposed as an alternative to the 2-MR for clustered traffic environments. An approximate analysis was presented. The simplifying assumptions cause deviations from the simulation results at high loads, however the results should still be sufficiently accurate for most uses.

It was shown that the DRDR slotted ring network had a significantly lower average packet transfer delay than the 2-MR slotted ring network, as well as far higher maximum throughputs due to spatial bandwidth reuse. This performance improvement was even greater for clustered traffic than for random traffic. Hence the DRDR network or derivatives thereof, such as the MetaRing, would be a good choice for clustered traffic environments.

Furthermore, the suggested immediate flow control scheme allows the protocol to retain the benefits of immediate responses from destination nodes without significantly degrading performance.

CHAPTER 5

EXACT RING NETWORK ANALYSES

Up to this point, all the analyses that have been performed have been approximate. The method that was used was fairly simple and in most cases gave accurate results. However, it was stated in chapter two that exact single token ring analyses have been performed in the past, but they are complicated and require solving large sets of equations. In section 5.1, we present an exact mean value analysis of a single token ring network with a 1-limited service discipline that was performed by the author. It clearly shows the complexities involved in analysing ring network protocols exactly. Nevertheless, we generate a set of m linear equations that can be solved for the mean queue lengths of all the nodes on the ring. We also derive an explicit solution for a token ring network with symmetrical load.

It has been stated on numerous occasions in the literature, for example [Bhuyan *et al*, 1989], that the general analysis of a multiple ring network is intractable. However, in section 5.2 we look at a novel Markov chain approach that gives exact results for multiple symmetrical token rings at near zero loads. Five examples are given, each one adding extra complexity to the problem, until a general double ring solution is arrived at. The same technique is applicable to networks with more than two rings.

5.1 Exact mean value analysis of a 1-limited token ring

The mean number in the system of an M/G/1 queue can be solved using a two-moment method [Kleinrock, 1975a]. Our method is based on this approach, but it is somewhat more involved. Recall from chapter two that the general M/G/1 model consists of a single queue with a Poisson arrival rate λ, being served by a single server with a service time generating function of $b(x)$, which is an independent variable. Kleinrock's analysis considers the imbedded Markov chain at packet departure instants. The variable q_n describes the queue length immediately after the nth packet has departed. It is described by the following equation.

$$q_{n+1} = q_n - \Delta_{q_n} + v_n \tag{5.1}$$

where

$$\Delta_{q_n} = \begin{cases} 0 & if \quad q_n = 0 \\ 1 & if \quad q_n > 0 \end{cases} \tag{5.2}$$

and v_n is the number of customers that arrive between departure instants n and $n + 1$. In the limit as n tends to infinity, i.e. at steady state, the probability distributions for the queue lengths at time n and $n + 1$ must be identical. Expressing this concept in terms of the expectations, we get

$$\bar{q} = \lim_{n\to\infty} \overline{q_{n+1}} = \lim_{n\to\infty} \overline{q_n} \tag{5.3}$$

The same equality will hold for the second moments. Exploiting this property, we can use equation (5.1) to solve for the mean number in the system.

If we take the expectations of both sides of equation (5.1), we arrive at the following expression.

$$\overline{\Delta_{q_n}} = \overline{v_n} \tag{5.4}$$

In the limit as n tends to infinity, we can drop the subscripts to obtain

$$\overline{\Delta_q} = \bar{v} = \rho \tag{5.5}$$

In other words, the mean number of arrivals per unit time is equal to the mean number of customers that are served per unit time. We have already come across this variable in earlier analyses. It is simply the utilisation factor ρ of the queue. This is a fairly obvious result that is not very helpful in solving for the expected number in the system, which is our goal. If, however, we square both sides of (5.1) we arrive at

$$q_{n+1}^{\ 2} = q_n^{\ 2} + \Delta_{q_n}^{\ 2} + v_n^{\ 2} - 2q_n\Delta_{q_n} + 2q_n v_n - 2\Delta_{q_n} v_n \tag{5.6}$$

Taking the expectations of both sides at steady state leads to

$$0 = \overline{\Delta_q^{\ 2}} + \overline{v^2} - 2\overline{q\Delta_q} + 2\overline{qv} - 2\overline{\Delta_q v} \tag{5.7}$$

which can be simplified to

$$0 = \overline{\Delta_q} + \overline{v^2} - 2\bar{q} + 2\overline{qv} - 2\overline{\Delta_q v} \tag{5.8}$$

It is this expression that we can now use to obtain the results we desire. Also, it is now evident why the method is called a two-moment method. Equation (5.7) is equivalent to having taken the second moments of both sides of equation (5.1). If v_n is independent of q_n, equation (5.8) can be further simplified to

$$0 = \overline{\Delta_q} + \overline{v^2} - 2\bar{q} + 2\bar{q}\bar{v} - 2\overline{\Delta_q}\,\bar{v} \tag{5.9}$$

Using equation (5.4), we can now easily solve for the mean number in the system, as follows

$$\bar{q} = \rho + \frac{\overline{v^2} - \bar{v}}{2(1-\rho)} \tag{5.10}$$

Unfortunately, equation (5.9) does not hold for the token ring case, because v_n is dependent upon q_n, as was discussed in chapter two. To make matters worse, if we have an m node

network, we need a separate version of equation (5.1) for each of the m nodes. Moreover, the v_n of each of the m equations is dependent upon each of the q_n's of all m equations. We now proceed to take all of this into account.

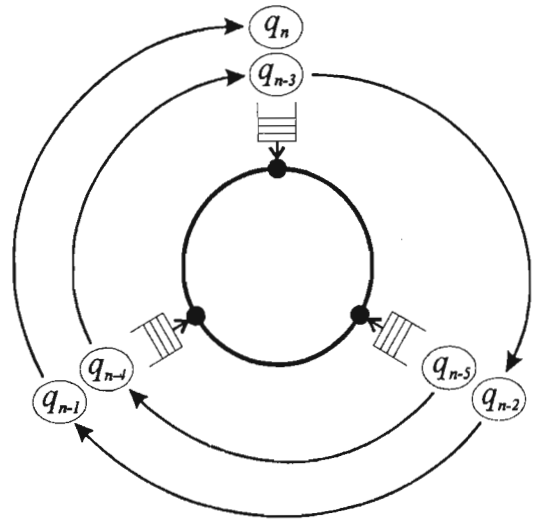


Figure 5-1: Three node token ring network

Figure 5-1 shows a three node token ring network as an illustrative example. New notation is introduced to simplify the following analysis. Time is discretised into token arrival instants. There are m such arrival instants per token rotation, as the token visits each of the m nodes. Each of the variables q_i shown in the circles refer to the queue length at the node at which the token arrives at time i. For a three node network, q_{n-3} and q_n refer to the queue length at the same node for successive token arrivals to that node. Furthermore, λ_n refers to the packet arrival rate (packets per bit time) to the node receiving the token at the nth token arrival. The packet length x_n is variable, with known mean and variance.

The queue length at time n is thus given by

$$q_n = q_{n-m} - \Delta_{q_{n-m}} + \sum_{i=1}^{m} v\left(\lambda_n, \Delta_{q_{n-i}}\right) + v_R\left(\lambda_n\right) \tag{5.11}$$

The variable v_n of equation (5.1) has now been split into $m + 1$ independent terms. The variable $v\left(\lambda_n, \Delta_{q_{n-i}}\right)$ is the number of packets that arrive at node n whilst packets are actively being transmitted by node $n - i$ and has the Poisson distribution

$$P\left(v\left(\lambda_n, \Delta_{q_{n-i}}\right) = k\right) = \frac{\left(\lambda_n \Delta_{q_{n-i}} x_n\right)^k e^{-\lambda_n \Delta_{q_{n-i}} x_n}}{k!} \tag{5.12}$$

The variable v_R is the total number of packets that arrive at the node whilst the token is being passed between nodes and has the Poisson distribution

$$P\left(v_R(\lambda)=k\right)=\frac{(\lambda R)^k e^{-\lambda R}}{k!} \tag{5.13}$$

where R is the ring delay. Summing over m consecutive time intervals gives

$$\sum_{k=0}^{m-1} q_{n-k} = \sum_{k=0}^{m-1}\left[q_{n-k-m} - \Delta_{q_{n-k-m}} + \sum_{i=1}^{m} v\left(\lambda_{n-k}, \Delta_{q_{n-k-i}}\right) + v_R\left(\lambda_{n-k}\right)\right] \tag{5.14}$$

Rearranging and grouping dependent terms gives

$$\begin{aligned} & q_n + \sum_{k=1}^{m-1}\left[q_{n-k} - \sum_{i=0}^{k-1} v\left(\lambda_{n-i}, \Delta_{q_{n-k}}\right)\right] \\ & = \sum_{k=m}^{2m-1}\left[q_{n-k} - \Delta_{q_{n-k}} + \sum_{i=k-m}^{m-1} v\left(\lambda_{n-i}, \Delta_{q_{n-k}}\right)\right] + \sum_{k=0}^{m-1} v_R\left(\lambda_{n-k}\right) \end{aligned} \tag{5.15}$$

Since packet arrivals are Poisson, this can be simplified to

$$\begin{aligned} & [q_n] + \sum_{k=1}^{m-1}\left[q_{n-k} - v\left(\sum_{i=0}^{k-1}\lambda_{n-i}, \Delta_{q_{n-k}}\right)\right] \\ & = \sum_{k=m}^{2m-1}\left[q_{n-k} - \Delta_{q_{n-k}} + v\left(\sum_{i=k-m}^{m-1}\lambda_{n-i}, \Delta_{q_{n-k}}\right)\right] + \left[v_R\left(\sum_{k=0}^{m-1}\lambda_{n-k}\right)\right] \end{aligned} \tag{5.16}$$

We now have an equation that fully describes the system. Furthermore, we have subdivided the equation into $2m + 1$ independent groups of terms. Each group is enclosed in square brackets. We could, at this point, square equation (5.16) and take the expectations of both sides. However, it is difficult to directly determine the expectations of each term group, so we take a slightly different approach – we take the z-transform of equation (5.16). Since each term group is independent, we can z-transform each one separately. We then proceed to take the second derivative of the transformed equation evaluated at $z = 1$, which is equivalent to determining the second moments.

