P-137

The TurboLAN Project

Final Report
Project Number NCC 2-621

Phase I: Protocol Choices for High Speed Local Area Networks

Phase II: TurboLAN Intelligent Network Adapter Card, (TINAC) Architecture

May 10, 1991

Principal Investigator

Hasan S. AlKhatib, Ph.D.
Associate Professor of Computer Engineering
Santa Clara University
Santa Clara, CA 95053

(408)554-4485

The TurboLAN Project [1]

Interim Report
Phase II

TurboLAN Intelligent Network Adapter Card (TINAC) Architecture

Principal Investigator

Hasan S. AlKhatib, Ph.D.
Associate Professor of Computer Engineering
Santa Clara University, Santa Clara, CA 95053

Research Assistants

Kenneth Majithia and Chi-Jiunn Jou

Research Scientist

Atsuhiko S. Suzuki

[1] This project has been sponsored under a joint grant from NASA and Furukawa Electric Company

1.0 Introduction

This document describes the hardware and the software architecture of the TurboLAN Intelligent Network Adapter Card (TINAC). The intent of this report is to present a high level as well as the detailed treatment of the workings of various components of the TINAC. The TINAC is divided into four major functional units.

A. **Network Access Unit (NAU):** This unit is comprised of Fiber Optic Transceivers, 8B/9B encoder/decoder, Clock/data separator, Pad data generator, and transmit clock generation and control.

B. **Buffer Management Unit (BMU):** This unit contains arbitration logic for buffer access, read/write and Region boundary error checking, DMA channels for the Node Processor (NP) and the Host Processor (HP).

C. **Host Interface Unit (HIU):** This unit provides a communication path between the HP and the BMU. It generates requests for buffer access, provides data path (through a DMA channel) for the movement of data to and from the HP.

D. **Node Processor Unit (NPU):** The Node Processor Unit provides initialization, programming and maintenance function for NAU, BMU, and HIU. It also monitors the network activity.

2.0 Block Diagram

The following is a block diagram of the four major functional Units of the TINAC.

Only major data paths are shown. Control signals are to be determined.

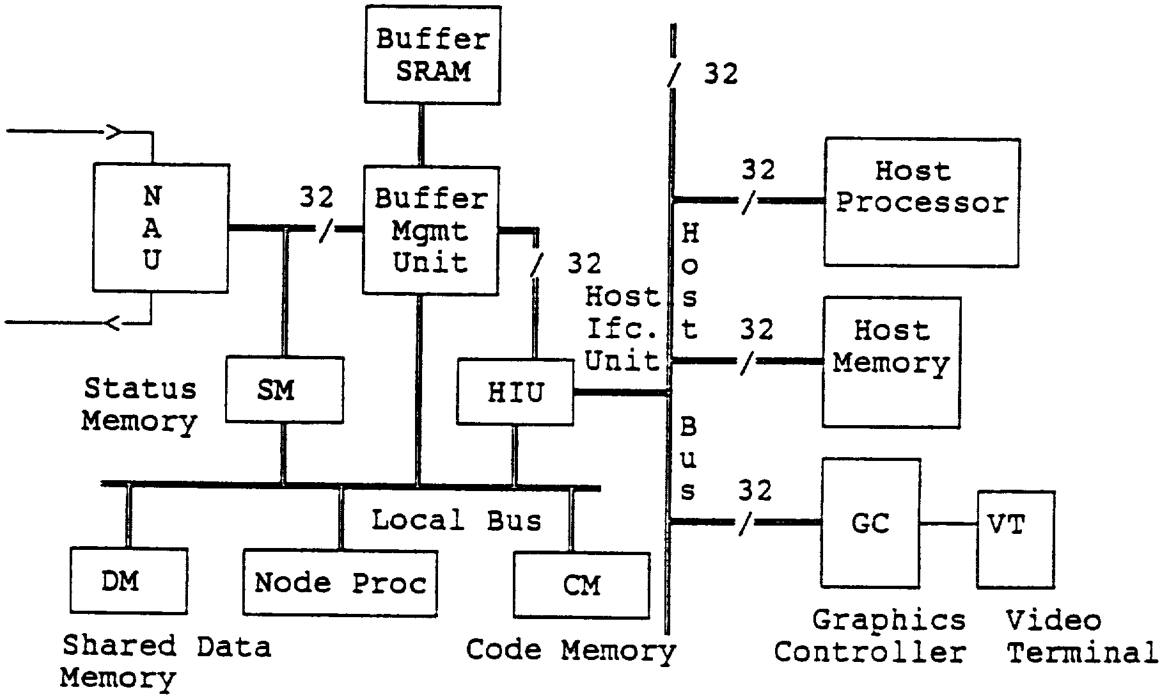


Figure 1. TINAC Block Diagram

3.0 Network Attachment Unit (NAU)

The Network Attachment Unit is responsible for error free transfer of data between the physical medium and the Buffer Management Unit. The NAU performs the following functions:

A. Data Flow Control

    1. Receive data from the media

        a. Strip start/end delimiter fields

        b. Convert frame data to bytes

        c. Recognize address

        d. Check CRC

    2. Transmit data to the media

        a. Handle tokens

        b. Add preamble, start/end delimiters, FCS and FC

        c. Generate CRC

B. 8B/9B encoding/decoding of clock and data

C. Compensate for variation in transmit clock of downstream station and the receive clock of the upstream station through elasticity buffer.

D. Generate pad or idle data during a frame reception.

4.0 Buffer Management Unit (BMU)

The Buffer Management Unit is responsible for error free access and transfer of data to and from NAU, NPU and HIU. The following are the major functions performed by this unit:

A. Arbitrate buffer access requests from NAU, NPU and HIU. NAU priority is the highest and HIU priority is the lowest.

B. Allow simultaneous access requests by three contending entities to different regions of the buffer.

C. If simultaneous requests are for the same region of the buffer, provide facility to queue the requests.

D. Maintain buffer memory region pointers and frame linking pointers.

E. Maintain byte ordering control.

F. Provide error checking (parity check and generate) during buffer reads and writes. Also perform Regional boundary checking.

G. Provide DMA channels to transfer data between buffer and NAU, NPU or HIU.

H. Communicate with Node Processor through processor instructions and/or status memory.

I. Communicate with the host processor through HIU.

J. Communicate with NAU through NAU interface.

5.0 Host Interface Unit (HIU)

Any communication between the Host Processor (HP) and the Buffer Management Unit (BMU) or the Node Processor Unit (NPU) is through the Host Interface Unit (HIU). It provides the following functions:

A. Decode commands from the Host Processor and generate appropriate buffer access requests.

B. Provide transceiver control through which data flows to and from the buffer and the Host Processor.

C. The Host Processor DMA transfers use this interface as well.

6.0 Node Processor Unit (NPU)

The Node Processor is a general purpose microprocessor that interfaces to BMU, HIU and NAU through the Local Bus. This unit provides the following functions:

A. The NP contains and controls its own code and data memory area.

B. Program HIU, NAU and BMU at initialization and when necessary.

C. Monitor the network functions by reading status registers/memory at appropriate times.

E. Perform network maintenance functions such as running background diagonistics etc.

F. Perform station management function by running CMP (Connection Management Protocol) an SMAP (System Management Application Protocol). The CMP consists of a number of tasks which have to do with managing the actual physical connection to the network. These include: logical attachment/detachment, physical attachment/detachment, establish ring configuration. The SMAP is required to communicate to a remote application process for the purpose of transferring the management information.

7.0 Buffer Management Unit Operation

The BMU interfaces with all three entities: NAU, NPU and HIU to efficiently manage the transfer of data in and out of the buffer. Before we consider the BMU operation in detail, it is important to understand the buffer memory organization. The buffer memory of 4 megabytes is organized into 8 Regions (RE), 512K bytes each as shown in the figure below:

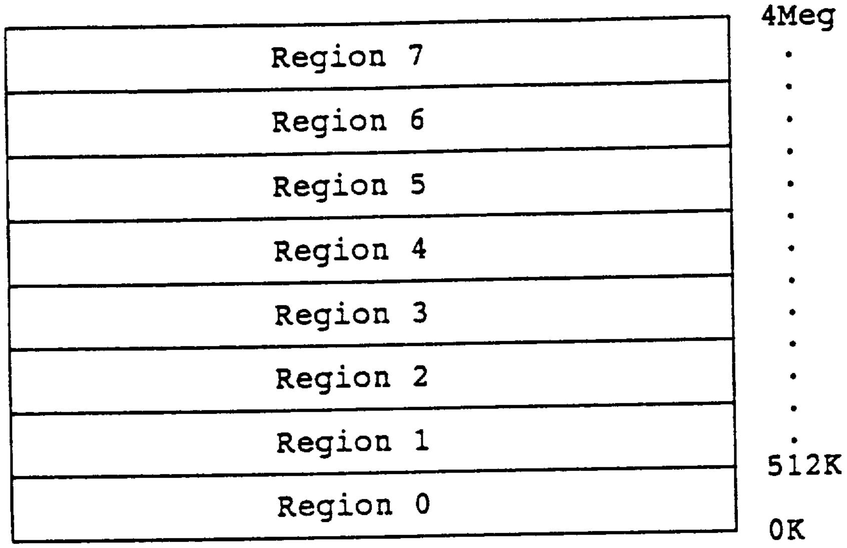


Figure 2. Buffer Memory Organization

Each Region is divided into 8 Frame Blocks (FB), 64K bytes each. In turn, each FB is divided into 256 Mini-Frame Blocks (MFB), 256 bytes each. The size of the MFB is chosen to match the size of the Mini-frame. The following figure shows the structure of a FB.

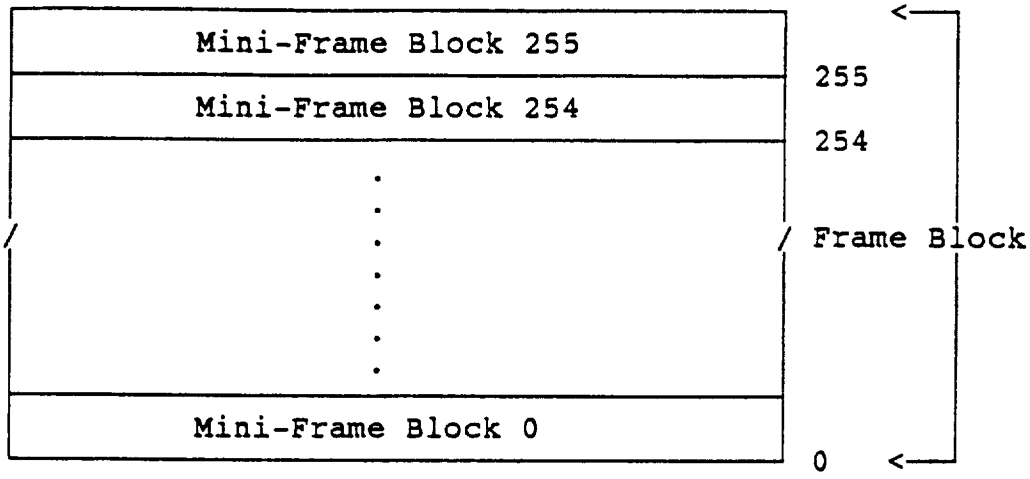


Figure 3. Buffer Frame Block Organization

The BMU transfer activity is tied to receive and transmit operations of the NAU. During the receive operation, data is loaded from NAU into one of the buffer FB. The data is then transferred to the host. Note that if the received data is larger than one FB (64K bytes), the next available Block in the same Region is used to load additional data. While data is being transferred from NAU to the second buffer FB, data from the first buffer FB is transferred to the host. This concurrency in data transfer of large amounts of data achieves a greater throughput. Note that this architecture uses off-the-shelf single-port RAM, thus taking advantage of low-cost memory. On the other hand, the decoding, control, and arbitration logic design in this approach is relatively complex. There is another possibility, where dual-port memory may be used. A typical dual-port memory comes with a standard parallel port and a serial port. The serial port is connected to NAU and the parallel port is shared by

NPU and HIU with HIU having the higher priority. Dual-port memory allows simultaneous access to both ports. Thus NAU can access the serial port at the same time HIU can access the parallel port. The only time the simultaneous operations are not allowed is when there is an internal RAM transfer cycle between the serial registers and the memory array taking place.

8.0 Communication Protocol Between HIU and NPU

First we will consider communication protocol between the HIU and the NPU since it significantly influences the other parts of the architecture. There are two possible approaches:

A. First approach uses a thread running on NPU to achieve various tasks.

B. The second approach uses command oriented protocol.

The first approach presents a number of problems:

1. The thread schedular for the network server task in the host adds complexity.
2. Synchronization of the thread running on NPU with the other threads running under the Network Server task.

3. Requires both the HP and the NP to be the same (or capable of running the same machine code). This may significantly increase the NP cost.

C. The command based protocol can lead to a straight forward implementation. However, let us look at all possible combinations based on where thread and data structure reside.

Data Struct. \ Thread		
	Case 1	Case 2
	Case 3	Case 4

Figure 4. Thread/Data Structure Combination Table

D. There are four possibilities. Let us investigate each in more detail.

1. Case 1: In this case, both the thread and the data structure are resident in the TINAC. This configuration has a number of disadvantages:

   a. NP needs code compatibility with the HP.

b. A separate schedular is needed for TINAC because TINAC and HP are independent.

c. Portability Problem - This configuration is hardware dependent.

2. **Case 2:** This is an unacceptable combination where the thread is running on TINAC and the data structure is in host memory. Since host memory is not available to TINAC, this configuration is not useful.

3. **Case 3:** In this case, the thread is running on HP and the data structure is in TINAC. This combination has a number of advantages:

   a. It does not require a separate schedular.

   b. It allows for a consistent and uniform implementation.

   c. Data Structure is visible.

   However, there is one disadvantage: Some bandwidth of host bus is taken up to manage the data structure in TINAC.

4. **Case 4:** In this case, both the thread and the data structure are in the host. Status memory is in the shared data memory area.

9.0 NAU/BMU Interface Protocol

The transfer of data to/from NAU and BMU is controlled through the protocol described below. The protocol is divided according to two major activities: transmit and receive. Transmit means transmission of data to the medium and receive means receiving of data from the medium.

A. **Transmit to Medium:** The following steps are required for a successful transmission of data to the medium. Note that even though some of the steps are listed sequentially, they may be executed in parallel.

1. A FB is reserved or locked (as contiguous buffer space)and added to the Frame Block Queue (FIFO).
2. The NPU sets up transmit control registers and appropriate DMA registers in the NPU.
3. The NPU generates transmit request to NAU (if FIFO in not empty), in turn, NAU generates READ REQUEST to BMU.
4. The NAU waits for free token. When it is received, the NAU sends READY signal to BMU.
5. The BMU returns READ ACK to NAU and begins transferring data using DMA.

6. When the last transfer occurs, DMA logic sends an interrupt to the NPU.

7. The NPU checks the transfer status and unreserves or unblocks the FB.

8. The Frame Buffer Queue is updated. If the Queue is not empty, the NPU sets up control registers and DMA and the process is repeated starting at step 2 and until FIFO becomes empty.

B. **Receive From Medium:** The following steps are required for a successful reception of data from the medium. Note that although some steps are listed sequentially, they may be actually executed in parallel. In the following steps it is assumed that two FBs are reserved for receive; i. e. there is no contention for these blocks and they are always available to receive data.

1. When the destination address match occurs at the receiving node, the NAU sends data to BMU. The data is transferred via a DMA channel. Note that the NAU strips physical header and CRC before sending data over to BMU.

2. The stripped physical address of each mini-frame is stored in the status memory (SM). At the end of a reception, this will show the last mini-frame of the current transaction.

3. When the last transfer occurs, The DMA channel will interrupt the NPU.

4. If an erroneous mini-frame is received by the receiver, the ACK bits in the mini-frame are turned off. When the sender sees these turned off ACK bits, it terminates the transmission of the current frame and retransmits beginning with the damaged mini-frame. The receiver, after resetting the ACK bits, waits for retransmission of the damaged mini-frame.

10.0 Frame Structure

There are at least two different types of frames; the DATA frame and the TOKEN frame.

A. **Data Frame:** The data frame is a frame through which a station executes data transactions with other stations on the network. The following figure shows fields that make up a data frame.

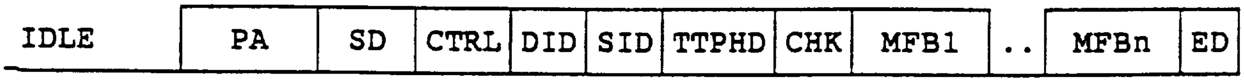


Figure 5. DATA Frame Structure

The following is a brief description of each field in the data frame:

PA: This is a preamble field for synchronization purposes. This field consists of 16 IDLE symbols. This field preceeds every data transmission.

SD: This is the Starting Delimiter. This field consists of two-symbol sequence (JK) that is uniquely recognizable. This field establishes the symbol boundary for the contents that follow.

CTRL: The control field contains several distinct pieces of information. It is divided into four sub-fields. They are as follows:

FT - Frame Type (1 symbol)
bit 0 - Control/Data Frame
bits 1,2,3 - Protocol Select: The NAU will decode these bits. If TTP is selected, TTP header will be copied from the data frame and put into the Status Memory.
MP - Message Priority (1 symbol)
FN - Mini Frame Number (2 symbols)
RE - Reserved (4 symbols)

SID and DID: Source and Destination IDs are 32 bits long. This may be an individual or group address.

TTPHD: This is the TTP (Turbo Transport Protocol) header field. See next section for a detailed description of this field.

CHK: This is the frame check field consisting of 32 bits. It is a cyclic redundancy check using the standard polynomial used in the IEEE 802 protocols.

MINI FRAME: Each mini frame is 256 bytes long and consists of two fields; The DATA field and the CHECK field. The DATA field is 255 bytes long and the CHECK field is one byte long. The CHECK field in turn is divided into two fields; a CRCCHK field (6 bits) and an ACK field (2 bits). The ACK field is used by the receiving station to flag a faulty mini frame.

ED: The ED field of a data frame is one delimiter symbol (T).

B. **Token Frame:** The token frame has all the fields that a data frame except the mini frame fields. The following figure shows the structure of a token frame.

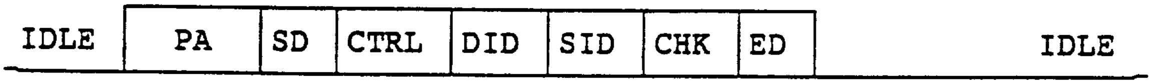


Figure 6. TOKEN Frame Structure

11.0 Turbo Transport Protocol (TTP)

Turbo Transport Protocol is in the process of being refined. However, a brief description will be provided here. The motivation behind creating yet another protocol is to improve the throughput and performance by reducing or removing certain system bottlenecks and overheads. The TTP attempts to combine Network, Transport and the Data Link Layers to achieve this goal. In this design an attempt is made to carefully design the frame to simplify parsing, reduce data copying and movement, and quick access to fields within the frame. All fields are aligned on 8, 16, or 32 bit boundaries.

TTP Header Format: The following figure shows the TTP header format.

<table>
<tr><td>0</td><td>7</td><td>15</td><td>31</td></tr>
<tr><td>VERS</td><td>LEN</td><td>SERVICE TYPE</td><td>MESSAGE ID</td></tr>
<tr><td colspan="3">FRAME ID</td><td>FLAGS</td></tr>
<tr><td colspan="4">FRAME LENGTH</td></tr>
<tr><td colspan="4">SOURCE IP ADDRESS</td></tr>
<tr><td colspan="4">DESTINATION IP ADDRESS</td></tr>
<tr><td colspan="4">SOURCE PORT ADDRESS</td></tr>
<tr><td colspan="4">DESTINATION PORT ADDRESS</td></tr>
<tr><td colspan="4">MESSAGE LENGTH</td></tr>
</table>

Figure 7. TTP Header Format

VERS: Current Protocol Version. This field is used to verify that the sender, receiver, and gateways agree on the format of the datagram. This field is 1 symbol long.

LEN: This is the length of the datagram header measured in 32-bit words.

SERVICE TYPE (ST): This is the Service Type field. It contains 4 bits of PRECEDENCE. These bits specify datagram precedence. Values range from 0 (normal precedence) to 15 (network control). This allows user to indicate the importance of each datagram. The other 4 bits are reserved.

MESSAGE ID (MID): This is the Message ID field. It is 16 bits long. This is required because there may be several frame per message. This allows multiple frame to be related to a single message.

FRAME ID (FID): This is a 16-bit field. This identifies the sequence number of the frame within a message.

FLAGS: This is a 16-bit field. The flag bits are to be determined.

FRAME LENGTH (FL): This is a full 32-bit field. This field contains the length of frame in bytes or in number of mini frames.

SOURCE IP ADDRESS (SIA) AND DESTINATION IP ADDRESS (DIP): Each of this field is a 32-bit field. They contain internet address of datagram sender and receiver respectively.

SOURCE PORT ADDRESS (SPA) & DESTINATION PORT ADDRESS (DPA): These addresses are compatible with MACH requirements. Each field is 32-bit long.

MESSAGE LENGTH (ML): This is a 32-bit field which contains the number of frames in the current message.

The TurboLAN Project [1]

Interim Report
Phase I

Protocol Choices for High Speed Local Area Networks

Principal Investigator
Hasan S. AlKhatib, Ph.D.
Associate Professor of Computer Engineering
Santa Clara University
Santa Clara, CA 95053, U.S.A.
(408)554-4485, halkhatib@scu.bitnet

Research Assistants
Chi–Jiunn Jou & Kenneth Majithia

February 15, 1990

[1] This project has been sponsored by NASA under grant number NCC89-352 and Furukawa Electric Company under grant number 5-28979.

Abstract

The TurboLAN project is aimed at investigating the necessary hardware and software technology to support Gbps local area networking. The project targets the development of a prototype 2.5 Gbps LAN by the end of its third phase, in 1992. In this first phase, four candidate protocols are examined thoroughly. These protocols are:

- The Token Passing Bus
- The Token Passing Ring
- The Santa Clara Ring
- A Star with an active switch

Since the behavior of a protocol at high speeds is significantly influenced by the propagation delay on the LAN and the frame length, the project had to define and identify the new conditions that will evolve under high speed networking, including the anticipated characteristics of the supported traffic. The four candidate protocols are analyzed under the new conditions. The TurboLAN project addresses a specific target application, namely graphic intensive simulations, including animations. The volume of data produced by system attached to the network, exceeds the conventional traffic volume by at least one or two orders of magnitude.

The goal of this phase is to evaluate candidate protocols for high speed networking and choose one to be used for the succeeding phases. A variety of performance measures, quantitative and qualitative, have been considered. This report presents the results of this evaluation. The study shows that two of these protocols exhibit significantly better performance in several aspects. These two protocols are *the Token Passing Ring and the Santa Clara Ring*. The final choice as indicated in the conclusion, has been extremely hard, since it was left to more subjective performance measures.

Contents

1 Introduction

Most of the currently available Local Area Networks (LANs) have been developed to support data services. It is, however, becoming quite apparent that we will soon have a strong demand for LANs which will support integrated video, image and graphics services. The video, image and graphic services incur a large amount of data transfer per transaction. This requires that the network supporting these services be a high speed network with very small transfer delays.

In recent years, High Speed Local Area Networks (HSLANs) with the data rates in excess of 100 Mbps and various network topologies and media access methods have been reported [1], [2], [3], [4], [5], [23], [22]. Without exception, the architectures of these new high speed networks are intertwined with the recent advances in the fiber optic technology. Proven to be extremely effective in the CATV and telephone trunk applications [6], optical fibers have yet to be exploited to the limits of their enormous information-carrying capacity.

The TurboLAN project sponsored by the National Aeronautics and Space Administration (NASA) and Furukawa Electric Company at Santa Clara University, to investigate the necessary technology and design an ultra high speed local area networks. A target raw data rate of 2.5 Gbps with an effective transfer rate of 250 Mbps per node has been set, using fiber optic technology.

In this report, four leading protocols based on three different topologies are compared. Token Passing Bus, Token Passing Ring, Santa Clara Ring and Star protocols

are selected as potential candidates for the implementation of the TurboLAN. The goal of this phase of research is to select a protocol for the implementation of the TurboLAN based on the evaluation of the following four performance parameters:

- Propagation and Network Access Delay
- Throughput
- Reliability and Maintainability
- Maturity of the Protocol

In section 2 of this report we presents an overview of the current state of the technology in ultra high speed local area networking. Since the TurboLAN project addresses a new application environment, it was necessary to develop new analysis models for the candidate protocols. This is done in section 3 of this report. Section 4 summarizes the results of the performance analysis and presents it in graphical form. Section 5 presents concluding arguments that lead to the choice of a protocol of adoption in the TurboLAN project.

2 Survey of High Speed LAN Projects

The concepts of high speed LANs are not new. However, only recently, with the progress and maturity of the fiber optic technology, it has been possible to realize these concepts into actual networks. Several successful implementations of high speed LANs with 60 Mbps to 400 Mbps data rates are published in the literature. In this section we will briefly describe key features of a select number of such implementations. Exclusively, with two exception, the projects are based on the Token Passing Ring and the Star protocols. Protocols such as Carrier Sense Multiple Access with Collision Detection (CSMA/CD) are not practical for high speed architectures. Even in the vicinity of 10 Mbps data rate, CSMA/CD exhibits saturation. In particular, the throughput of a CSMA/CD network is poor when the duration of the average packet transmitted is short compared to the end to end delay on the network channel. The following is a brief description of several published projects or implementations.

2.1 The LAN-DTH 140 Mbps Token Ring

This network system was developed at the Technical University of Denmark, Lyngby, Denmark [7]. It uses a 140 Mbps fiber optic Token Ring within a hierarchical heterogeneous local area network system called LAN-DTH. In this architecture, the high speed LAN plays a role of a main backbone net to which heterogeneous hosts and sub-nets of varying topologies are attached. The nodes within the main net can be used for the attachment of hosts or sub-nets, nodes within each subnet for the attachment of hosts or sub-subnet, and so on. In this connection the term "host" refers to any terminal equipment with independent processing power, thus including not only mainframe computers but also smaller machines, special processors (for graphics, image analysis etc.) workstations, as well as PCs. A reduction in bandwidth of about a factor of 10–15 for each level in hierarchy is expected. Under reasonable assumptions regarding the utilization of nodes within each component net, this factor also allows up to 100 nodes in each component sub-net.

Within the main net, a new token protocol is used. It uses a combination of priority and an interrupt mechanism [7] to offer high priority traffic, very short access times, making it suitable for carrying synchronous data streams, without the complex token circulation time control seen in other systems such as FDDI. It is designed to allow long transfers to proceed at full speed when no other station wishes to transmit. A token is used to control channel access, and is generated by the last station which has had access to the ring. There is no limit to the length of transmission from any station. But other stations may request the current transfer to be interrupted. The current sender then must pass the token. A fully loaded network functions like a simple Token Ring with non-exhaustive service and with a packet size equal to the ring size. For a lightly loaded network, users are offered exhaustive service resulting in good utilization. Since the token is passed immediately after the end of the data, utilization is further improved for small messages by propagation parallelism.

A prototype LAN-DTH network system has been set-up at the Technical University of Denmark, and is currently being tested and evaluated. It contains seven nodes within the main net, whose circumference is approximately 3 kilometers. Of these nodes, one is a management station, two support VAX hosts, one supports an HP1000 host and three support sub-nets of different types.

2.2 System FINEX

System FINEX is the first network controller product implemented based on FDDI. [9] It was developed by the R & D Group of N.T.M. Advanced Technology Center, Fibronics Ltd. of Israel. This controller is an implementation of the FDDI standard as published by ANSI X3T9.5 sub-committee. The host side of the controller connects to a VME bus (12.5 Mbytes per second) and on the network side to two pairs of 62.5/125 micrometer fibers. On the network side, the controller provides PHY level interfacing and MAC level functions. For example, the MAC level functions include transmit/receive, 32-bit CRC generation and checking, ring initialization with beacon and claim frames, frame insertion, stripping and repetition. The Physical level interfacing functions include symbol encoding/decoding, serialization and de-serialization of symbols, electrical to optical and optical to electrical signal conversion, clock recovery and frequency matching, as well as connection establishment and termination.

The Frame Buffer Unit (FBU) prevents the controller from becoming a throughput bottleneck. FBU is a 256 KB of dual ported RAM. The MAC hardware is designed to transfer data at 12.5 Mbytes per second, which matches with the 100 Mbit/s raw data rate of the network. If host channels are slower than 12.5 Mbytes per second, then FBU must be able to store many frames. The controller connects to a Network Management Station through a Protocol Processor (PP), which has direct access to the various elements of the controller. It performs such functions as configure controller parameters, monitor status and gather statistics about the number of frames transmitted/received, number of errors, memory allocation. It executes distributed management algorithms such as ring mapping and fault isolation. The PP is a Motorola 68010 12.5 MHz microprocessor with 512 KB of RAM. The controller supports data rates of 50 Mbps. The limiting factor is the processing power of the Protocol Processor, which handles the parameters, LLC headers, linked lists, and queues.

