DQDB MEDIA ACCESS CONTROL PROTOCOL: PERFORMANCE EVALUATION AND UNFAIRNESS ANALYSIS

Internal Report C89-06

22/March/1989

Marco Conti
Enrico Gregori
Luciano Lenzini

DQDB Media Access Control Protocol: Performance Evaluation and Unfairness Analysis.

M. Conti, E. Gregori, L. Lenzini

CNR - Istituto CNUCE
Via S. Maria, 36
56100 Pisa
Italy

Tel: (50) 593111
Telex: 500371
Telefax: (50) 576751

ABSTRACT

This paper reports on an extended simulation analysis of the Distributed Queue Dual Bus (DQDB) MAC protocol. The simulation analysis is aimed at catching the most relevant protocol mechanisms (requests, empty and busy slots, distributed queue, etc.) which are responsible for the DQDB behaviour. First, the results obtained under several levels and characterization of the offered load to each station of a MAN made up of 50 stations are shown. The DQDB analysis is then complemented by extending the DQDB study to include 1.2 Gbps, which is the speed envisaged for MANs in the very near future.

1.0 Introduction

DQDB is a very promising Media Access Control (MAC) protocol which is being developed in IEEE 802 as an 802.6 MAN Standard proposal. Details on DQDB can be found in [IEEE802.6].

DQDB provides synchronous and asynchronous transport services to its users. This paper focuses on the asynchronous services.The rationale behind this choice is that these asynchronous services are of the same type as those being considered in the ongoing studies on ATM for B-ISDN ([HIRO88], [VORS88]). This leads, among other things, to a simplified integration between these two emerging technologies.

DQDB, as any other MAC for MAN, is suitable for the interconnection of LANs, PABXs, or other current communication equipment, as well as new, higher data rate equipment, and it is a natural candidate for providing integrated services (video, voice, data), a requirement made possible for all new telecommunication networks (i.e., B-ISDN) by the increasing bandwidth of optical fibers.

The paper focuses on the extent to which DQDB is suitable to perform service integration. This suitability is expressed by means of several performance diagrams representing access delay vs. throughput under various workload conditions. Such results are then interpreted in terms of DQDB internal mechanisms.

At the moment, there is no precise characterization of the traffic generated by such integrated services. For this reason, the simulation analysis was performed varying the packet interarrival distribution, message length and the distribution of the overall load among the stations.

This analysis is then complemented by extending the DQDB study to include results obtained at 1.2 Gbps, which is the speed envisaged for MANs in the very near future.

The paper is organized as follows. In Section 2 the DQDB MAC protocol is sketched. Section 3 briefly shows the DQDB stochastic queuing model. Section 4 focuses on the workload characterization and performance indices. Section 5 shows and discusses the results obtained at 150 Mbps, while Section 6 extends the DQDB analysis to 1.2 Gbps. The conclusions are drawn in Section 7.

2.0 DQDB Basic Access Mechanism

In this section only the main features of DQDB will be highlighted.

The principal components of the DQDB architecture are the nodes responsible for generating frame synchronization (Head and End), two contra-directional buses, and a multiplicity of intermediate nodes - addressed by an integer number ranging from 1 to N - which access both buses as shown in Figure 2.1. The Head node generates frame synchronization on the forward bus (bus A) and the End node generates frame synchronization at the same rate on the reverse bus (bus B). The frame interval is 125 microseconds, matching that of digital telephony. The frame is subdivided into a fixed number of equally sized units called slots. Slots are used to carry segments. Each slot contains an access control field which is used to control the writing of segments into a slot and the reading of a segment from an occupied, or busy, slot. There are two types of slots: queued-arbitrated and non-arbitrated. Each type of slot is used to transfer a different type of segment. Queued-arbitrated slots are used to transfer asynchronous segments, while non-arbitrated slots are used to transfer isochronous-sample segments. There are different access mechanisms for each of the slot types. This paper deals with the queued-arbitrated slot access. The segmentation of packets, provided by a user node, into fragments (which can be accommodated within a slot) and the reassembly of the packets at the destination node is an additional feature of the DQDB protocol which will not be discussed in this paper. Details on fragmentation/reassembly can be found in [IEEE802.6].

Queued-Arbitrated Slot Access

Asynchronous segments are written into empty, queued-arbitrated slots under the control of the Distributed Queue. The Distributed Queue is controlled by counters at each station. A separate Distributed Queue is operated for each of the two contra flowing buses, with separate counters. There can be a maximum of only two asynchronous segments queued within the Distributed Queue for access at each node, one for each bus. In addition to the Distributed Queue, each node may maintain, for each bus, a local queue for segments which cannot be placed in the Distributed Queue.

The DQDB protocol is specified by the Distributed Queue State Machine (DQSM). Before describing the various DQSM transitions, it is necessary to highlight the roles played by Busy and Request (REQ) bits, which are both located in the access control field of each slot.

The Busy bit indicates whether the slot is full or empty. When a node, by means of the ATC-DATA request service primitives [IEEE802.6], puts a segment on the Distributed Queue for (for example) bus A or forward bus, the node itself sets a REQ bit on the control part of a slot traveling on bus B (or reverse bus). The REQ bit informs the upstream nodes (upstream being defined in relation to the flow on the forward bus) that an additional asynchronous segment occupies a position in the Distributed Queue. Each node, by counting the number of REQs it receives, can determine the number of segments queued ahead of it. This operation can be achieved by means of the request (RQ) counter, which is increased by one for each REQ passing on bus B. On the other hand, the RQ counter is decreased by one each time an empty slot passes on bus A. This empty slot will be used by one of the downstream queued segments. Hence the RQ counter keeps a precise record of the number of segments queued downstream.

Operation of the Distributed Queue State Machine

Figure 2.2 describes the DQSM states and transitions. There is a DQSM for each node and bus (A and B). The DQSM for access to bus x (x=A or B) can be in one of the following states: Idle, Countdown and Standby. The other bus is defined as bus y (y=B or A, respectively).

The DQSM is in the **Idle** state when it has no segments to be transferred. While in this state, the DQSM maintains the RQ counter according to transitions 1 and 2, shown in Figure 2.2. The DQSM remains in the Idle state until it receives an ATC-DATA request, requesting transfer of a segment on bus x. The next state to be entered is decided by the DQSM according to the result of a check performed on the RQ counter. Then:

- If the content of the RQ counter is greater than zero, the DQSM transfers the current value of the RQ counter into the countdown (CD) counter, clears the RQ counter, sends a REQ on bus y and enters the Countdown state. The CD counter maintains the number of segments downstream which were queued for access to the bus x before the segment from the node underway was queued.

- If the content of the RQ counter equals zero than the DQSM enters the Standby state.

The DQSM is in the **Countdown** state when it has a segment queued for transfer on bus x, but is not yet permitted to access empty slots passing on bus x as the RQ counter reads different from zero. The DQSM remains in this state until the segment which is queued for sending is written into an empty slot (transition 7). The purpose of the Countdown state is to allow those segments previously queued to access the bus first. To achieve this, the node decreases the CD counter value by one for each empty slot that passes on bus x (transition 5). These empty slots are used by those segments previously queued downstream. The new REQs received during the Countdown state go to increment the RQ counter (transition 5). These REQs do not affect the CD counter, since they arrive after the segment is queued. Moreover, the empty slots passing by the node while it is in this state only decrease the CD counter value, and not the RQ counter value, since the empty slots are serving the prior REQs. When the content of the CD counter equals zero, the node waits for the first empty slot on bus x to transmit its queued segment. Once the segment is transmitted on bus x the Idle state is entered again (transition 7).

The DQSM is in the **Standby** state if, at the time the segment is ready for transmission, RQ counter value equals zero and there is no segment queued for access to bus x at the node. A zero RQ counter value implies that there are no nodes downstream that have segments queued, and therefore, the bus utilization is likely to be low. The DQSM remains in this state while waiting for the next slot to arrive on bus x. If the slot is empty, then the segment which is queued for sending is written into the slot and the DQSM returns to the Idle state (transition 8) without sending a REQ. If the next slot is Busy, then the DQSM sends a REQ on bus y and moves to the Countdown state (transition 9). Whenever the DQSM reads a REQ on bus y, the DQSM increments the RQ counter by one, sends a REQ on bus y and, finally, moves to the Countdown state (transition 10).

3.0 DQDB Queueing Model

In this section, a stochastic queueing model (Figure 3.1) of a DQDB MAN is described. Each unidirectional bus is represented by a sequence of servers {Delay(i,i+1)}.

Each server models the signal propagation delay between a consecutive pair of nodes. Slots traveling on each bus are generated by a source according to a deterministic distribution. The interarrival time between slots is equal to the slot length. Each slot is absorbed by a sink once it reaches the end of the bus. For the proper modelling of the slot propagation delay, the Infinite Server (IS) policy was adopted for each server. Each node of the DQDB MAN is modeled - for each transmission bus - by the following components:

- a DQDB MAC;
- a Local User Queue;
- a Traffic Generator.

The DQDB MAC model is not described in this paper. Details on it can be found in [CONTI89]. The Local User Queue models the queue where a node maintains segments which cannot be placed in the Distributed Queue (see Section 2). The traffic generator models the user packet interarrival time. In our simulation experiments, a user packet coincides with a segment (i.e., there is no need to break down a packet into segments). In the following sections, *segment* and a *packet* can thus be used as synonyms.

Segment transmission on each bus is controlled through an interaction between the DQDB MAC protocol and the slot variables (Busy, REQ). In the model, a segment transmission is performed by setting Busy=1 on a free slot while the segment itself is absorbed by a sink.

4.0 Workload and Behaviour Characterization

4.1 Workload Characterization

The following parameters:

- Network Utilization (ro);
- Workload Type;
- Traffic Types,

define the conditions under which the DQDB queueing model (Section 3) was solved.

Network Utilization

Network utilization defines the bandwidth portion which is used to transmit traffic generated by users. In our simulation experiments, no overhead coming from the control part of a slot was taken into consideration. Therefore, Busy slots and used bandwidth coincide. In the following sections, any given value of ro refers to both buses.

Workload Type

The workload type defines how the nodes contribute to network utilization. The workload type for each node has been split up into two components: one which is kept equal for all nodes and one which is dependent upon the node index (the rationale behind this choice is that in an internetworking environment the fixed component represents the internet traffic, while the other represents the intranet traffic). The policies selected for the latter component for the simulation experiments have been symmetric (S), asymmetric (AS), and uniform (EQ). In all policies, the traffic generated by each node is the same. The differences between the policies is in the way destination addresses (of segments) are generated.

In workload type S, the space containing the destination addresses is uniformly distributed among the nodes. In other words, the probability that a node transmits a segment to any other node is constant. As a consequence, the probability that a node transmits a segment on a given bus is proportional to the number of the downstream nodes.

Workload type AS can be derived from workload type S by swapping the traffic of the two buses. The probability that a node transmits a segment on a given bus is then proportional to the number of the upstream nodes.

In workload type EQ, each node transmits the same average number of segments on each bus.

Traffic Types

Within an integrated service environment, such as a DQDB MAN environment, users might generate highly different types of traffic which, for our purposes, can be categorized into real-time and non real-time. Here are two meaningful examples:

- non real-time: file transfer which might require transfer rates up to 10 Mbit/sec.
- real-time: high-quality sound and digital moving pictures that will need transfer rates ranging from 64 Kbit/sec to some 20 Mbit/sec or even 100 Mbps.

In our model, the various traffic characteristics have been represented by two (one for each bus) traffic generators in each node. More precisely, the generators (GA and GB) represent the interarrival times of segments arriving to the Local User queues. The real-time traffic considered here is mainly composed by packetized voice and video; the latter normally requires transmission of streams of contiguous packets (bursts). Due to the lack of statistics on interarrival times of packets generated by a variable bit rate (VBR), video codecs [Torino 88], the global traffic has been approximated by an exponential (coefficient of variation, C, equal to 1), or a hyperexponential distribution of interarrival times (C=2 and C=4). Bursts have been approximated by means of messages of 1, 4 and 8 packets.

It must be underlined that most of the characteristics of the new services are not yet well understood (for example, variable bitrate coding for video services [TORINO88])

4.2 Behaviour characterization

Performance Indices

In order to analyze the ability of the DQDB MAN to integrate services (data, voice and video), several performance indices have been identified.