We now consider each term group separately. By definition

$$Z\left\{q_n\right\} = \sum_{k=-\infty}^{\infty} P\left(q_n = k\right) z^k = Q_n(z) \tag{5.17}$$

The second "type" of term group on the left hand side of (5.16), we will refer to as $f_{n,k}$ and define as follows

$$f_{n,k} = q_{n-k} - v\left(\lambda_{n,k}, \Delta_{q_{n-k}}\right) \tag{5.18}$$

where

$$\lambda_{n,k} = \sum_{i=0}^{k-1}\lambda_{n-i} \tag{5.19}$$

Z-transforming $f_{n,k}$ gives

$$F_{n,k}(z) = \phi_{n-k} + [Q_{n-k}(z) - \phi_{n-k}]B^*(\lambda_{n,k} - \lambda_{n,k}z^{-1}) \tag{5.20}$$

where

$$\phi_n = P(q_n = 0) \tag{5.21}$$

The utilisation of the node served at time n is given by

$$\rho_n = 1 - \phi_n \tag{5.22}$$

$B^*(s)$ is the Laplace transform of the time spent serving the queue visited at time n. If $q_n = 0$, the service time will be zero. If, however, $q_n > 0$, the service time will be equal to the length of the packet that is transmitted, which is x_n. It is a well known queuing theory result that

$$V_{n,k}(z) = B^*(\lambda_{n,k} - \lambda_{n,k}z) \tag{5.23}$$

where $V_{n,k}(z)$ is the z-transform of $v(\lambda_{n,k}, \Delta_{q_{n-k}})$.

Later stages of the analysis require both the first and second derivatives of the z-transforms evaluated at $z = 1$ (i.e. the first and second moments), which are given by

$$F_{n,k}'(1) = Q_{n-k}'(1) + \gamma_{n,k} \tag{5.24}$$

$$F_{n,k}''(1) = Q_{n-k}''(1) + \alpha_{n,k}Q_{n-k}'(1) + \beta_{n,k} \tag{5.25}$$

where

$$\gamma_{n,k} = -\rho_n\lambda_{n,k}\bar{x} \tag{5.26}$$

$$\alpha_{n,k} = -2\lambda_{n,k}\bar{x} \tag{5.27}$$

$$\beta_{n,k} = \rho_n(\lambda_{n,k}{}^2\overline{x^2} + 2\lambda_{n,k}\bar{x}) \tag{5.28}$$

Now let us turn to the first type of term group on the right hand side of (5.16), namely

$$g_{n,k} = q_{n-k} - \Delta_{q_{n-k}} + v(\tilde{\lambda}_{n,k}, \Delta_{q_{n-k}}) \tag{5.29}$$

where

$$\tilde{\lambda}_{n,k} = \sum_{i=k-m}^{m-1} \lambda_{n-i} \tag{5.30}$$

It has a z-transform as follows

$$G_{n,k}(z) = \phi_n + \tfrac{1}{z}[Q_n(z) - \phi_n]B^*(\tilde{\lambda}_{n,k} - \tilde{\lambda}_{n,k}z) \tag{5.31}$$

Taking the first and second derivatives of $G_{n,k}(z)$ at $z = 1$, gives

$$G_{n,k}'(1) = Q_{n-k}'(1) + \kappa_{n,k} \tag{5.32}$$

$$G_{n,k}''(1) = Q_{n-k}''(1) + \tau_{n,k}Q_{n-k}'(1) + \omega_{n,k} \tag{5.33}$$

where

$$\kappa_{n,k} = \rho_n\left(\tilde{\lambda}_{n,k}\bar{x} - 1\right) \tag{5.34}$$

$$\tau_{n,k} = 2\tilde{\lambda}_{n,k}\bar{x} - 2 \tag{5.35}$$

$$\omega_{n,k} = \rho_n\left(\tilde{\lambda}_{n,k}^{\,2}\overline{x^2} - 2\tilde{\lambda}_{n,k}\bar{x} + 2\right) \tag{5.36}$$

The z-transform of the final term of the right hand side is given by,

$$V_R(z) = e^{-\lambda_T R(1-z)} \tag{5.37}$$

where

$$\lambda_T = \sum_{k=0}^{m-1}\lambda_{n-k} \tag{5.38}$$

The first and second derivatives evaluated at $z = 1$ are thus given by,

$$V_R'(1) = \lambda_T R \tag{5.39}$$

$$V_R''(1) = \lambda_T^{\,2} R^2 \tag{5.40}$$

The z-transform of (5.16) is therefore given by,

$$Q_n(z)\left[\prod_{k=1}^{m-1}F_{n,k}(z)\right] = \left[\prod_{k=m}^{2m-1}G_{n,k}(z)\right]V_R(z) \tag{5.41}$$

In the limit as n tends to infinity,

$$G_{n,k}(z) = G_{n-m,k}(z) \tag{5.42}$$

if we redefine $\tilde{\lambda}_{n,k}$ accordingly. We will still retain the same subscripts, but from now on, we assume that the system is in steady state. Hence

$$Q_n(z)\left[\prod_{k=1}^{m-1}F_{n,k}(z)\right] = \left[\prod_{k=0}^{m-1}G_{n,k}(z)\right]V_R(z) \tag{5.43}$$

where

$$\tilde{\lambda}_{n,k} = \sum_{i=k}^{m-1}\lambda_{n-k} \tag{5.44}$$

Taking the second derivative of the left hand side of (5.43) gives,

$$\frac{d^2}{dz^2}\left[Q_n(z)\prod_{k=1}^{m-1}F_{n,k}(z)\right]_{z=1}$$

$$= Q_n''(1) + \sum_{k=1}^{m-1}\left[F_{n,k}''(1) + 2Q_n'(1)F_{n,k}'(1) + 2\sum_{j=k+1}^{m-1}F_{n,k}'(1)F_{n,j}'(1)\right] \tag{5.45}$$

Substituting (5.24) and (5.25) into (5.45) gives

$$= Q_n''(1) + \sum_{k=1}^{m-1}\left[\begin{array}{l} Q_{n-k}''(1) + \alpha_{n,k}Q_{n-k}'(1) + \beta_{n,k} + 2Q_n'(1)\left(Q_{n-k}'(1) + \gamma_{n,k}\right) \\ +2\sum_{j=k+1}^{m-1}\left[Q_{n-k}'(1)Q_{n-j}'(1) + \gamma_{n,k}Q_{n-j}'(1) + \gamma_{n,j}Q_{n-k}'(1) + \gamma_{n,j}\gamma_{n,k}\right] \end{array}\right] \quad (5.46)$$

which, noting that $\alpha_{n,0} = 0$, can be rearranged to

$$\begin{aligned} &= \sum_{k=0}^{m-1}\left[Q_{n-k}''(1) + 2\sum_{j=k+1}^{m-1}Q_{n-k}'(1)Q_{n-j}'(1)\right] \\ &+ \sum_{k=0}^{m-1}\left[Q_{n-k}'(1)\left(\alpha_{n,k} + \sum_{\substack{j=0 \\ j\neq k}}^{m-1}\gamma_{n,j}\right)\right] + \sum_{k=1}^{m-1}\left[\beta_{n,k} + 2\gamma_{n,k}\sum_{j=k+1}^{m-1}\gamma_{n,j}\right] \end{aligned} \quad (5.47)$$

Taking the second derivative of the right hand side of (5.16) evaluated at $z = 1$ gives,

$$\frac{d^2}{dz^2}\left[\prod_{k=0}^{m-1}G_{n,k}(1)\right]V_R(1)$$

$$= \sum_{k=0}^{m-1}\left[G_{n,k}''(1) + 2G_{n,k}'(1)V_R'(1) + 2\sum_{j=k+1}^{m-1}G_{n,k}'(1)G_{n,j}'(1)\right] + V_R''(1) \quad (5.48)$$