2.3 A High-capacity Multi-service Local Area Network LION

The project LION (Local Integrated Optical Network) is completed within ESPRIT (European Strategic Program for Research and Development in Information Technologies), promoted by EEC. [10] Physically, LION is a double ring topology (like FDDI) which allows for reconfiguring in case of failure. But it operates like a unidirectional bus [11]. The information is written and transmitted to the 'write channel' fiber then it is retransmitted by the HB and FP (head bus and folding point) node on the 'read channel' fiber. In case of failure, the (HB and FP) node becomes a standard one and its function is shared between the nodes upstream and downstream the faulty area.

Each node on the LION network uses its own clock for transmission. However, the receiver is synchronized to the upstream transmitter. Frequency differences are compensated for with padding bits. The transmission system is integrated in the MAU (Medium Attachment Unit) block. The MAU performs such functions as transmitting/receiving optical signals, encoding/decoding, frame synchronization with local clock, and error detection.

The Network Communication Unit (NCU) contains several hardware and software functional blocks. The ACM (Access Protocol Manager) implements the hybrid access protocol and handles the interface to the MAU. The Management Module (MM) which is in connection with the Network Control Center (NCC) uses the management protocol for controlling the node. The Circuit Module (CM) is in charge of the data transfer between the access interface handler and ACM via a dual-ported buffer CDM (Circuit Data Memory). The Circuit Transfer Control (CTC) manages the memory transfers. The data stored in the memory is directly sent to the MAU or the circuit interface handler without any further buffering step. The Burst Data Module (BDM) is dedicated to the data transfer in the form of packets including the signaling and management primitives. The management software is ported by a microcomputer connected via a node to the network. This is labeled as the Network Control Center (NCC). NCC performs such functions as network initialization and reconfiguration, maintenance, user administration for controlling the resource sharing.

2.4 A Synchronous Fiber Optic Ring LAN for Multi-gigabit per second Mixed-Traffic Communication

This fiber optic LAN protocol uses time division multiplexing (TDM) to assign a set of individual channels to each data frame [12]. Phase and frequency synchronism of ring is achieved by skewing the system clock so that an integral number of data frames are suspended in transit about the ring's circumference. This technique is practical only when data rates are in the gigabit range. This concept demonstrates that with appropriate system design, logic complexity can be minimized and clock distribution can be simplified.

This architecture allows, in addition to high channel efficiency, future bandwidth growth. The system can interface to both low-speed and high-speed user peripheral systems through TDM bandwidth partitioning. It can also accommodate the speed upgrades prompted by faster opto-electronic devices by increasing the frame bit density. The clock recovery circuit remains unaffected. The LAN suffers from the problem inherent to all single ring topologies, namely a single point of failure. Its main performance disadvantage is that unless the traffic is uniformly distributed over all channels, the aggregate channel utilization is low. This problem is common to all TDM channels.

Each station in the network functions as an active repeater and uses a TDM architecture to combine independent channels into one data frame. A laboratory prototype was implemented at 100 MHz.

The fiber optic elements consist of a 850 nm transverse junction strip (TJS) single-mode laser diode (Mitsubishi ML-2307), a 1.5 Km long single-mode fiber, and a small-area silicon avalanche photo-diode detector (APD). The detector output is amplified 35 dB by a 0.1-4.5 GHz wideband amplifier (Avantek AMG-4045M120) to about 2 Volts. The repeater is detector band-limited to 5 Gbps. A two-station breadboard network was constructed to verify the network feasibility.

2.5 A 200 Mbps Synchronous TDM LAN Suitable for Multi-Service Integration

This prototype was implemented at Fujitsu Laboratories Ltd., Kawasaki, Japan [13]. It is a 200 Mbps multi-service optical LAN using a synchronous TDM loop structure. The LAN consists of a central supervisory node (SV node) and multiple service nodes called ordinary nodes (OD nodes) connected via a double optical fiber loop. The TDM frames circulate continuously around the loop. The SV node has central supervisory functions, including TDM frame generation, total loop length adjustment, and network operation control. The OD nodes provide communication channels for various communication paths virtually overlaid on the loop. Connected to the SV node is the PC based monitor. The main role of the monitor is to provide a human operator interface to the LAN. The operator is able to control the

LAN through the monitor. A duplicate optical fiber loop is used to provide extra reliability for OD node or fiber cable failures. Each OD node has a transmission line switching function, which allows three kinds of loop reconfigurations to isolate a faulty portion of the LAN: [Line Switch, Bypass, and Loopback.]

Multiple independent communication paths of various speeds up to 140 Mbps and various modes including point-to-point ring, and multicast, can be provided on the loop. The structure of this LAN is suitable for integration of multiple services, including, video, image, data, and voice, since each service can independently choose its speed, access method and mode. The video service is operated at 50 Mbps. The TV signal from the TV camera is broadcast to multiple TV monitors. Functions of the video access unit include conversion between analog and digital signals, as well as TV signal bandwidth compression.

The image/data service, operated at 50 Mbps, takes a ring path, in which multiple workstations and a database, communicate with each other using a token passing access method. The 2 Mbps voice service supports multiplexed voice channels for which the LAN provides a fixed path between the voice multiplexer and the PBX. The PBX acts as a gateway to public networks or ISDN. Telephones connected to the voice multiplexer can talk with each other through the PBX.

This network has the advantage of multimode service, but it suffers from low channel utilization like all TDM channels.

2.6 A 1.2 Gbps Optical Loop LAN for Wideband Office Communications

This is an ultra high speed optical loop LAN serving as a backbone network for the next generation service integrated office communication systems [14]. It is developed by C&C Systems Research Laboratories of NEC Corporation. The network has a hierarchical structure in that 1.2 Gbps total capacity is shared between a 400 Mbps multiaccess packet switching system and a 700 Mbps wideband circuit switching system. The 400 Mbps packet switching system in itself has its own hierarchical structure with the provision of real-time information handling capability. By virtue of this network structure, a wide variety of communication systems, from CSMA/CD systems to distributed video conferencing systems can be economically integrated in the network. The optical loop uses a 1.3 micrometer multimode fiber. The 400 Mbps packet switching multiaccess system and the 700 Mbps circuit switching system are multiplexed so that the network can handle a large class of information services. The 700 Mbps circuit switching system is mainly used for video communication. The 400 Mbps packet switching multiaccess system can be shared not only by high-speed terminals but also by a variety of subsystems such as CSMA/CD networks, which enables low-end terminals to economically access the backbone network.

2.7 A Robust 100 Mbps Network for Avionics Application

This LAN was developed to demonstrate the feasibility of a 32-node fiber optic LAN that operates at 100 Mbps suitable for space applications. [15] The network incorporates a high bandwidth fiber optic communication medium that interconnects up to 32 nodes. The network is a packet data system. The topology is that of a star coupler that broadcasts optical signals from any network node to all network nodes. The media access protocol uses a Carrier Sense Multiple Access with Collision Detection and contention resolution via allocation of Time Slots, CSMA/CD/TS. The protocol uses a statistical relationship between collision and data traffic to provide dynamic performance adjustment. The access protocol provides for collision avoidance, collision detection, and contention resolution. The protocol consists of two modes, a random access mode and an ordered access mode. In the random access mode, if the media is not active a BIU with a packet for transmission may access the network by beginning the transmission. If the media is active, BIUs must defer network access until the network becomes available. All BIUs monitor the network for activity and the media is considered not active when a carrier has not been sensed for T(gap) micro-seconds. T(gap) is derived from the time required between the end of a message and the beginning of an acknowledge as seen by the furthest node, in the FODS. The T(gap) is 1.2 microseconds. Collisions are recognized by BIUs when two transmissions occurring at a node cause an invalid Bi-phase signal.

The bus interface unit, (BIU), of a node provides support both for the Physical and Data Link layer protocols, as well as the Front End Processor function that implements higher layer protocols. The higher layers include user interface functions, BIU diagnostic functions, system monitoring, fault recovery, and user packet routing besides others. The system provides a high degree of robustness to keep the system operational under adverse conditions in space. The features incorporated provide self-start, restart, automatic adjustment to changing load conditions, built-in-test equipment, statistics monitoring, automatic redundancy, and remote or local configuration control. A unique access protocol combines the performance advantages of token and contention protocols. The protocol supports the self-start features.

The CSMA/CD/TS protocol performance parallels that of other random access protocols, but as the data traffic increases this protocol does not experience inordinate transfer delays caused by the overhead associated with repetitive collisions and random backoffs. This protocol incurs a bounded transfer delay due to the deterministic manner in which contentions are resolved and the fact that, in the random mode, any time two or more nodes are waiting to access the network the nodes will collide when the media is available, Therefore, statistically, as the data traffic load increases, more collisions will occur causing the network to realize the performance of a token network.

2.8 The Distributed Memory Network (DMN): An 8 Gbps Fiber Optic Tightly Coupled System

This network has been developed at Harris Government Aerospace System Division, Melbourne, Florida [16]. The DMN consists of nodes that are interconnected in a ring topology. Each node consists of a local Bus Arbitration Unit (BAU) and a processor host. The host is a source of data to the network and/or a sink of data from the network. These hosts could be combinations of data processors, sensor interfaces or signal processors. The local host processor communicates with the network via a BAU, which provides rate buffering, control logic and access to distributed shared memory.

DMN has several unique properties. One of the most important properties is that accesses to the network appear to the host processor as memory accesses. In other words, a host processor transfers data to another processor within the network by writing the data to a specific memory address. The hardware which implements the network provides the physical transfer of data in a manner which is transparent to the host processor. Each host processor accesses the network as if it were a local memory.

DMN is fundamentally a packet switching network with a packet size of one word. The bus operation within DMN is based upon the transfer of parallel words which contain address, data and control fields. All messages are transfers of words from one node to the memory of another node.

2.9 IMAGENET: A High Speed Local Area Network

This R & D project was conducted at Ford Aerospace Corporation, San Jose, California [17]. IMAGENET is a high speed fiber optic LAN based on a Star topology. The Star is a 16 × 16 intelligent non-blocking switch connecting 16 nodes. When a link is established between two nodes, the Star circuit switches a dedicated 50 Mbps path between nodes. Since the Star is non-blocking, up to eight simultaneous links can be connected at one time giving the switch an aggregate capacity of 400 Mbps. The Star operates in two modes: control mode and data transfer mode. In the control mode, small command messages are passed between the Star and the Network Interface Units (NIUs). The messages are used to set up connections, pass acknowledgments, and check on the network status. If an NIU needs to access the network, it sends a message to the Star requesting network access and specifying the address of the destination NIU. The Star then either establishes a connection if the destination NIU is free, or puts the request on a queue. When the source NIU is finally served, the Star switches into the data transfer mode, establishes a direct connection between the source NIU and the destination NIU, and transfers data at 50 Mbps.

Such a Star suffers form the problem of having a single point of failure, requiring the use of expensive redundancy techniques to compensate for the weak reliability of the central switch.

2.10 VectorNet

VectorNet is a very high speed Token Ring network developed by Scientific Computer Systems Corporation [22]. Either fiber optic or copper media can be used supporting a network span of 1 km or 50 meters, respectively. Either media supports a data rate of 1.4 Gbps on a ring. Several counter rotating rings are supported for reliability. Fiber optic and copper media can be mixed in the same network allowing a flexible cost-effective implementation.

An original light weight high level protocol has been introduced with VectorNet to facilitate high performance interaction among nodes. Each ring interface has a protocol processor that implements the special light weight protocol. Four different services are supported with the light weight protocol: [*DATAGRAM, REQUEST, GET & PUT, and REPLY*]. The *DATAGRAM* message service support protocols that require datagram services, such as IP. The *REQUEST* message service is used to request an I/O operation across the network. The *GET & PUT* message service are used to directly access another node's memory across the network. The *REPLY* message service is used to notify a node that a *REQUEST* operation has been completed.

2.11 Ultra Network

Ultra Network is a product of Ultra Network Technologies of Santa Clara, California [23]. The network uses a Star topology where nodes are connected over data links to a central Ultra-Bus. The Ultra-Bus is a high speed parallel bus, 5-foot long, acting as a central hub for the Star. The bus supports a transfer speed of up to 1 Gbps. The data links to the Ultra-Bus can carry information at a whole range of speeds depending on the media used. Data links from as low as 56 Mbps up to 1Gbps are supported using either coaxial cables or fiber optic links.

A protocol processor is responsible for making routing decisions on a packet-by-packet basis on the Ultra-Bus. It is also responsible for the set-up and release of connections. The protocol processor is a large bit-slice machine, with extensive

ASIC support.

Ultra Network is designed to serve supercomputer class machines. Interfaces to Cray HSX channels are available as well as to HSC busses. High speed FIFO buffers are used for the attachment of data links to the hub.

3 Performance Analysis of Protocols

In this section the analysis of four candidate protocols for the TurboLAN project are described. The protocols examined are the Token Passing Bus, the Token Passing Ring, the Santa Clara Ring and a Star. The performance measures are defined and explained. The notation used in the analysis is given and the parameters are outlined.

3.1 Performance Measures

Four local area network protocols have been selected for investigation due to their apparent suitability to high speed networking. Since the goal of this project is to develop the technology for a very high speed LAN to run at several Gbps, it was very important to exclude those protocols that may present further limitation to the effective throughput of the overall LAN system. An *effective* data rate of 250 Mbps per graphic workstation has been set as a goal for this project. A typical example for the target application environment is a configuration where a large number of graphic workstations are interconnected via a local area network as front-end machines to a supercomputer in the back-end. Such an environment will facilitate applications that will require running large simulations on a supercomputer while passing graphics images in real-time to the graphics workstations for animation.

From the user's point of view, an obvious consideration of performance is the *Network Response-Time*. This network response-time is the time a message takes to be transferred from a user on one system to another user on a second system and get acknowledged. In other words, the *Response-Time* consists of the following components:

1. The time for the message to be transferred from the user space to the network buffer space in the host memory;
2. The time to copy the message from the host memory to the network interface buffer;
3. The time for the message waiting in the interface buffer, until the station manages to access the network media;
4. The time for the message to propagate from the source station to the destination station;
5. The time to transmit the entire message on the network;
6. The time for the destination station to acknowledge the reception of the message;
7. The time to copy the message from the controller buffer to the network buffer in the destination host memory;
8. The time to copy the message from the network buffer to destination user's space within the destination host memory.

Since the time to generate an acknowledgment may depend on the application as well as the implementation, it may be more useful to measure the *one-way delay* in transferring a message. The data transfer time in step 1, 2, 7 and 8 are not affected by network media access protocols, since they are dependent on the high level protocols, the operating system and the device-oriented features such as the speed of the host bus. Thus, the *one-way delay* consists of the time to go through steps 3-5 of *Response-Time*.

The great majority of the published research has addressed the performance of LAN media access protocols under steady state conditions. Analysis of the transient state performance tends to be more complicated. The information one learns from the steady state analysis, may not answer some critical questions, such as those one encounters when designing a high speed LAN. Since the desire to construct a high speed LAN is invariably associated with the desire to achieve a high performance LAN, it is often important to answer the question: *What effective throughput will a user see?* In addition, the traffic on a typical LAN tends to be of a bursty nature. In this study, we examine the transient state behavior of the different media access protocols, instead of the steady state behavior. As mentioned in the beginning of this section, we are mainly interested in very high speed data communication such as the target 250 Mbps effective data rate for graphic workstations. In this configuration, a typical message is a graphic image from a screen. Assuming a 1250x1250-pixel with 8-bit color, a screen bit map consists of 12.5 Mbits. During animation, a message with a full graphic image will have to be transferred at such a speed to meet the needs of real-time presentation on the destination station. In order to support such requirements, several innovations have to be achieved in removing the bottlenecks in the communication path. In particular, pipelining techniques can be used, which would result into a flow-through protocol. Such a protocol allows the overlap of data copying from the user space to interface with data transmission from interface to the network media. In this case, there is no waiting time for a message in the interface buffer, for instance.

In order to guarantee a target throughput of 250 Mbps, from workstation to workstation, it is important to examine the transient behavior of each protocol rather than the steady state behavior, as indicated above. The conditions under which we guarantee to provide the effective data rate of 250 Mbps are:

- *Data transfers are user-memory to user-memory.*
- *The message size is 12.5 Gbits.*
- *There are at most 10 active users producing data on the LAN at any time.*
- *The media data rate is at least 2.5 Gbps.*

The network delay time in the transient-state analysis can be defined as the time from the arrival of a message into the network interface of a station until the last bit of this message is delivered through the channel to the network interface of its destination station. In order to evaluate and compare the four different protocols, it was important to isolate station queuing and implementation influences from the pure media access network protocol behavior. To achieve this, we will limit our analysis to examining the delays in steps 3, 4, and 5 of the *Response-Time*.

An obvious performance consideration from the systems point of view, is the LAN aggregate throughput. Throughput measures the number of bits per second that are successfully transferred through the network. Frequently, throughput is normalized by dividing it by the channel capacity. Transient normalized throughput can be computed as the time for transmitting the messages of all active stations divided by the total delay time of these active stations.

In summary, the two common and important performance measures for a local area network are delay time and normalized throughput. The transient-state behavior is more significant than steady-state behavior in the target configuration environment. The delay time consists of the network access time, message transmission time, and message propagation time.

3.2 Notations

The following terms and parameters are used as the framework for performance comparison among the candidate protocols. These terms are common for most protocols. The terms for a particular protocol are described, when that protocol is analyzed.

R: data rate.
N: number of stations.
k: number of active stations.
L_m: message size.
L_t: token size.
t_{ee}: end-to-end propagation time.
t_h: token holding time.
t_t: total time to transmit a token frame.
t_b: propagation delay per station.
T: delay time.
T_{min}: lower limit of delay time.
T_{max}: upper limit of delay time.
T_{ave}: average delay time.
T_{cycle}: sum of the delay times of all active stations.
S: normalized throughput.

The target data rate, R, for the TurboLAN is 2.5 Gbps. Nevertheless, in this study we examine the performance of the different protocols under a wide range of data rates. R is varied between 0.5 Gbps and 4.5 Gbps. The number of stations, N, is varied from 20 to 1000. The number of active stations, k, is varied from 1 to 20.

The propagation of signals along an optical fiber is dependent on the refraction index of that fiber. The typical speed of propagation through the optical fibers is about 2/3 the speed of light. Thus, signals need about 5 microseconds to propagate over one kilometer of fiber. The TurboLAN will be designed to extend over a distance of 100 Kilometers. This will result into an end-to-end propagation delay of approximately 500 μsec.

As discussed earlier, the target message size, L_m is 12.5 Megabits. Although a message may consists of several frames, a station transmits one full message at a time to satisfy the needs of real time animation. The total time to transmit a message equals L_m/R, or 5 $msec$ at $R = 2.5$ Gbps. In the Token Bus or the Token Ring protocol, the token holding time t_h is the maximum time for which a station is allowed t hold the token, transmit the data and then release the token. In this analysis, we set t_h to be equal to the message transmission time, to allow transmitting a full message. To be consistent, we also use t_h as the message transmission time in other protocols such as the Santa Clara Ring and the Star.

The token size, L_t, is selected as 15 bytes, or 120 bits. This is similar to the token size used in FDDI, which has at least 11 bytes [18]. The total time to transmit a token frame, t_t, equals to L_t/R or 48 $nsec$ when $R = 2.5$ Gbps.

The Delay time, T, is the sum of network access time, message transmission time, and message propagation time. That is

$$T = T_{networkaccess} + T_{transmission} + T_{propagation} \tag{1}$$

Since there are many overlaps among these components of delay time due to the interaction among different stations, the actual computation of delay time is different for each protocol.

Normalized throughput, S, is computed as the channel utilization. Consider a cycle time T_{cycle}, during which k active stations transmit messages. This time includes the transmission time as well as contention time for k messages.

Therefore, normalized throughput can be computed as follows:

$$S = \frac{kt_h}{T_{cycle}} \tag{2}$$

3.3 Token Passing Bus Protocol

In the Token Bus protocol, access to the shared bus is controlled by means of a token which is passed from station to station. A station wishing to transmit a message must wait until it receives a free token. The station transmits its message, then generates a new token and sends it to its successor on the logical ring. This structure is shown in Figure 1.

If there is not any message for transmission, a station still needs to transmit a new token to its successor when it acquires the token from its predecessor. This media access protocol is deterministic rather than contention-based, thus providing an upper bound on the access time and has predictable delays for varying loads.

The average propagation delay, for a message on a bus is approximately 1/3 the length of the bus when the number of stations is large [20],[21].

The minimum delay time occurs when a station seizes a token and is ready to transmit its message. The delay time will only consist of message transmission time and message propagation time. That is,

$$T_{min} = t_h + \frac{1}{3}t_{ee} \tag{3}$$

The maximum delay time occurs when a station has just missed the token. In this case, the station needs to wait for an entire cycle to allow a token to come again. Every time a token is passed from one station to another, the full token has to be transmitted over the bus. The resulting delay will consist of the token propagation on the bus plus the token transmission. Idle stations will have to pass the token to succeeding stations as described above. Active stations on the other hand will not only cause token propagation and transmission delays, but will also cause message transmission and message propagation delays. Hence, a station that has just missed the token will have to wait the delay caused by all active stations on the bus as well as the delay of passing the token by inactive stations, namely,

$$T_{max} = (k-1)(\frac{1}{3}t_{ee} + t_h + t_t) + (N-k)(\frac{1}{3}t_{ee} + t_t) + (\frac{1}{3}t_{ee} + t_h + \frac{1}{3}t_{ee})$$

The first term in the above equation is the total time spent by the other $k-1$

active stations. The second term is the total time for $N - k$ inactive stations. The third term is due to the delay encountered by the station under consideration. The station waits $\frac{1}{3}t_{ee}$ to receive the token and spends t_h to transmit its message. The last $\frac{1}{3}t_{ee}$ term is due to the propagation time for this transmitted message to reach its destination station.

The above equation can be simplified as

$$T_{max} = kt_h + (N-1)t_t + (N+1)\frac{1}{3}t_{ee} \tag{4}$$

For k active stations, the average delay time can be computed as follows: If an active station is the $(i+1)$-th station seizing a token from the k active stations, it will suffer the delay time caused by i active stations preceding it in the loop plus the delay time caused by half the number of inactive stations on the average. That is,

$$T_{ave} = \frac{1}{k}\sum_{i=0}^{k-1}[i(t_h + t_t + \frac{1}{3}t_{ee}) + \frac{N-k}{2}(t_t + \frac{1}{3}t_{ee}) + (t_h + \frac{1}{3}t_{ee})]$$

The first term inside the summation is the delay time for $i - 1$ active stations, and the second term is for $(N-k)/2$ inactive stations. The third term is due to the message transmission and message propagation time for the $i - th$ active station. The above equation can be simplified to:

$$T_{ave} = \frac{k+1}{2}t_h + \frac{N-1}{2}t_t + \frac{N+1}{6}t_{ee} \tag{5}$$

The normalized throughput, S, can be computed by using equation 2. The cycle time, T_{cycle}, for k active stations in the system is as follows

$$T_{cycle} = k(t_h + t_t + \frac{1}{3}t_{ee}) + (N - k)(t_t + \frac{1}{3}t_{ee})$$

or

$$T_{cycle} = kt_h + N(t_t + \frac{1}{3}t_{ee}) \tag{6}$$

The first term of the above equation is the delay time for k active stations. The second term is the delay for $N - k$ inactive stations. Hence the normalized throughput is:

$$S = \frac{kt_h}{kt_h + N(t_t + \frac{1}{3}t_{ee})} \tag{7}$$

3.4 Token Passing Ring Protocol

The Token Passing Ring protocol, resembles that of the Token Passing bus, except the token here circulates around a physical ring rather than a logical ring. Figure 2 shows this structure. The token does not need to be stored and forwarded as the case is with the Token Passing bus. Transmission over the ring is done in a flow-through manner through the ring interfaces, with one bit delay per ring interface (or node). A station waits for a free token, converts it to a busy one through the modification of one bit, then appends its data frame to the end of the busy token.

The message on the ring will make a round trip and be removed by the transmitting station. The transmitting station then inserts a new free token on the ring when it has completed transmission of its own message.

There are two modes, for the logical operation of the interface; listen and transmit. The structures for both modes are shown in Figure 3. A station enters the transmit mode only when a free token arrives and is converted into a busy token. A station returns back to the listen mode at the completion of a data frame transmission. Stations are normally in the listen mode, in which a station can detect its address in the messages circulating around the ring, and read the contents of messages addressed to it. Each station in the listen mode causes 1 bit delay to the traffic passing around the ring. To facilitate the analysis, we define a new term for this delay:

t_b: bit delay per station in the listen mode.

Note that t_b is the time used to allow the conversion of a free token into a busy token to initiate a data transmission cycle.

The token propagation delay, from one station to its successor, is only $1/N$ of the length of the ring. Assuming a uniform geographical distribution of the stations around the physical ring. In other words, the token propagation time is t_{ee}/N. The message propagation time, from the transmitting station to its destination station, is $t_{ee}/2$ on the average.

The minimum delay time occurs when an active station has just received a free token ready to be used. Before starting to transmit its message, this active station needs a delay of t_b to change the free token into the busy token. As the message travels around the ring, it will suffer a delay of t_b per station. On the average, there are $(N-1)/2$ stations betweeen a transmitting station and its receiver. . Therefore, we have

$$T_{min} = t_b + t_h + \frac{N-1}{2}t_b + \frac{1}{2}t_{ee}$$

or

$$T_{min} = t_h + \frac{N+1}{2}t_b + \frac{1}{2}t_{ee} \tag{8}$$

The maximum delay time occurs when a station has just missed the token. The station needs to wait for an entire cycle to allow a free token to come again. An inactive station will cause a delay of t_{ee}/N to wait for the propagation of a free token from the station preceding it, and a delay of t_b for the bit propagation delay through the ring interface. An active station will cause a delay of t_{ee}/N due to the token propagation time, a delay of t_b to change the free token into a busy one, t_h to transmit its message, and t_t to transmit a new free token. The transmitting station will suffer delays of t_{ee}/N to wait for a free token arrival, t_b to change the free token into a busy one, and t_h to transmit its message. In addition, the transmitted message will suffer a delay of $(N-1)t_b$ to go through all other stations and $t_{ee}/2$ from the transmitting station to its destination station. So, we have

$$T_{max} = (k-1)(\frac{t_{ee}}{N} + t_b + t_h + t_t) + (N-k)(\frac{t_{ee}}{N} + t_b) + \frac{t_{ee}}{N} + t_b + t_h + \frac{N-1}{2}t_b + \frac{1}{2}t_{ee}$$

or

$$T_{max} = kt_h + (k-1)t_t + \frac{3N-1}{2}t_b + \frac{3}{2}t_{ee} \tag{9}$$

The average delay time seen by k active stations can be computed as follows: If an active station is the $(i+1)$-th station acquiring a free token among the k active stations, it will suffer the delay time caused by i active stations, plus the delay caused by half the number of inactive stations, on average. Thus,

$$T_{ave} = \frac{1}{k}\sum_{i=0}^{k-1}[i(t_b + t_h + t_t + \frac{t_{ee}}{N}) + \frac{N-k}{2}(t_b + \frac{t_{ee}}{N}) + t_b + t_h + \frac{N-1}{2}t_b + \frac{1}{2}t_{ee}]$$

or

$$T_{ave} = \frac{k+1}{2}t_h + \frac{k-1}{2}t_t + Nt_b + \frac{2N-1}{2N}t_{ee} \tag{10}$$

The normalized throughput, S, can be computed by using equation 2. The cycle time T_{cycle} is

$$T_{cycle} = k(t_b + t_h + t_t + \frac{t_{ee}}{N}) + (N-k)(t_b + \frac{t_{ee}}{N})$$

or

$$T_{cycle} = k(t_h + t_t) + Nt_b + t_{ee} \tag{11}$$

Thus, the normalized throughput is

$$S = \frac{kt_h}{k(t_h + t_t) + Nt_b + t_{ee}} \tag{12}$$

3.5 Santa Clara Ring Protocol

The Santa Clara Ring protocol is a tokenless ring media access protocol. Contention for acquisition of the ring is resolved through a dynamic priority scheme. Under heavy load conditions, this priority scheme results in round-robin service.

The ring interface,as shown in Figure 4, contains two shift registers: one for incoming messages and the other for outgoing messages. The input shift register is accessed for output via a dynamic pointer. There is a switch at the output of the ring interface. This switch is connected to the input shift register pointer when the station has no outgoing message. A message from the channel is propagated through the ring interface via the input shift register and placed back onto the channel. The message is also delivered from the input line to the host computer when the host station is the destination. When a station transmits its message, the output switch is connected to the output shift register and data bits are directed onto the ring channel.