For non-real time traffic (data), it is sufficient to guarantee that the average end-to-end delay will not exceed (for example) 500 msec. [Terada88]. Segments lost due to buffer overflow (for example) can be retransmitted end-to-end. For this type of traffic, a trade-off between the workload and the average access delay is required.

On the other hand, we have real-time traffic whose major requirement is a constant user-to-user delay. In conventional circuit-switching networks, once the bandwidth is allocated, constant end-to-end delay implicitly follows; this is not the case in the MAN environment. In fact, in a MAN, the end-to-end segment delay D (i.e., the delay that a segment experiences from source to destination nodes) can be broken down into several components:

$$D=\delta acc + \delta prop$$

where:

δacc is the bus access delay;
$\delta prop$ is the propagation delay.

The bus access delay depends on the DQDB MAC protocol.

The propagation delay does not depend on traffic conditions. Hence it gives a constant contribution to the end-to-end delay that is easily evaluated in terms of the end-to-end distance.

The bus access delay is obviously variable and a compensation is therefore required to guarantee constant user-to-user delay. A possibility is to defer the delivery of the first segment by a quantity of time high enough to ensure that a synchronous receiver is not exposed to the variability of the end-to-end delay (jitter). This induces an additional variable delay, δsmooth, before every packet is delivered. Hence the user-to-user delay is:

δacc+ δprop+ δsmooth

The additional delay δsmooth can be computed knowing that some real-time applications are not sensitive to a given segment loss percentage, n (for example, voice channels can tolerate a 1% segment loss). We can use this peculiarity in order to evaluate a minimum value of this delay, guaranteeing that (100-n)% of segments will be correctly received. The loss of segments is due both to excessive delay and buffer overflow.

Fairness Characterization

Metrics must be chosen to represent fairness. For DQDB, we shall restrict our analysis to the two simplest metrics that have found much favour in previous work [......] :

- the node average access delay (defined as the weighted sum of the access delays of bus x and bus y) in normal operation;
- the average bandwidth sharing among nodes in overload condition.

This study focuses on the DQDB behaviour in stationary conditions. Since, in our simulation experiments, each node generates the same traffic (see above), it follows that in a fully fair system, each node should have the same node average access delay.

5.0 DQDB Performance Evaluation Analysis

The DQDB performance analysis was performed by means of the RESQ2 [SAUER82] simulation tool with the assumptions reported in the following table:

- Nodes were spaced equally along the two buses;
- Capacity of each bus = 150 Mbit/sec;
- Slot size = 1000 bits;
- Length of each bus = 100Km;
- Propagation delay = .4msec

- Total number of nodes = 50

Table 5.1: Network Configuration Parameters

Since many factors influence the DQDB performance, it was decided to divide the experiments into three classes:

1 - behaviour analysis under various workload types (Section 5.1):
- workload types: S, AS and EQ
- C=1, Msg=1, ro=.8

2 - behaviour analysis under various traffic types (Section 5.2):
2.1- Msg=1, 4 and 8
C=1, ro=.8, workload type=S
2.2- C=1, 2 and 3
Msg=1, ro=.8, workload type=S

3 - behaviour analysis under different network utilizations (Section 5.3)
ro=.8, .9 and .95
Msg=1, C=1, workload type=S

Results are reported in several figures, labelled with the respective workload characterizations.

5.1 Behaviour Under Various Workload Types

Figure 5.1 shows the DQDB node average access delay E(Tq) as a function of the node index for the S, AS and EQ workload types. The figure shows that the DQDB unfairness strongly depends on the specific workload type. In order to understand such a behaviour, Figure 5.2 must be examined. Figure 5.2 plots the bus A average access delay as ordinate against the node index as abscissa. As can be seen from this figure, the same type of behaviour is exhibited by all workload types. Moreover, the slope of each curve tends to increase as the node index approaches 50 (i.e., the end node). This observation plus the definition of the node access delay justify the shapes of the three curves shown in Figure 5.1. The curves for workload types AS and EQ (see Figures 5.4 and 5.5, respectively) exhibit the same behaviour for the nodes downstream of the 25th.

Figures 5.3, 5.4 and 5.5 show the distribution functions related to the total number (Local User&Distributed Queue length) of segments waiting for transmission on each bus, with the understanding that node i has, for bus A, the same statistics as node 50-(i+1) has for bus B. It is interesting to note that for all workload types the 99th percentile never exceeds 4. Moreover, the probability is fairly high that an arriving segment finds an empty queue.

This is consistent with the high probability of finding the DQSM in the Standby state (see below).

The performance results reported above in Figures 5.1-5.5, are those experienced by the DQDB protocol user. In the following discussion these performance indices are explained in terms of DQDB protocol mechanisms. This is done with the aim of singling out those protocol mechanisms which are responsible for the DQDB unfairness.

Figure 5.6 shows the CD mean value vs. node index for the three workload types already defined. The main information gleaned from the figure is that the maximum value for the three curves is always lower than one. The station for which this maximum is achieved depends on the particular workload type. These low values might appear unreasonable for ro=.80. In order to justify these results, Figure 5.7 - showing the probability that a given node transmits a segment while in the Standby state - must be analyzed. As can be noted, this probability is still high at ro=.8, and this means that the number of REQs sent by a node is low compared to the number of segments transmitted.

Figure 5.8 shows the mean waiting time (measured in consecutive Busy slots) experienced by a segment when the CD counter is cleared (from now on, referred to as E(Busy) vs. the node index. This figure shows that the number of consecutive Busy slots increases with the node index. More precisely, the curve is a stright line before turning rapidly at the *knee,* which is located around the 30th node. This behaviour leads to the conclusion that correlations between Busy slots (which are observed as trains of consecutive Busy slots) are almost absent in the first 20 nodes, and then tend to be established when the End node is approached. To get some intuitive insight into the relationship between correlation and node index, we should note that the bus access time is influenced by:

- the value of the CD counter, which is always less than one (as shown in Figure 5.6);
- the number of consecutive Busy slots (seen by the node once the CD counter gets to zero), which is greater than one for nodes ahead of the 20th and reaches a value of 4-6 for nodes close to the End node.

From the discussion above it seems, then, reasonable to draw the following general conclusion for a DQDB protocol exercised with the S, ES, and EQ workload types and ro=0.8: the major factor influencing the DQDB unfairness is the correlation between Busy slots.

5.2 Behaviour Analysis Under Various Traffic Types

5.2.1 Influence of Message Length

Figure 5.2.1 shows the DQDB node average access delay E(Tq) as a function of message length. The figure shows that the DQDB unfairness strongly increases with message length. Figure 5.2.2 plots the bus A average access delay as ordinate against the node index as abscissa for message length=1, 4 and 8. The increments in the bus average access delay are due to the fact that DQDB allows for only one packet in the distributed queue.

Figure 5.2.3 shows the CD mean value vs node index for different message lengths. The main information visible from the figure is that E(CD) changes significantly when passing from Msg=1 to Msg=4, while it remains almost the same for further increases.

Figure 5.2.4 shows the mean value of the number of consecutive busy slots vs. node index for different message lengths. The behaviour exhibited by the three curves can be explained by the fact that for Msg=4 and beyond, the request mechanism tends to reach a stationary state where each station having packets to transmit is allowed to use one slot every N slots (N being the number of active stations). This is due to the fact that, although the effect of the requests depends on their propagation times, time shifts are not meaningfull in stationary conditions. From this observation, we can also justify the results shown in Figure 5.2.3 for Msg=4 and Msg=8.

5.2.2 Influence of Interarrival Distribution

Figure 5.2.5 depicts the DQDB node average access delay E(Tq) as a function of the interarrival distribution. The figure shows that incrementing the C parameter makes DQDB more unfair. Figure 5.2.6 plots the bus A average access delay as ordinate against the node index as abscissa for C=1, 2 and 3.

Each of Figures 5.2.7 , 5.2.8 and 5.2.9 shows the CD mean value and the mean values plus standard deviations vs. node index for C=1, 2 and 3, respectively.

Each of Figures 5.2.10, 5.2.11 and 5.2.12 shows the mean value and the mean values plus standard deviations of the number of consecutive busy slots vs. node index for C=1, 2 and 3, respectively.

The main information to be inferred from these figures is that E(CD) changes significantly for all network nodes when passing from C=1 to C=3. The number of consecutive busy slots changes significantly only in nodes very close to the end and, in any case, the percentage increase is not very significant.

explained by analyzing Figure 6.2, which plots the bus A average access delay as ordinate against the node index as abscissa for ro=.8, .9 and .95. The irregularity of the DQDB E(Tq) can now be explained by means of the bump in the curve of the bus average access delay. The behaviour of the bus access delay can in turn be explained by examining Figures 6.3-6.8. Each of Figures 6.3, 6.4 and 6.5 shows the CD mean value and the mean value plus standard deviation vs. the node index for ro=.8, .9 and .95, respectively. Each of Figures 6.6, 6.7 and 6.8 shows the mean value and the mean value plus standard deviation of the number of consecutive busy slots vs. the node index for ro=.8, .9 and .95, respectively. It must be noted that the behaviour of the CD is much more regular than the behaviour of the E(Busy). In addition, the CD value cannot be neglected (as it is greater than or equal to one), except for the last two or three stations. On the other hand, the number of consecutive busy slots has a sharp increase in the last 10 nodes. The combined effect of these two mechanisms will result in the bumb shown in Figure 6.2.

7.0 Conclusions

This paper has focused specifically on the analysis of the DQDB protocol. The following conclusions can be drawn:

1- The average node access delay E(Tq) depends on a number of factors. DQDB is designed to manage one single segment at a time; this is the reason why messages made up of several segments introduce a significant degradation in the performance indices.

2- Although E(Tq) depends on node index and workload characterization, its values always seem to be adequate for managing real-time applications.

3- Moving from 150 Mbps to 1.2 Gbps results in a higher variability of E(Tq), although its absolute values decrease.

Until real-time traffic for ATM environments has been characterized more thoroughly, it is impossible to draw absolute conclusions about the suitability of DQDB for managing this type of traffic.

This study shows that DQDB is unfair, and this unfairness depends on the workload. This implies, for example, that each node should be sized according to the worst values of the performance indices.

In addition, DQDB may introduce a strong correlation among busy slots. This results in very long trains of busy slots, observed in particular by the

End station. This property is undesirable in an interconnected MAN environment where the End node plays the gateway role.

DQDB seems to perform much better than centralized MAC protocols such as FDDI [SCHI87] and FASNET [BOND88]. However, our analysis has shown that before drawing absolute conclusions, further research is needed.

REFERENCES

[BOND88] Bondavalli, A., Conti, M., Gregori, E., Lenzini, L., Strigini, L., "Medium Access Protocol for High-speed, Long-distance MANs: Performance Comparisons for a Family of FASNET-based Protocols", submitted for publication.

[HIRO88] Ohnishi, H., Okada, T., Noguchi, K., "Flow Control Schemes and Delay/ Loss Tradeoff in ATM Networks", IEEE JSAC, Vol. 6, No. 9, December 1988, pp. 1609-1616.

[IEEE802.6] IEEE802.6 Distributed Queue Dual Bus - Metropolitan Area Network - Draft Standard - Version D.0 - June 1988.

[SAUER82] Sauer, C., MacNair, E., " The Research Queueing Package Version 2: CMS User Guide", IBM Research Report, Yorktown 1982.

[SCHI87] Schill, A., Zieher, M., "Performance Analysis of the FDDI 100 Mbit/s Optical Token Ring", Proceedings on High Speed Local Area Networks, O. Spaniol and A. Danthine (Editors), Elsevier Science Publishers B.V. (North-Holland), IFIP, 1987, pp. 53-74.

[TERADA88] Terada, Y., "Trends in packetized Multimedia Communication Network Technologies in Japan", Proceedings on International Workshop on Packet Video, 8-9th September 1988, Torino, Italy.

[TORINO88] Proceedings on International Workshop on Packet Video, 8-9th September 1988, Torino, Italy.

[VORS88] Vorstermans, J. P., De Vleeschouwer, A. P., "Layered ATM Systems and Architectural Concepts for Subscribers' Premises Networks", IEEE JSAC, Vol. 6, No. 9, December 1988, pp. 1545-1554.