Substituting (5.32) and (5.33) in (5.48) gives

$$= \sum_{k=0}^{m-1}\left[\begin{array}{l} Q_{n-k}''(1) + \tau_{n,k}Q_{n-k}'(1) + \omega_{n,k} + 2V_R'(1)\left(Q_{n-k}'(1) + \kappa_{n,k}\right) \\ +2\sum_{j=k+1}^{m-1}\left[Q_{n-k}'(1)Q_{n-j}'(1) + \kappa_{n,k}Q_{n-j}'(1) + \kappa_{n,j}Q_{n-k}'(1) + \kappa_{n,j}\kappa_{n,k}\right] \end{array}\right] + V_R''(1) \quad (5.49)$$

which can be rearranged to,

$$\begin{aligned} &= \sum_{k=0}^{m-1}\left[Q_{n-k}''(1) + 2\sum_{j=k+1}^{m-1}Q_{n-k}'(1)Q_{n-j}'(1)\right] + V_R''(1) \\ &+ \sum_{k=0}^{m-1}\left[Q_{n-k}'(1)\left(2V_R'(1) + \tau_{n,k} + 2\sum_{\substack{j=0 \\ j\neq k}}^{m-1}\kappa_{n,j}\right)\right] + \sum_{k=0}^{m-1}\left[\omega_{n,k} + 2\kappa_{n,k}\left(V_R'(1) + \sum_{j=k+1}^{m-1}\kappa_{n,j}\right)\right] \end{aligned} \quad (5.50)$$

The first summation terms cancel when equating (5.47) and (5.50), giving us

$$\begin{aligned} &\sum_{k=0}^{m-1}\left[Q_{n-k}'(1)\left(2V_R'(1) + \tau_{n,k} + 2\sum_{\substack{j=0 \\ j\neq k}}^{m-1}\kappa_{n,j}\right)\right] - \sum_{k=0}^{m-1}\left[Q_{n-k}'(1)\left(\alpha_{n,k} + 2\sum_{\substack{j=0 \\ j\neq k}}^{m-1}\gamma_{n,j}\right)\right] \\ &+ \sum_{k=0}^{m-1}\left[\omega_{n,k} + 2\kappa_{n,k}\left(V_R'(1) + \sum_{j=k+1}^{m-1}\kappa_{n,j}\right)\right] + V_R''(1) - \sum_{k=1}^{m-1}\left[\beta_{n,k} + 2\gamma_{n,k}\sum_{j=k+1}^{m-1}\gamma_{n,j}\right] = 0 \end{aligned} \quad (5.51)$$

This can be further simplified to,

$$\sum_{k=0}^{m-1}\left[Q_{n-k}'(1)\left(2V_R'(1)+\tau_{n,k}-\alpha_{n,k}+2\sum_{\substack{j=0\\ j\neq k}}^{m-1}(\kappa_{n,j}-\gamma_{n,j})\right)\right]=$$
$$\sum_{k=1}^{m-1}\left[\beta_{n,k}+2\gamma_{n,k}\sum_{j=k+1}^{m-1}\gamma_{n,j}\right]-\sum_{k=0}^{m-1}\left[\omega_{n,k}+2\kappa_{n,k}\left(V_R'(1)+\sum_{j=k+1}^{m-1}\kappa_{n,j}\right)\right]-V_R''(1) \quad (5.52)$$

What we actually have here is a system of m linear equations with m unknowns that can be solved to obtain the mean queue lengths.

5.1.1 Solving for ϕ_n

To solve for ϕ_n we need to take the first moments of (5.11) as follows

$$\overline{q_n}=\overline{q_{n-m}}-\overline{\Delta_{q_{n-m}}}+\sum_{i=1}^{m}\overline{v(\lambda_n,\Delta_{q_{n-i}})}+\overline{v_R(\lambda_n)} \quad (5.53)$$

In the limit as n tends to infinity

$$\overline{q_n}=\overline{q_{n-m}}$$

Hence

$$0=-\overline{\Delta_{q_n}}+\sum_{i=1}^{m}\overline{v(\lambda_n,\Delta_{q_{n-i}})}+\overline{v_R(\lambda_n)} \quad (5.54)$$

Now

$$\overline{\Delta_{q_n}}=1-\phi_n \quad (5.55)$$

$$\overline{v(\lambda_n,\Delta_{q_{n-i}})}=\lambda_n\overline{x\Delta_{q_{n-i}}}=\lambda_n\bar{x}(1-\phi_{n-i}) \quad (5.56)$$

$$\overline{v_R(\lambda_n)}=\lambda_n R \quad (5.57)$$

Substituting (5.55) to (5.57) back into (5.54) gives

$$\phi_n-1+\sum_{i=1}^{m}\left[\lambda_n\bar{x}(1-\phi_{n-i})\right]+\lambda_n R=0 \quad (5.58)$$

This can be rearranged to

$$\sum_{i=0}^{m-1}\left[\delta(0)-\lambda_n\bar{x}\right]\phi_{n-i}=1-\lambda_n(R+m\bar{x}) \quad (5.59)$$

Once again, we have a system of m linear equations that can be solved for ϕ_n.

5.1.2 Solving for the mean queue length

Note that we have solved for the number of packets in the *system* immediately after *token arrival* instants. This includes the packet, if any, that is currently being transmitted. The value

we really want is the mean number in the *queue* immediately after *packet departure* instants because, according to Little's theorem, this will be equal to the mean queue length at a random point in time.

The probability that a packet is transmitted when the token arrives, is simply the probability that the number in the system is non-zero, which by definition is given by ρ. Thus, the mean number in the system for node n immediately after a packet departure, is given by

$$\frac{\overline{q_n}}{\rho_n} \tag{5.60}$$

The mean queue length (excluding the packet currently being transmitted) is hence given by

$$\overline{Q_n} = \frac{\overline{q_n}}{\rho_n} - 1 \tag{5.61}$$

This is the result we desire.

5.1.3 Solving for the mean packet transfer delay

Now that we have solved for the mean queue lengths, it is a simple matter to solve for the packet transfer delay. According to Little's Theorem,

$$\overline{W_n} = \frac{\overline{Q_n}}{\lambda_n} \tag{5.62}$$

The mean packet transfer delay for node n is then given by

$$\overline{D_n} = \overline{W_n} + \bar{x} + \bar{\gamma}_n \tag{5.63}$$

where $\bar{\gamma}_n$ is the mean propagation delay of packets originating from node n. If the nodes are distributed evenly around the ring, the mean propagation delay will be half the ring delay for all nodes. The overall average packet transfer delay is then given by the weighted sum of the mean propagation delays for each of the nodes, where the weights are proportional to the packet arrival rates to the nodes.

$$\overline{D_{avg}} = \sum_{i=0}^{m-1} \frac{\lambda_{n-i}}{\lambda} \overline{D_{n-i}} \tag{5.64}$$

5.1.4 Solving for a symmetrical load

When the packet arrival rate to each node is the same, i.e. $\lambda_i = \lambda_j$ for all i and j, the queue length statistics for all the nodes will also be the same, i.e.

$$Q'(1) = Q_i'(1) = Q_j'(1) \tag{5.65}$$

Hence (5.52) can be rewritten as

$$Q'(1)\left[\sum_{k=0}^{m-1}\left(2V_R'(1) + \tau_{n,k} - \alpha_{n,k} + 2\sum_{\substack{j=0 \\ j\neq k}}^{m-1}\left(\kappa_{n,j} - \gamma_{n,j}\right)\right)\right] = \tag{5.66}$$

$$\sum_{k=1}^{m-1}\left[\beta_{n,k} + 2\gamma_{n,k}\sum_{j=k+1}^{m-1}\gamma_{n,j}\right] - \sum_{k=0}^{m-1}\left[\omega_{n,k} + 2\kappa_{n,k}\left(V_R'(1) + \sum_{j=k+1}^{m-1}\kappa_{n,j}\right)\right] - V_R''(1)$$

Solving for the mean number in the system gives

$$\bar{q} = \frac{\sum_{k=1}^{m-1}\left(\beta_{n,k} + 2\gamma_{n,k}\sum_{j=k+1}^{m-1}\gamma_{n,j}\right) - \sum_{k=0}^{m-1}\left(\omega_{n,k} + 2\kappa_{n,k}\left(V_R'(1) + \sum_{j=k+1}^{m-1}\kappa_{n,j}\right)\right) - V_R''(1)}{\sum_{k=0}^{m-1}\left(2V_R'(1) + \tau_{n,k} - \alpha_{n,k} + 2\sum_{\substack{j=0 \\ j\neq k}}^{m-1}\left(\kappa_{n,j} - \gamma_{n,j}\right)\right)} \tag{5.67}$$

Now that we have the general form of the solution for the mean number in the system, following further simplifications are possible.