The message format consists of a priority field, a packet length, a source identification number, a destination identification number, a data field, a cyclic redundant code field, and an acknowledgment field. Each ring interface is assigned a unique ID number. These ID numbers are the basis for calculating transmission priorities. The priority is always a non-negative value such that the larger the value the lower the priority. The priority of a message is the current priority of the transmitting station when starting to be transmitted. The priority of a station is calculated as $(station\_ID - source\_ID + N) \bmod N$. This dynamic priority of a station is calculated every time a new frame passes by the station's ring interface. Each station has a unique station_ID number. The source_ID is the station identification for the source of the current message on the ring. Additionally, when a ring interface detects an idle channel and the host station has nothing to transmit, the priority for that ring interface is reset to the default priority which is the lowest priority.

A station can transmit its message if the channel is idle. Contention occurs when a message from one station reaches the ring interface of another station, while that other station is trying to gain access to the ring.

Before the contention is resolved, the incoming message is buffered in the input shift register and the associate dynamic pointer is moved toward the output direction as data bits propagate through. The priorities of both incoming message and the local station are compared. If the incoming message has a lower priority, it will be aborted. Otherwise, the station will abort its own transmission and begins repeating the winning message. In order for other stations to delineate between the aborted and following message, some amount of idle bits are inserted to make the separation known as silence. When losing the contention, a station will insert two bits of silence before repeating the winning message. When a message is successfully transmitted, the transmitting station will insert four bits of silence at the end of the transmission.

Fair access to the ring is achieved through a dynamic reassignment of priorities, that results into round-robin scheduling.

For the purpose of performance analysis, we define some new terms and parameters as follows

t_{ba}: the delay time in integral number of bits for detecting an aborted transmission.

t_{bi}: the delay time in integral number of bits for detecting idle transmission.

t_{bp}: the delay time in integral number of bits for resolving contention through a bit by bit comparison of the priority fields of the two contending messages.

t_{ba} is defined as two bits of silence. t_{bi} is defined as four bits of silence.

t_{bp} has a minimum value of 1 bit and a maximum value equal to the size of the priority field. In this protocol, the priority field is defined to have 16 bits. On the average, the number of bits required to be compared is no more than two! In this analysis, we use t_{bp} to represent those different values.

On the average, the propagation time for a message transmitted from a source station to a destination station is $t_{ee}/2$. There is no delay to wait for a token. However, there are some delays caused by the contention resolution algorithm during the transmission of messages.

When there are k active stations, the station with the highest priority has the minimum delay time. In addition to the message transmission time and message propagation time, this station suffers the delay caused by the resolution of the contention at other active stations. Other active stations will buffer the incoming message, adding an extra delay of $t_{bp} + t_{ba}$. This is the sum of the delay time for priority comparison and the delay for the insertion of two bits of silence. Assume that the k active stations are uniformly distributed on the ring. On the average, there are $(k - 1)/2$ active stations between the transmitting station and its destination station. Therefore, we have

$$T_{min} = t_h + \frac{1}{2}t_{ee} + \frac{k-1}{2}(t_{bp} + t_{ba}) \tag{13}$$

The maximum delay time is encountered by the active station that has the lowest priority. The lowest priority station will suffer the delay caused by other $k-1$ active stations. Each active station will insert four bits of silence, t_{bi}, and transmits its own message. Each active station will buffer the incoming message to allow for resolving the contention. This is equal to $t_{bp} + t_{ba}$. The message with the highest priority will be buffered at all other $k-1$ stations. The message with the second highest priority will be buffered at $k-2$ stations, and so on. There is no extra delay of $(t_{bp} + t_{ba})$ the lowest priority station will suffer with its transmission, since it is the last station to transmit among the k contending stations. On the average, there is a propagation delay of t_{ee}/k for an active station until it detects the end of four bits of silence inserted by its predecessor. Therefore, we have

$$T_{max} = \sum_{i=1}^{k-1}[t_h + t_{bi} + (k-i)(t_{bp} + t_{ba}) + \frac{t_{ee}}{k}] + t_h + \frac{1}{2}t_{ee}$$

or

$$T_{max} = kt_h + (k-1)t_{bi} + \frac{k(k-1)}{2}(t_{bp} + t_{ba}) + \frac{3k-2}{2k}t_{ee} \tag{14}$$

The average delay for the k active stations can be computed as follows: An active station with the i-th priority will suffer the delay caused by the other $i-1$ active stations with higher priorities. Similar to the case of maximum delay time, the total delay caused by $i-1$ active stations is

$$\sum_{j=1}^{i-1}[t_h + t_{bi} + (k-j)(t_{bp} + t_{ba}) + \frac{t_{ee}}{k}]$$

Since only $(k-i)$ active stations are left when the message with the i-th priority is transmitted, there are $(k-i)/2$ active stations between the transmitting station and its destination station. Therefore, we have

$$T_{ave} = \frac{1}{k}\sum_{i=1}^{k}\{\sum_{j=1}^{i-1}[t_h + t_{bi} + (k-j)(t_{bp}+t_{ba}) + \frac{t_{ee}}{k}] + t_h + \frac{1}{2}t_{ee} + \frac{k-i}{2}(t_{bp}+t_{ba})\}$$

After simplification, we get

$$T_{ave} = \frac{k+1}{2}t_h + \frac{k-1}{2}t_{bi} + \frac{2k-1}{2k}t_{ee} + \frac{(k-1)(4k+1)}{12}(t_{bp}+t_{ba}) \qquad (15)$$

There are two different cases for computing the normalized throughput S. When $k=1$, the cycle time T_{cycle} is just the message transmission time t_h. This is due to the fact that no contention occurs. Thus

$$S = \frac{t_h}{t_h} = 1 \qquad (16)$$

When $k>1$, the duration of transmission and propagation of the message with the i-th priority is

$$t_h + t_{bi} + \frac{t_{ee}}{k} + (k-i)(t_{bp}+t_{ba})$$

This duration begins from the instant the message with i-th priority starts to transmit until the message with the $(i+1)$-th priority starts to transmit its message. This expression is just the delay time caused by the message with the i-th priority as mentioned in the case of maximum delay time. Therefore, we have

$$T_{cycle} = \sum_{i=1}^{k}[t_h + t_{bi} + \frac{t_{ee}}{k} + (k-i)(t_{bp}+t_{ba})]$$

or

$$T_{cycle} = k(t_h + t_{bi}) + t_{ee} + \frac{k(k-1)}{2}(t_{bp} + t_{ba})$$

Thus

$$S = \frac{kt_h}{k(t_h + t_{bi}) + t_{ee} + \frac{k(k-1)}{2}(t_{bp} + t_{ba})} \tag{17}$$

3.6 Star Protocol

In general, it is hard to compare the Star with the above three protocols. Instead of packet switching used by others, the Star protocol uses circuit-switching. In a Star topology network, a collection of stations are attached to a central switching unit. The central switch establishes a dedicated path between any two stations that wish to communicate.

An important characteristic of a Star topology network is whether it is blocking or non-blocking. Blocking occurs when there is no available path through the switch to connect a source station with a destination station. Blocking occurs only during the request set-up phase of a circuit-switched connection. A blocking network is one in which such blocking is possible. While, a non-blocking network permits all stations to be connected at once and grants all possible connection requests as long as the destination station is available.

A crossbar switch is usually used to realize a non-blocking network. To be able to make n simultaneous connections, the crossbar switch needs $n \times n$ crosspoints. Since the number of crosspoints grows with n^2, it is costly to have a crossbar switch with large n. In practice, the number of crosspoints of a crossbar switch is up to 10×10. In order to accommodate a large number of stations, such as 100, one needs to use a Star protocol with multiple layers. The number of layers needed depends on the number of stations N. For example, a two-layer Star protocol is required for a network with 100 stations when 10×10 crossbar switches are used.

In order to do a comparative performance analysis with the other three protocols, we need to make some reasonable assumptions about the Star protocol. One such assumption is to allow a multiple-layer Star topology with 10×10 crossbar switches resulting into a hierarchical structure. Hence, with a system with 20 or less stations, we only need a single-layer Star.

A switch has 20 links which can be used to connect stations or to other switches. In a two-layered Star topology, there is one switch in the top layer. This switch connects other switches in the lower layer. There is no connection linked to a station directly from the top switch. A switch in the lower layer uses a link to connect the higher-layered switch and the other 19 links to connect to stations. A two-layered Star structure is shown in Figure 5. A special case for the two-layered Star is when there are less than 38 stations in the system. In this case, two switches are enough to connect 38 stations. These two switches can be connected directly to each other without a second-layer switch.

A two-layered Star can hold a maximum of 380 stations, which corresponds to 20 local switches with each holding 19 stations. A three-layered Star is required for more than 380 stations. A similar approach is used for the three-layered Star as the one used in a two-layered Star. A three-layered Star can hold a maximum of 7220 stations. This corresponds to 20 middle-layered switches with each linking 19 local switches. Each local switch in turn links 19 stations.

Another important assumption about the Star is the equivalent data rate. To be fair with the other protocols, a single star is assumed to have the same aggregate data rate as the ones in other protocols. Hence, the bandwidth in a single link is dependent on the number of connections that can be made simultaneously. In a single 10×10 Star, ten simultaneous connections are possible. If the aggregate date rate is R, the data rate in each link is therefore $R/10$.

In general, the number of simultaneous connections is dependent on the number of stations. Each connection links two stations. Hence, the maximum number of simultaneous connections is $N/2$, where N is the number of stations. For example, when N equals 100, the data rate of each link becomes $R/50$, where R is the aggregate data rate.

The length of each link will affect the message propagation time. Hence, links are assumed to have the same length L/N. L is the length of a ring or a bus in the other protocols. This will give the propagation delay of t_{ee}/N when a message propagates along a single link. There are two links required to connect two stations within a single star. More links are needed for connections beyond one single star.

Circuit switched networks require a connection set-up time, before the transmission of data commences. We use the term switching time to represent the delay due to a connection setup.

We define the following new terms and parameters to facilitate the analysis.

N_c: the cluster size for a single star.
t_w: the switching time of a Star.
t'_h: the effective holding time.

N_c is the number of links that a switch has. It is the same as the number of stations that a single star cluster can accommodate. In our analysis, N_c is 20. t_w is the switching time of a Star, and assumed to be 1 μsec. t'_h is the message transmission time for a Star protocol. Since the data rate of each link in a Star protocol is different from the one used in other protocols, we distinguish them by using t'_h and t_h. t'_h is equivalent to $\frac{N}{2}t_h$.

In this analysis, we assume that the Star is non-blocking. In reality, a Star does not have the probability for blocking caused by the contention of the links connecting a switch to a higher-layer switch. Since the message propagation time changes as the number of stations is varied, we analyze delay for four different cases corresponding to $N = 20, 100, 500,$ and 1000.

For case 1, $N = 20$, we only need a single switch. Before a message is transmitted, t_w is needed to set up the connection. Each connection requires two links, thus the message propagation time is $2t_{ee}/N$. Therefore, the minimum delay time is

$$T_{min} = t_w + t'_h + 2\frac{t_{ee}}{N}$$

or

$$T_{min} = t_w + \frac{N}{2} t_h + 2\frac{t_{ee}}{N} \tag{18}$$

When k is not more than 10, the maximum delay is the same as the minimum delay. When k is greater than 10, a station may delay until other stations finish transmitting their messages. This is because the maximum capability that a single switch can support is 10 simultaneous connections. The next connections may be set up just after messages pass the central switch. Thus, the maximum delay is

$$T_{max} = t_w + t_h^{'} + \frac{t_{ee}}{N} + t_w + t_h^{'} + 2\frac{t_{ee}}{N}$$

or

$$T_{max} = 2t_w + Nt_h + 3\frac{t_{ee}}{N} \tag{19}$$

The average delay time needs to take account of the probability that a station waits for other stations. When k is not more than 10, all stations can transmit messages immediately. The average delay is the same as minimum delay. Note that we ignore the probability that a certain active station is both a sender and a receiver. If a station is both a sender and a receiver, it must suffer twice the minimum delay – one for receiving a message from another station, and the other for transmitting its own message. When k is greater than 10, at least one station needs to wait until other stations almost finish transmitting their messages. The probability that a station does not need to wait is $10/k$. The probability of waiting is then $(k - N)/k$.

Hence, the average delay time is

$$T_{ave} = \frac{10}{k}(t_w + \frac{N}{2}t_h + 2\frac{t_{ee}}{N}) + \frac{k-10}{k}(2t_w + Nt_h + 3\frac{t_{ee}}{N})$$

or

$$T_{ave} = \frac{2k-10}{k}(t_w + \frac{N}{2}t_h + \frac{t_{ee}}{N}) + \frac{t_{ee}}{N} \tag{20}$$

For case 2, $N = 100$, we need a two-layered Star. Except a single second-layered switch, there are six local switches. The minimum delay time occurs when a connection linking two stations within a single local star. This is equivalent to the delay time in case 1, i.e. equation 18.

The maximum delay time occurs when a connection linking two stations in two different local star clusters. This connection will go through four links and three switches. Two links are needed to connect a station to the second-layered switch, and two links are for the second-layered switch to the destination station in other cluster. Thus, we have

$$T_{max} = 3t_w + t_h' + 4\frac{t_{ee}}{N}$$

or

$$T_{max} = 3t_w + \frac{N}{2}t_h + 4\frac{t_{ee}}{N} \tag{21}$$

The average delay time is based on the probabilities of the occurrence of the above two kinds of connections. For a particular station, there are $(N_c - 2)$ paths that can connect to other stations in the same local star cluster. The total number of paths that a particular station can connect with is $(N - 1)$. Thus, the probability of a local connection is $(N_c - 2)/(N-1)$. While the probability of a remote connection is $1-(N_c-2)/(N-1)$, or $(N-N_c+1)/(N-1)$. Therefore, we have

$$T_{ave} = \frac{N_c - 2}{N-1}(t_w + t_h' + 2\frac{t_{ee}}{N}) + \frac{N - N_c + 1}{N-1}(3t_w + t_h' + 4\frac{t_{ee}}{N})$$

After simplification, we have

$$T_{ave} = t_w + \frac{N}{2}t_h + 2\frac{t_{ee}}{N} + \frac{2N - 2N_c + 2}{N-1}(t_w + \frac{t_{ee}}{N}) \qquad (22)$$

For case 3, $N = 500$, we need two second-layer switches. We can directly connect these two switches without going through a third-layered switch. The minimum delay time is still the same as in case 1, expressed in equation 18.

The maximum delay time occurs when a connection going through the two second-layered switches. Two links are required to connect the source station to one of the second-layered switches, and two links are needed to connect the destination station to the other second-layered switch. In addition to this, there is one link needed between the two second-layered switches. This requires five links and four switches. Thus, we have

$$T_{max} = 4t_w + t_h' + 5\frac{t_{ee}}{N}$$

or

$$T_{max} = 4t_w + \frac{N}{2}t_h + 5\frac{t_{ee}}{N} \qquad (23)$$

The average delay time is also based on the probabilities of these events. There are three kinds of connections. One is the minimum delay time as in equation 18, another is the maximum delay time as in equation 23, and third is the delay time as in equation 21. Same as in case 2, the probability of the minimum delay time is $(N_c - 2)/(N-1)$. The probability of the third kind of connection is $(N_c - 1)(N_c - 2)/(N-1)$. The term $(N_c - 2)$ corresponds to the number of stations other than the particular one within a local cluster. The term $(N_c - 1)$ corresponds to the number of links connecting to local switches for a second-layered switch. The probability of the maximum delay time is $1 - (N_c - 2)/(N-1) - (N_c - 1)(N_c - 2)/(N-1)$, or $(N - N_c^2 + 2N_c - 1)/(N-1)$. Therefore, we have

$$\begin{aligned} T_{ave} &= \frac{N_c - 2}{N-1}(t_w + t_h' + 2\frac{t_{ee}}{N}) + \frac{(N_c - 1)(N_c - 2)}{N-1}(3t_w + t_h' + 4\frac{t_{ee}}{N}) \\ &+ \frac{N - N_c^2 + 2N_c - 1}{N-1}(4t_w + t_h' + 5\frac{t_{ee}}{N}) \end{aligned}$$

After simplification, we have

$$T_{ave} = t_w + \frac{N}{2}t_h + 2\frac{t_{ee}}{N} + \frac{3N - N_c^2 + 1}{N-1}(t_w + \frac{t_{ee}}{N}) \qquad (24)$$

For case 4, $N = 1000$, we need a three-layered Star. A third-layer switch is used to connect three second-layer switches. The minimum delay is still the same as in case 1, or equation 18.

The maximum delay occurs when a connection goes through the third-layer switch. Three links are required to connect the source station to the third-layer switch. Another three links are needed to connect the third-layer switch to the destination station. This requires six links and five switches. Thus, we have

$$T_{max} = 5t_w + t_h' + 6\frac{t_{ee}}{N}$$

or

$$T_{max} = 5t_w + \frac{N}{2}t_h + 6\frac{t_{ee}}{N} \tag{25}$$

Similar to case 3, the average delay time can be computed based on the probabilities of the different events. There are also three kinds of connections. They are the same as the ones in case 3 except that the maximum delay is as expressed in equation 25, The probability of each kind of connection is exactly the same as the corresponding one in case 3. Therefore, we have

$$\begin{aligned} T_{ave} &= \frac{N_c - 2}{N-1}(t_w + t_h' + 2\frac{t_{ee}}{N}) + \frac{(N_c - 1)(N_c - 2)}{N-1}(3t_w + t_h' + 4\frac{t_{ee}}{N}) \\ &+ \frac{N - N_c^2 + 2N_c - 1}{N-1}(5t_w + t_h' + 6\frac{t_{ee}}{N}) \end{aligned}$$

After simplification, we have

$$T_{ave} = t_w + \frac{N}{2}t_h + 2\frac{t_{ee}}{N} + \frac{4N - 2N_c^2 + 2N_c}{N-1}(t_w + \frac{t_{ee}}{N}) \tag{26}$$

The normalized throughput, S, can also be computed via equation 2. Since all messages can be transmitted simultaneously, the cycle time T_{cycle} is the same as the maximum delay time. Hence, for case 1, $N = 20$ and $k \leq 10$, we have

$$S = \frac{kt_h}{\frac{N}{2}t_h + t_w + 2\frac{t_{ee}}{N}} \tag{27}$$

For $N = 20$ and $k > 10$, we have

$$S = \frac{kt_h}{Nt_h + 2t_w + 3\frac{t_{ee}}{N}} \tag{28}$$

Similarly, for case 2, $N = 100$, we have

$$S = \frac{kt_h}{\frac{N}{2}t_h + 3t_w + 4\frac{t_{ee}}{N}} \tag{29}$$

For case 3, $N = 500$, we have

$$S = \frac{kt_h}{\frac{N}{2}t_h + 4t_w + 5\frac{t_{ee}}{N}} \tag{30}$$

For case 4, $N = 1000$, we have

$$S = \frac{kt_h}{\frac{N}{2}t_h + 5t_w + 6\frac{t_{ee}}{N}} \tag{31}$$

3.7 Messages With Short Holding Time

In the above sections, the performance analysis is based on the assumption of long size of messages and long holding time. The size of message is about $12.5Mbits$, and the holding time is $5msec$, which is long enough to transmit the whole message at one time. In reality, a typical local are network environment has mixed sizes of messages. The long size of messages are only used for transmitting long-sized files or special purpose applications such as an entire graph image. While, the short-sized messages will often appear on the network. Even in the case of transmitting an entire graph image, it needs several control messages which have short sizes to establish and terminate the connection. Long holding time may cause intolerant delay for transmitting a short message. So, a short holding time is sometimes an appropriate alternative. The performance analysis for the protocols with mixed-sized messages will be discussed in the next section. The analysis of delay time and throughput for the protocols with long messages and short holding time are described in this section.

When using a short holding time, a long message needs to be divided into several segments which are transmitted at different time. The holding time, for example, is $1msec$, a long message which needs $5msec$ to transmit will be divided into five segments. Each segment is transmitted $1msec$. Before transmitting the next segment, a station may be interrupted and wait for messages transmitted by other stations. We define a new term to represent this parameter.

N_m: the number times of transmission needed to transmit a long-sized message.

$N_m = 1$ when the holding time is equal to or less than message transmission time. Otherwise, $N_m = t_{ml}/t_h$, where t_{ml} is the transmission time for a long message.

3.7.1 Token Passing Bus Protocol

When holding time is less than message transmission time, a station will release its token upon holding time is expired and wait for an entire cycle to allow a token to come again. This cycle time, T_{cycle}, is shown in equation 6. The delay times for the first $(N_m - 1)$ cycles are the same and equal to T_{cycle}. The delay time for the last cycle are the same as the ones in section 3.3. Hence, we have

$$T_{min} = t_h + \frac{1}{3}t_{ee} + (N_m - 1)[kt_h + N(t_t + \frac{1}{3}t_{ee})] \quad (32)$$

$$T_{max} = kt_h + (N-1)t_t + (N+1)\frac{1}{3}t_{ee} + (N_m - 1)[kt_h + N(t_t + \frac{1}{3}t_{ee})] \quad (33)$$

$$T_{ave} = \frac{k+1}{2}t_h + \frac{N-1}{2}t_t + \frac{N+1}{6}t_{ee} + (N_m - 1)[kt_h + N(t_t + \frac{1}{3}t_{ee})] \quad (34)$$

For throughput, the cycle time to transmit all k messages is $N_m T_{cycle}$. The total effective messages transmission time is $k(N_m t_h)$. Thus

$$S = \frac{k(N_m t_h)}{N_m T_{cycle}} \tag{35}$$

or

$$S = \frac{kt_h}{kt_h + N(t_t + \frac{1}{3}t_{ee})} \tag{36}$$

Note that the above equation is the same as equation 7. In other words, the throughput is not affected by varying the holding time.

3.7.2 Token Passing Ring Protocol

Similar to Token Passing Bus protocol, the Token Passing Ring protocol has the same effect as the holding time is reduced. The delay times are the same as the ones in section 3.4 except the additional term $(N_m - 1)T_{cycle}$, where T_{cycle} is equation 6. Thus, we have

$$\begin{aligned} T_{min} &= t_h + \frac{N+1}{2}t_b + \frac{1}{2}t_{ee} \\ &\quad + (N_m - 1)[k(t_h + t_t) + Nt_b + t_{ee}] \qquad (37) \\ T_{max} &= kt_h + (k-1)t_t + \frac{3N-1}{2}t_b + \frac{3}{2}t_{ee} \\ &\quad + (N_m - 1)[k(t_h + t_t) + Nt_b + t_{ee}] \qquad (38) \\ T_{ave} &= \frac{k+1}{2}t_h + \frac{k-1}{2}t_t + Nt_b + \frac{2N-1}{2N}t_{ee} \\ &\quad + (N_m - 1)[k(t_h + t_t) + Nt_b + t_{ee}] \qquad (39) \end{aligned}$$

Similarly, the throughput can be expressed as the equation 35, where T_{cycle} is shown in equation 6. After simplification, it is also reduced to the one in section 3.4. That is,

$$S = \frac{kt_h}{k(t_h + t_t) + Nt_b + t_{ee}} \tag{40}$$

3.7.3 Santa Clara Ring Protocol

In the Santa Clara Ring protocol, the delay times for the last cycle are the same as the ones in section 3.5. The delay times for the rest $(N_m - 1)$ cycles are the same. However, this cycle time is different from the one in section 3.5. All k stations remain active when a message is transmitted in the first $(N_m - 1)$ cycles. Hence, this cycle time is

$$\sum_{i=1}^{k}[t_h + t_{bi} + \frac{t_{ee}}{k} + (k - i)(t_{bp} + t_{ba})]$$

After simplification, it becomes

$$k(t_h + t_{bi}) + t_{ee} + k(k-1)(t_{bp} + t_{ba})$$

Hence, we have

$$\begin{aligned}
T_{min} &= t_h + \frac{1}{2}t_{ee} + \frac{k-1}{2}(t_{bp} + t_{ba}) \\
&\quad + (N_m - 1)[k(t_h + t_{bi}) + t_{ee} + k(k-1)(t_{bp} + t_{ba})] && (41) \\
T_{max} &= kt_h + (k-1)t_{bi} + \frac{k(k-1)}{2}(t_{bp} + t_{ba}) + \frac{3k-2}{2k}t_{ee} \\
&\quad + (N_m - 1)[k(t_h + t_{bi}) + t_{ee} + k(k-1)(t_{bp} + t_{ba})] && (42) \\
T_{ave} &= \frac{k+1}{2}t_h + \frac{k-1}{2}t_{bi} + \frac{2k-1}{2k}t_{ee} + \frac{(k-1)(4k+1)}{12}(t_{bp} + t_{ba})
\end{aligned}$$

$$+ (N_m - 1)[k(t_h + t_{bi}) + t_{ee} + k(k-1)(t_{bp} + t_{ba})] \tag{43}$$

For throughput, $S = 1$ or the equation 16 when $k = 1$. When $k > 1$, the cycle time to transmit all k messages becomes

$$(N_m - 1)[k(t_h + t_{bi}) + t_{ee} + k(k-1)(t_{bp} + t_{ba})] + \sum_{i=1}^{k}[t_h + t_{bi} + \frac{t_{ee}}{k} + (k-i)(t_{bp} + t_{ba})]$$

The first term is the delay time for the first $(N_m - 1)$ cycles, and the second term is for the last cycle. After simplification, it becomes

$$kN_m(t_h + t_{bi}) + N_m t_{ee} + (N_m - \frac{1}{2})k(k-1)(t_{bp} + t_{ba})$$

Therefore, we have

$$S = \frac{kN_m t_h}{kN_m(t_h + t_{bi}) + N_m t_{ee} + (N_m - \frac{1}{2})k(k-1)(t_{bp} + t_{ba})} \tag{44}$$

3.7.4 Star Protocol

The Star protocol uses circuit switching, hence, has no effect when the holding time is reduced. Since we choose a non-blocking network, all messages will be transmitted without being interrupted when the connections are set up. Hence, the delay times and throughput are the same as the ones in section 3.6. except that t_h is replaced by $N_m t_h$. Therefore, we have

$$T_{min} = t_w + \frac{N}{2}N_m t_h + 2\frac{t_{ee}}{N} \tag{45}$$

For case 1, $N = 20$, T_{max} and T_{ave} are the same as T_{min}, or equation 45, when $k \leq 10$, In this condition, throughput is

$$S = \frac{kN_m t_h}{\frac{N}{2}N_m t_h + t_w + 2\frac{t_{ee}}{N}} \tag{46}$$

When $k > 10$, we have

$$\begin{aligned} T_{max} &= 2t_w + NN_m t_h + 3\frac{t_{ee}}{N} && (47) \\ T_{ave} &= \frac{2k-10}{k}(t_w + \frac{N}{2}N_m t_h + \frac{t_{ee}}{N}) + \frac{t_{ee}}{N} && (48) \\ S &= \frac{kN_m t_h}{NN_m t_h + 2t_w + 3\frac{t_{ee}}{N}} && (49) \end{aligned}$$

For case 2, $N = 100$, we have

$$\begin{aligned} T_{max} &= 3t_w + \frac{N}{2}N_m t_h + 4\frac{t_{ee}}{N} && (50) \\ T_{ave} &= t_w + \frac{N}{2}N_m t_h + 2\frac{t_{ee}}{N} + \frac{2N - 2N_c + 2}{N-1}(t_w + \frac{t_{ee}}{N}) && (51) \\ S &= \frac{kN_m t_h}{\frac{N}{2}N_m t_h + 3t_w + 4\frac{t_{ee}}{N}} && (52) \end{aligned}$$

For case 3, $N = 500$, we have

$$\begin{aligned} T_{max} &= 4t_w + \frac{N}{2}N_m t_h + 5\frac{t_{ee}}{N} && (53) \\ T_{ave} &= t_w + \frac{N}{2}N_m t_h + 2\frac{t_{ee}}{N} + \frac{3N - N_c^2 + 1}{N-1}(t_w + \frac{t_{ee}}{N}) && (54) \\ S &= \frac{kN_m t_h}{\frac{N}{2}N_m t_h + 4t_w + 5\frac{t_{ee}}{N}} && (55) \end{aligned}$$

For case 4, $N = 1000$, we have

$$T_{max} = 5t_w + \frac{N}{2}N_m t_h + 6\frac{t_{ee}}{N} \tag{56}$$

$$T_{ave} = t_w + \frac{N}{2}N_m t_h + 2\frac{t_{ee}}{N} + \frac{4N - 2N_c^2 + 2N_c}{N-1}(t_w + \frac{t_{ee}}{N}) \tag{57}$$

$$S = \frac{kN_m t_h}{\frac{N}{2}N_m t_h + 5t_w + 6\frac{t_{ee}}{N}} \tag{58}$$

3.8 Messages With Mixed Sizes

In this section, we consider that the messages may have different sizes. Long messages represent the graph image of an entire screen, while short messages represent the control messages or other network traffic. The former has the typical size of 12.5 *Mbits*, while the latter has 36 *kbits*. Short messages come more freqent than the long ones. A typical system may have ninty percent of short messages and only ten percent of long messages. Before the analysis, we define some new terms and paramters.

t_{ml}: transmission time for a long message.
t_{ms}: transmission time for a short message.
w_l: percentage of long messages.
w_s: percentage of short messages.