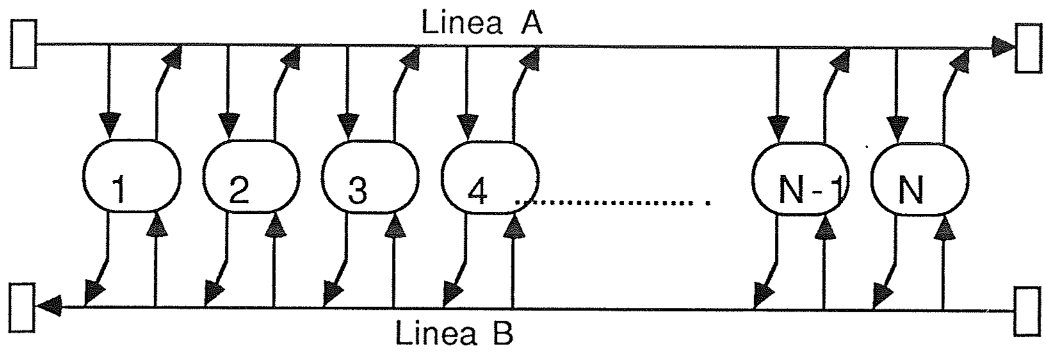

Figure 2.1

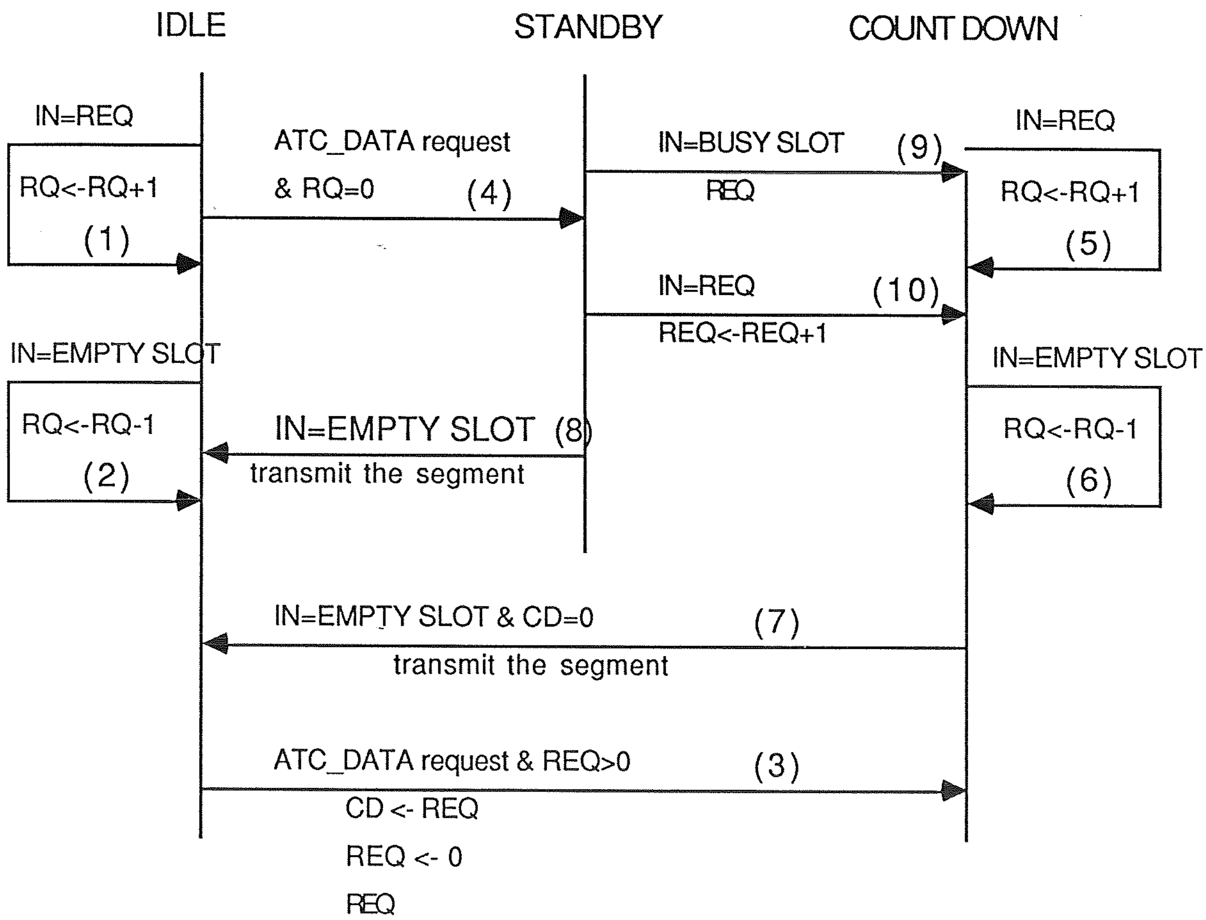

Figure 2.2

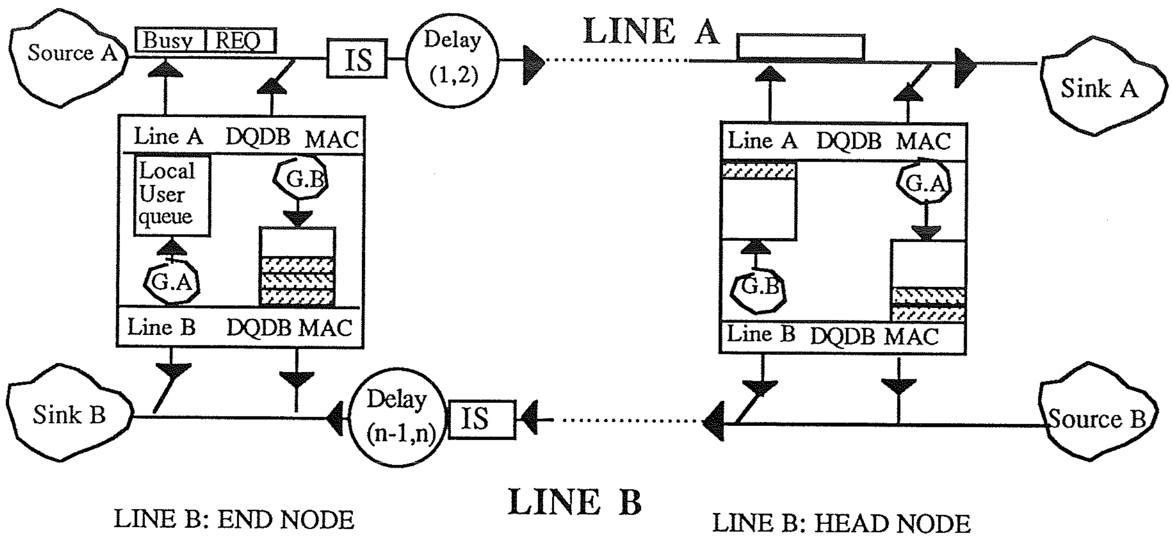

Figure 3.1

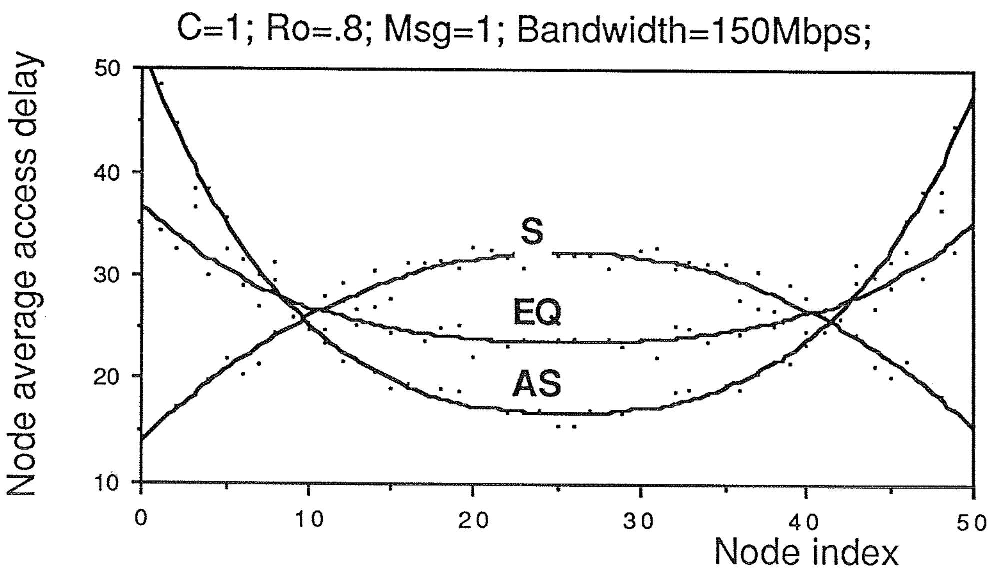

Figure 5.1

Figure -2

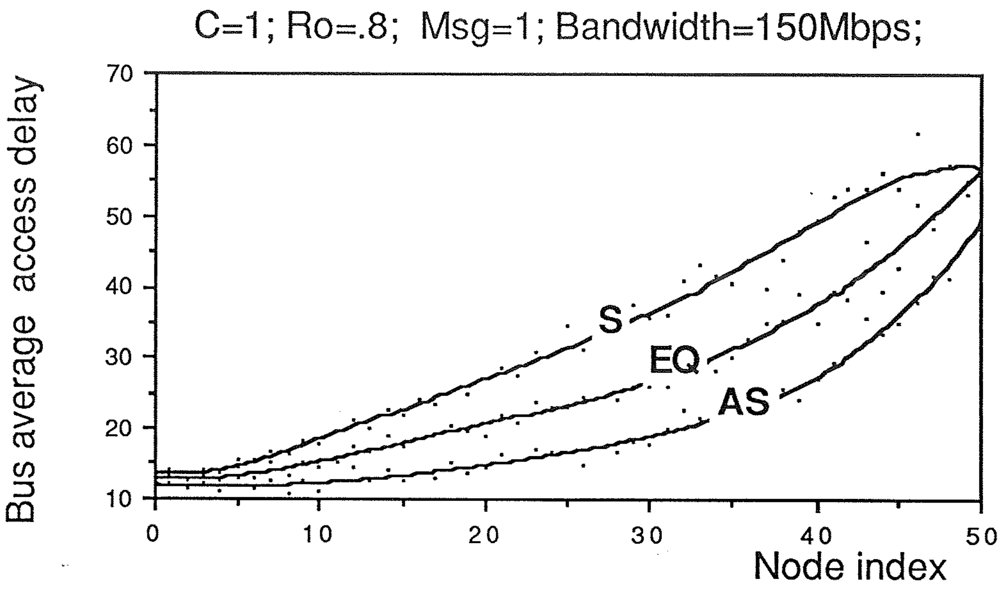
C=1; Ro=.8; Msg=1; Bandwidth=150Mbps;
Bus average access delay
70
60
50
40
30
20
10
0
10
20
30
40
50
Node index
S
EQ
AS