$$\phi = \frac{1 - \lambda\left(R + m\bar{x}\right)}{1 - m\lambda\bar{x}} \tag{5.68}$$

$$\lambda_{n,k} = k\lambda \tag{5.69}$$

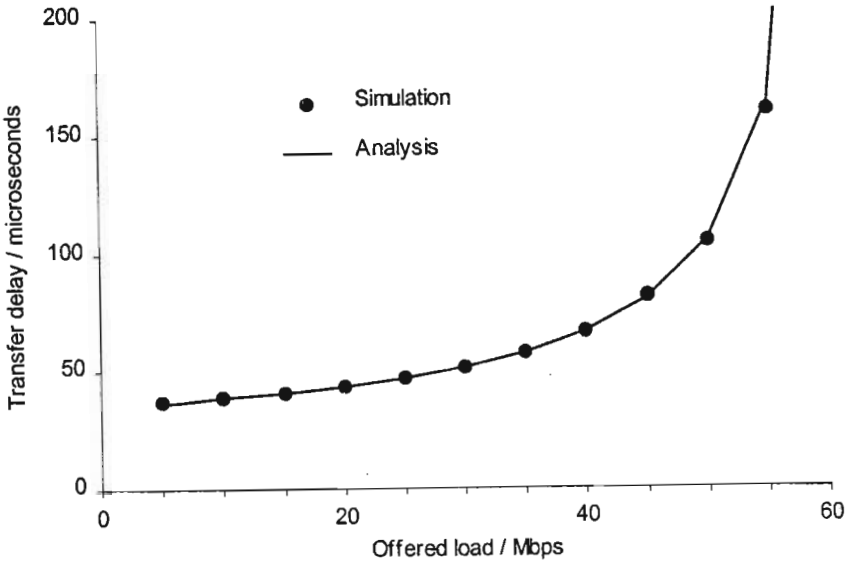
$$\tilde{\lambda}_{n,k} = (m-k)\lambda \tag{5.70}$$

The results of sections 5.1.2 and 5.1.3 can then be applied to obtain the mean packet transfer delay.

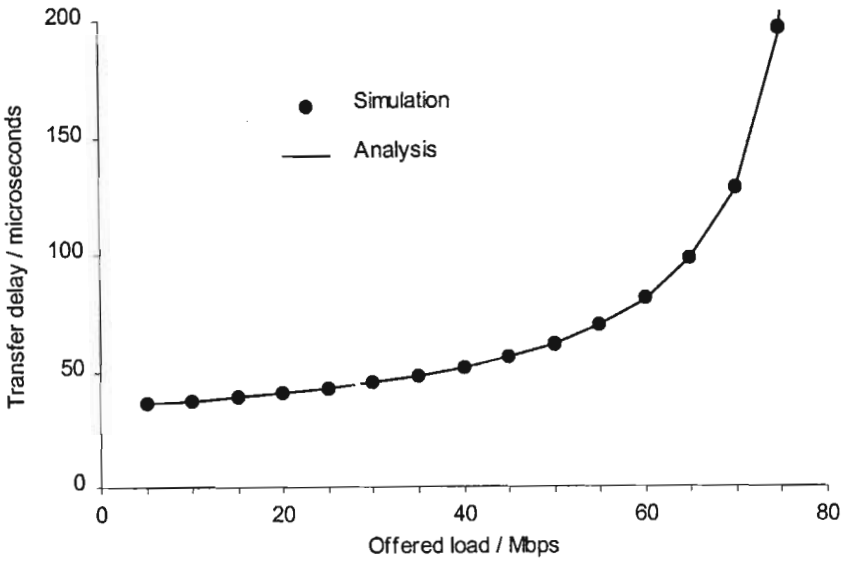
5.1.5 Results

We now compare the results of the above analysis to computer simulations. Two examples are considered. In the first, we assume a 10 node network operating at 100Mbps. The packet length is fixed at 500 bits and the ring delay is 3000 bits. Graph 5-1 shows the mean packet transfer delay for this network.

Graph 5-2 shows the second example, where we assume a 30 node network operating at 100Mbps. Again, the packet length is fixed at 500 bits and the ring delay is 3000 bits. The results in both cases clearly show that the analysis closely agrees with the simulation results, which validates its authenticity.



Graph 5-1: Comparison of exact single token ring analysis to simulation
$m = 10, x = 500, R = 3000$



Graph 5-2: Comparison of exact single token ring analysis to simulation
$m = 30, x = 500, R = 3000$

5.2 Applying Markov chains to multiple ring networks at near zero loads

It was pointed out in chapter two that an exact analysis of a symmetric multiple token ring with asynchronous tokens (or the closely related slotted ring) has never been performed and in fact, many sources state that it is indeed intractable. In this section, we apply a Markov chain approach to analyse such networks with fixed packet lengths at infinitesimal offered loads.

As the offered load to the network tends to zero

- The time intervals between arriving packets tend to infinity
- The number of token rotations between packets tends to infinity
- The probability of a queue containing more than one packet tends to zero
- The probability of more than one queue having a packet to transmit during the same token rotation tends to zero
- The token rotation time of each token tends to the ring delay

The discussion will now be restricted to a double ring network. The ring delays are assumed to be identical for both rings at R bit times and the packet lengths are fixed at x bits. One token can always be considered to be leading the other by a "distance" of half the ring or less. We denote this "distance" with the variable k, which we measure in packet lengths. We also introduce the variable R_k to denote the ring delay, measured again in packet lengths. The analytical results we present only apply to rational values of R_k.

Thus, the leading token will arrive, followed by the lagging token at a time k packet lengths later. The leading token will return after a further R_k - k packet lengths. The token interarrival times will alternate between k and R_k - k until another packet arrives at the network at some distant time in the future.

We now consider such a packet arrival to a node. Since the arrival is a random process independent of the token positions on the ring, the probability that the leading token will be received first is given by

$$P_{leading} = \frac{R_k - k}{R_k} \tag{5.71}$$

In this case, the "gap" between the leading and lagging tokens is decreased by a packet length. In other words, k is decremented by 1. On the other hand, the probability that the lagging token will be received first is given by

$$P_{lagging} = \frac{k}{R_k} \tag{5.72}$$

This results in k increasing by 1. If R_k is rational, we can obtain a Markov chain with k describing the state. We will now proceed to look at five examples of how the probability density of k can be found by solving the Markov chain, which then allows us to obtain the token interarrival statistics. Each example adds some additional complexity until all eventualities are covered.

Example 1: $R_k = 6k$

Figure 5-2 depicts the Markov chain that describes the system when $R_k = 6k$, including all transition probabilities. Obviously k can only vary between a minimum of zero and a maximum of three. We assume that both tokens are generated by the same node simultaneously, i.e. the system starts in the state $k = 0$.

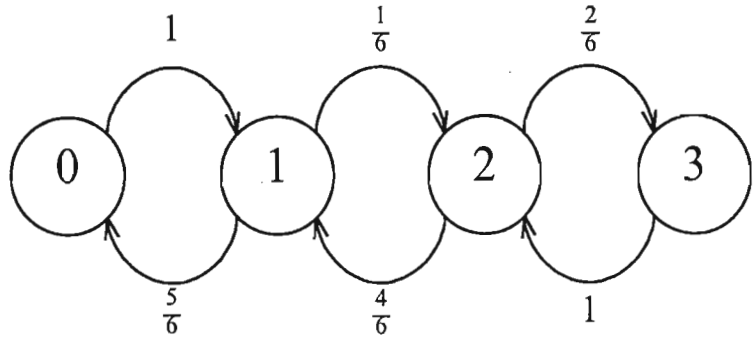


Figure 5-2: Markov chain for k where $R_k = 6k$

Consider the rare event of a packet arrival at a node.

- If $k = 0$, the two tokens will arrive at the node simultaneously. When the packet is transmitted, the token on one of the two rings will be delayed by a packet length, incrementing k to a value of 1. This is the only transition possible and hence has a probability of one.
- If $k = 1$ or $k = 2$, the transition probabilities are given by the equations for $P_{leading}$ and $P_{lagging}$.
- If $k = 3$, the tokens are exactly opposite each other on the ring. No matter which of the two tokens is delayed, k will be reduced to a value of 2. Since this is the only possible transition, it has a probability of one.