When $w_l = 1.0$, it becomes the case where all the messages has long size, i.e. the ones considered in the previous subsections. When $w_s = 1.0$, it is the other extreme case where all the messages has short size. The performance for the latter case is the same as the former one except that the holding time is replaced by the short message transmission time. These two extreme cases are special cases of the results derived in this section.

3.8.1 Token Passing Bus Protocol

The minimum delay time is the one when a short message is transmitted without any delay. That is,

$$T_{min} = t_{ms} + \frac{1}{3}t_{ee} \tag{59}$$

The maximum delay time is the one when all messages are long ones and a station just misses the token. The maximum delay time is the same as the one in section 3.7.1. or equation 33.

The average delay time is considered in two seperate cases. When $N_m = 1$, each message is wholly transmitted at one time. The average delay time is the same as equation 5. in section 3.3 except that t_h is replaced by $w_s t_{ms} + w_l t_h$. This change reflects the effects of the mixed messages loads over the network. Therefore, we have,

$$T_{ave} = \frac{k+1}{2}(w_s t_{ms} + w_l t_h) + \frac{N-1}{2}t_t + \frac{N+1}{6}t_{ee} \tag{60}$$

When $N_m > 1$, all short messages are completely transmitted within the first cycle. In this cycle, the average delay caused by any other active station becomes

$$w_s t_{ms} + w_l t_h + t_t + \frac{1}{3}t_{ee}$$

$w_l k$ stations are still active and transmit their long messages in the remaining $(N_m - 1)$ cycles. Except the last cycle, each station will suffer delay of $t_h + t_t + \frac{1}{3}t_{ee}$ caused by $w_l k$ active stations and of $t_t + \frac{1}{3}t_{ee}$ casused by $(N - w_l k)$ inactive stations. In the last cycle, a station which is $(i+1)$-th station seizing a token will suffer the delay time by active stations preceding it in the loop plus the delay time by half the number of inactive stations on the average. That is

$$\begin{aligned}
T_{ave} \quad = \quad & \frac{1}{k}\sum_{i=0}^{k-1}\{w_s[i(w_s t_{ms} + w_l t_h + t_t + \frac{1}{3}t_{ee}) + \frac{N-k}{2}(t_t + \frac{1}{3}t_{ee}) + t_{ms} + \frac{1}{3}t_{ee}] \\
& + w_l\{k(w_s t_{ms} + w_l t_h + t_t + \frac{1}{3}t_{ee}) + (N-k)(t_t + \frac{1}{3}t_{ee}) \\
& + (N_m - 2)[w_l k(t_h + t_t + \frac{1}{3}t_{ee}) + (N - w_l k)(t_t + \frac{1}{3}t_{ee})] \\
& + w_l i(t_h + t_t + \frac{1}{3}t_{ee}) + \frac{N - w_l k}{2}(t_t + \frac{1}{3}t_{ee}) + t_h + \frac{1}{3}t_{ee}\}\}
\end{aligned}$$

After simplification, we have

$$\begin{aligned}
T_{ave} \quad = \quad & \frac{k+1}{2}(w_s t_{ms} + w_l t_h) + \frac{N-1}{2}t_t + \frac{N+1}{6}t_{ee} \\
& + w_l(N_m - 1)[w_l k t_h + N(t_t + \frac{1}{3}t_{ee})] \\
& + w_s w_l[\frac{k+1}{2}t_{ms} + \frac{1}{2}(t_t + \frac{1}{3}t_{ee})] \qquad (61)
\end{aligned}$$

For throughput, the cycle time to transmit all k messages is

$$\begin{aligned}
& k(w_s t_{ms} + w_l t_h + t_t + \tfrac{1}{3}t_{ee}) + (N-k)(t_t + \tfrac{1}{3}t_{ee}) \\
& + (N_m - 1)[w_l k(t_h + t_t + \tfrac{1}{3}t_{ee}) + (N - w_l k)(t_t + \tfrac{1}{3}t_{ee})]
\end{aligned}$$

After simplification, it becomes

$$k(w_s t_{ms} + w_l N_m t_h) + N N_m(t_t + \frac{1}{3}t_{ee})$$

Thus,

$$S = \frac{k(w_s t_{ms} + w_l N_m t_h)}{k(w_s t_{ms} + w_l N_m t_h) + N N_m(t_t + \frac{1}{3}t_{ee})} \qquad (62)$$

3.8.2 Token Passing Ring Protocol

Similar to the Token Passing Bus protocol, the minimum delay time is the same as the one when all messages are short ones, and the maximum delay time is the same as the one when all messages are long ones. The latter is the same as the one in section 3.7.2. or equation 38. The former is

$$T_{min} = t_{ms} + \frac{N+1}{2}t_b + \frac{1}{2}t_{ee} \qquad (63)$$

The average delay time is also considered in two seperate cases. When $N_m = 1$, the average delay time is the same as equation 10. in section 3.4 except that t_h is replaced by $w_s t_{ms} + w_l t_h$. That is,

$$T_{ave} = \frac{k+1}{2}(w_s t_{ms} + w_l t_h) + \frac{k-1}{2}t_t + Nt_b + \frac{2N-1}{2N}t_{ee} \qquad (64)$$

When $N_m > 1$, the average delay caused by any other active station in the first cycle is

$$t_b + w_s t_{ms} + w_l t_h + t_t + \frac{t_{ee}}{N}$$

In each of the next $(N_m - 2)$ cycles, each station will suffer delay of $(t_b + t_h + t_t + \frac{t_{ee}}{N})$ caused by $w_l k$ active stations and of $(t_b + \frac{t_{ee}}{N})$ casused by the rest. In the last cycle, a station which is $(i+1)$-th station seizing a token will suffer the delay time by $w_l i$ active stations preceding it on the loop and by half the number of inactive stations on the average. Hence, we have

$$T_{ave} \;=\; \frac{1}{k}\sum_{i=0}^{k-1}\{w_s[i(t_b + w_s t_{ms} + w_l t_h + t_t + \frac{t_{ee}}{N}) + \frac{N-k}{2}(t_b + \frac{t_{ee}}{N})$$

$$
\begin{aligned}
&+ t_b + t_{ms} + \frac{N-1}{2}t_b + \frac{1}{2}t_{ee}] \\
&+ w_l\{k(t_b + w_s t_{ms} + w_l t_h + t_t + \frac{t_{ee}}{N}) + (N-k)(t_b + \frac{t_{ee}}{N}) \\
&+ (N_m - 2)[w_l k(t_b + t_h + t_t + \frac{t_{ee}}{N}) + (N - w_l k)(t_b + \frac{t_{ee}}{N})] \\
&+ w_l i(t_b + t_h + t_t + \frac{t_{ee}}{N}) + \frac{N - w_l k}{2}(t_b + \frac{t_{ee}}{N}) + t_b + t_h + \frac{N-1}{2}t_b + \frac{1}{2}t_{ee}\}\}
\end{aligned}
$$

After simplification, we have

$$
\begin{aligned}
T_{ave} \quad = \quad & \frac{k+1}{2}(w_s t_{ms} + w_l t_h) + \frac{k-1}{2}t_t + N t_b + \frac{2N-1}{2N}t_{ee} \\
& + w_l(N_m - 1)[w_l k(t_h + t_t) + N t_b + t_{ee}] \\
& + w_s w_l(\frac{k+1}{2}t_{ms} + \frac{k+1}{2}t_t + \frac{1}{2}t_b + \frac{1}{2N}t_{ee}) \qquad (65)
\end{aligned}
$$

For throughput, the cycle time to transmit all k messages is

$$
\begin{aligned}
& k(t_b + w_s t_{ms} + w_l t_h + t_t + \tfrac{1}{N}t_{ee}) + (N-k)(t_t + \tfrac{1}{N}t_{ee}) \\
& + (N_m - 1)[w_l k(t_b + t_h + t_t + \tfrac{1}{N}t_{ee}) + (N - w_l k)(t_b + \tfrac{1}{N}t_{ee})]
\end{aligned}
$$

After simplification, it becomes

$$k(w_s t_{ms} + w_l N_m t_h + N_m t_t) + N_m(N t_b + t_{ee})$$

Thus,

$$S = \frac{k(w_s t_{ms} + w_l N_m t_h)}{k(w_s t_{ms} + w_l N_m t_h + N_m t_t) + N_m(N t_b + t_{ee})} \qquad (66)$$

3.8.3 Santa Clara Ring Protocol

Similar to the above protocols, the minimum delay time is the same as the one when all messages are short ones, and the maximum delay time is the same as the one when all messages are long ones. The latter is the same as the one in section 3.7.3. or equation 42. The former is

$$T_{min} = t_{ms} + \frac{1}{2}t_{ee} + \frac{k-1}{2}(t_{bp} + t_{ba}) \qquad (67)$$

When $N_m = 1$, the average delay time is the same as equation 15. in section 3.5 except that t_h is replaced by $w_s t_{ms} + w_l t_h$. That is,

$$T_{ave} = \frac{k+1}{2}(w_s t_{ms} + w_l t_h) + \frac{k-1}{2}t_{bi} + \frac{2k-1}{2k}t_{ee} + \frac{(k-1)(4k+1)}{12}(t_{bp} + t_{ba}) \quad (68)$$

When $N_m > 1$, the transmissions of short messages are terminated in the first cycle. In this first cycle, there are $[k - i + w_l(i-1)]$ active stations are left when the message with the i-th priority is transmitted. Hence, the delay caused by the message with the j-th priority is

$$w_s t_{ms} + w_l t_h + t_{bi} + \frac{1}{k}t_{ee} + [k - j + w_l(j-1)](t_{bp} + t_{ba})$$

If the i-th message is short one, it will suffer the delay caused by the other $(i-1)$ active stations with higher priorities. In addition, there is extra delay of

$$t_{ms} + \frac{1}{2}t_{ee} + \frac{k - i + w_l(i-1)}{2}(t_{bp} + t_{ba})$$

before it reaches the destination. Otherwise, this message will suffer the delay caused by all k active stations, and go through the other $(N_m - 1)$ cycles. In the remaining

$(N_m - 1)$ cycles, there are only $w_l k$ active stations are left. If $w_l k \geq 1$, the cycle time for each of the next $(N_m - 2)$ cycles is

$$w_l k[t_h + t_{bi} + \frac{t_{ee}}{w_l k} + w_l(k-1)(t_{bp} + t_{ba})]$$

Since the minimum propagation delay is t_{ee} for any cycle, the cycle time becomes

$$w_l k[t_h + t_{bi} + t_{ee} + w_l(k-1)(t_{bp} + t_{ba})]$$

if $w_l k < 1$. In the last cycle, there is probability of w_l that i-th message suffers delay cuased by each other active station with higher priority. This delay is

$$t_h + t_{bi} + \frac{t_{ee}}{w_l k} + (k-j)(t_{bp} + t_{ba})$$

when $w_l k \geq 1$, If $w_l k < 1$, the last term becomes t_{ee}. The delay for transmitting the i-th message and its associated propagation is

$$t_h + \frac{1}{2}t_{ee} + \frac{w_l k - w_l i}{2}(t_{bp} + t_{ba})$$

Combining the above delays, we have

$$\begin{aligned}
T_{ave} \;=\; & \frac{1}{k}\sum_{i=1}^{k}\{w_s \sum_{j=1}^{i-1}[w_s t_{ms} + w_l t_h + t_{bi} + \frac{1}{k}t_{ee} + (k - j + w_l(j-1))(t_{bp} + t_{ba})] \\
& + t_{ms} + \frac{1}{2}t_{ee} + \frac{k - i + w_l(i-1)}{2}(t_{bp} + t_{ba}) \\
& + w_l\{\sum_{j=1}^{k}[w_s t_{ms} + w_l t_h + t_{bi} + \frac{1}{k}t_{ee} + (k - j + w_l(j-1))(t_{bp} + t_{ba})] \\
& + (N_m - 2)w_l k[t_h + t_{bi} + \frac{t_{ee}}{w_l k} + w_l(k-1)(t_{bp} + t_{ba})] \\
& + w_l \sum_{j=1}^{i-1}[t_h + t_{bi} + \frac{t_{ee}}{w_l k} + (k-j)(t_{bp} + t_{ba})] + t_h + \frac{1}{2}t_{ee} + \frac{w_l k - w_l i}{2}(t_{bp} + t_{ba})\}\}
\end{aligned}$$

for $w_l k \geq 1$. Or,

$$\begin{aligned} T_{ave} &= \frac{1}{k}\sum_{i=1}^{k}\{w_s\sum_{j=1}^{i-1}[w_s t_{ms} + w_l t_h + t_{bi} + \frac{1}{k}t_{ee} + (k-j+w_l(j-1))(t_{bp}+t_{ba})] \\ &+ t_{ms} + \frac{1}{2}t_{ee} + \frac{k-i+w_l(i-1)}{2}(t_{bp}+t_{ba})\} \\ &+ w_l\{\sum_{j=1}^{k}[w_s t_{ms} + w_l t_h + t_{bi} + \frac{1}{k}t_{ee} + (k-j+w_l(j-1))(t_{bp}+t_{ba})) \\ &+ (N_m-2)w_l k[t_h + t_{bi} + t_{ee} + w_l(k-1)(t_{bp}+t_{ba})] \\ &+ w_l\sum_{j=1}^{i-1}[t_h + t_{bi} + t_{ee} + (k-j)(t_{bp}+t_{ba})] + t_h + \frac{1}{2}t_{ee} + \frac{w_l k - w_l i}{2}(t_{bp}+t_{ba})\}\} \end{aligned}$$

for $w_l k < 1$. After simplification, we have

$$\begin{aligned} T_{ave} &= \frac{k+1}{2}(w_s t_{ms} + w_l t_h) + \frac{k-1}{2}t_{bi} + \frac{2k-1}{2k}t_{ee} + \frac{(k-1)(4k+1)}{12}(t_{bp}+t_{ba}) \\ &+ w_l(N_m-1)[w_l k(t_h+t_{bi}) + t_{ee} + w_l^2 k(k-1)(t_{bp}+t_{ba})] \\ &+ w_s w_l[\frac{k+1}{2}(t_{ms}+t_{bi}) + \frac{k-1}{12}(4k-2+12w_l k)(t_{bp}+t_{ba})] \end{aligned} \qquad (69)$$

for $w_l k \geq 1$. Or,

$$\begin{aligned} T_{ave} &= \frac{k+1}{2}(w_s t_{ms} + w_l t_h) + \frac{k-1}{2}t_{bi} + \frac{2k-1}{2k}t_{ee} + \frac{(k-1)(4k+1)}{12}(t_{bp}+t_{ba}) \\ &+ w_l(N_m-1)[w_l k(t_h+t_{bi}) + w_l k t_{ee} + w_l^2 k(k-1)(t_{bp}+t_{ba})] \\ &+ w_s w_l[\frac{k+1}{2}(t_{ms}+t_{bi}) + \frac{k-1}{12}(4k-2+12w_l k)(t_{bp}+t_{ba}) \\ &+ w_l(\frac{1}{k}-w_l)\frac{k+1}{2}t_{ee}] \end{aligned} \qquad (70)$$

for $w_l k < 1$.

For throughput, $S = 1$ or the equation 16. when $k = 1$. Otherwise, there are two cases depending on the value of N_m. When $N_m = 1$, the cycle time becomes

$$\sum_{i=1}^{k}[w_s t_{ms} + w_l t_h + t_{bi} + \frac{t_{ee}}{k} + (k-i)(t_{bp} + t_{ba})]$$

or

$$k(w_s t_{ms} + w_l t_h + t_{bi}) + t_{ee} + \frac{k(k-1)}{2}(t_{bp} + t_{ba})$$

Thus,

$$S = \frac{k(w_s t_{ms} + w_l t_h)}{k(w_s t_{ms} + w_l t_h + t_{bi}) + t_{ee} + \frac{k(k-1)}{2}(t_{bp} + t_{ba})} \tag{71}$$

When $N_m > 1$, the cycle time to transmit all k messages becomes

$$\sum_{i=1}^{k}\{w_s t_{ms} + w_l t_h + t_{bi} + \frac{1}{k}t_{ee} + [k - i + w_l(i-1)](t_{bp} + t_{ba})\}$$

$$+(N_m - 2)w_l k[t_h + t_{bi} + \frac{t_{ee}}{w_l k} +$$

$$+w_l \textstyle\sum_{i=1}^{k}[t_h + t_{bi} + \frac{t_{ee}}{w_l k} + ($$

for $w_l k \geq 1$. Or,

$$\sum_{i=1}^{k}\{w_s t_{ms} + w_l t_h + t_{bi} + \frac{1}{k}t_{ee} + [k - i + w_l(i-1)](t_{bp} + t_{ba})\}$$

$$+(N_m - 2)w_l k[t_h + t_{bi} + t_{ee} + \imath$$

$$+w_l \textstyle\sum_{i=1}^{k}[t_h + t_{bi} + t_{ee} + (k$$

for $w_l k < 1$. After simplification, this cycle time, T'_{cycle}, becomes

$$T'_{cycle} = k(w_s t_{ms} + w_l t_h + t_{bi}) + t_{ee} + \frac{k(k-1)}{2}(t_{bp} + t_{ba})$$

$$+ (N_m - 1)[w_l k(t_h + t_{bi}) + t_{ee} + w_l^2 k(k-1)(t_{bp} + t_{ba})]$$

$$+ w_s w_l k(k-1)(t_{bp} + t_{ba})$$

for $w_l k \geq 1$. Or,

$$\begin{aligned} T'_{cycle} &= k(w_s t_{ms} + w_l t_h + t_{bi}) + t_{ee} + \frac{k(k-1)}{2}(t_{bp} + t_{ba}) \\ &\quad + (N_m - 1)[w_l k(t_h + t_{bi}) + w_l k t_{ee} + w_l^2 k(k-1)(t_{bp} + t_{ba})] \\ &\quad + w_s w_l k(k-1)(t_{bp} + t_{ba}) \end{aligned}$$

Therefore, we have

$$S = \frac{k(w_s t_{ms} + w_l N_m t_h)}{T'_{cycle}} \tag{72}$$

3.8.4 Star Protocol

Since circuit switching and non-blocking network are used for Star protocol, the performance is simliar to the ones in the previous sections. The minumum delay time is the same as the one in section 3.6. except that t_h is replaced by t_{ms}. That is,

$$T_{min} = t_w + \frac{N}{2} t_{ms} + 2\frac{t_{ee}}{N} \tag{73}$$

The maximun delay times are exactly the same as the ones in section 3.7.4.

The average delay time are the same as the ones in section 3.6 except that t_h is replaced by $w_s t_{ms} + w_l N_m t_h$.

$$T_{ave} = t_w + \frac{N}{2}(w_s t_{ms} + w_l N_m t_h) + 2\frac{t_{ee}}{N}$$

$$\text{for } N = 20,\ k \le 10 \tag{74}$$

$$T_{ave} = \frac{2k-10}{k}[t_w + \frac{N}{2}(w_s t_{ms} + w_l N_m t_h) + \frac{t_{ee}}{N}] + \frac{t_{ee}}{N}$$

$$\text{for } N = 20,\ k > 10 \tag{75}$$

$$T_{ave} = t_w + \frac{N}{2}(w_s t_{ms} + w_l N_m t_h) + 2\frac{t_{ee}}{N} + \frac{2N - 2N_c + 2}{N-1}(t_w + \frac{t_{ee}}{N})$$

$$\text{for } N = 100 \tag{76}$$

$$T_{ave} = t_w + N_m \frac{N}{2}(w_s t_{ms} + w_l N_m t_h) + 2\frac{t_{ee}}{N} + \frac{3N - N_c^2 + 1}{N-1}(t_w + \frac{t_{ee}}{N})$$

$$\text{for } N = 500 \tag{77}$$

$$T_{ave} = t_w + N_m \frac{N}{2}(w_s t_{ms} + w_l N_m t_h) + 2\frac{t_{ee}}{N} + \frac{4N - 2N_c^2 + 2N_c}{N-1}(t_w + \frac{t_{ee}}{N})$$

$$\text{for } N = 1000 \tag{78}$$

The throughput are also the same as the ones in section 3.6 except that t_h is replaced by $w_s t_{ms} + w_l N_m t_h$. That is,

$$S = \frac{k(w_s t_{ms} + w_l N_m t_h)}{\frac{N}{2}(w_s t_{ms} + w_l N_m t_h) + t_w + 2\frac{t_{ee}}{N}} \qquad \text{for } N = 20,\ k \le 10 \tag{79}$$

$$S = \frac{k(w_s t_{ms} + w_l N_m t_h)}{N(w_s t_{ms} + w_l N_m t_h) + 2t_w + 3\frac{t_{ee}}{N}} \qquad \text{for } N = 20,\ k > 10 \tag{80}$$

$$S = \frac{k(w_s t_{ms} + w_l N_m t_h)}{\frac{N}{2}(w_s t_{ms} + w_l N_m t_h) + 3t_w + 4\frac{t_{ee}}{N}} \qquad \text{for } N = 100 \tag{81}$$

$$S = \frac{k(w_s t_{ms} + w_l N_m t_h)}{\frac{N}{2}(w_s t_{ms} + w_l N_m t_h) + 4t_w + 5\frac{t_{ee}}{N}} \qquad \text{for } N = 500 \tag{82}$$

$$S = \frac{k(w_s t_{ms} + w_l N_m t_h)}{\frac{N}{2}(w_s t_{ms} + w_l N_m t_h) + 5t_w + 6\frac{t_{ee}}{N}} \qquad \text{for } N = 1000 \tag{83}$$

4 Comparison of Protocols

This section presents a comparison of the four high speed protocols under consideration: the Token bus, the Token Ring, the Santa Clara Ring, and the Star. These protocols are compared in light of the following four parameters: delay time, throughput, reliability/maintainability, and maturity of standards and tools.

4.1 Delay Time

The delay time is a measure of performance from the user's point of view. The analysis of each protocol is described in section 3. In this subsection, we show the results computed from the equations derived in the previous section. Table 1 and 2 show the average delay for the four protocols for R = 2.5 Gbps and N = 100. These tables also reflect the effect on delay when the holding time and message size change.

Figure 6, 7, and 8 show the maximum, average, and minimum delays, respectively, for $R = 2.5$ Gbps, $N = 100$, and $w_l = 1.0$. From these figures, it is clear that the delay performance of the Star is very poor compared to the other three protocols. This is due to the fact that the data rate of each link is small as compared to other protocols. As mentioned in section 3.6, the data rate of each link is only $2R/N$, an assumption that had to be made to allow fair comparison of the Star to the other three protocols. Notice that the aggregate data rate of a single star is maintained at a value equal to the data rates used in the other three protocols. A smaller data rate

k	Token Bus		Token Ring		Santa Clara Ring		Star	
	t_h		t_h		t_h		t_h	
	5 ms	0.5 ms	5 ms	0.5 ms	5 ms	0.5 ms	5 ms	0.5 ms
1	13.42	163.46	5.50	10.00	5.25	9.75	250.02	250.02
5	23.42	182.46	15.50	29.00	15.45	28.95	250.02	250.02
10	35.92	206.21	28.00	52.75	27.98	52.73	250.02	250.02
15	48.42	229.96	40.50	76.50	40.48	76.49	250.02	250.02
20	60.92	253.71	53.00	100.26	52.99	100.24	250.02	250.02

Table 1. Average delay in $msec$, $R = 2.5$ Gbps, $N = 100$, and $w_l = 1.0$

k	Token Bus		Token Ring		Santa Clara Ring		Star	
	t_h		t_h		t_h		t_h	
	5 ms	0.5 ms	5 ms	0.5 ms	5 ms	0.5 ms	5 ms	0.5 ms
1	8.93	23.54	1.01	1.06	0.76	0.45	25.67	25.67
5	9.96	23.85	2.04	1.37	1.99	1.11	25.67	25.67
10	11.24	24.23	3.32	1.75	3.30	1.73	25.67	25.67
15	12.52	24.62	4.60	2.14	4.59	2.12	25.67	25.67
20	13.81	25.01	5.88	2.52	5.87	2.51	25.67	25.67

Table 2. Average delay in $msec$, $R = 2.5$ Gbps, $N = 100$, and $w_l = 0.1$

in turn gives a longer message transmission time. In our application environment, the message transmission time is the dominant component of delay, due to the long message size. Hence, the delay time of a Star protocol is affected by N and becomes longer when N increases. Figure 9, 10, and 11 show the average delay for $R = 2.5$ Gbps, $w_l = 1.0$, and different values of $N = [500, 1000, 20]$.

The figures also point out that the delay performance of the Token Bus protocol is much poorer than the ones for the Token Ring or the Santa Clara Ring. This longer delay is due to the longer network access time. Each station needs to wait for the token to arrive fully before transmitting a message. The walk-time for a token from one station to its successor is equal to the token transmission time plus 1/3 the end-to-end propagation time. The delay will be affected by the accumulated walk-times for the token going through all the N stations.

Figure 12 through 15 show the average delay for $N = 100$, $w_l = 1.0$, and different values of $R = [0.5, 1.5, 3.5, 4.5]$ Gbps.

These figures indicate that the delay performance for both the Token Ring and the Santa Clara Ring are almost the same. However, they have much difference when the holding time is reduced or when there are messages with mixed sizes. Figure 16 through 18 show the average delay for $R = 2.5$ Gbps, $N = 100$, $w_l = 1.0$, and different values of $t_h = [0.5, 1.0, 2.5]$ $msec$. Figure 19 through 22 show the average delay for $R = 2.5$ Gbps, $N = 100$, $w_l = 0.1$, and different values of $t_h = [0.5, 1.0, 2.5, 5.0]$ $msec$. In the above figures, t_{ms} is 14.4 μsec and corresponds to the message with 4500 $k$bytes.

Under light traffic conditions, each station in Token Ring protocol needs to wait for the token to arrive before transmitting a message. While stations in Santa Clara Ring can transmit messages as soon as the ring is free. Under heavy traffic conditions, both protocols provide a round-robin service. However, the delay time of the Token Ring is affected by N, since every station on the ring will cause a 1 bit delay to traveling message. The delay for the Santa Clara Ring is affected by k, and is independent of N. Only active stations may cause delays to a traveling message around the ring.

4.2 Throughput

Throughput is a measure of performance from the systems point of view. Section 3 shows the analysis of the four candidate protocols. Table 3 and 4 show a summary of the normalized throughput for the four protocols under study.

Figure 23 compares the normalized throughput of the four protocols for $R = 2.5$ Gbps, $N = 100$, and $w_l = 1.0$. The throughput of a Star is significantly inferior to the other three protocols. The Star will do well when the majority of stations attached to a single star cluster are active!

Figure 24, 25, and 26 show the throughputs of the protocols for $R = 2.5$ Gbps, $w_l = 1.0$, and different $N = [500, 1000, 20]$. Figure 27 through 30 show the throughputs when $N = 100$, $w_l = 1.0$, and for different $R = [0.5, 1.5, 3.5, 4.5]$ Gbps.

From these figures, we can see that the throughput of Token Bus protocol is

k	Token Bus		Token Ring		Santa Clara Ring		Star	
	t_h		t_h		t_h		t_h	
	5 ms	0.5 ms	5 ms	0.5 ms	5 ms	0.5 ms	5 ms	0.5 ms
1	0.231	0.029	0.909	0.500	1.000	1.000	0.020	0.020
5	0.600	0.130	0.980	0.833	0.980	0.833	0.100	0.100
10	0.750	0.231	0.990	0.909	0.990	0.909	0.200	0.200
15	0.818	0.310	0.993	0.937	0.993	0.937	0.300	0.300
20	0.857	0.375	0.995	0.952	0.995	0.952	0.400	0.400

Table 3. Normalized throughput for $R = 2.5$ Gbps, $N = 100$, and $w_l = 1.0$

k	Token Bus		Token Ring		Santa Clara Ring		Star	
	t_h		t_h		t_h		t_h	
	5 ms	0.5 ms	5 ms	0.5 ms	5 ms	0.5 ms	5 ms	0.5 ms
1	0.030	0.003	0.506	0.093	1.000	1.000	0.020	0.020
5	0.133	0.015	0.837	0.339	0.837	0.483	0.100	0.100
10	0.235	0.030	0.911	0.506	0.911	0.506	0.200	0.200
15	0.316	0.044	0.939	0.606	0.939	0.606	0.300	0.300
20	0.381	0.058	0.953	0.672	0.953	0.672	0.400	0.400

Table 4. Normalized throughput for $R = 2.5$ Gbps, $N = 100$, and $w_l = 0.1$

inferior to the ones in both the Token Ring and Santa Clara Ring protocols.