Figure 5.2

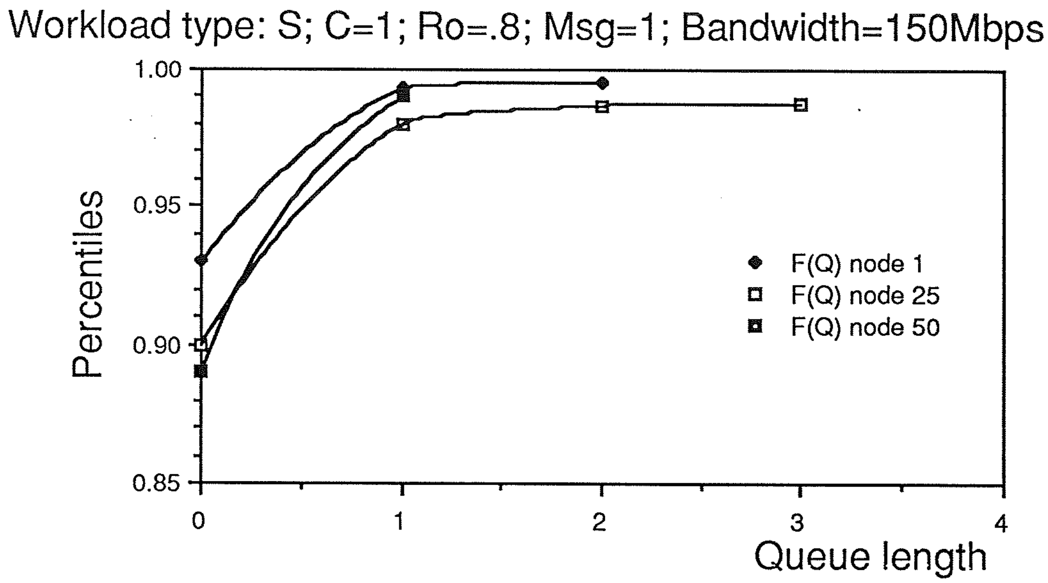

Figure 5.3

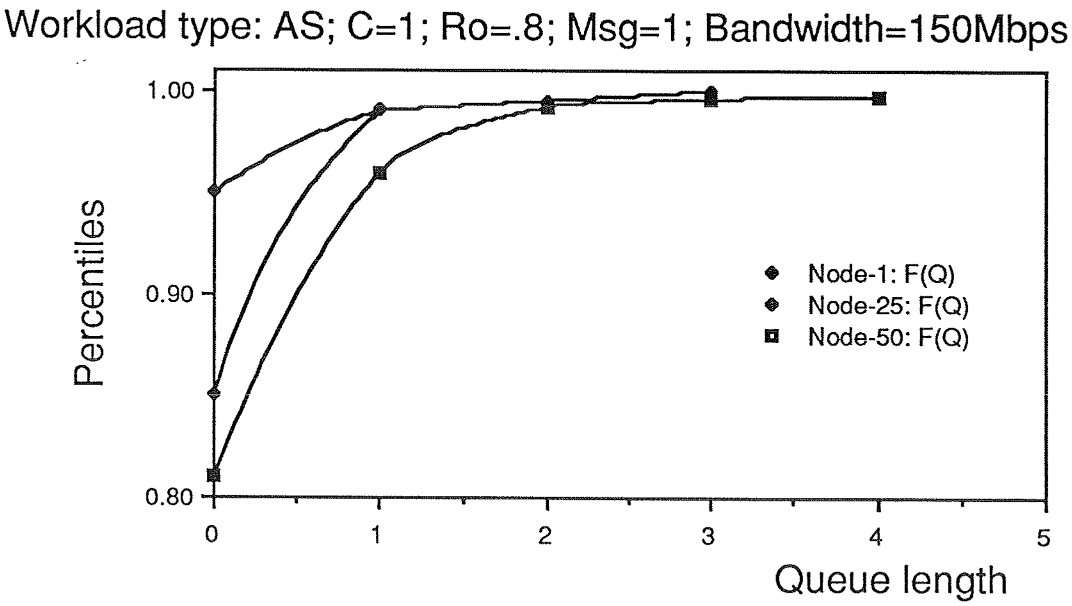

Figure 5.4

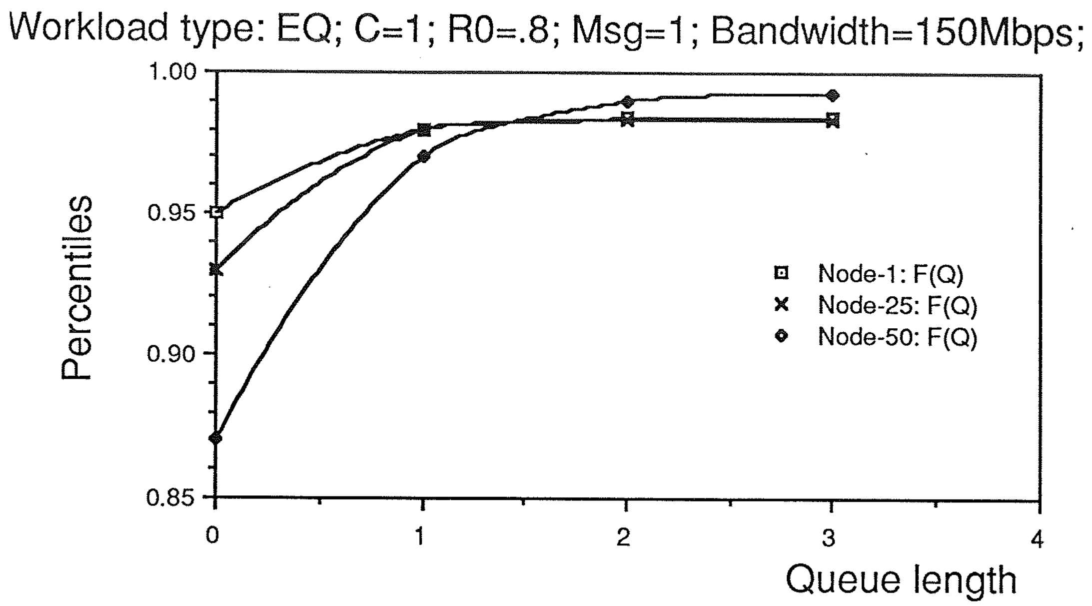

Figure 5.5

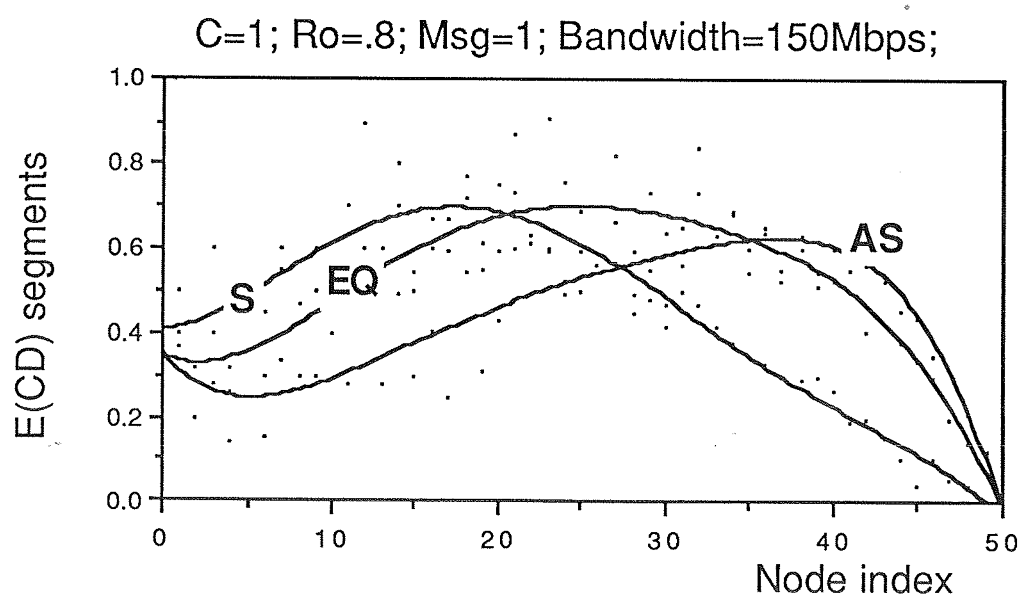

Figure 5.6

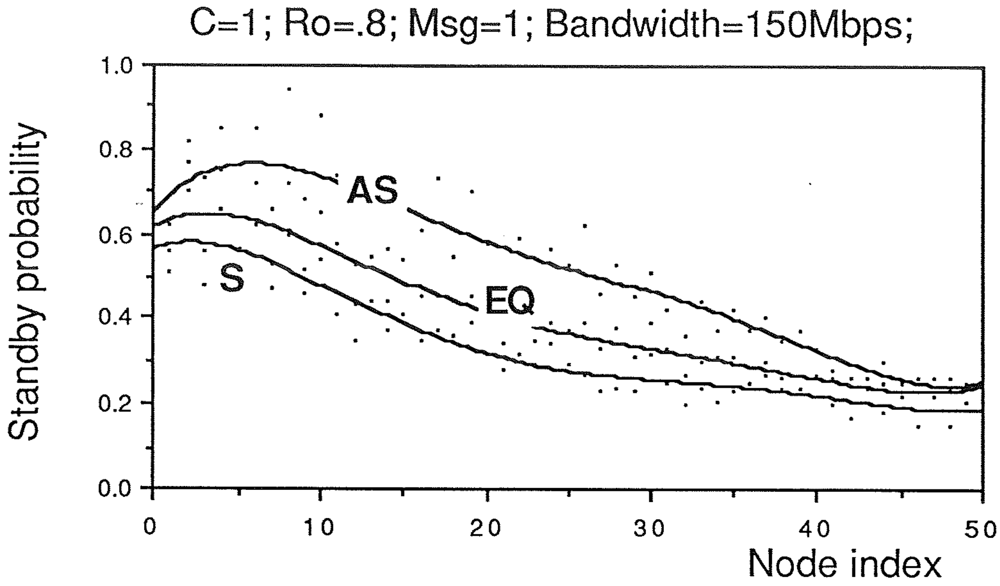

Figure 5.7

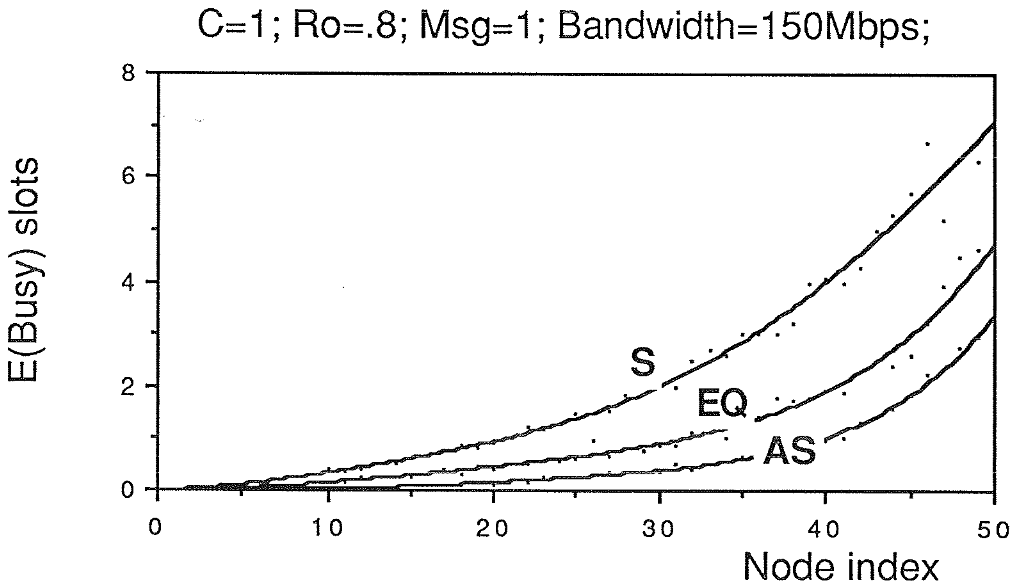

Figure 5.8

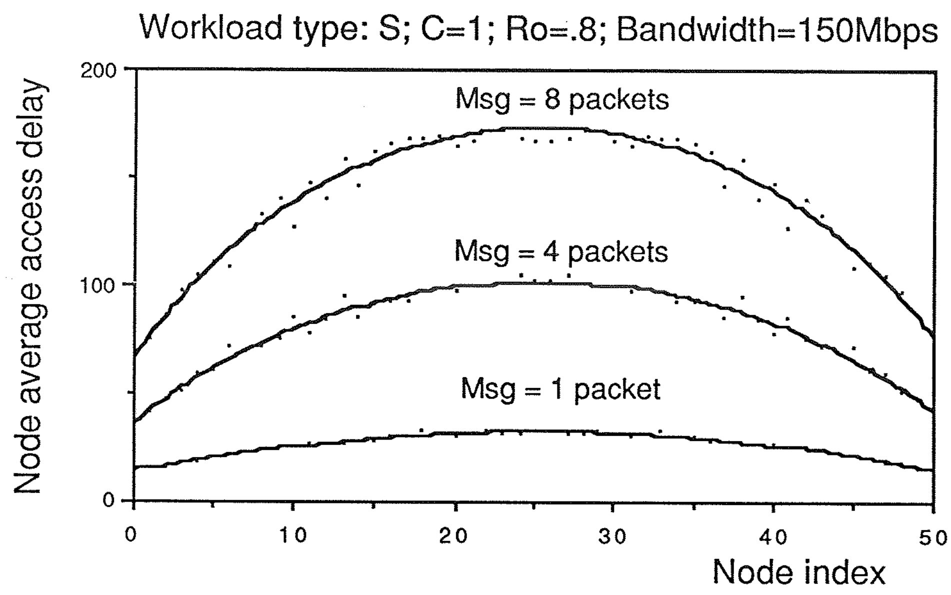

Figure 5.2.1