The chain can be easily solved as follows

$$P(k=0) = \tfrac{5}{6} P(k=1) \tag{5.73}$$

$$P(k=2) = \tfrac{1}{4} P(k=1) \tag{5.74}$$

$$P(k=3) = \tfrac{1}{3} P(k=2) = \tfrac{1}{12} P(k=1) \tag{5.75}$$

But

$$\sum_i P(k=i) = 1 \tag{5.76}$$

Hence

$$P(k=1)\left(\tfrac{5}{6} + 1 + \tfrac{1}{4} + \tfrac{1}{12}\right) = \tfrac{26}{12} P(k=1) = 1 \tag{5.77}$$

$$\Rightarrow P(k=1) = \tfrac{12}{26} \tag{5.78}$$

This leads to the probability density of k, as shown in table 5-1.

k	0	1	2	3
Probability	$\frac{10}{26}$	$\frac{12}{26}$	$\frac{3}{26}$	$\frac{1}{26}$

Table 5-1: Probability density of k

This probability density can now be used to solve for the mean and residual token interarrival times. The mean token interarrival time is given by

$$\overline{T} = \sum_k \tfrac{1}{2}\left[kp_k + (R_k - k)p_k\right]x = \tfrac{1}{2}R \tag{5.79}$$

which is independent of the probability distribution of k and holds for all values of R_k. However, this is not the case with the mean residual token interarrival time, since it is dependent on the second moment of the token interarrival time, which is given by

$$\overline{T^2} = \sum_k \tfrac{1}{2} p_k\left[k^2 + (R_k - k)^2\right]x \tag{5.80}$$

Substituting into equation (5.80) from table 5-1 gives

$$\overline{T^2} = \left[\left(\tfrac{10}{26}\right)\left(0^2 + 6^2\right) + \left(\tfrac{12}{26}\right)\left(1^2 + 5^2\right) + \left(\tfrac{3}{26}\right)\left(2^2 + 4^2\right) + \left(\tfrac{1}{26}\right)\left(3^2 + 3^2\right)\right]x^2 = \tfrac{125}{312}R^2 \tag{5.81}$$

This can now be substituted into the standard formula to obtain

$$\overline{T_R} = \frac{\overline{T^2}}{2\overline{T}} = \tfrac{125}{312}R = 0.40R \tag{5.82}$$

The result differs somewhat from the approximate expression in equation (2.48) from section 2.5, which gave

$$\overline{T_R} = \tfrac{2}{3}\overline{T} = \tfrac{1}{3}R = \tfrac{104}{312}R = 0.33R \tag{5.83}$$

Example 2: $R_k = \frac{8}{5}k$

We now look at a second example where R_k is no longer an integer. The same concept holds true, although the situation becomes slightly more complicated, as shown in figure 5-3.

If R_k is rational, it can be expressed in its simplest form as

$$R_k = \frac{s}{l} \tag{5.84}$$

where s and l are integers. The physical interpretation of this formula is that the ring is divided into s segments and a packet on the ring extends over l of these. In our example, $s = 8$ and $l = 5$.

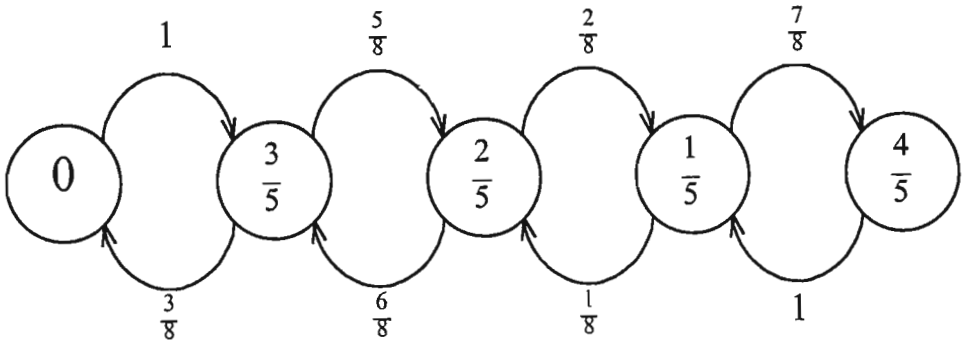


Figure 5-3: Markov chain for k where $R_k = \frac{8}{5}k$

We once again start with state $k = 0$. When one of the tokens is delayed by an arriving packet, we expect k to be incremented to a value of 1. This is not possible, however, since it is greater than $\frac{1}{2}R_k$. The token is actually delayed five out of the eight segments on the ring. The delayed token will now be leading the other token by three segments.

We introduce the notation $state_n$ to refer to the nth state from the left in the chain (as it is drawn in the figure), i.e. the value of k. In a system where $state_0 = 0$, the general formula for $state_n$ is given by

$$state_n = \frac{\frac{1}{2}s - \left|nl \bmod s - \frac{1}{2}s\right|}{l} \tag{5.85}$$

Also, the number of states is given by

$$Number\ of\ states = \left\lfloor \frac{s}{2} \right\rfloor + 1 \tag{5.86}$$

Notice that the states are no longer in ascending numerical order. Nevertheless, we still arrive at a linear (1-dimensional) birth-death model, which looks very similar to figure 5-2. There is a consistent pattern evident in the transition probabilities. Equations (5.71) and (5.72) still hold, however the following two formulae remove the need to determine which one of (5.71) and (5.72) is appropriate for a given transition. The first formula (5.87) returns the probability of moving one state to the right in the diagram.

$$P(state_n \rightarrow state_{n+1}) = \frac{nl \bmod s}{s} \tag{5.87}$$

The exception is when $n = 0$. In this case, the transition probability is 1. The second formula (5.88) returns the probability of moving one state to the left in the diagram.

$$P(state_n \leftarrow state_{n+1}) = \frac{(s - n - 1)l \bmod s}{s} \tag{5.88}$$

There is again an exception when n = ½s, where the transition probability is 1.

Solving this Markov chain gives the probability density of k, as shown in table 5-3.

k	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
Probability	$\frac{9}{128}$	$\frac{40}{128}$	$\frac{20}{128}$	$\frac{24}{128}$	$\frac{35}{128}$

Table 5-2: Probability density of k

Once again, we can use the distribution for k to solve for the token interarrival statistics, which can be calculated to be

$$\overline{T^2} = \tfrac{39}{128} R^2 \tag{5.89}$$

$$\overline{T_R} = \tfrac{39}{128} R = 0.30R \tag{5.90}$$

Example 3: $R_k = \frac{7}{3}k$

This example differs from the previous two in that s (the number of ring segments) is now an odd rather than an even number. Equations (5.85) to (5.88) still hold, however you will notice from figure 5-4 that the right hand side of the chain looks a little different. Equation (5.85) will generate two consecutive states of k = 2/3, for n = 3 and n = 4. Furthermore, equation (5.87) gives a transition probability of 2/7 from $state_3$ to $state_4$. We no longer have the exception to equation (5.88) where the transition probability from $state_{½s}$ to $state_{½s\,-\,1}$ is 1. The reason being that there is no $state_{½s}$, because s is odd. If the system is in $state_{½(s\,-\,1)}$ and the lagging token is received first, the transition will be back to $state_{½(s\,-\,1)}$.

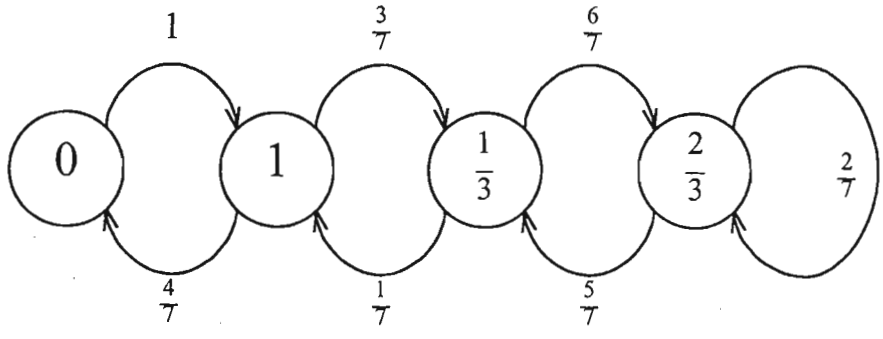


Figure 5-4: Markov chain for k where $R_k = \frac{7}{3}k$

Solving this Markov chain gives the probability density of k, as shown in table 5-3.

k	0	$\frac{1}{3}$	$\frac{2}{3}$	1
Probability	$\frac{20}{256}$	$\frac{105}{256}$	$\frac{126}{256}$	$\frac{35}{256}$

Table 5-3: Probability density of k

Once again, we can use the distribution for k to solve for the token interarrival statistics, which can be calculated to be

$$\overline{T^2} = \tfrac{671}{1792} R^2 \tag{5.91}$$

$$\overline{T_R} = \tfrac{671}{1792} R = 0.37R \tag{5.92}$$

Example 4: $R_k = 6k$, $state_0 = \frac{1}{2}$

Up to this point, we have assumed that the system is initialised with $k = 0$, i.e. the two tokens are generated by the same node simultaneously. This may not be the case in practice, however, especially if a fault occurs and one of the tokens is regenerated. The general model thus has to allow for an initial state where

$$state_0 = \frac{a}{l} \tag{5.93}$$

We have to appropriately adjust our other equations. Firstly, we need to divide the ring into more segments so that in $state_0$, k comprises an integral number of segments. Hence, if

$$R_k = \frac{s_0}{l_0} \tag{5.94}$$

in its simplest form, we need to choose s such that it is the lowest common multiple of a and s_0. Assuming this is the case, the general formula for $state_n$ is given by

$$state_n = \frac{\frac{1}{2}s - \left|(nl + a) \bmod s - \frac{1}{2}s\right|}{l} \tag{5.95}$$

This formula will generate every possible state, even if $state_0$ does not happen to be at one of the ends of the chain. What will happen is that new states will be generated until one end of the chain is reached. The formula will then go backwards through the same states until $state_0$ is again reached, at which time it will start generating new states in the opposite direction to before. We can further replace equations (5.87) and (5.88) with

$$P(state_{n-1} \rightarrow state_n) = \frac{m}{s} \Rightarrow P(state_n \rightarrow state_{n+1}) = \frac{(m+l) \bmod s}{s} \tag{5.96}$$

which can be used around the entire chain, i.e. both left and right. Therefore, to obtain the transition probabilities we merely need one initial probability to start with, namely

$$P(state_0 \rightarrow state_1) = \frac{a}{s} \tag{5.97}$$

where $state_0$ and $state_1$ have been generated by (5.95). It is important to note that the exceptions to equations (5.87) and (5.88) still apply, i.e. if we have states $k = 0$ or $k = \frac{1}{2}s$, we need to set the appropriate transition probabilities to 1.