Again, the throughput performance of both the Token Ring and the Santa Clara Ring is almost a match, especially for large k. Under light traffic, though, each station in Token Ring protocol spends extra time waiting for the token to arrive, while in the Santa Clara Ring protocol stations do not have to wait, hence resulting in a better throughput performance. In particular, the normalized throughput of the Santa Clara Ring protocol is 1, when there is only one active station on the ring.

Figure 31 through 33 show the normalized throughput for $R = 2.5$ Gbps, $N = 100$, $w_l = 1.0$, and different values of $t_h = [0.5, 1.0, 2.5]\, msec$. Figure 34 through 37 show the average delay for $R = 2.5$ Gbps, $N = 100$, $w_l = 0.1$, and different values of $t_h = [0.5, 1.0, 2.5, 5.0]\, msec$.

4.3 Reliability/Maintainability

- Token Bus: The Token Bus forms a logical ring. This ring must be initialized. Some cooperative, decentralized algorithm is needed to determine who goes first. Addition and deletion of stations can be achieved easily if the bus is implemented using a coaxial cable and passive couplers and taps. This can be done without disrupting the network. A number of errors can occur during token transfer. Lost and duplicate tokens must be properly handled with suitable algorithms. This adds to the cost of the bus interface hardware/software.

- Token Ring: Traditionally the Token Ring, since its first inception, had three engineering problems. However, over the years, they have been solved through elegant and straightforward methods. The first problem is the reliability of the repeater string. A failure in any one repeater can disrupt the entire local network. This is solved by arranging the transmission links between successive nodes so that each internode link loops through a central point, a wire center. At the wire center, bypass relays that are energized remotely by the network stations can do the majority of the reconfiguration operations automatically. The resulting configuration, a star-shaped ring, creates a centralized location for maintenance and reliability, without compromising the distributed nature of the ring control. The second problem is that of distributed initialization and recovery. An algorithm is required whereby all active repeaters can quickly and simply agree upon the need for initialization and recovery. One approach is to assign a priority as to which station should attempt to reinitialize the ring first. This technique also applies in case of duplicate or lost tokens on the ring. The third problem is that of closed-loop clock coordination. The issue here is that not only must the collection of repeaters agree on a common clock rate, but that a clock rate must result in a integral number of bit times of delay when traversing the closed ring. This problem is easily solved by opening the ring when originating a message, and thereby allowing all non-originating repeaters to track the originator. Nevertheless, during the times when a free token circulates around the ring, one is left with the choice of either closing the ring or supporting the circulating token through the last station that released it. A sophisticated approach has been introduced using a phase-locked-loop in each repeater tracking its preceding neighbor, and a loop filter so that the resulting ring of PLL's is stable.

- Santa Clara Ring: The Santa Clara Ring is a tokenless protocol. Hence it does not exhibit token related problems found in the Token Ring protocol. This include the problem of closed loop clock coordination, since any time there is a transmission on the ring there is an originator. The only problem the Santa Clara Ring suffers from is the problem of the reliability of the repeater string. Again, this problem is solved through the use of the star-shaped ring with active by-passing relays.

- Star: In a Star architecture, all nodes are joined at a single point called the central node or hub. The central node establishes a dedicated path between any two devices that wish to communicate. The obvious problem with any Star network is the single point of failure. If the central switch fails, it can bring down the entire network. Addition and deletion of nodes on a Star network is disruptive. It requires a change in central switch configuration. Generally, the central node consists of a crossbar switch. The crossbar switch has a number of disadvantages: The number of crosspoints grows with the square of the number of connected nodes. This is costly for a large number of nodes and results in high capacitive loading on any message path. The loss of a crosspoint prevents the connection between the two devices involved. These problems can be overcome, at additional cost, by adding multiple stages as well as copies of the switch.

4.4 Maturity of Standards and Tools

- Token Bus: The Token Bus protocol is well established and the IEEE 802.4 Token Bus standard is accepted both by manufactures of the network components and the users. Tools and equipments to design and monitor networks based on this protocol are readily available.

- Token Ring: As in the Token Bus, the Token Ring is also well established and the IEEE 802.5 Token Ring standard is accepted by manufacturers of the network components and the users. A number of network products based on IEEE 802.5 are readily available. Tools and equipments to design and monitor networks based on this protocol are readily available as well.

- Santa Clara Ring: The Santa Clara Ring is a new protocol and has no standard. It is home grown at Santa Clara University. Tools and equipments used for Token Ring protocol can be used to design and monitor networks based on this protocol. Our local experience with the details of the Santa Clara Ring protocols makes it more appealing than the other three protocols.

- Star: The Star is the oldest of all the protocols described here. Different configurations have been established and used by the telephone industry. There is no standard though. Many different configurations are possible and could lead to different performance and reliability. Tools and equipments are readily available to develop and monitor networks based on this protocol.

5 Conclusion

We have presented four different protocols that are candidates for adoption for ultra high speed local area networking. The performance evaluation has led us to an interesting conclusion. The Token Passing Bus is definitely a loser under the conditions of our test. The Star configuration depends to a great extent on the number of arms it has. It also is very sensitive to the eventual hierarchical topology that one has to choose, once the number of nodes on the LAN exceeds the number of arms on a single star. It is clear that if one would assume that each arm in the Star configuration carries the same bandwidth of the other LAN topologies under study, the Star configuration would have an unfair advantage in providing an aggregate speed way higher than the single channel speed. This implies that the active device in the center of the Star would be significantly more expensive than a typical LAN controller attached to any node. The bandwidth for each arm in the Star was chosen to be equal to $\frac{1}{a}$ of total capacity of the LAN, where a is the number of star arms in a single star. Under such conditions, the Star loses to the other three protocols, namely: the Santa Clara Ring, the Token Passing Ring, and the Token Bus.

The quantitative performance measures show both the Santa Clara Ring and the Token Passing Ring offering almost exact results! The Santa Clara Ring does slightly better with smaller configurations than the Token Passing Ring. As the number of active stations increases, the performance is the same for both protocols! In order to make the final decision, it is important to examine the other qualitative issues, such as maturity of the protocol, reliability, and ease of implementation.

Our extensive analysis and design work that has been done during previous projects with the Santa Clara Ring, gives the Santa Clara Ring the advantage in that respect. It will mean that the development cycle for a Santa Clara Ring network will be shorter for us! The Token Passing Ring on the other hand is a mature standard as opposed to the Santa Clara Ring that was only implemented and simulated at SCU. The Santa Clara Ring has the advantage of eliminating the use of a token that puts an extra burden on the LAN for recovering from lost and duplicate tokens. One may then argue that the Santa Clara Ring may have an extra level of inherent reliability. This would result into a reduction of the complexity of the implemented controller, hence a more cost effective choice. The Santa Clara Ring does not suffer from the problem of closed loop clock coordination, the Token Passing Ring suffers from.

We conclude that the Santa Clara Ring should be the target protocol for this project, admitting that we could have equally chosen the Token Passing Ring as an alternative. Choosing the Santa Clara Ring will allow us to introduce to the community a new protocol that has excellent performance, with possibly a more cost effective implementation than the Token Passing Ring.

References

[1] S. Joshi, V. Iyer. "New Standards for Local Networks Push Upper Limits for Lightwave Data", *Data Communications* (July 1984), pp. 127-138.

[2] A. Brosio ,F. Gagliardi, L. Lambarelli, G. Panatotto, D. Roffinella, and M. Sposini. "A Reconfigurable High-Speed Optical System for Integrated Local Communication", *IEEE Journal on Selected Areas in Communication*, SAC-3.6, p. 825-834 (1985)

[3] A. Danthine. "A Backbone Wideband Network for LAN Interconnection", *Proc. of EFOC/LAN '86*, Orlando (November 1986).

[4] A. Luvison, G. Roullet, and F. Toft. "A high-Capacity Multi-Service Local Area Network", *Proc. of EFOC/LAN '86* , Orlando (November 1986).

[5] W. K. Giloi, P. Behr, and G. Zuber. "UPPERBUS: A High-speed Backbone for Metropolitan Area Networks", *Proc. of EFOC/LAN '86*, Amsterdam (June 1986).

[6] W. Chou. Computer Communications, Volume I, Principles, Prentice-Hall, Englewood Cliffs, NJ, 1983, pp. 299-300.

[7] R. I. Sharp, "The LAN-DTH 140 Mbit/s Token Ring", *High Speed Local Networks*, Elsevier Science Publishers, B. V. (North-Holland) IFIP 1987, pp. 213-225

[8] R. I. Sharp, N. N. Pedesen, and C. Jensen. "Services and Protocols in High-speed LANs", *Proc. of Euteco*, Varese, October 1983, pp. 425-433.

[9] D. V. Mierop. "System FINEX: The First Implementation of a High-Speed Networking (FDDI) Standard", *High Speed Local Area Networks*, Elsevier Science Publishers B.V., (North-Holland), IFIP 1987, pp. 241-249

[10] A. Luvison. "A High-capacity Multiservice Local Area Network, LION", *High Speed Local Area Networks*, Elsevier Science Publishers B.V., (North-Holland), IFIP, 1987, pp. 251-269.

[11] F. Ferrero. "Architectures and System Design of a High Bandwidth Multi-Service LAN", *Proc. of the 4th EFOC and LAN Exposition, EFOC/LAN '86*, Amsterdam, the Netherlands, June 23-27, 1986, pp. 88-92.

[12] L. A. Bergman. "A Synchronous Fiber Optic Ring Local Area Network for Multi-Gigabit/s Mixed-Traffic Communication", *IEEE Journal on Selected Areas in Communication*, Vol. SAC-3, No. 6, November 1985, pp 842-848.

[13] T. Minami. "200 Mbit/s Synchronous TDM Loop Optical LAN Suitable for Multi-Service Integration", *Proc. of the IEEE Globecom '85*, New Orleans, pp.15-20.

[14] H. Goto. "A 1.2 Gbit/s Optical Loop LAN for Wideband Office Communications", *Proc. of the IEEE Globecom '85*, New Orleans, pp. 462-467.

[15] M. P. Varga. "A Robust 100 Mbps Network for Avionics Application, *Proc. of the IEEE NAECOM'85*, Dayton, Ohio, May 1985.

[16] R. J. Folmar. "Distributed Memory Network: An 8 Gigabit Fiber Optic Tightly Coupled System", *Proc. of the IEEE NAECOM'85*, Dayton, Ohio, May 1985, pp 91-94.

[17] P. C. Fetteroff, K. J. Ho, and J. J. Shintani. "IMAGENET: A High Speed Local Area Network", *Proc. of the IEEE MILCOM'88*, San Diego, California, 1988, Vol. 1, pp 19-23.

[18] Floyed E. Ross, "FDDI - an Overview", *Proc. of the Compcon Spring'87*, San Francisco, Feb. 1987, pp 434-440.

[19] Howard C. Salwen, Linda C. Perry, and Nathan K. Salwen. "Design Considerations for High Speed Local Area Networks", *Proc. of the IEEE International Conference on Communications 85*, Chicago, 1985, pp. 672-680.

[20] Joseph L. Hammond, and Peter J. P. O'Reilly, Performance Analysis of Local Computer Networks, Addison-Wesley, 1986.

[21] Sonia R. Sachs, Kai-li Kan, and John A. Silvester. "Token-Bus Protocol Performance Analysis and Comparison with Other LAN Protocols", *Proc. of the IEEE Globecom'85, New Orleans*, 1985, pp. 1492-1498.

[22] D. Deel. "VectorNet", *Proc. of the Compcon Spring'89*, San Francisco, Feb. 1989, pp 311-313.

[23] Howard W. Johnson, "Effective Performance in High Speed Networking", *Proc. of the Compcon Spring'89*, San Francisco, Feb. 1989, pp 306-310.

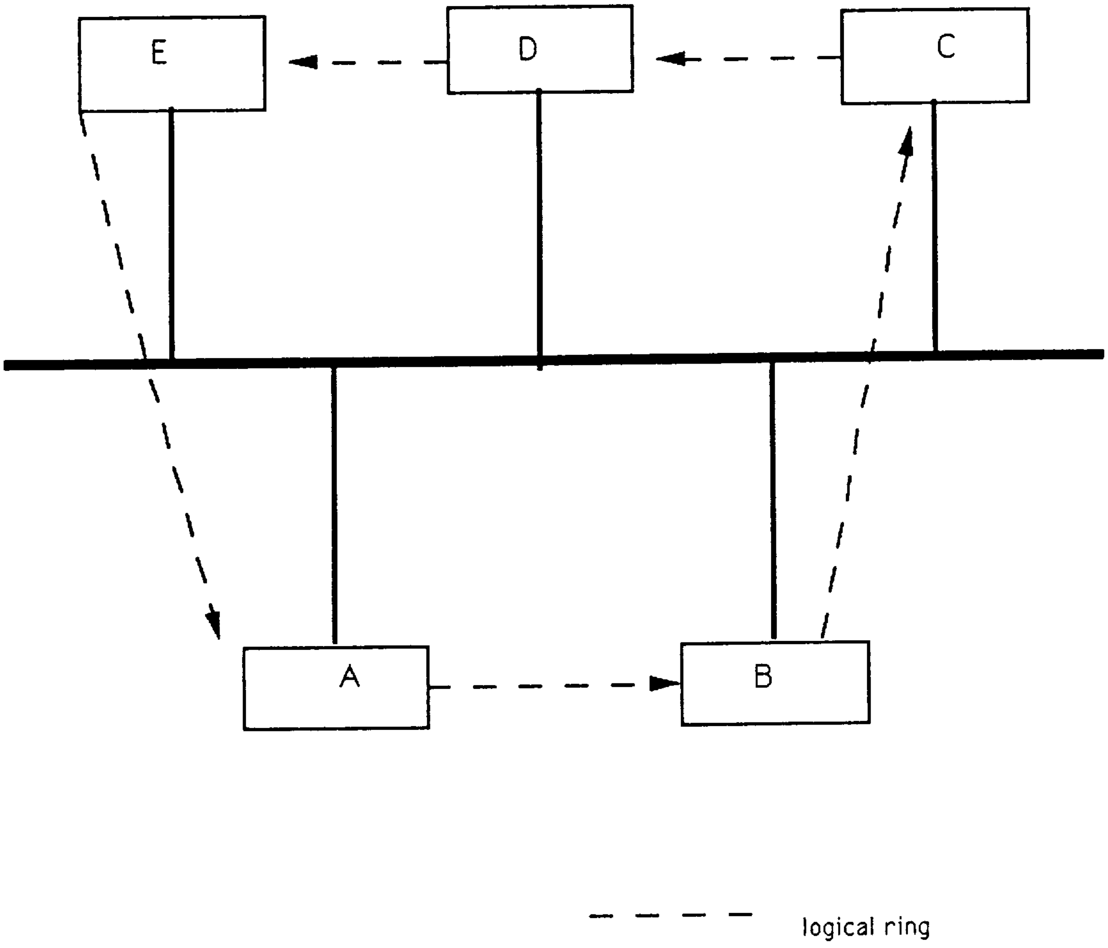


Figure 1. The structure for token-passing bus protocol.

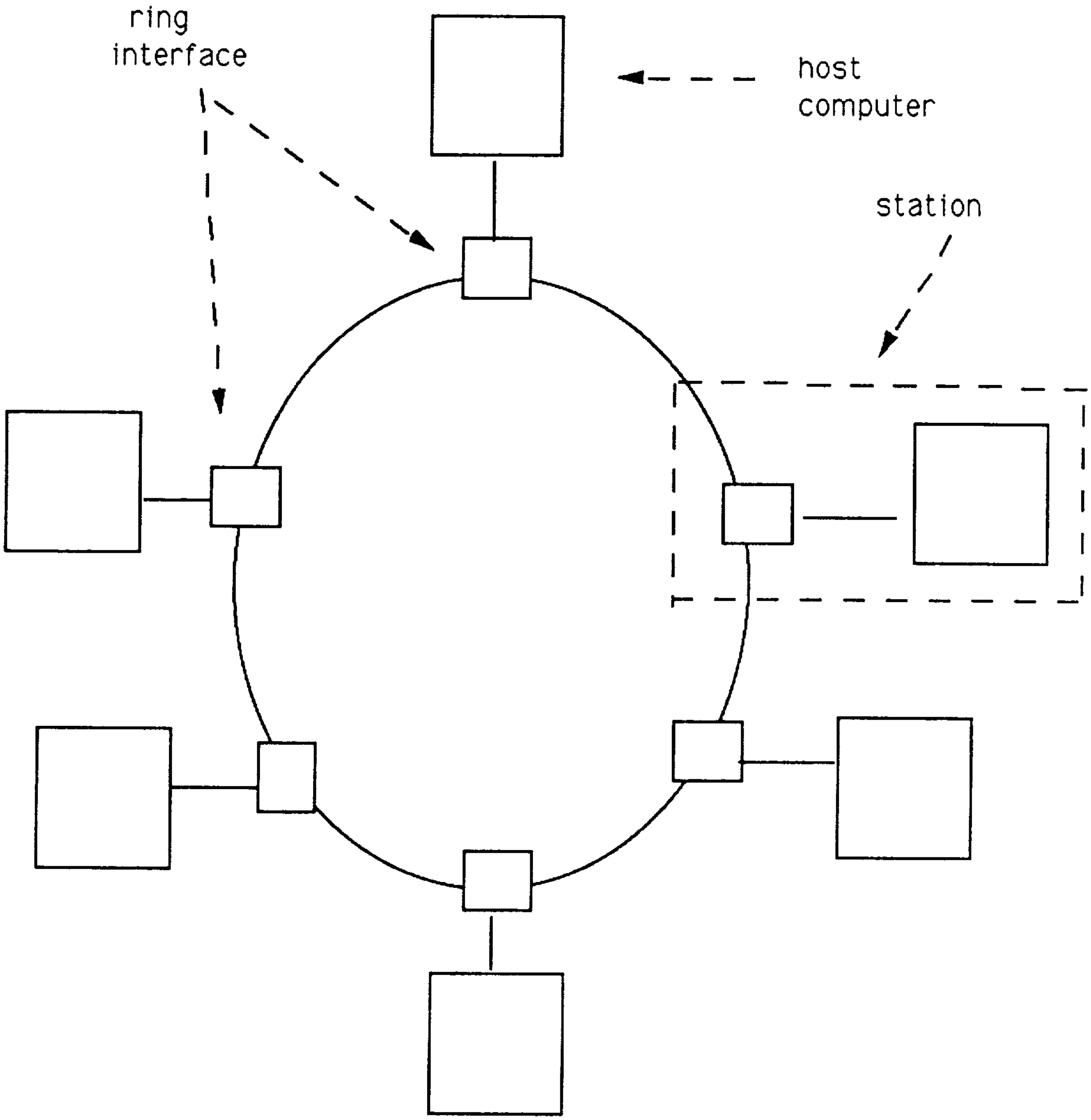
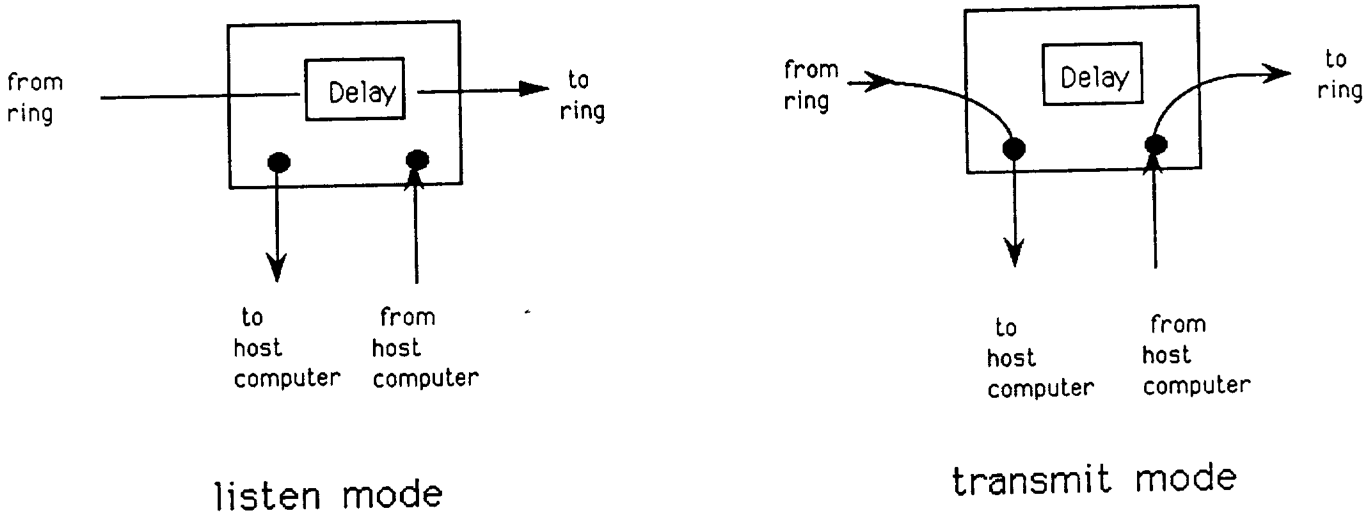


Figure 2. Structure for token-passing ring protocol



listen mode

transmit mode

Figure 3. Structure of ring interface in listen mode and transmit mode for token-passing ring protocol.

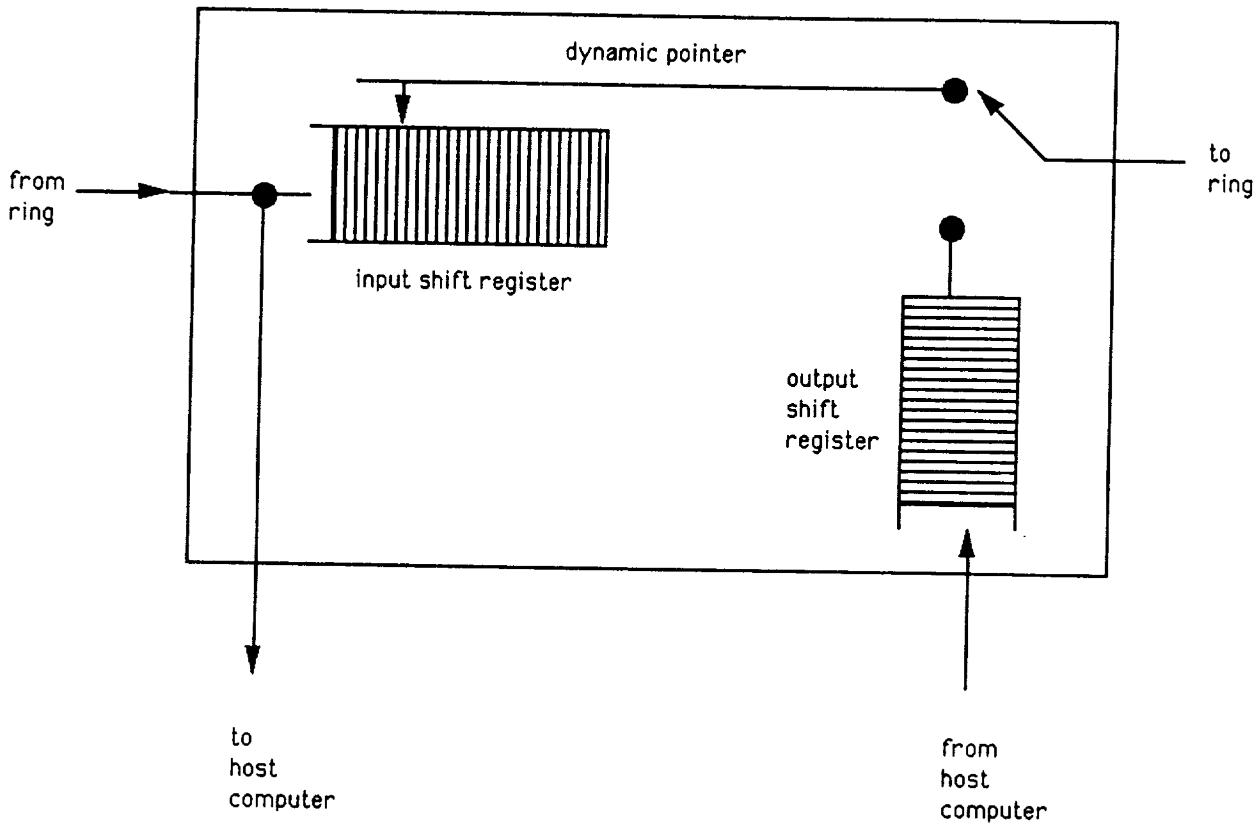


Figure 4. Structure of ring interface for Santa Clara ring protocol

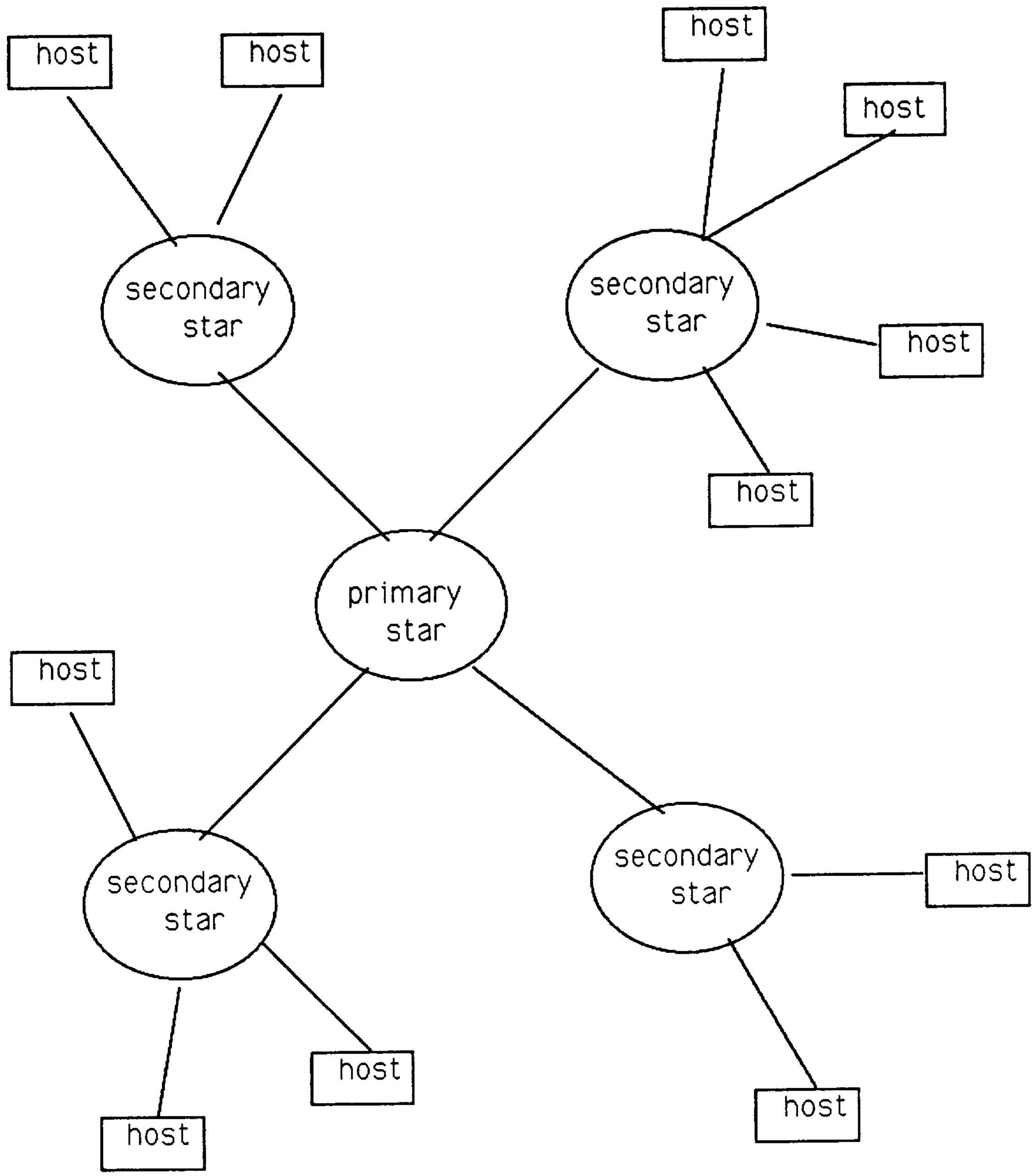


Figure 5. Structure for star protocol.