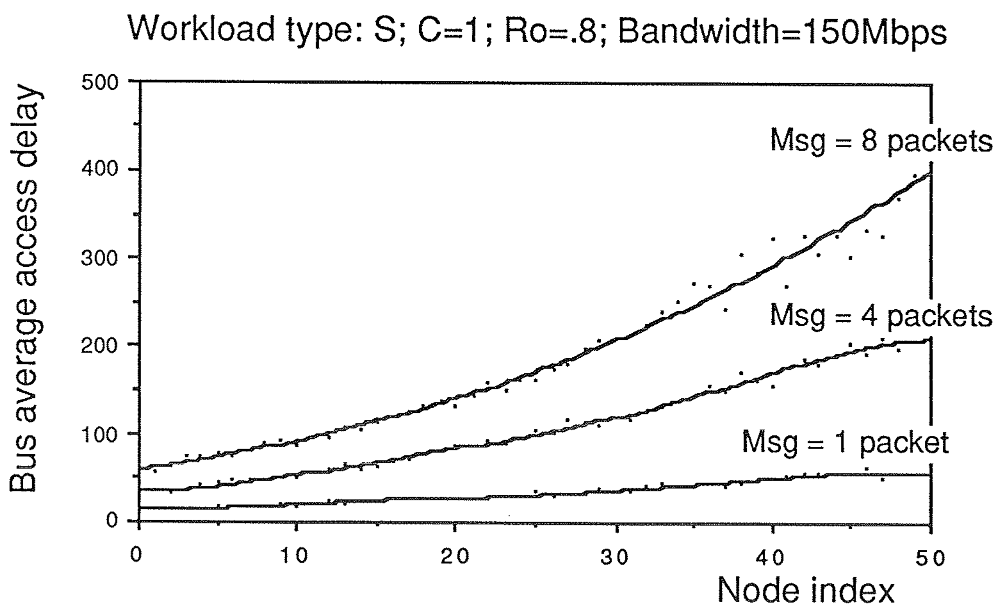

Figure 5.2.2

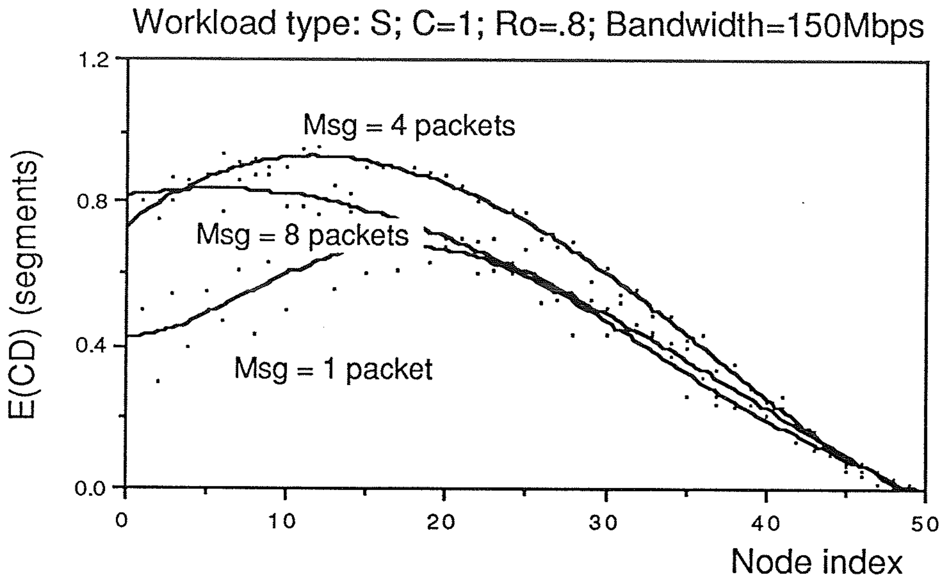

Figure 5.2.3

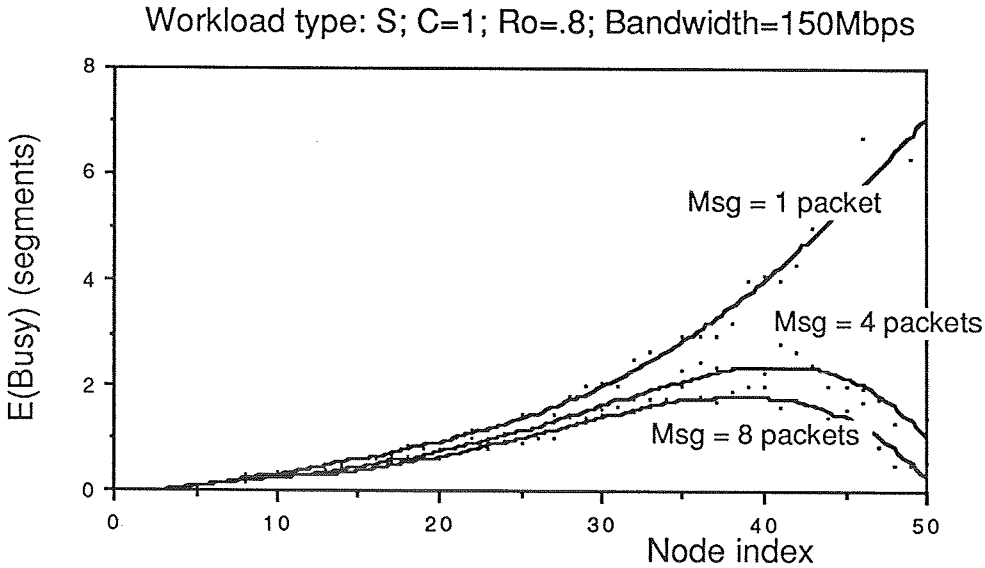

Figure 5.2.4

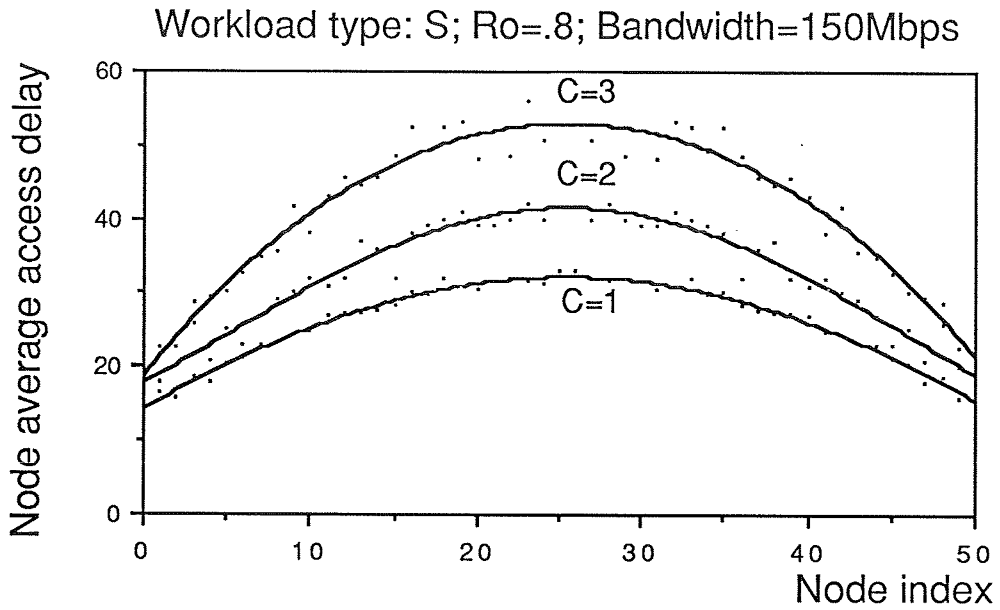

Figure 5.2.5

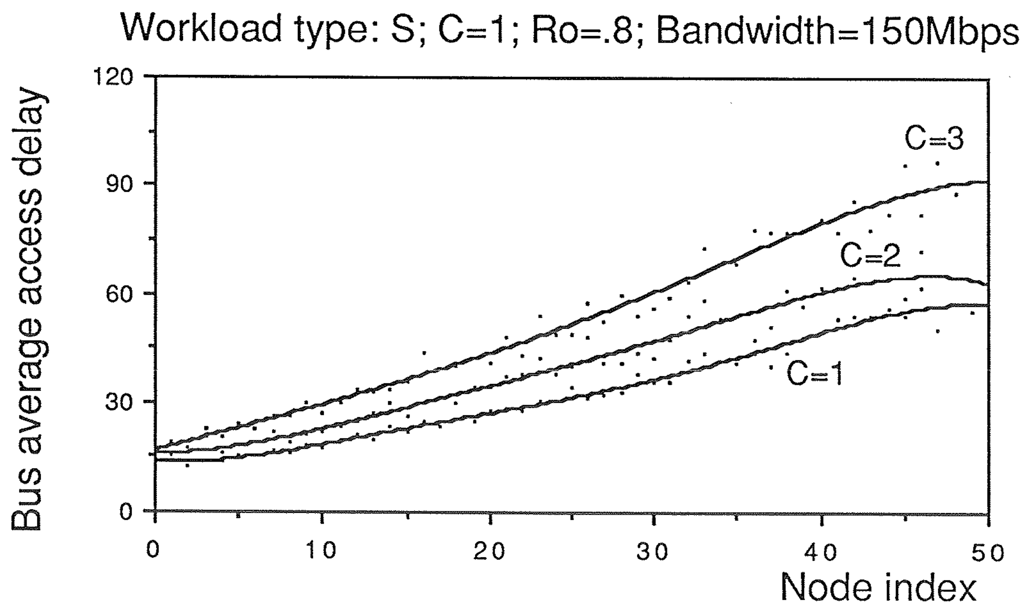

Figure 5.2.6

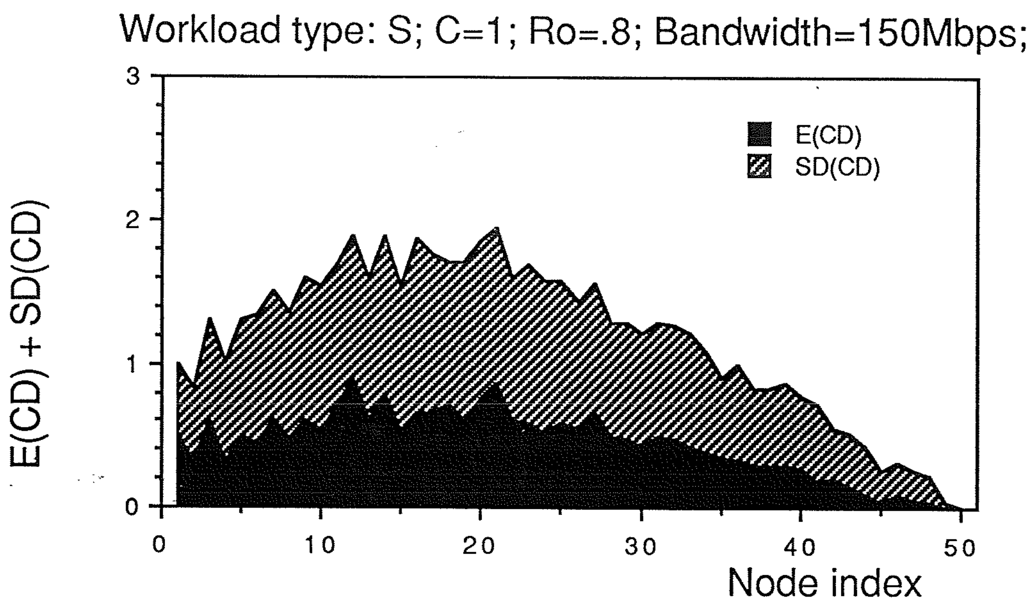

Figure 5.2.7

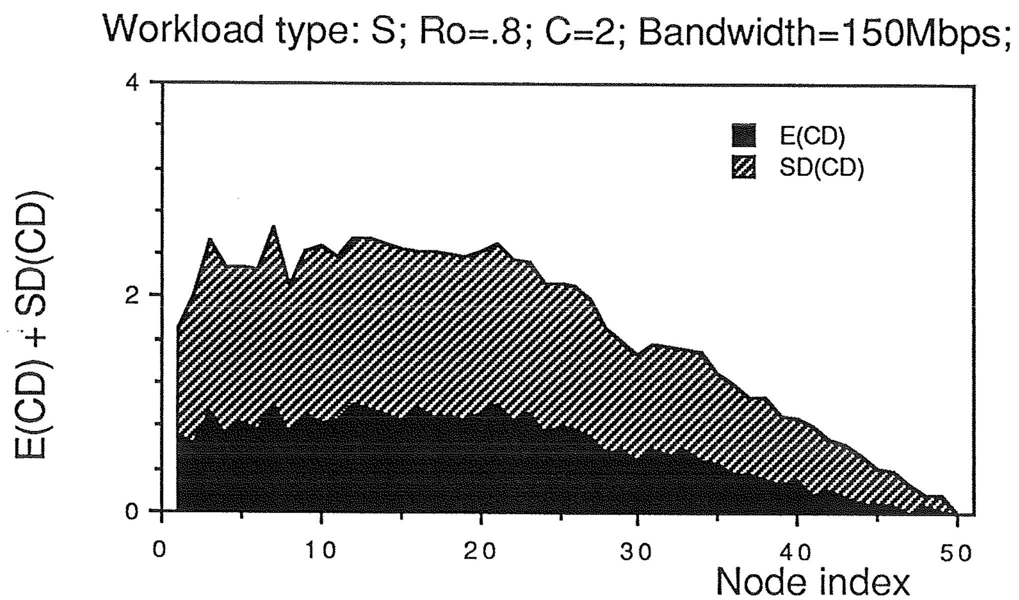

Figure 5.2.8