Applying the above equations to our example leads to the following Markov chain.

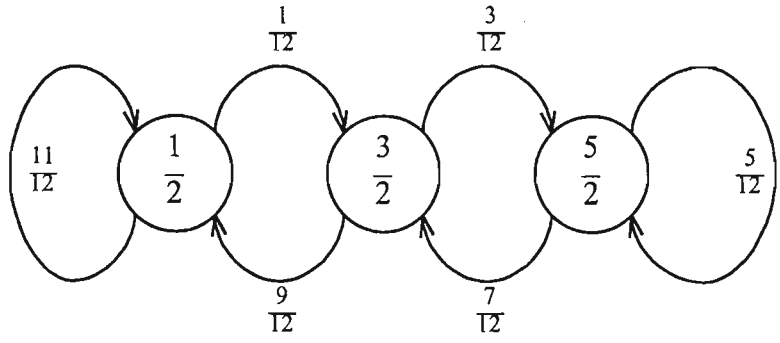


Figure 5-5: Markov chain for k where $R_k = 6k$, $state_0 = \frac{1}{2}$

Solving this Markov chain gives the probability density of k, as shown in table 5-4.

k	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{5}{2}$
Probability	$\frac{63}{73}$	$\frac{7}{73}$	$\frac{3}{73}$

Table 5-4: Probability density of k

Once again, we can use the distribution for k to solve for the token interarrival statistics, which can be calculated to be

$$\overline{T^2} = \tfrac{1423}{3504} R^2 \tag{5.98}$$

$$\overline{T_R} = \tfrac{1423}{3504} R = 0.41R \tag{5.99}$$

Example 5: $R_k = 4k$, $state_0 = \frac{1}{3}$

In this final example, we introduce the final complexity. Not all chains will be strictly birth-death models. Many will be circular, as shown in our example in figure 5-6. This will occur whenever there is no state where $k = 0$ or $k = \frac{1}{2}$. Equations (5.95) to (5.97) still operate

correctly, except that only the transitions in one direction (i.e. either clockwise or counter-clockwise) will be generated. To start the other "ring" of transition probabilities, the following formula can be used.

$$P(state_0 \rightarrow state_{-1}) = \frac{s-a}{s} \tag{5.100}$$

$State_{-1}$ refers to the state in the opposite direction around the ring to $state_1$, when starting from $state_0$. The Markov chain for example 5 is thus given by figure 5-6.

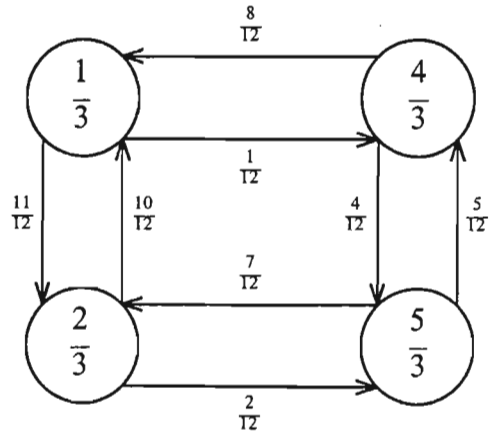


Figure 5-6: Markov chain for k where $R_k = 4k$, $state_0 = \frac{1}{3}$

It is not as simple to solve this chain as it was to solve the earlier chains because of its circular nature. We need to resort to some basic Markov chain theory. First, we need to set up a transition matrix **T**, where entry t_{ij} is the probability of moving from $state_j$ to $state_i$.

$$\mathbf{T} = \begin{bmatrix} 0 & \frac{8}{12} & 0 & \frac{10}{12} \\ \frac{1}{12} & 0 & \frac{5}{12} & 0 \\ 0 & \frac{4}{12} & 0 & \frac{2}{12} \\ \frac{11}{12} & 0 & \frac{7}{12} & 0 \end{bmatrix} \tag{5.101}$$

Our state vector **p** is given by

$$\mathbf{p} = \begin{bmatrix} P(k=\frac{1}{3}) \\ P(k=\frac{4}{3}) \\ P(k=\frac{5}{3}) \\ P(k=\frac{2}{3}) \end{bmatrix} \tag{5.102}$$

and must satisfy

$$\mathbf{Tp} = \mathbf{p} \tag{5.103}$$

$$(\mathbf{T} - \mathbf{I})\mathbf{p} = \mathbf{0} \tag{5.104}$$

where **I** is the identity matrix. We can now row reduce the matrix (**T** - **I**) and together with the conservation of probability (all the entries in **p** must sum to 1) we can solve for the probability density of k, as shown in table 5-5.

k	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{4}{3}$	$\frac{5}{3}$
Probability	$\frac{55}{136}$	$\frac{58}{136}$	$\frac{10}{136}$	$\frac{13}{136}$

Table 5-5: Probability density of k

Once again, we can use the distribution for k to solve for the token interarrival statistics, which can be calculated to be

$$\overline{T^2} = \tfrac{404}{1091} R^2 \tag{5.105}$$

$$\overline{T_R} = \tfrac{404}{1091} R = 0.37R \tag{5.106}$$

This technique will work for any double token ring network where the tokens rotate in the same direction and R_k is rational. In the case that R_k is irrational, a rational approximation can be carefully chosen to give any degree of required accuracy. It is interesting to note that the exact value for the residual token interarrival time obtained using this technique is usually significantly larger than the approximate value from [Bhuyan *et al*, 1989]. The reason for this discrepancy is that the tokens tend to coalesce rather than distributing evenly around the ring, which was the assumption made in [Bhuyan *et al*, 1989]. An explanation for this effect can be found in the transition probabilities of the Markov chains – the gap between the leading and lagging tokens is more likely to reduce in size than to increase, because the leading token is more likely to be received first. The larger the value of R_k, the more pronounced will be the coalescing effect. The more closely bunched the tokens become, the higher the variance in the token interarrival statistics becomes. Consequently, the mean residual token interarrival times become larger.

A similar technique (at near zero loads) can be applied to networks with more than two rings. An extra "dimension" of the Markov chain is required for each additional ring. The chain will become very large and complicated if there are many rings.

The same technique can also be applied to double token ring networks with 1-limited service policies when the network utilisation is very close to unity. In this case, since all nodes will usually have packets to transmit, the token rotation times are given by

$$\lim_{\rho \to 1} \overline{T_w} = mx + R \tag{5.107}$$

where m is the number of nodes in the network, x is the packet length and R is the ring delay. The token interarrival times will cycle between k and R_k - k, where R_k is now given by

$$R_k = \frac{mx + R}{x} \tag{5.108}$$

The state of the system will change in the rare event that a node does not have a packet to transmit.

Unfortunately, this approach only works for infinitesimal loads (or loads infinitesimally less than the maximum useable bandwidth). As soon as a non-zero load is considered, the problem becomes seemingly impossible to solve. All the initial assumptions are no longer valid. The system becomes multi-dimensional, since all the current queue lengths add an additional variable. As a result, we are forced to make use of approximate analyses and for most uses, simple is probably better.

5.3 Summary

Before chapter five, all the analyses that had been performed were approximate. The method that was used was fairly simple and in most cases gave accurate results. In section 5.1, however, we presented an exact mean value analysis of a single token ring network with a 1-limited service discipline. It clearly showed the complexities involved in analysing ring network protocols exactly. A set of m linear equations was derived that can be solved for the mean queue lengths of all the nodes on the ring. We also derived an explicit solution for a token ring network with symmetrical load. Two examples were given that validated the analysis, due to their close agreement with the simulation results.

We then looked at a novel Markov chain approach that gives exact results for multiple symmetrical token ring networks at near zero loads. Five examples were given, each one adding extra complexity to the problem, until a general double ring solution was arrived at. It was noted that the technique is applicable to networks with more than two rings, but an additional "dimension" of the chain is required for each additional ring.