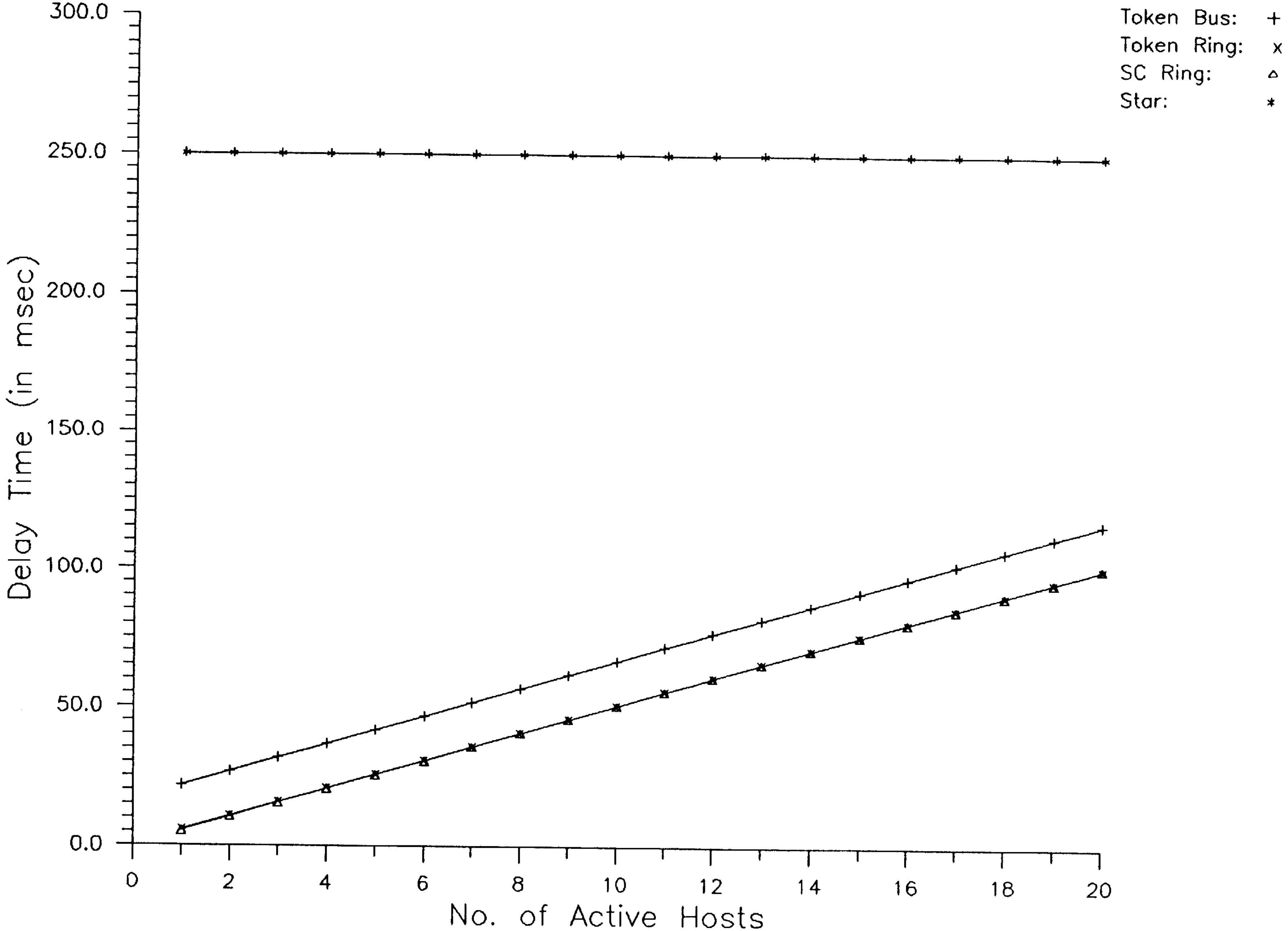
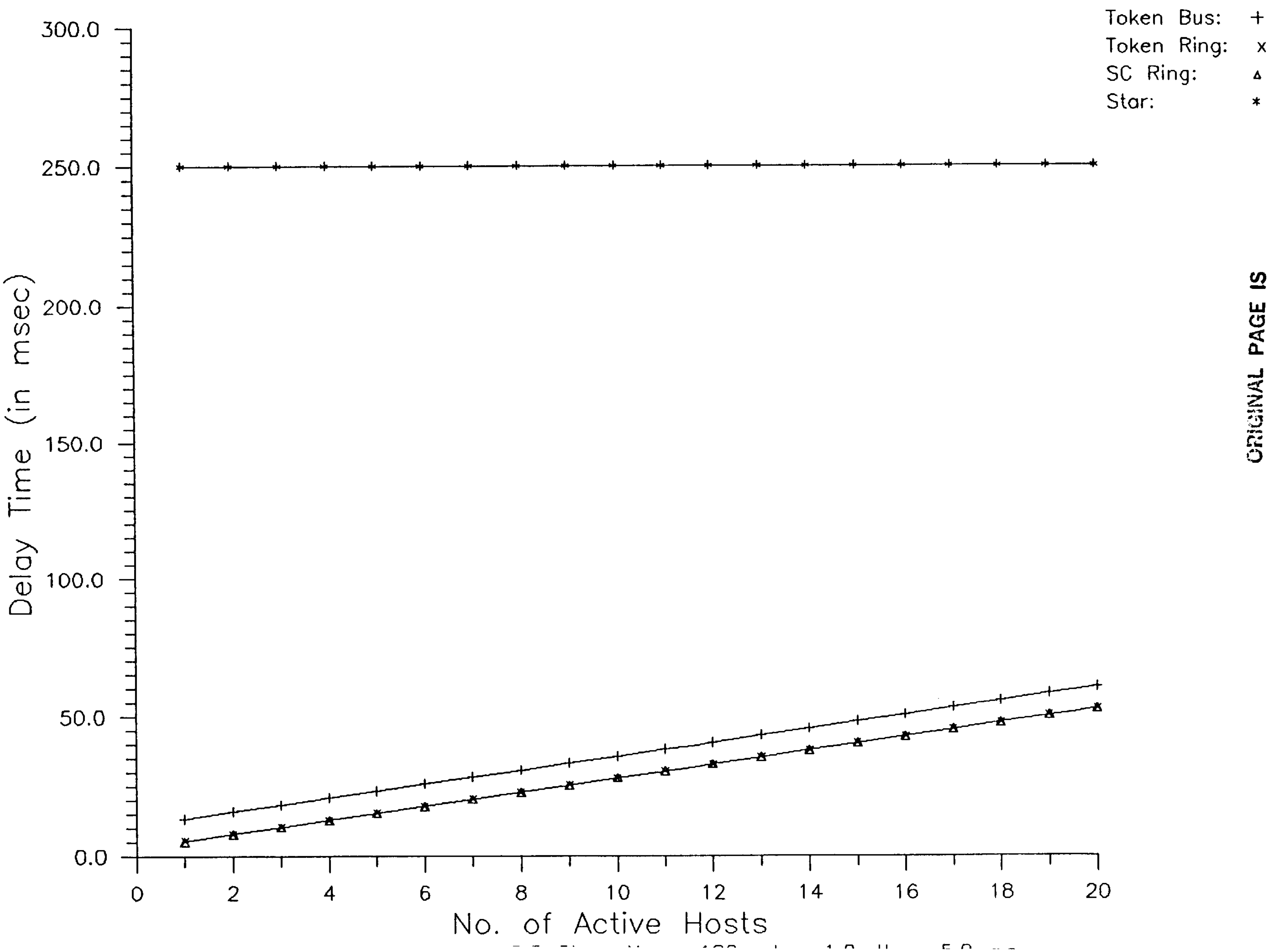


Figure 6. Maximum delay time for R = 2.5 Gbps, N = 100, wl = 1.0, th = 5.0 ms.


Token Bus: +
Token Ring: x
SC Ring: ▵
Star: *
Delay Time (in msec)
300.0
250.0
200.0
150.0
100.0
50.0
0.0
0
2
4
6
8
10
12
14
16
18
20
No. of Active Hosts

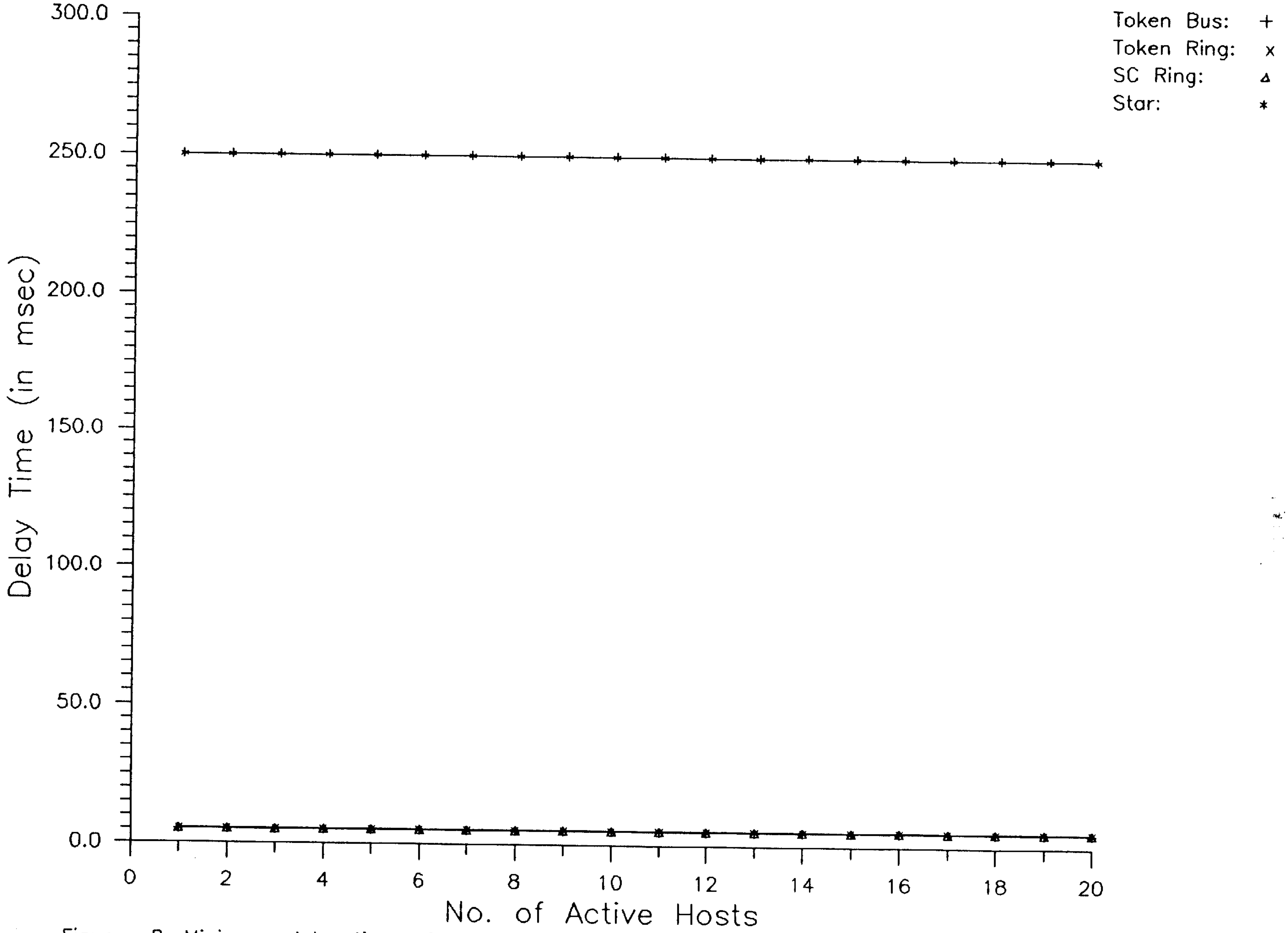
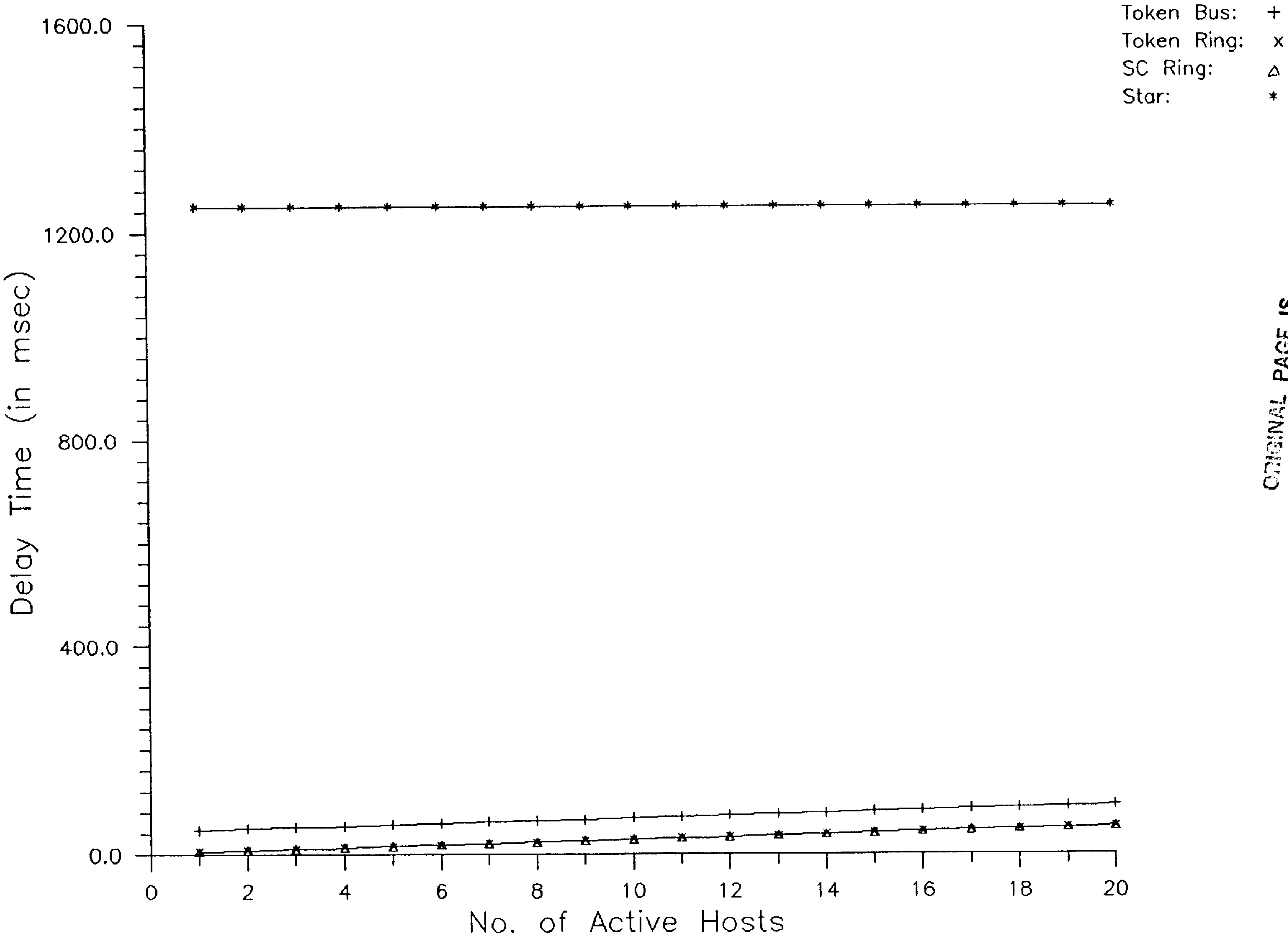


Figure 8. Minimum delay time for R = 2.5 Gbps, N =100, wl = 1.0, th = 5.0 ms.


Delay Time (in msec)
1600.0
1200.0
800.0
400.0
0.0
0
2
4
6
8
10
12
14
16
18
20
No. of Active Hosts
Token Bus: +
Token Ring: x
SC Ring: △
Star: *

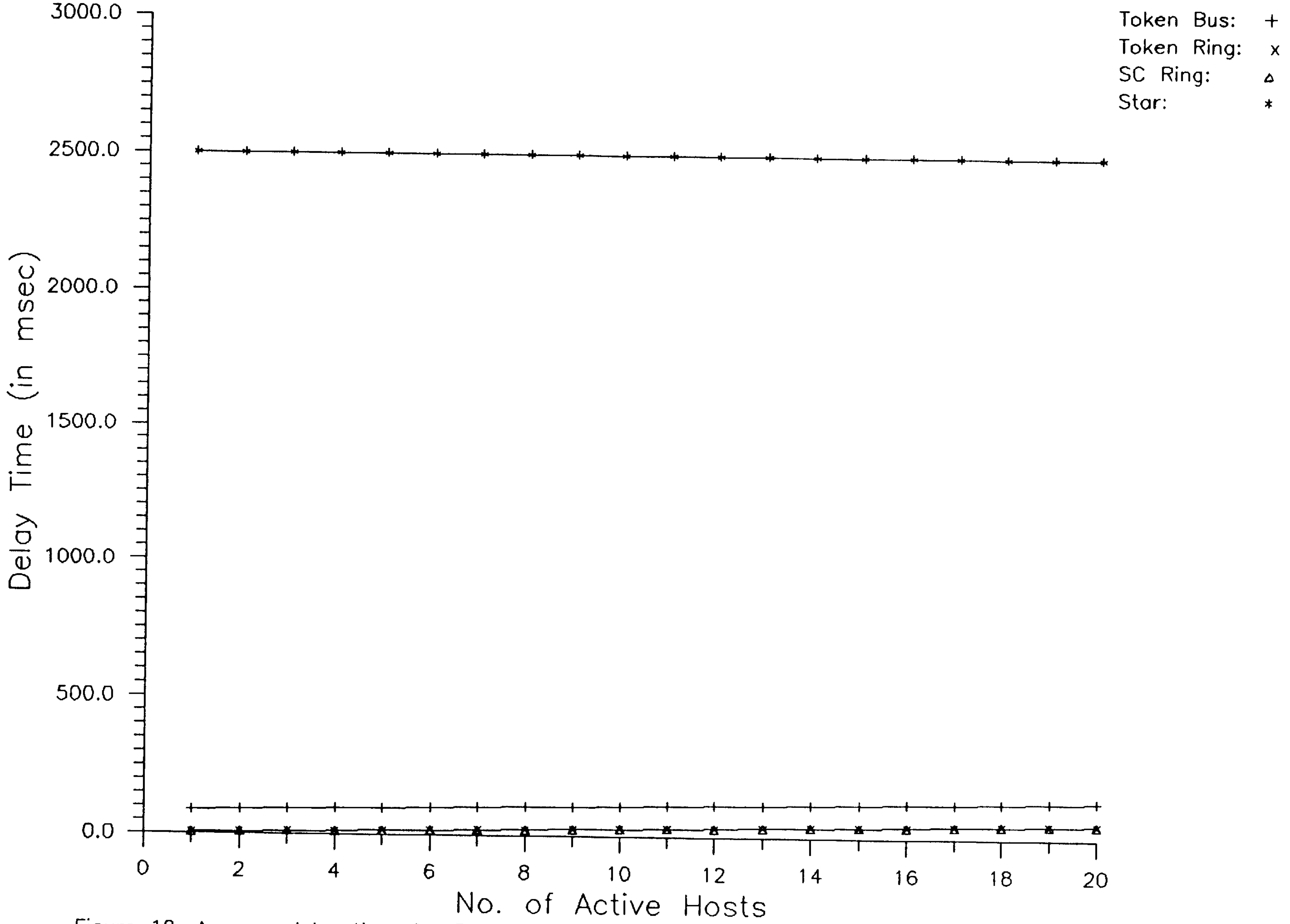
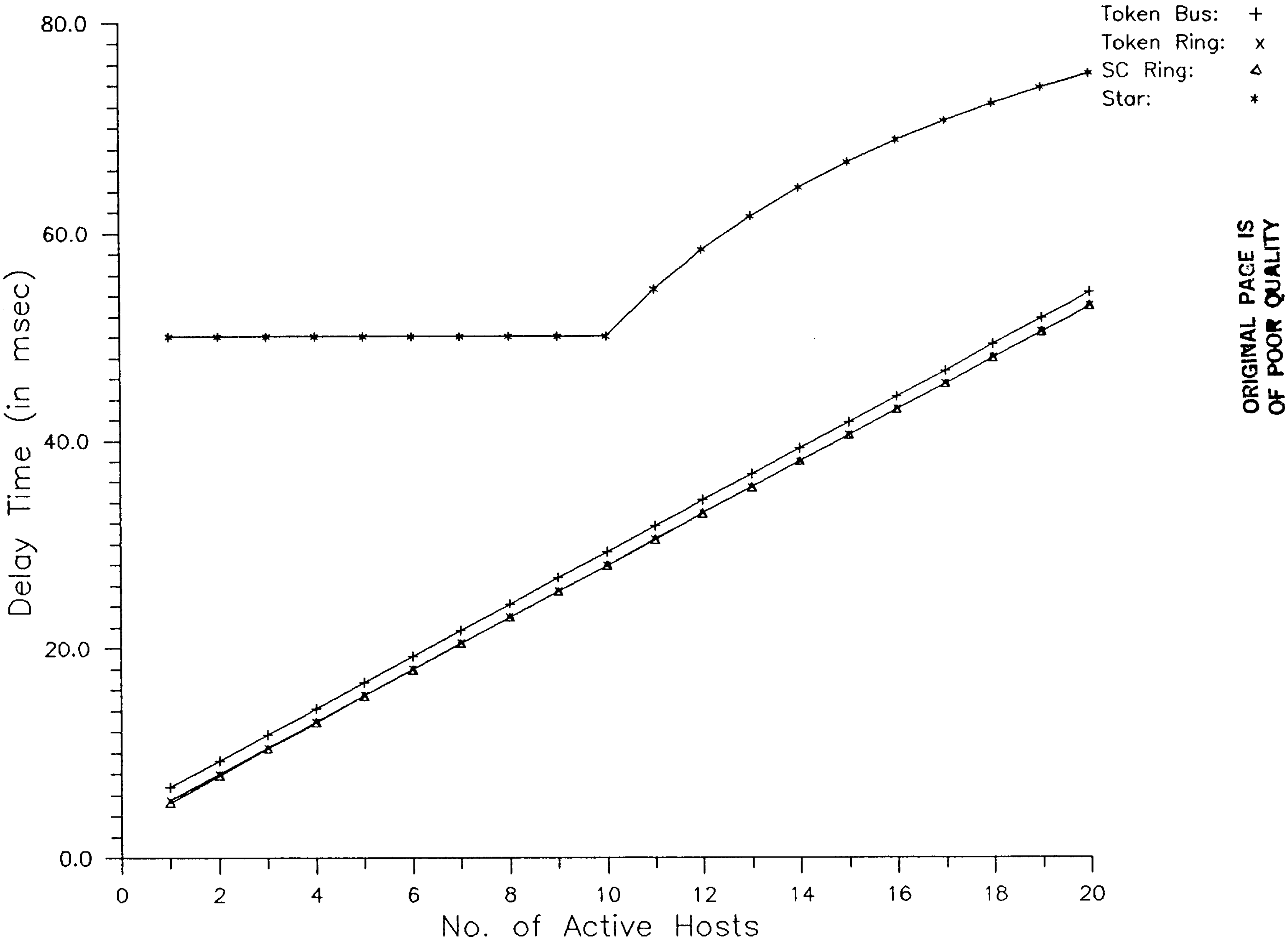


Figure 10. Average delay time for R = 2.5 Gbps, N = 1000, wl = 1.0, th = 5.0 ms.


Delay Time (in msec)
80.0
60.0
40.0
20.0
0.0
0
2
4
6
8
10
12
14
16
18
20
No. of Active Hosts
Token Bus: +
Token Ring: x
SC Ring: ∆
Star: *

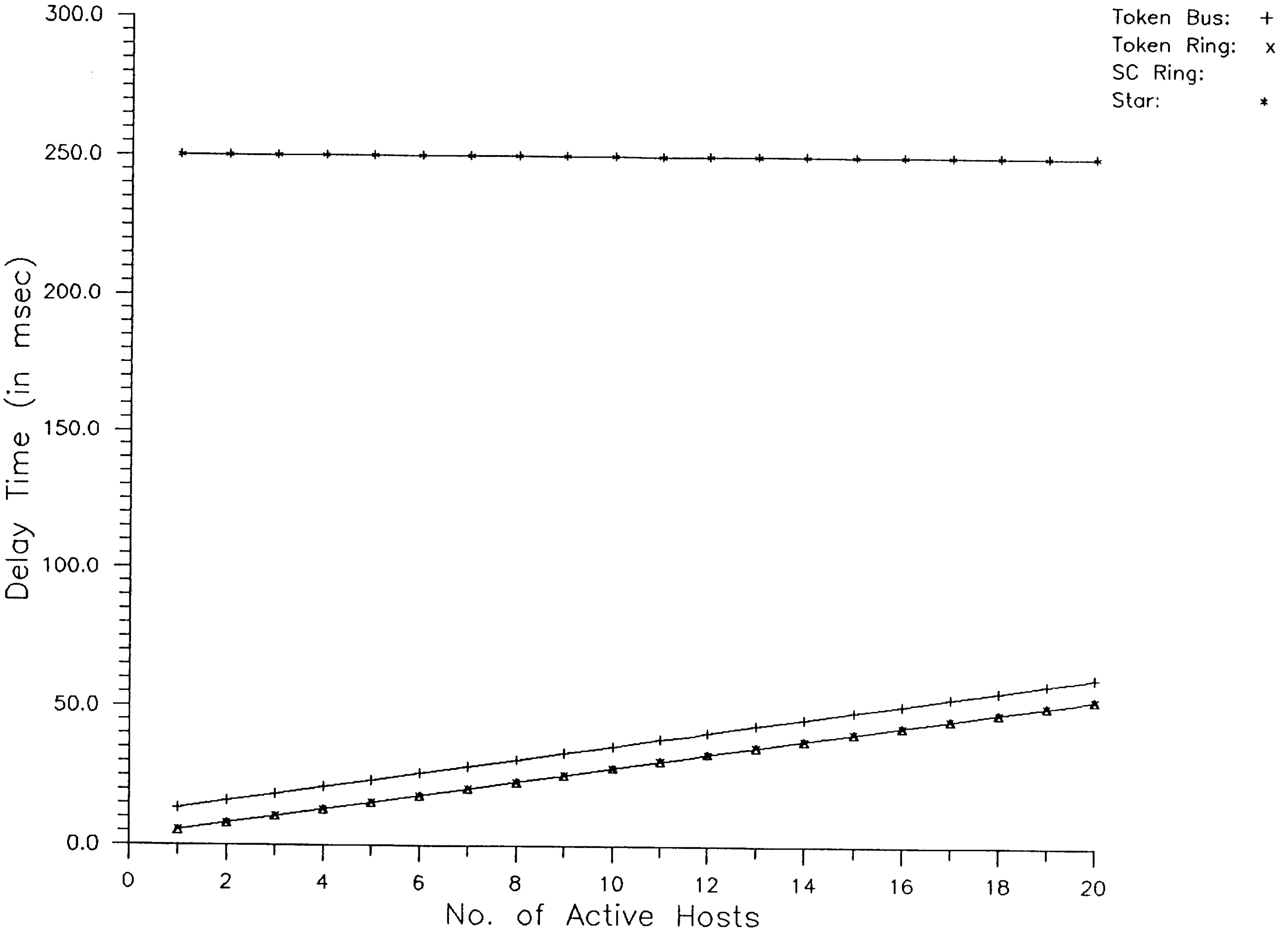
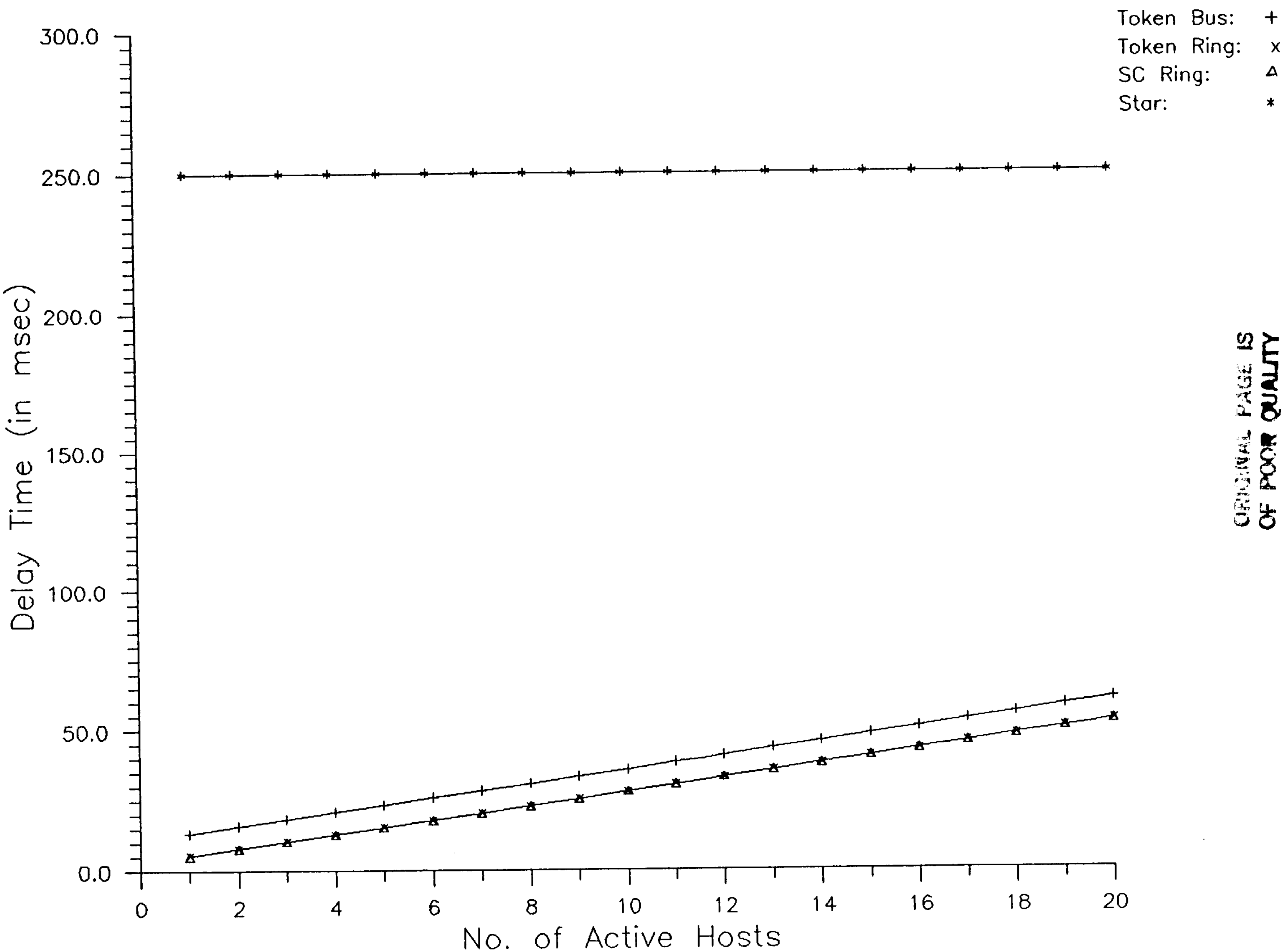


Figure 14. Average delay time for R = 3.5 Gbps, N = 100, wl = 1.0, th = 5.0 ms.