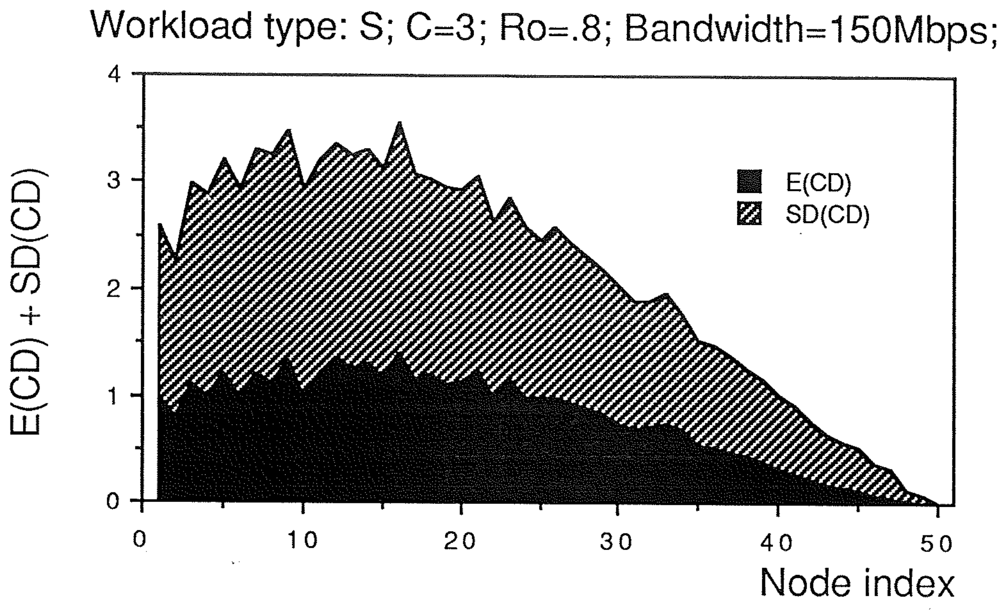

Figure 5.2.9

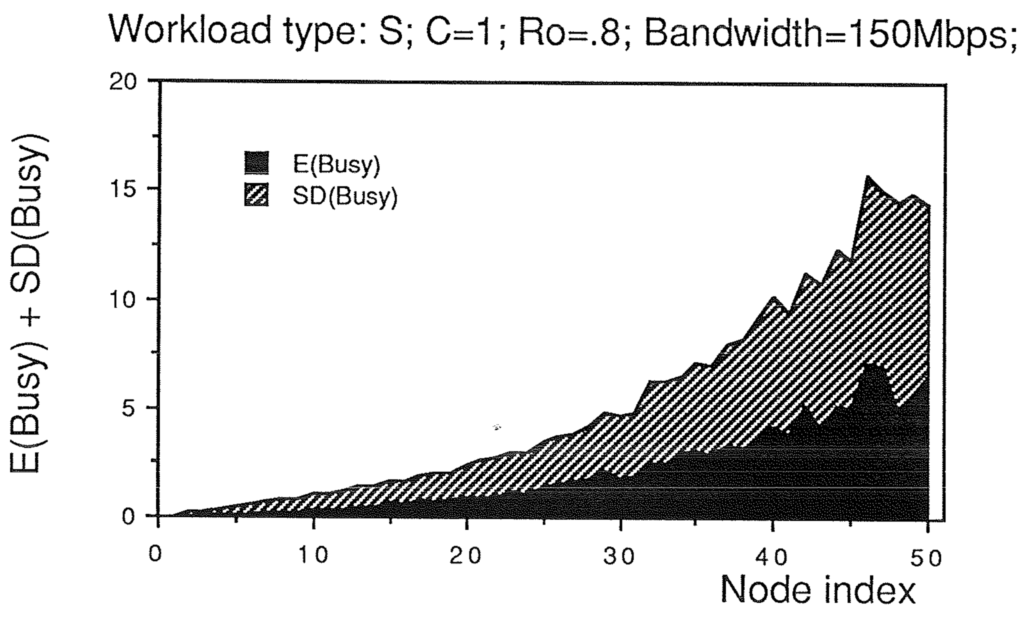

Figure 5.2.10

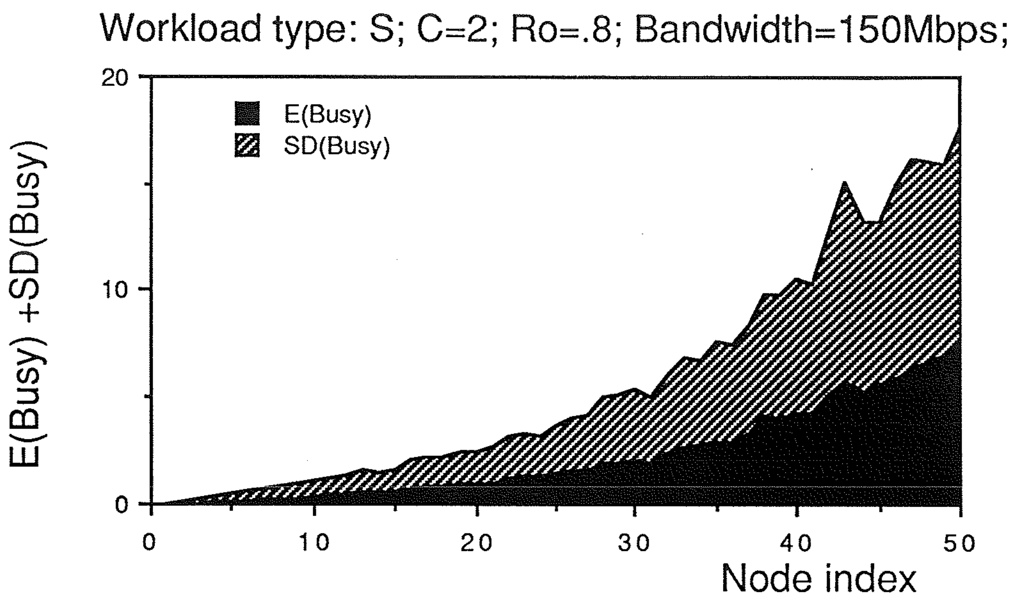

Figure 5.2.11

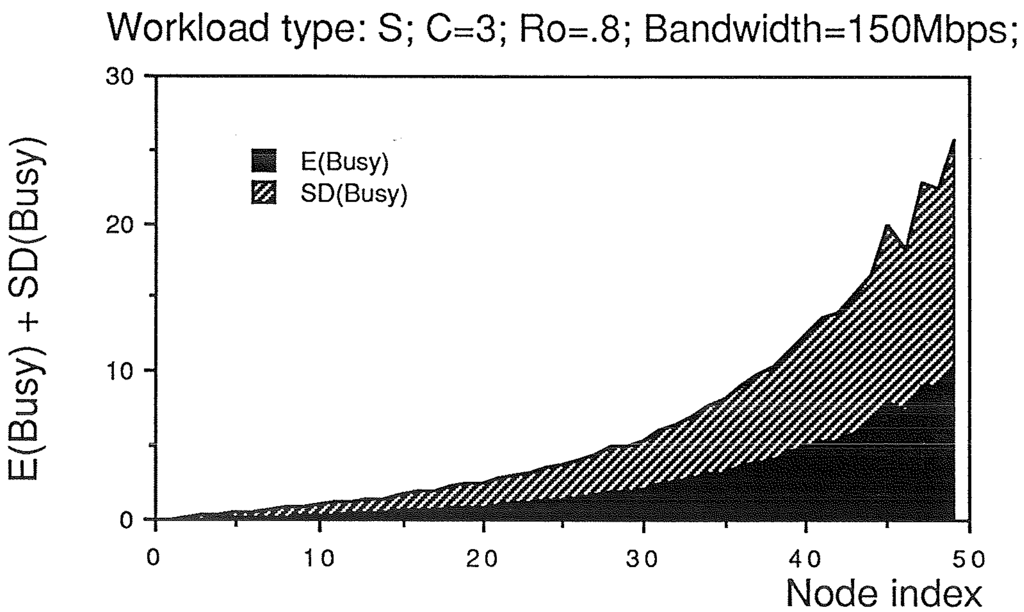

Figure 5.2.12

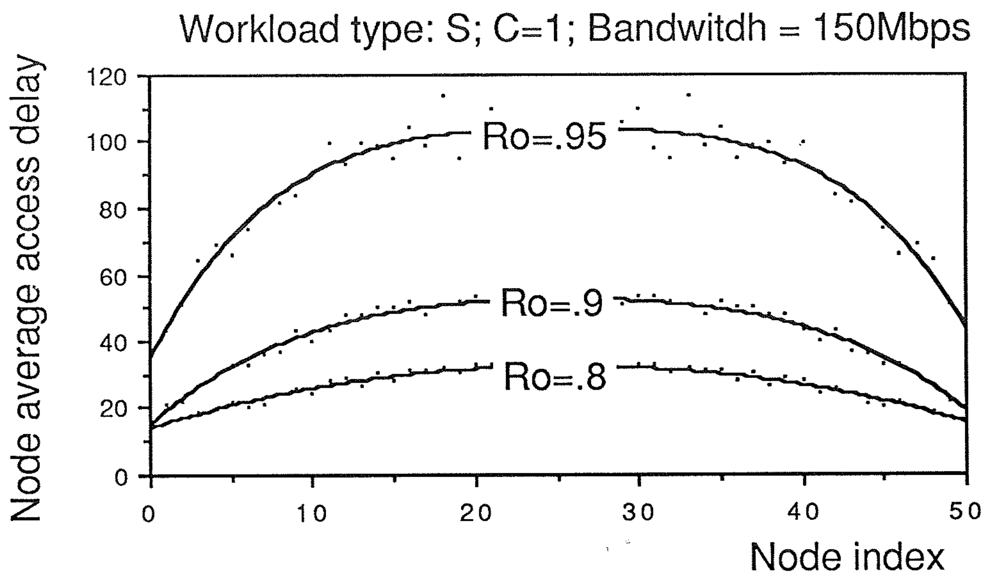

Figure 5.3.1

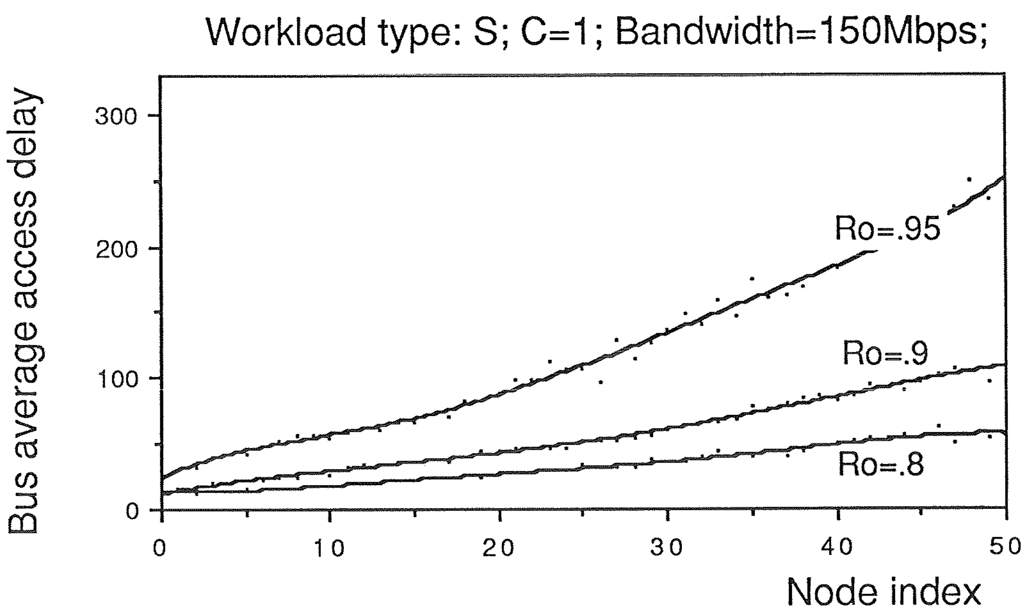

Figure 5.3.2

Workload type: S; C=1; Ro=.8; Bandwidth=150Mbps;