The residual token interarrival times obtained in the examples differed somewhat from the approximate expression of [Bhuyan *et al*, 1989]. The reason for this discrepancy is that tokens on different rings tend to coalesce rather than distribute evenly around the network. An explanation for this effect can be found in the transition probabilities of the Markov chains – the

gap between the leading and lagging tokens is more likely to reduce in size than to increase, because the leading token is more likely to be received first. Unfortunately, the technique is only applicable at infinitesimal loads and cannot be easily extended to accommodate finite loads. It may, however, give insight into a more accurate approximate analysis.

CHAPTER 6

CONCLUSION

The first chapter introduced the basic concepts of ring networks and some of the issues they involve. It described what they are and how they compare to other network types. Their advantages, as well as limitations, were discussed. The different basic protocols were outlined. The focus of this thesis is multiple ring networks and hence some time was spent looking at the various issues they involve. Other aspects, including wiring, spatial bandwidth, costs, traffic patterns and performance evaluation, were also discussed.

Chapter two was a review of the relevant existing literature on ring networks. Firstly, we examined the single token, slotted and buffer insertion ring protocols, including the two most common implementations (IEEE 802.5 and FDDI) as well as the many analyses that have been performed thereof. We outlined an approximate method for analysing ring networks in general and proceeded to give the specifics of approximate token and slotted ring analyses.

We then looked at two multiple token ring networks. The first of these was the symmetrical multiple token ring, for which we gave some approximate analytical results. Importantly, we then reviewed the 2-MR network. This multiple token ring network was proposed to provide improved performance in clustered traffic environments. The original conclusions of the proposers, that it significantly outperforms a double ring network, were verified.

Next we examined four protocols that implement spatial bandwidth reuse: the parallelring, the multiple-token protocol, the PLAYTHROUGH ring and the MetaRing. These destination removal networks provide improved performance over their source removal counterparts by allowing multiple concurrent transmissions over the various segments of the ring. Finally we discussed four fairness algorithms: the timed token protocol, the helical window token ring and the MetaRing fairness algorithms. Fairness algorithms are especially important in ring networks that reuse the available spatial bandwidth because *starvation* of nodes can occur.

Chapter three included the major portion of new work – an investigation into the Two-Connected Multiple Ring. The original proposal by [Lye *et al*, 1995] utilised a l-limited token ring protocol where intracluster packets have access priority over intercluster packets to the

main ring. This protocol is obviously non-optimal. Intercluster packets suffer an unfair performance penalty, especially under clustered traffic, which not only leads to higher intercluster packet transfer delays, but also a higher overall average.

We considered four alternative protocols in an attempt to find a protocol that optimises the performance of the topology and evaluated the performance of each via both analysis and computer simulation. Both random and clustered traffic patterns were considered. Each protocol introduced some performance improvement over the previous proposal, until a near optimum was reached.

The first improvement simply involved changing the access priority to the main ring from the intracluster queue to the intercluster queue. The random traffic results improved marginally, but there was a vast improvement in intercluster and hence the overall average performance under clustered traffic. In the second protocol, the service policies were changed from 1-limited to gated. As expected, this led to significantly higher throughputs on all the rings, with a corresponding reduction in packet transfer delays at high loads for all traffic classes.

It was then pointed out that the ideal scenario would be to have the load balanced as equally as possible over all the rings. An analysis was performed assuming a known traffic pattern and optimally chosen a_i's. Random traffic performance once again improved marginally. Under clustered traffic, the main improvement was evident in the performance of the class with the highest delay. The delay performance of all the classes proved very similar. A practical protocol that included a load-balancing algorithm was then proposed. It was shown that this algorithm produced very similar results to the ideally balanced case and, in fact, even resulted in marginally lower packet transfer delays. Finally, we considered a 2-MR network based on a slotted ring protocol. As is this case for single ring networks, the slotted protocol reduced the packet transfer delays considerably in all cases. It did, however, allow a slightly lower maximum useable throughput than the gated and algorithm based token ring protocols proposed earlier.

In chapter four, we considered an alternative network that can be used in clustered traffic environments: a destination removal double ring. By reusing the available spatial bandwidth and having counter-rotating rings, this network provides significant performance improvements. A general approximate analysis was presented that can accommodate any traffic pattern. The DRDR analysis presented is approximate and the simplifying assumptions cause deviations from the simulation results at high loads. The results should still be sufficiently accurate for most uses, however.

We then specifically considered the performance of the network under random traffic and clustered traffic, where nodes in the same cluster are assumed adjacent. It was shown that the DRDR slotted ring network had a significantly lower average packet transfer delay than the 2-MR slotted ring network, as well as far higher maximum throughputs due to spatial bandwidth reuse. This performance improvement was even greater for clustered traffic than for random traffic. Hence the DRDR network or derivatives thereof, such as the MetaRing, would be a good choice for clustered traffic environments.

Furthermore, a scheme that still allows immediate flow control (a benefit of source removal ring networks) was proposed, which allows the DRDR protocol to retain the benefits of immediate responses from the destination nodes without significantly reducing overall system performance.

Before chapter five, all the analyses that had been performed were approximate. The method that was used was fairly simple and in most cases gave accurate results. In this chapter, however, we presented some exact methods for analysing ring networks. An exact mean value analysis of a single token ring network with a 1-limited service discipline was given. It clearly showed the complexities involved in analysing ring network protocols exactly. A set of m linear equations was derived that can be solved for the mean queue lengths of all the nodes on the ring. We also derived an explicit solution for a token ring network with symmetrical load. Two examples were given that validated the analysis, due to their close agreement with the simulation results.

We then looked at a novel Markov chain approach that gives exact results for multiple symmetrical token ring networks at near zero loads. Five examples were given, each one adding extra complexity to the problem, until a general double ring solution was arrived at. It was noted that the technique is applicable to networks with more than two rings, but an additional "dimension" of the chain is required for each additional ring.

The residual token interarrival times obtained in the examples differed somewhat from the approximate expression of [Bhuyan *et al*, 1989]. The reason for this discrepancy is that tokens on different rings tend to coalesce rather than distribute evenly around the network. An explanation for this effect can be found in the transition probabilities of the Markov chains – the gap between the leading and lagging tokens is more likely to reduce in size than to increase, because the leading token is more likely to be received first. Unfortunately, the technique is only applicable at infinitesimal loads and cannot be easily extended to accommodate finite loads. It may, however, give insight into a more accurate approximate analysis.

REFERENCES

Agrawal, G, Chen, B, Zhao, W and Davarai, S (1994). "Guaranteeing synchronous message dealines with the timed token medium access control protocol," *IEEE Transactions on Computers*, Vol. 43, No.3, pp. 327-339, March 1994

Aminetzah, Y (1975). "An exact approach to the polling system," Ph. D. Dissertation, McGill University, Montreal, P.Q., Canada, March 1975

Bhuyan, L, Ghosal, D and Yang, Q (1989). "Approximate Analysis of Single and Multiple Ring Networks," *IEEE Transactions on Computers*, Vol. 38, No. 7, July 1989

Blanc, JPC (1992). "An Algorithmic Solution of Polling Models with Limited Service Disciplines," *IEEE Transactions on Communications*, Vol. 40, No. 7, pp. 1152-1155, July 1992

Burr, WE (1986). "The FDDI optical data link," *IEEE Communications Magazine*, Vol. 24, No. 5, pp. 18-23, May 1986

Bux, W (1981). "Local-area subnetworks: A performance comparison," *IEEE Transactions on Communications*, Vol. COM-29, pp. 1465-1473, October 1981

Bux, W and Schlatter, M (1983). "An approximate method for the performance analysis of buffer insertion rings," *IEEE Transactions on communications*, Vol. COM-31, pp. 50-55, January 1983

Bux, W and Truong, HL (1983). "Token-ring performance: Mean delay approximation," *Proceedings of 10th International Teletraffic Congress*, pp. 3.3.3-1 – 3.3.3-7, 1981

Carsten, RT, Newhall, EE and Posner, MJM (1977). "A simplified analysis of scan times in an asymmetrical Newhall loop with exhaustive service," *IEEE Transactions on Communications*, Vol. COM-25, pp. 951-957, September 1977

Chen, J, Ahmadi, H and Ofek, Y (1991). "Performance study of a Gb/s MetaRing," *16th Conference on Local Computer Networks*, pp. 136-147, 1991

Chen, J, Cidon, I and Ofek, Y (1993). "A local fairness algorithm for gigabit LAN's/MAN's with spatial reuse," *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 8, pp. 1183 - 1191, August 1993

Cidon, I and Ofek, Y (1993). "MetaRing – A full-duplex ring with fairness and spatial reuse," *IEEE Transactions on Communications*, Vol. 41, No. 1, pp. 110 – 120, January 1993

Coffman, EG and Gilbert, EN (1986). "A Continuous Polling System with Constant Service Times," *IEEE Transactions on Information Theory*, Vol. IT-32, No. 4, pp. 584-591, July 1986