Token Bus: +
Token Ring: x
SC Ring: ∆
Star: *
Delay Time (in msec)
300.0
250.0
200.0
150.0
100.0
50.0
0.0
0
2
4
6
8
10
12
14
16
18
20
No. of Active Hosts

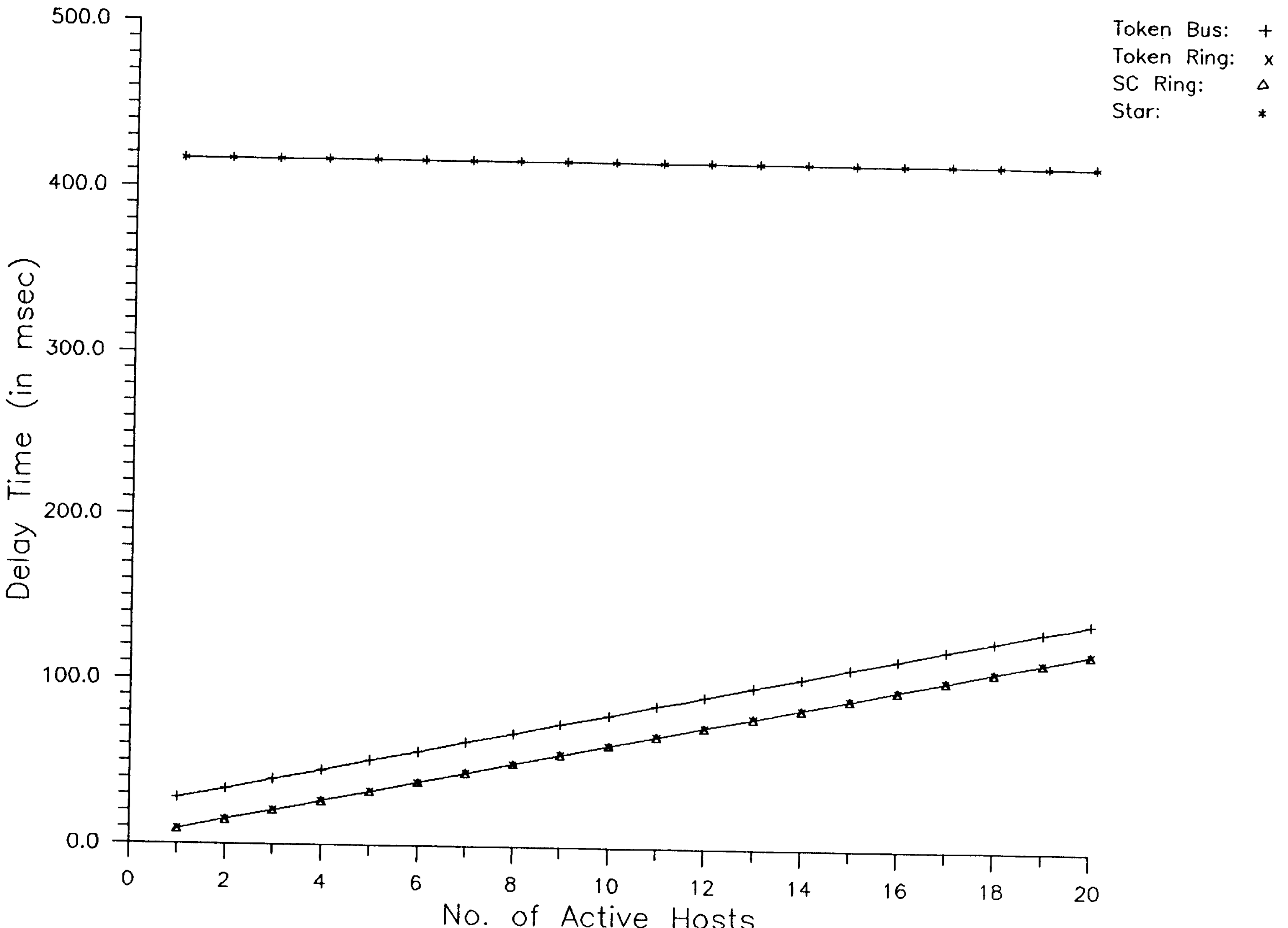
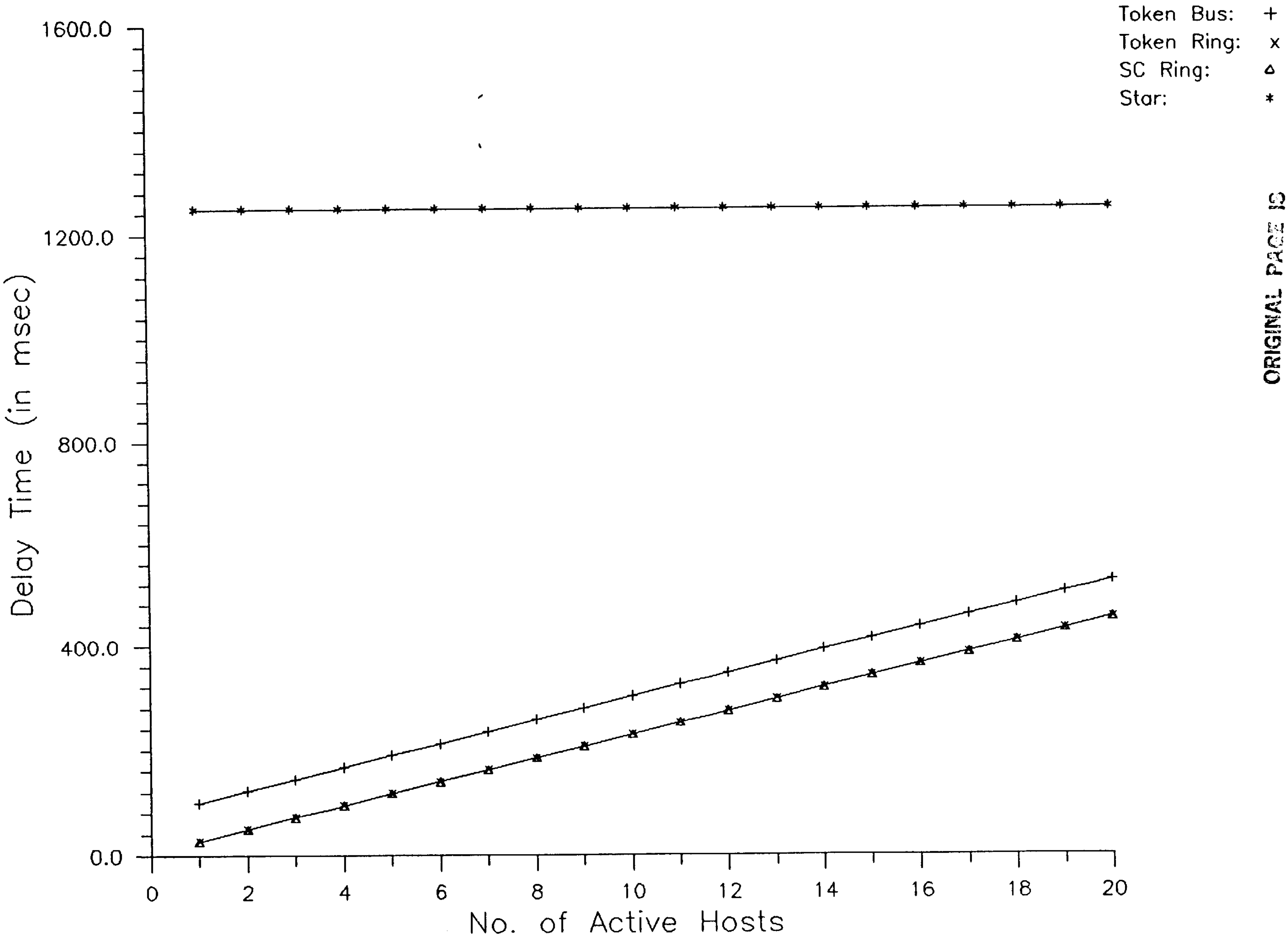


Figure 13. Average delay time for R = 1.5 Gbps, N = 100, wl = 1.0, th = 5.0 ms.



[illegible] for B = 2.5 Gbps N = 100 wl = 1.0 th = 5.0 ms

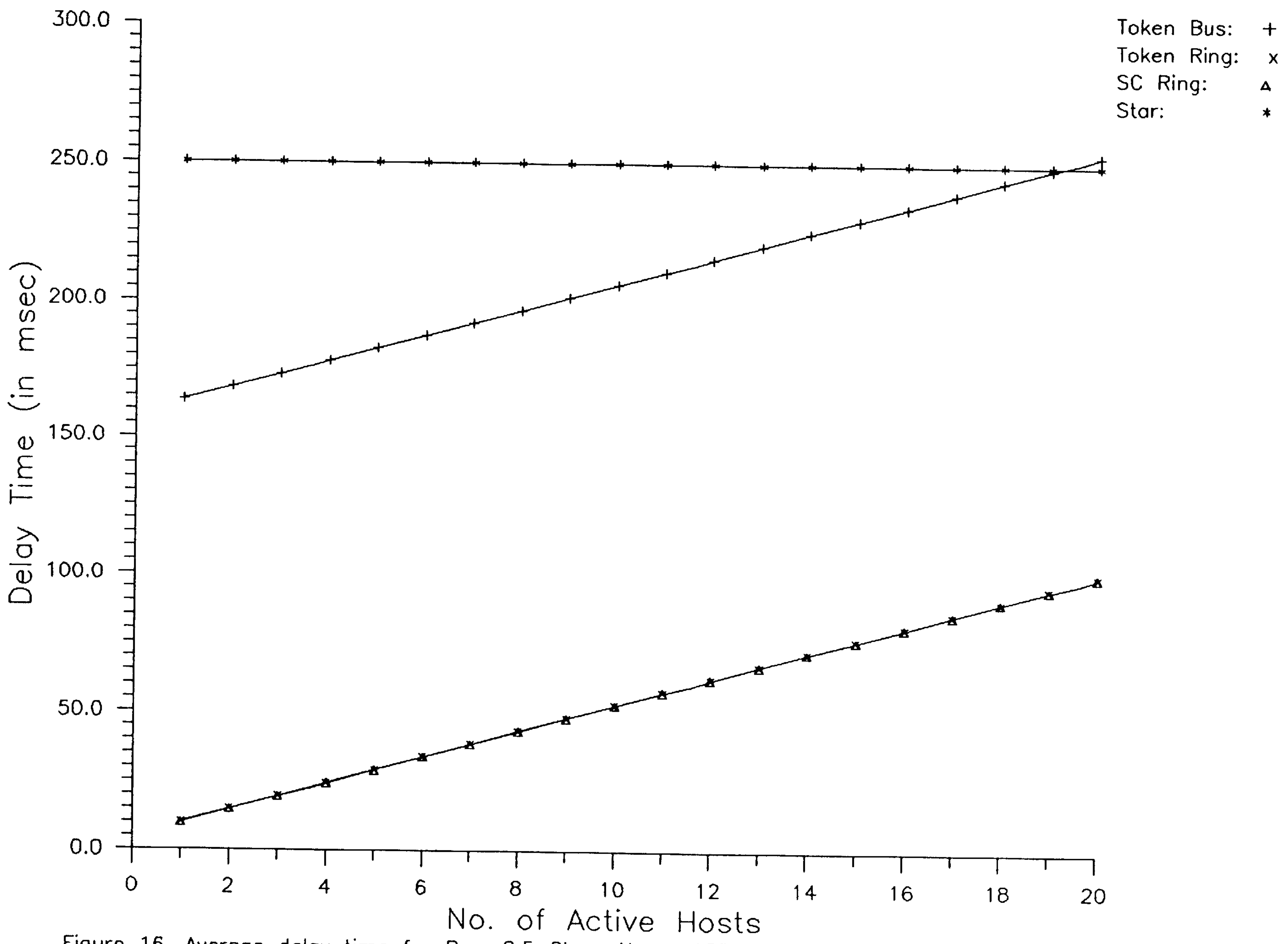
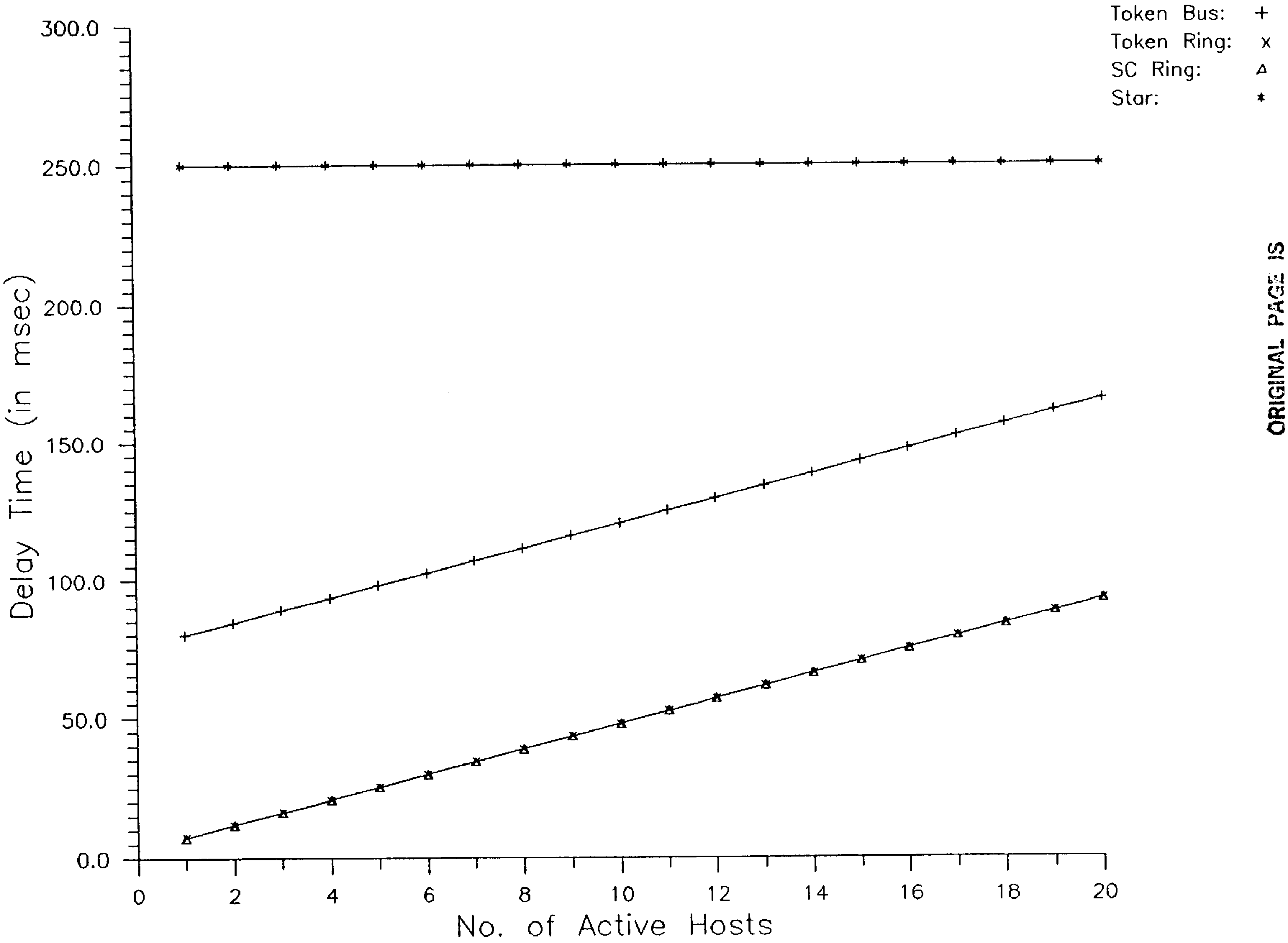


Figure 16. Average delay time for R = 2.5 Gbps, N = 100, wl = 1.0, th = 0.5 ms.


Token Bus: +
Token Ring: x
SC Ring: Δ
Star: *
Delay Time (in msec)
300.0
250.0
200.0
150.0
100.0
50.0
0.0
0
2
4
6
8
10
12
14
16
18
20
No. of Active Hosts

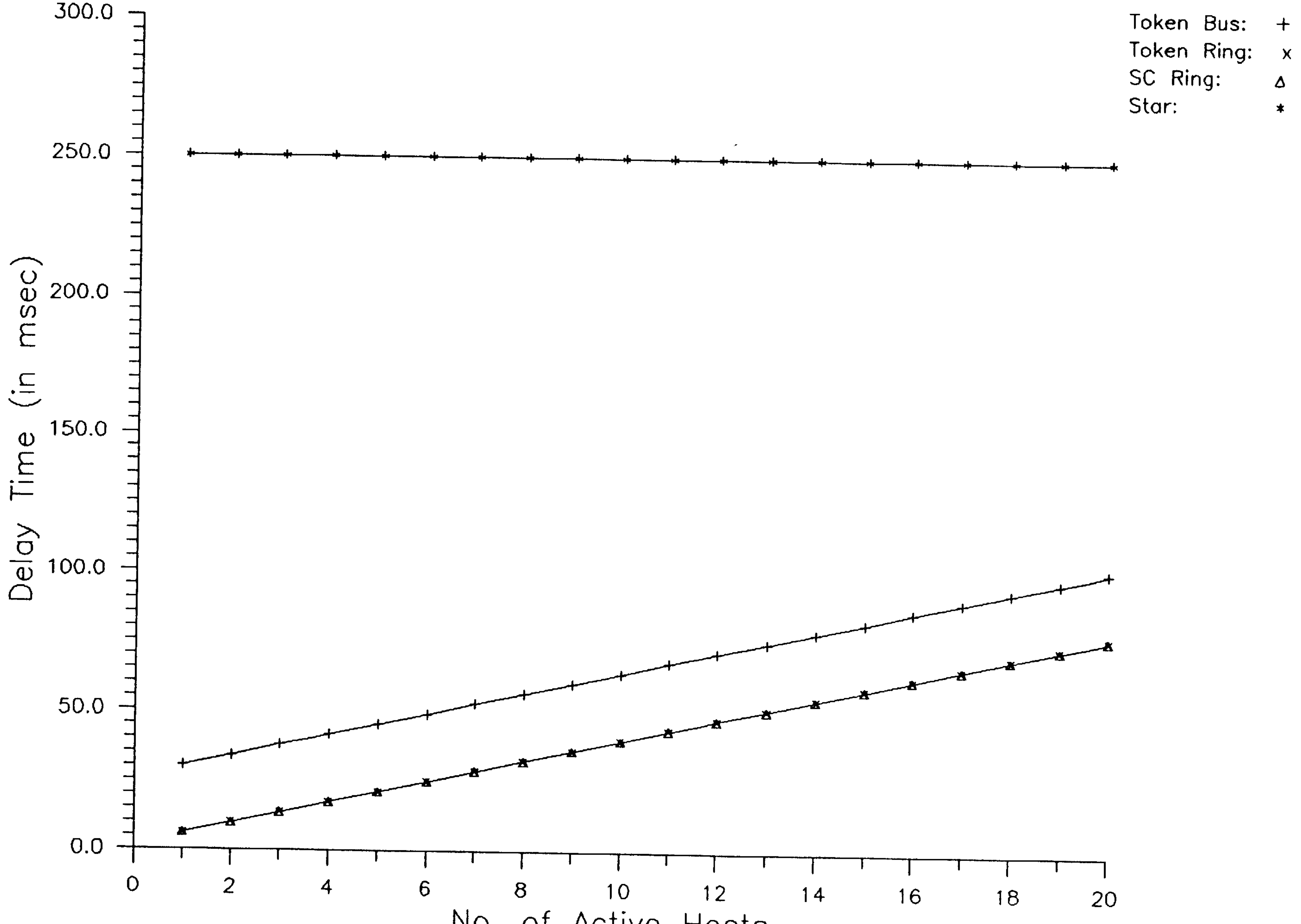
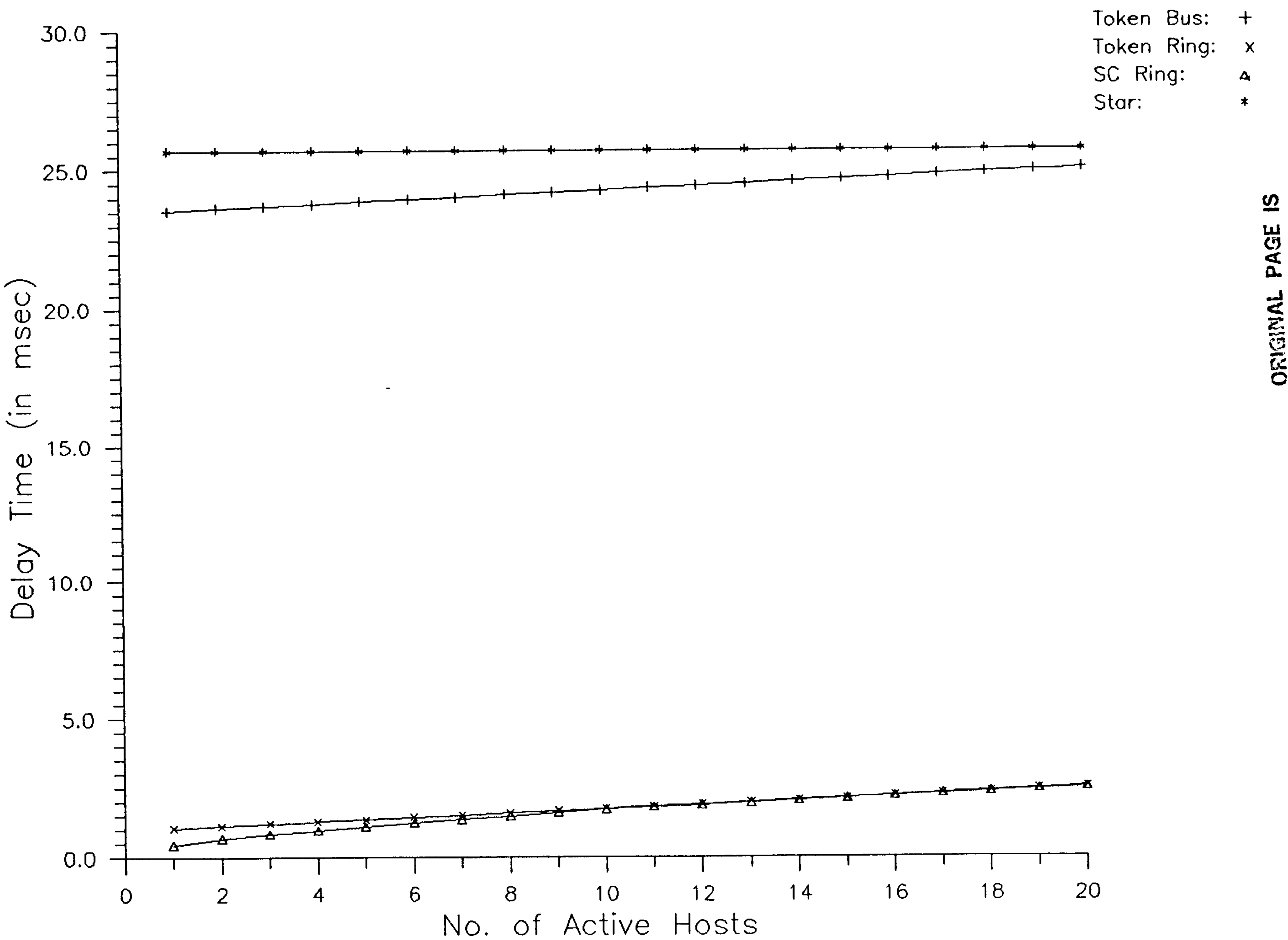


Figure 18. Average delay time for R = 2.5 Gbps, N = 100, wl = 1.0, th = 2.5 ms.