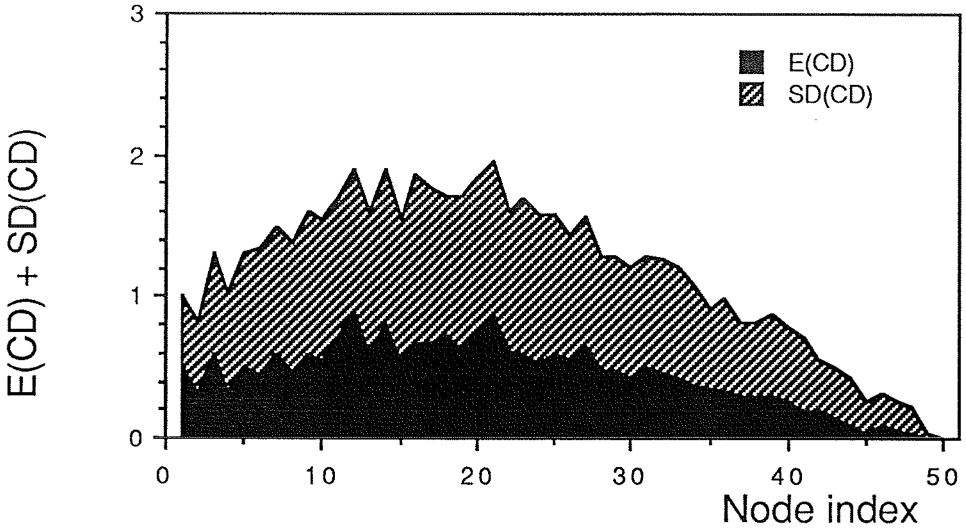

Figure 5.3.3

Workload type: S; C=1; Ro=.9; Bandwidth=150Mbps;