Cohen, R and Segall, A (1994). "Multiple logical token-rings in a single high-speed ring," *IEEE Transactions on Communications*, Vol. 42, No. 2/3/4, pp. 1712-1721, February/March/April 1994

Falconer, RM and Adams, JL (1985). "Orwell: A protocol for an integrated services local network," *British Telecom Technology Journal*, Vol. 3, No. 2, pp. 27-35, October 1985

Ferguson, MJ and Aminetzah, YJ (1985). "Exact Results for Nonsymmetric Token Ring Systems," *IEEE Transactions on Communications*, Vol. COM-33, No. 3, pp. 223-231, March 1985

Ganz, A and Chlamtac, I (1988). "Queuing analysis of finite buffer token networks", *Performance Evaluation Revue*, Vol. 16, pp. 30-36

Ghafir, H and Silio, CB Jr (1993). "Performance analysis of a multiple access ring network," *IEEE Transactions on communications*, Vol. 41, No. 10, pp 1494-1505, October 1993

Gross, D and Harris, CM (1985). *Fundamentals of Queuing Theory,* Second Edition, John Wiley and Sons 1985

Hammond, JL and O'Reilly, PJP (1986). "Performance analysis of local computer networks," Reading, MA: Addison-Wesley, 1986

Hopper, A and Needham, RM (1988). "The Cambridge fast ring networking system," *IEEE Transactions on Computers*, Vol. 37, No. 10, pp. 1214-1223, October 1988

Hubber, DE, Steinlin, W and Wild, PJ (1983). "SILK: An implementation of a buffer insertion ring," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-1, pp. 766-774, November 1983

Humblet, P (1978). "Source coding for communication concentrators," *Electron. Syst. Lab.*, Massachusetts Institute of Technology, Cambridge, ESLR-798, January 1978

Ibe, OC and Trivedi, KS (1990). "Stochastic Petri net models of polling systems," *IEEE Journal Selected Areas in Communications*, Vol.8, No. 12, pp. 1649-1657, December 1990

Jafari, H, Lewis, TG and Spragins, JD (1980). "Simulation of a class of ring structured networks," *IEEE Transactions on Computers*, Vol. C-29, pp. 385-392, May 1980

Johnson, MJ (1987). "Proof that the timing requirements of the FDDI token ring protocols are satisfied," *IEEE Transactions on Communications*, Vol. COM-35, No. 6, pp. 620-625, June 1987

Jung, WY and Un, CK (1994). "Analysis of a Finite-Buffer Polling System with Exhaustive Service Based on Virtual Buffering," *IEEE Transactions on Communications*, Vol. 42, No. 12, pp. 3144-3149, December 1994

Kamal, AE and Hamacher, VC (1989). "Approximate analysis of nonexhaustive multiserver polling systems with applications to local area networks," *Computer Networks and ISDN Systems*, Vol. 17, pp. 15-27, 1989

Kamal, AE and Hamacher, VC (1989). "Utilizing bandwidth sharing in the slotted ring," *IEEE Transactions on Computers*, Vol. C-39, No. 3, March 1990

Kervelas, G and Leon-Garcia, A (1993). "Delay analysis of various service disciplines in symmetric token passing networks," IEEE Transactions on Communications, Vol. 41, No. 9, pp. 1342-1355, September 1993

King, PJB and Mitrani, *I* (1987). "Modelling a slotted ring local area network," *IEEE Transactions on Computers*, Vol. C-36, May 1987

Kleinrock, L (1975a). *Queuing Systems. Volume I: Theory*, John Wiley and Sons, Wiley-Interscience Publication 1975

Kleinrock, L (1975b). *Queuing Systems. Volume II: Computer Applications*, John Wiley and Sons, Wiley-Interscience Publication 1975

Kschischang, FR and Molle, ML (1989). "The helical window token ring," *IEEE Transactions on Information Theory*, Vol. 35, No. 3, pp. 626-636, May 1989

Leventis, S, Papadopoulos, G and Koubias, S (1982). "A new experimental computer network and its simulated performance," *IEEE Proceedings of INFOCOM '82*, Las Vegas, NV, pp. 113-121, March 1982

Lye, KM, Wong, TC and Chua, KC (1995). "Two-connected multiple-token multiple-ring network," *IEE Proceedings on Communications*, Vol. 142, No. 6, December 1995

Mack, C (1957). "The efficiency of *N* machines uni-directionally patrolled by one operative when walking time is constant and repair times are variable," *Journal Royal Stat. Society Series B*, Vol. 19, No. 1, pp. 173-178, 1957

Mack, C, Murphy, T and Webb, NL (1957). "The efficiency of *N* machines uni-directionally patrolled by one operative when walking time and repair times are constant," *Journal Royal Stat. Society Series B*, Vol. 19, No. 1, pp. 166-172, 1957

Ofek, Y (1994). "Overview of the MetaRing architecture," *Computer Networks and ISDN Systems*, Vol. 26, No. 6, pp. 817-829, 1994

Ohnishi, H, Morita, N and Suzuki, S (1989). "ATM ring protocol and performance," *Proceedings ICC'89*, pp. 394-398, 1989

Proteon Associates, Inc (1982). "Operation and maintenance manual for the PRONET local network interface", issue No. 5, 30 April 1982

Reames, CC and Liu, MT (1976). "Design and simulation of the distributed loop computer network (DLCN)," *Proceedings of 3rd Annual Symposium on Computer Architectures*, Clearwater, FL, pp. 124-129, January 1976

Robillard, PN (1974). "An analysis of a loop switching system with multirank buffers based on the Markov process", *IEEE Transactions on Communications*, Volume 22, pp. 1772-1778

Ross, FE (1986). "FDDI – A tutorial," *IEEE Communications Magazine*, Vol. 24, No. 5, pp. 10-17, May 1986

Schwartz, M (1987). *Telecommunications Networks: Protocols, Modeling and Analysis*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.

Sethi, AS and Saydam, T (1985). "Performance analysis of token ring local area networks," *Comput. Netw. ISDN Syst.*, Vol. 9, No. 3, 1985, pp. 158-166

Sevcik, KC and Johnson, MJ (1987). "Cycle time properties of the FDDI token ring protocol," *IEEE Transactions on Software Engineering*, Vol. SE-13, No. 3, pp. 376-385, 1987

Silio, CB Jr and Wilson, TC (1987). "A performance model for double PLAYTHROUGH circuit-switched rings," IEEE Proceedings of INFOCOM '87, San Francisco, CA, pp. 745-753, April 1987

Silio, CB Jr, Ghafir, H and Parikh, R (1992). "An approximate method for the performance analysis of PLAYTHROUGH rings," *IEEE Transactions on Computers*, Vol. 41, No. 9, pp 1137-1155, September 1992

Simha, R and Ofek, Y (1991). "A starvation-free access protocol for a full-duplex buffer insertion ring local area network," *Computer Networks and ISDN Systems*, Vol. 21, No. 2, pp. 109 - 120, April 1991

Takagi, H (1985). "On the analysis of a symmetric polling system with single-message buffers," *Performance Evaluation*, Vol. 5, No. 3, pp. 149-157, 1985

Takagi, H (1986). *Analysis of Polling Systems*. The Massachusetts Institute of Technology Press, Cambridge, Massachusetts, 1985

Takagi, H (1991). "Analysis of finite-capacity polling systems", *Advanced Applications of Probability*, Vol. 23, pp. 373-387, June 1991

Takine, T, Takahashi, Y and Hasegawa, T (1988). "Exact analysis of asymmetric polling systems with single buffers," *IEEE Transactions on Communications*, Vol. 36, No. 10, pp. 1119-1126, October 1988

Tran-Gian, P and Raith, T (1988). "Performance analysis of finite capacity polling systems with nonexhaustive service", *Performance Evaluation*, Vol. 9, pp. 1-16, 1998

Van Arem, B and Van Doorn, EA (1990). "Analysis of a queuing model for slotted ring networks," *Computer Networks and ISDN Systems*, Vol. 20, pp. 309-314, 1990

Wilkes, MV and Wheeler, DJ (1979). "The Cambridge digital communication ring," *Proceedings of Local Area Communications Networks Symposium*, Boston, MA, pp. 47-61, May 1979

Wilson, TC and Silio, CB Jr (1979). "Loop interconnection of processors using a 'play through' protocol," *Proceedings of the 9the International Symposium on Mini and Micro Computers, MIMI '79*, Montreal, P.Q. Canada, and in *Mini and Micro Computers*, Vol. 5, No. 1, ACTA Press, Anaheim, CA, pp. 175-180, September 1979

Yung, M and Ofek, Y (1995). "MetaNet: Principles of an arbitrary topology LAN," *IEEE/ACM Transactions on Networking*, Vol. 3, No. 2, pp. 168 - 180, February 1995

Zafirovic-Vukotic, M, Niemegeers, IG and Valk, DS (1988). "Performance analysis of slotted ring protocols in HSLAN's," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-6, No. 6, July 1988