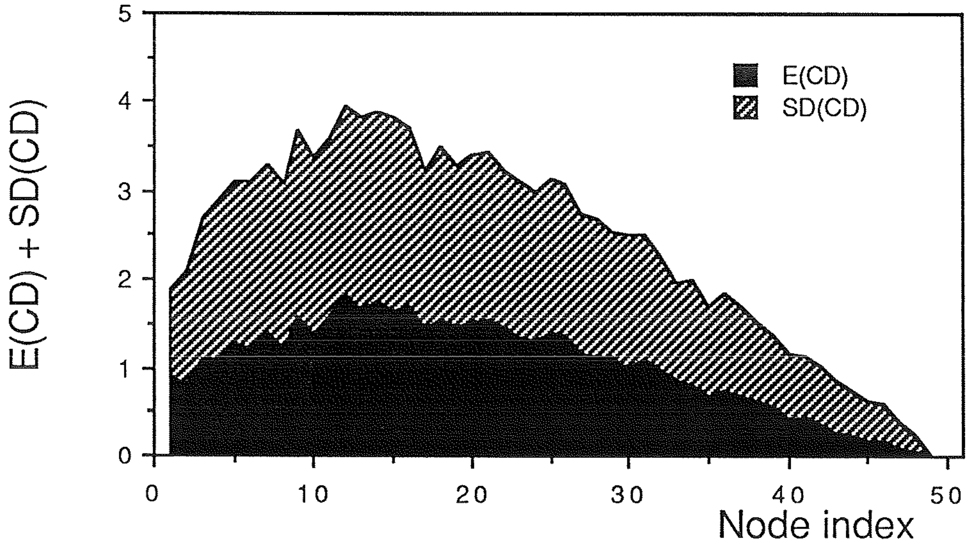

Figure 5.3.4

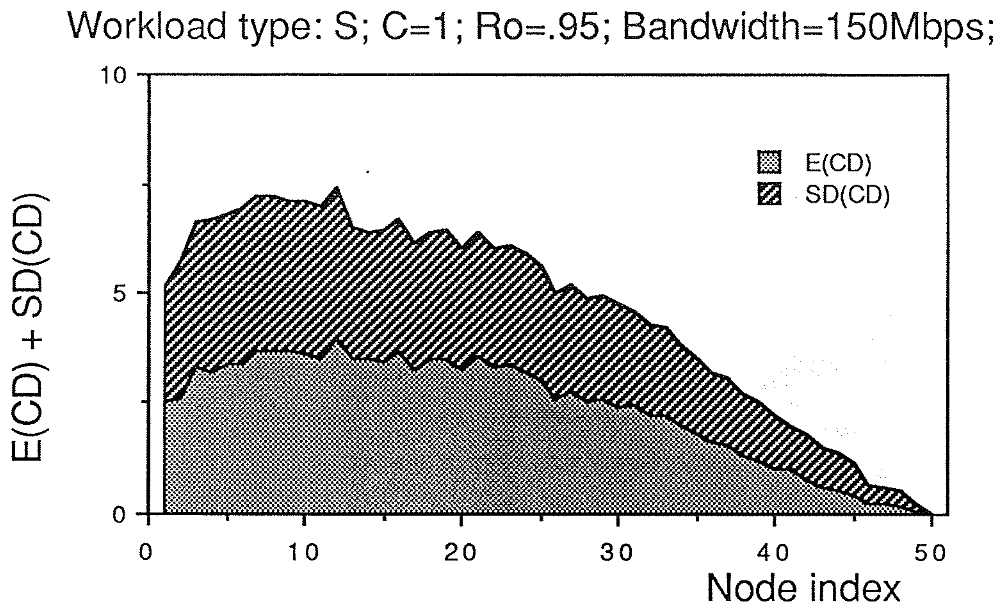

Figure 5.3.5

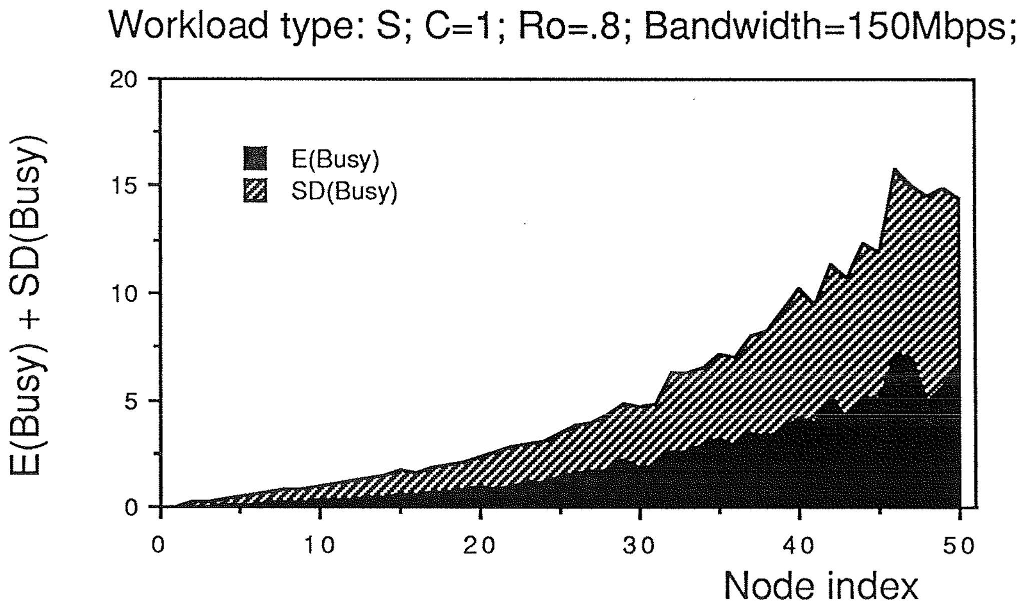

Figure 5.3.6

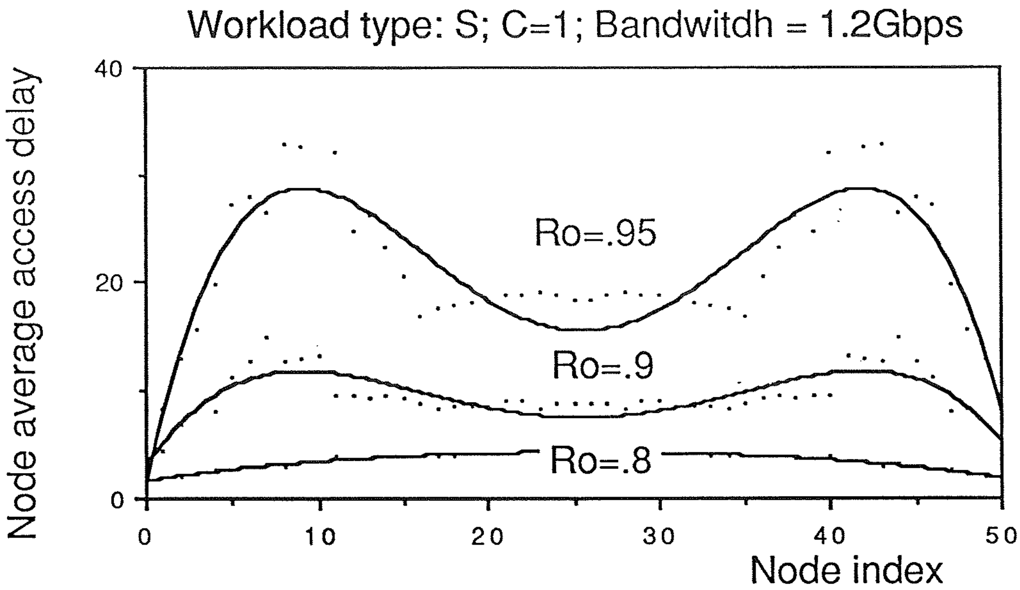

Figure 6.1

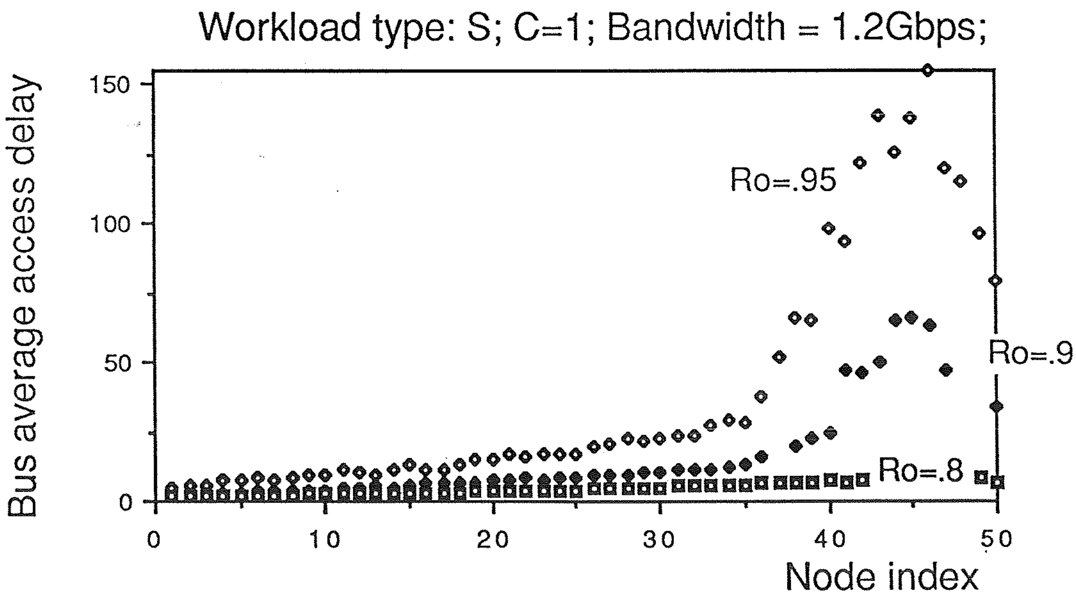

Figure 6.2

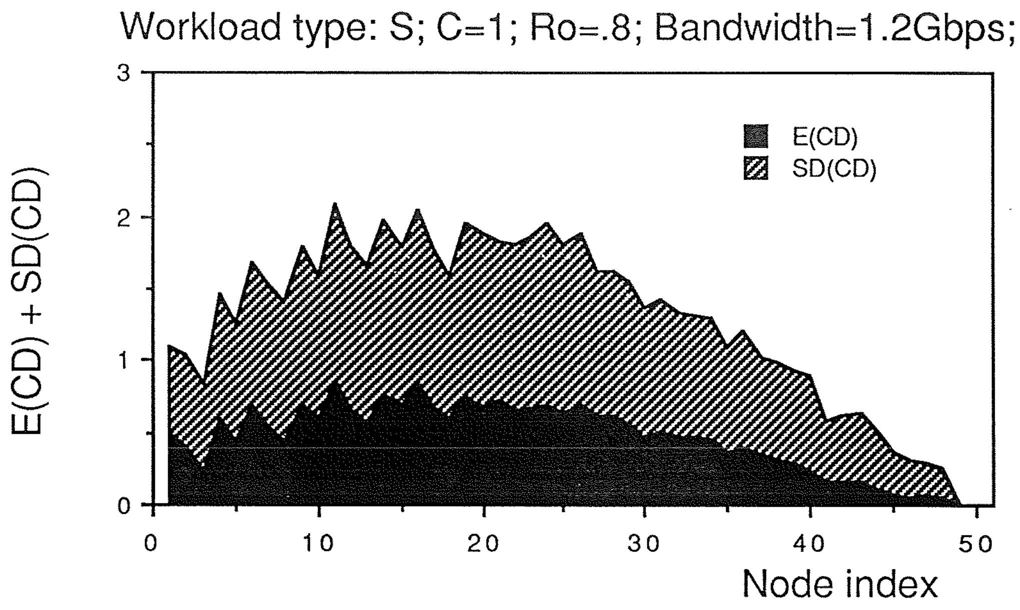

Figure 6.3

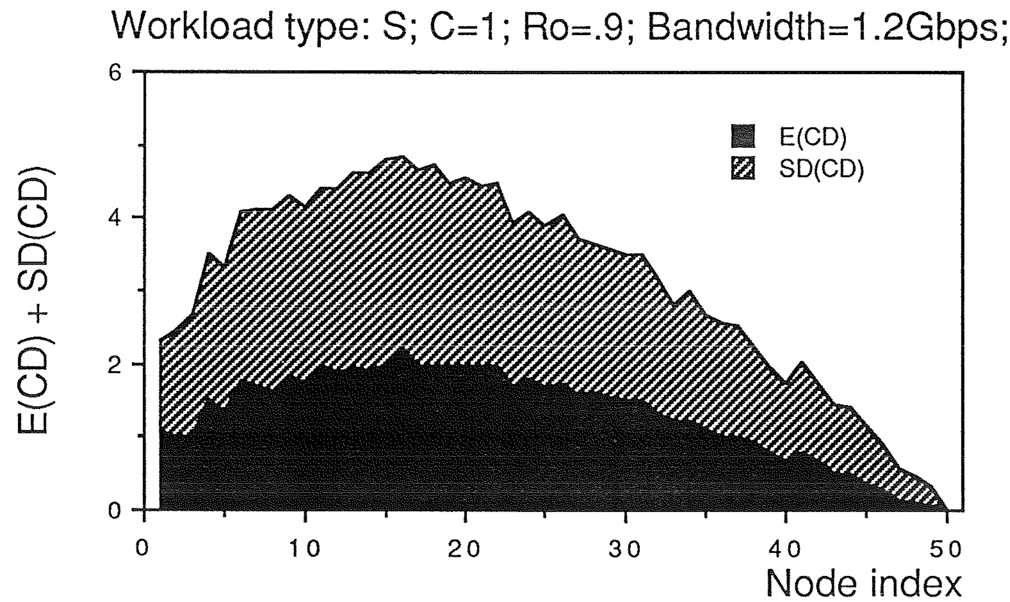

Figure 6.4

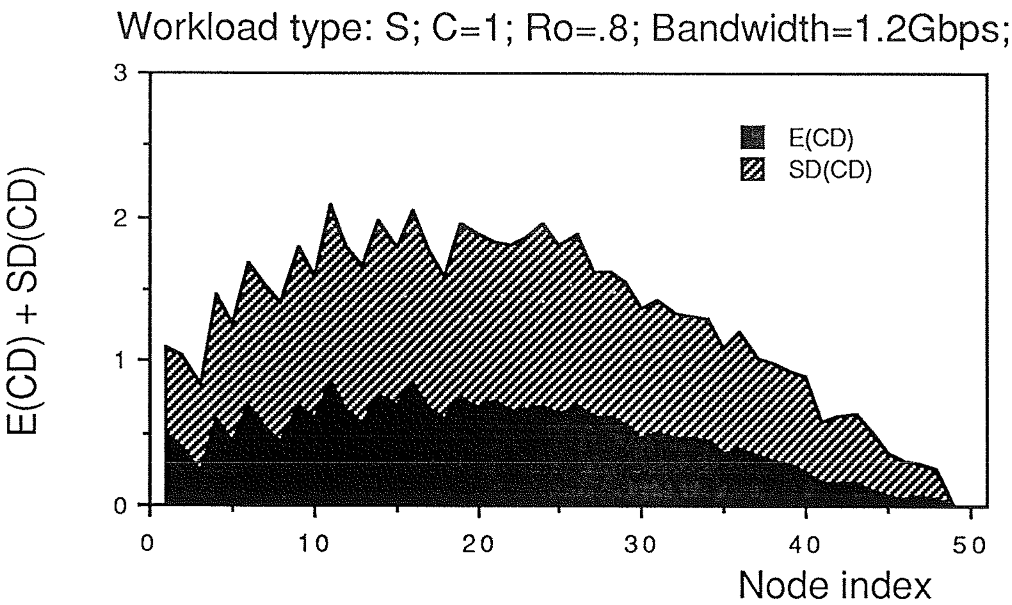

Figure 6.3

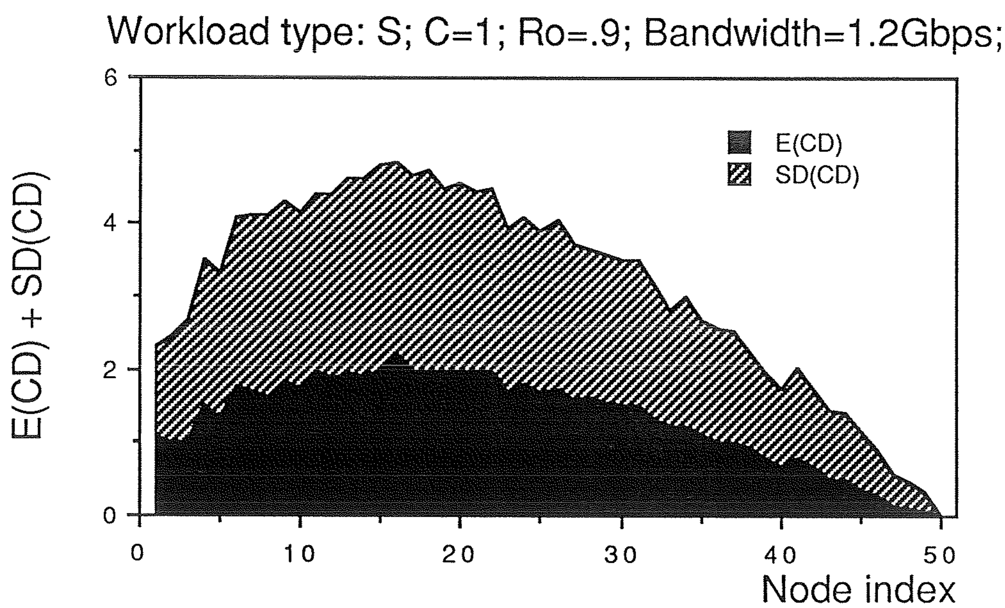

Figure 6.4

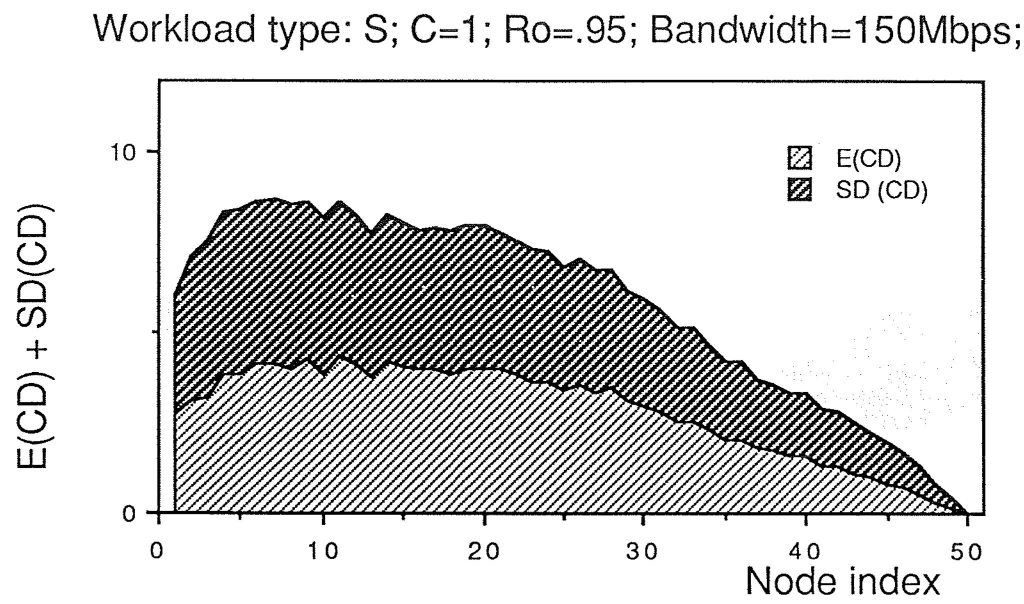

Figure 6.5

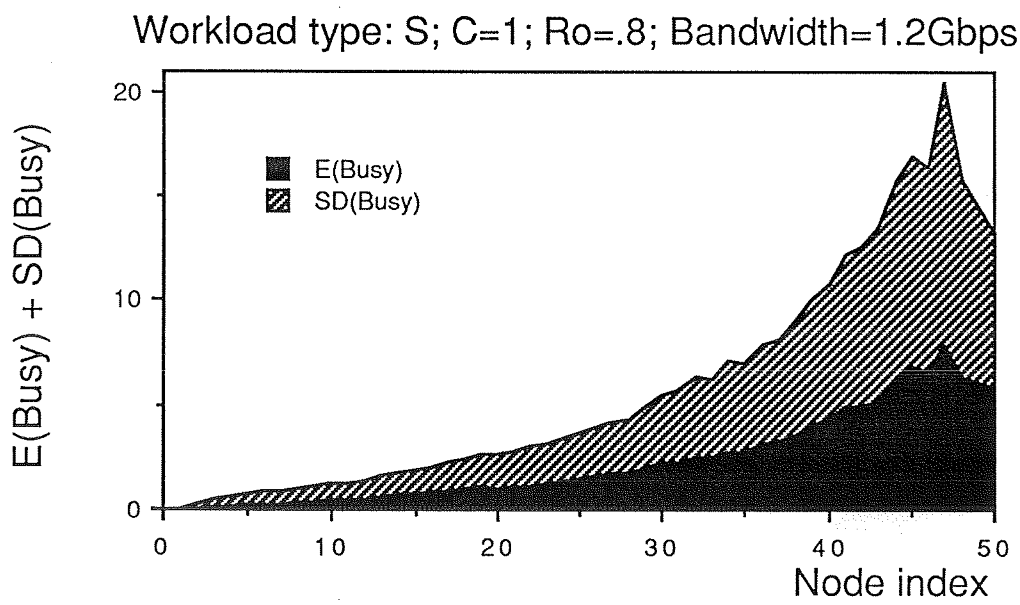

Figure 6.6