Delay Time (in msec)
30.0
25.0
20.0
15.0
10.0
5.0
0.0
0
2
4
6
8
10
12
14
16
18
20
No. of Active Hosts
Token Bus: +
Token Ring: x
SC Ring: ▵
Star: *

[illegible]

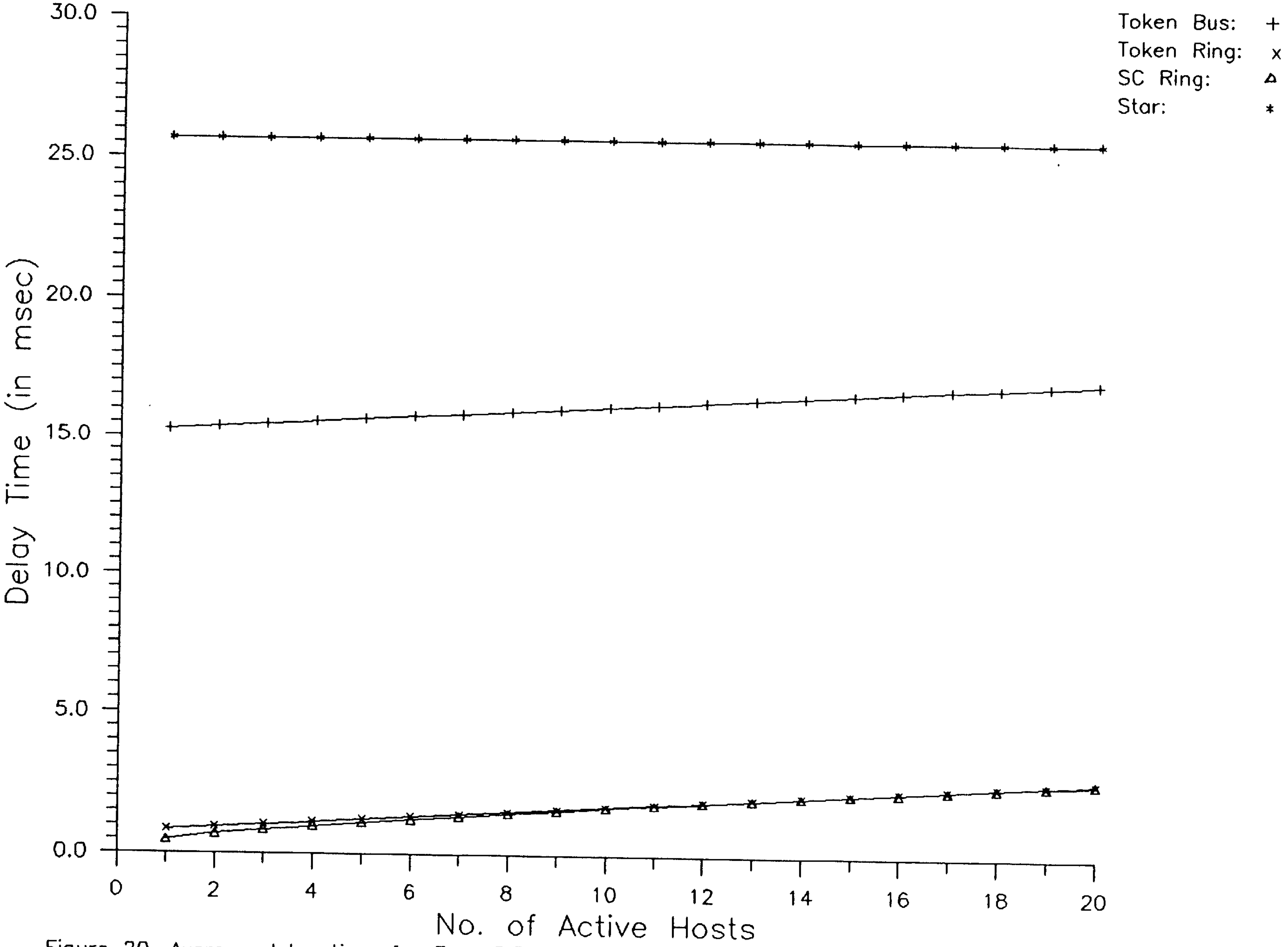
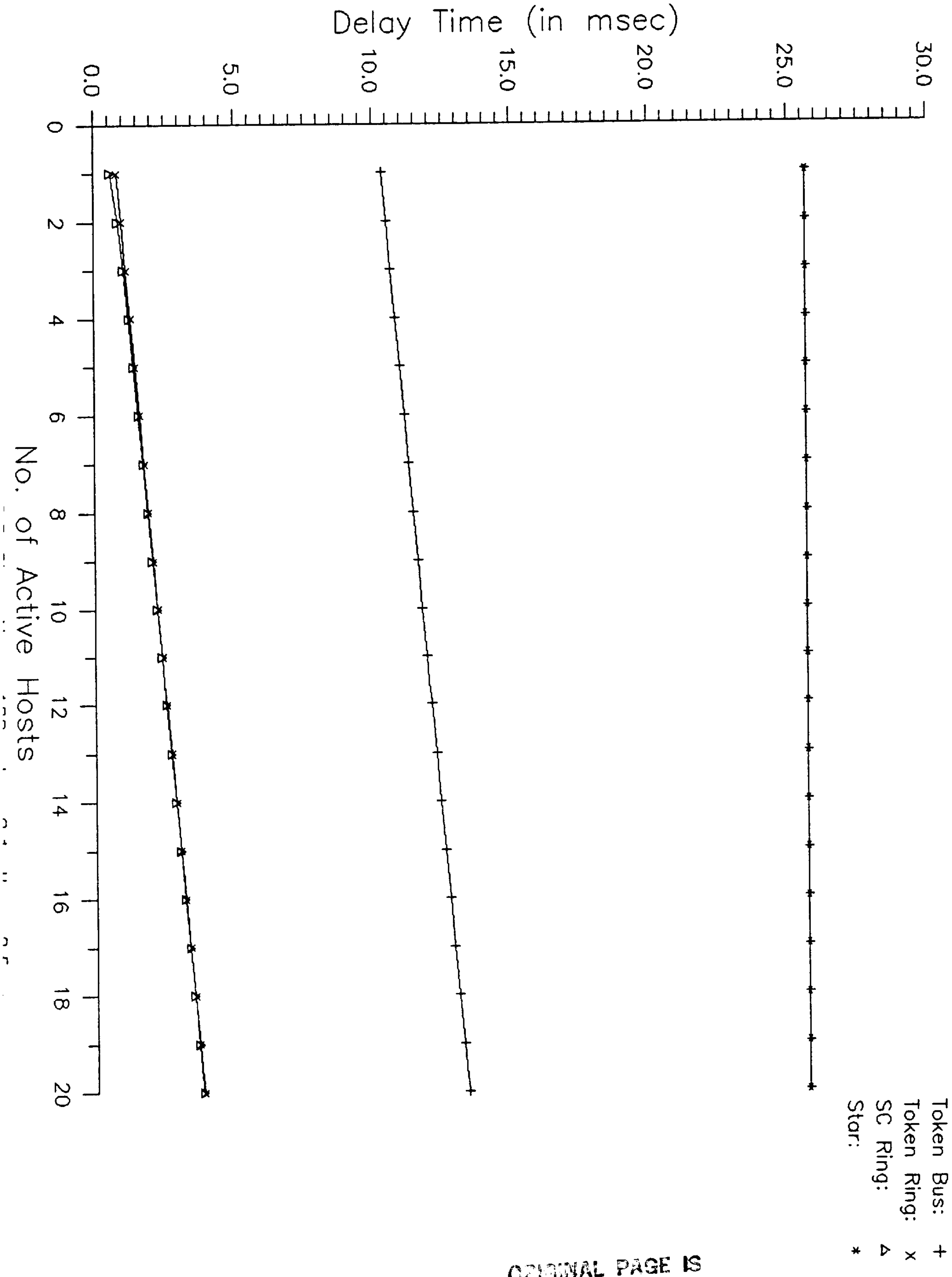


Figure 20. Average delay time for R = 2.5 Gbps, N = 100, wl = 0.1, th = 1.0 ms.


Delay Time (in msec)
0.0
5.0
10.0
15.0
20.0
25.0
30.0
No. of Active Hosts
0
2
4
6
8
10
12
14
16
18
20
Token Bus: +
Token Ring: ×
SC Ring: △
Star: *

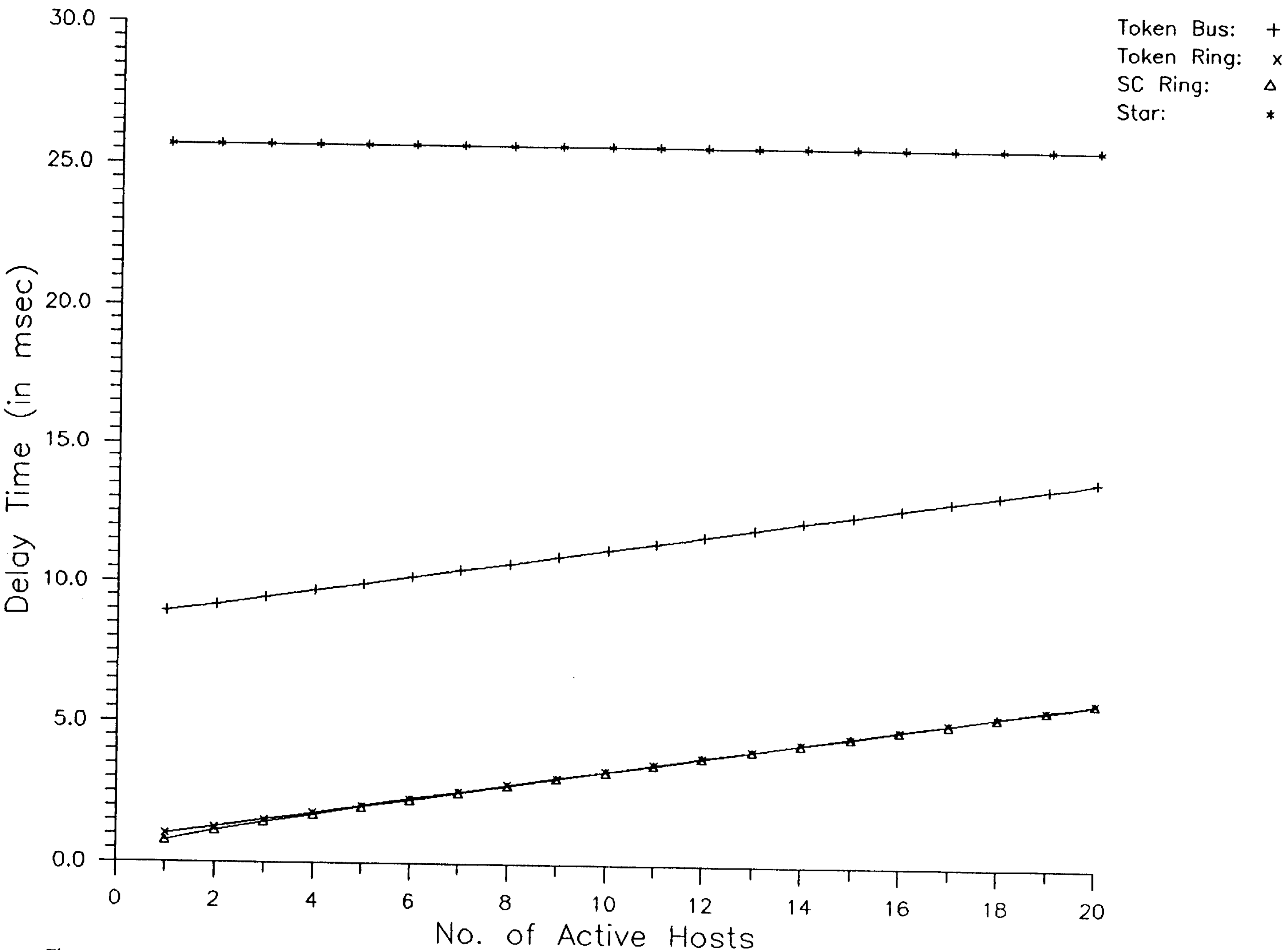
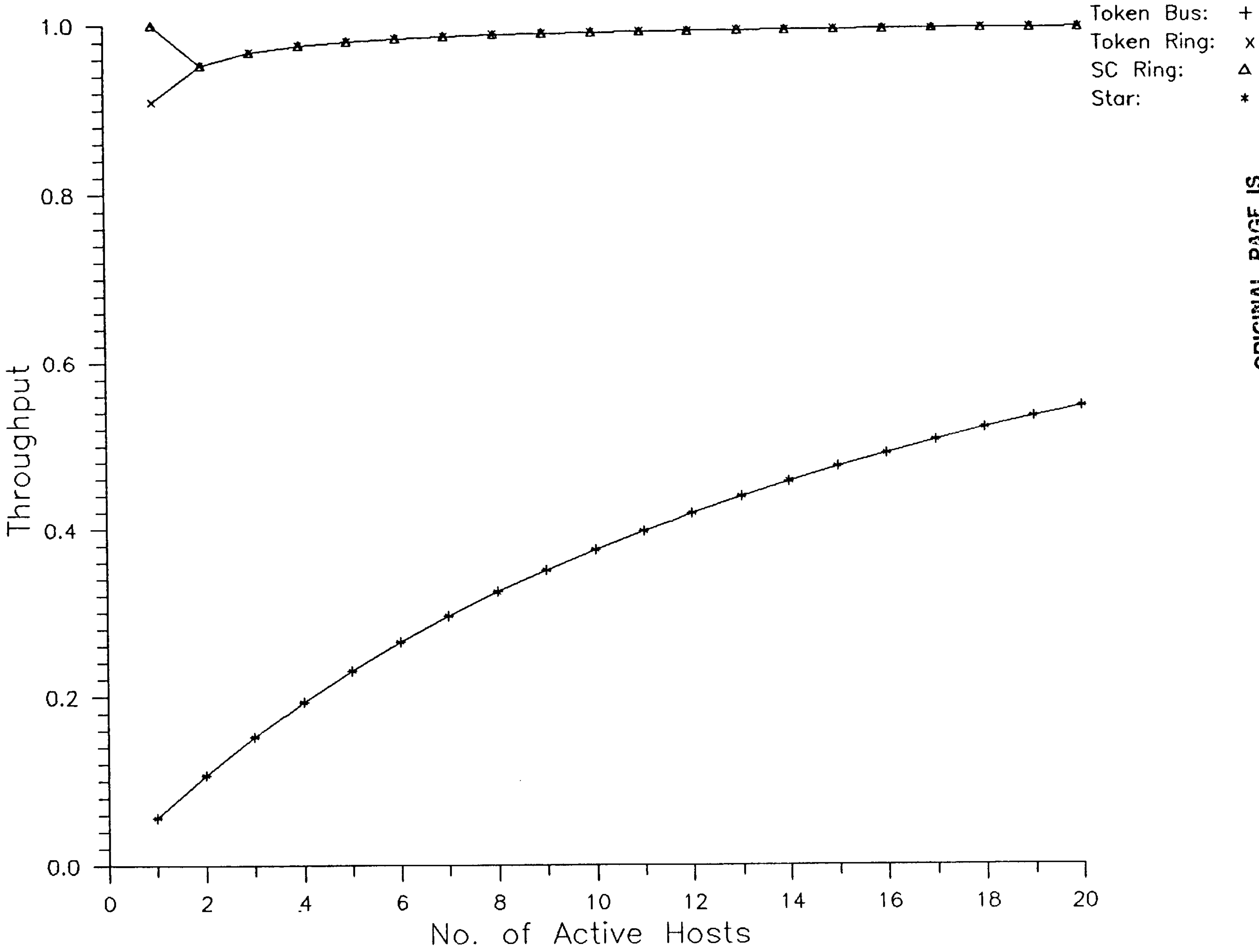


Figure 22. Average delay time for R = 2.5 Gbps, N = 100, wl = 0.1, th = 5.0 ms.


Throughput
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
16
18
20
No. of Active Hosts
Token Bus: +
Token Ring: x
SC Ring: Δ
Star: *

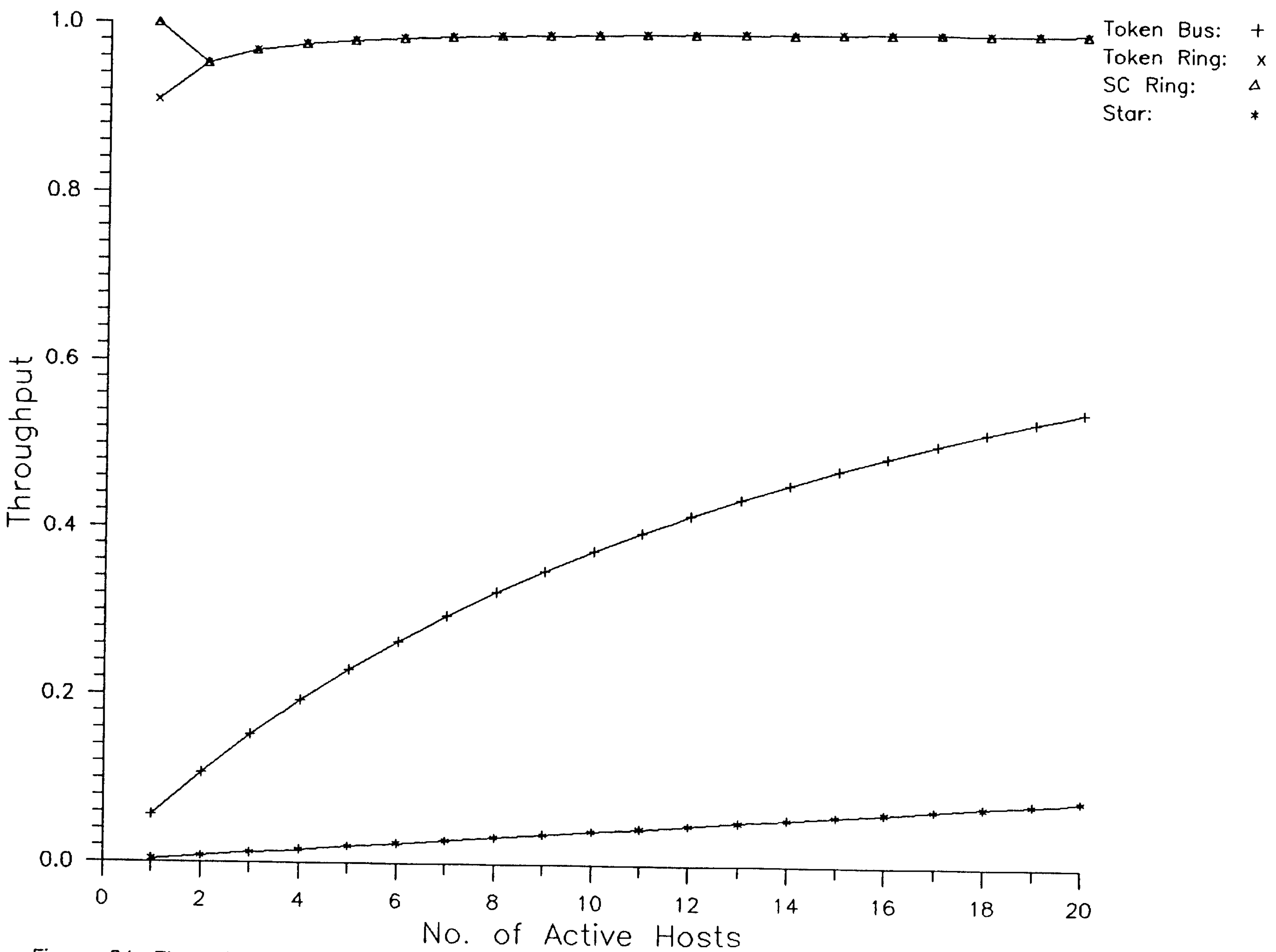
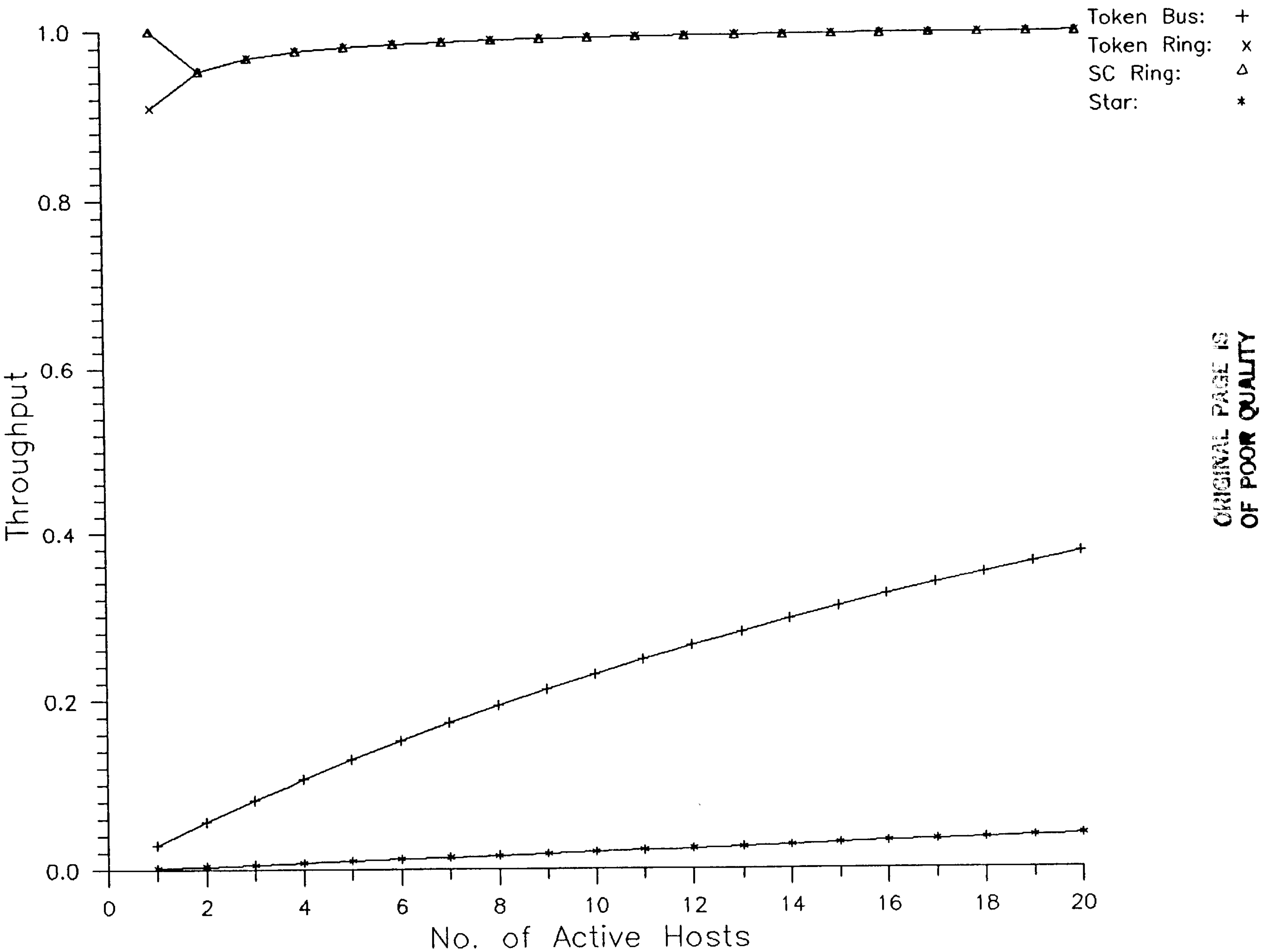


Figure 24. Throughput for R = 2.5 Gbps, N = 500, wl = 1.0, th = 5.0 ms.



[illegible]

ORIGINAL PAGE IS
OF POOR QUALITY

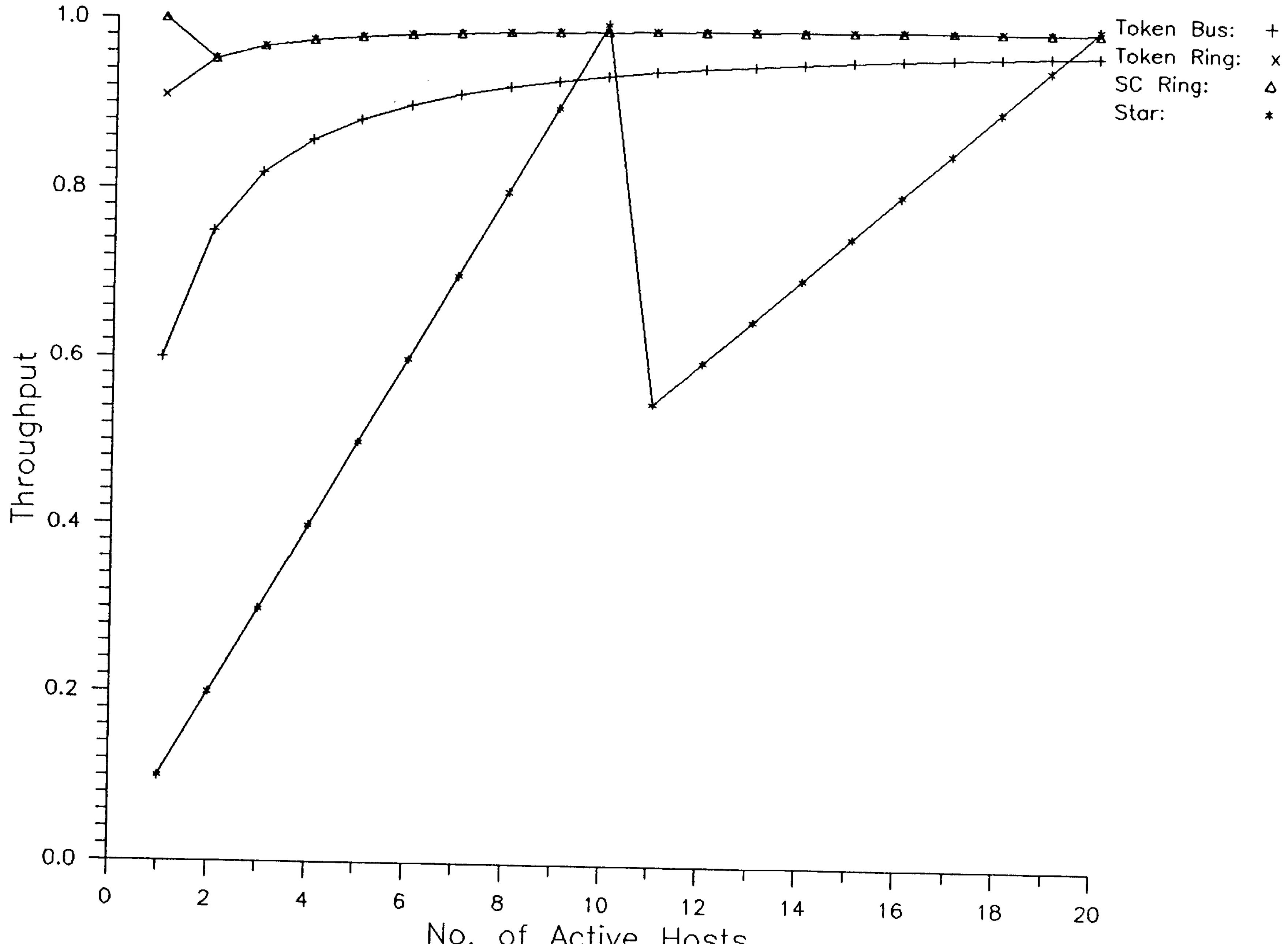
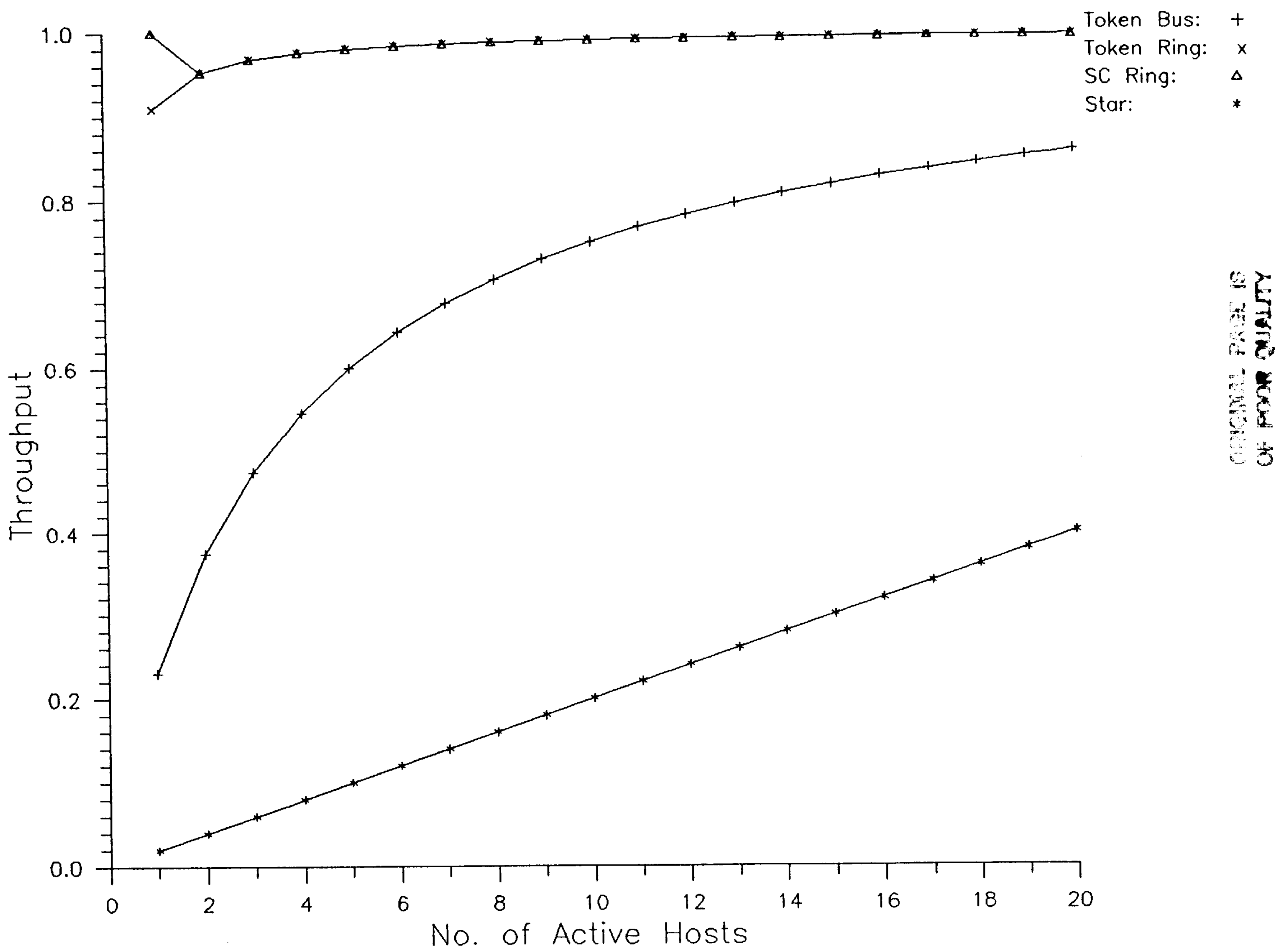


Figure 26. Throughput for R = 2.5 Gbps, N = 20, wl = 1.0, th = 5.0 ms.


Token Bus: +
Token Ring: x
SC Ring: △
Star: *
1.0
0.8
0.6
0.4
0.2
0.0
Throughput
0
2
4
6
8
10
12
14
16
18
20
No. of Active Hosts

[illegible]

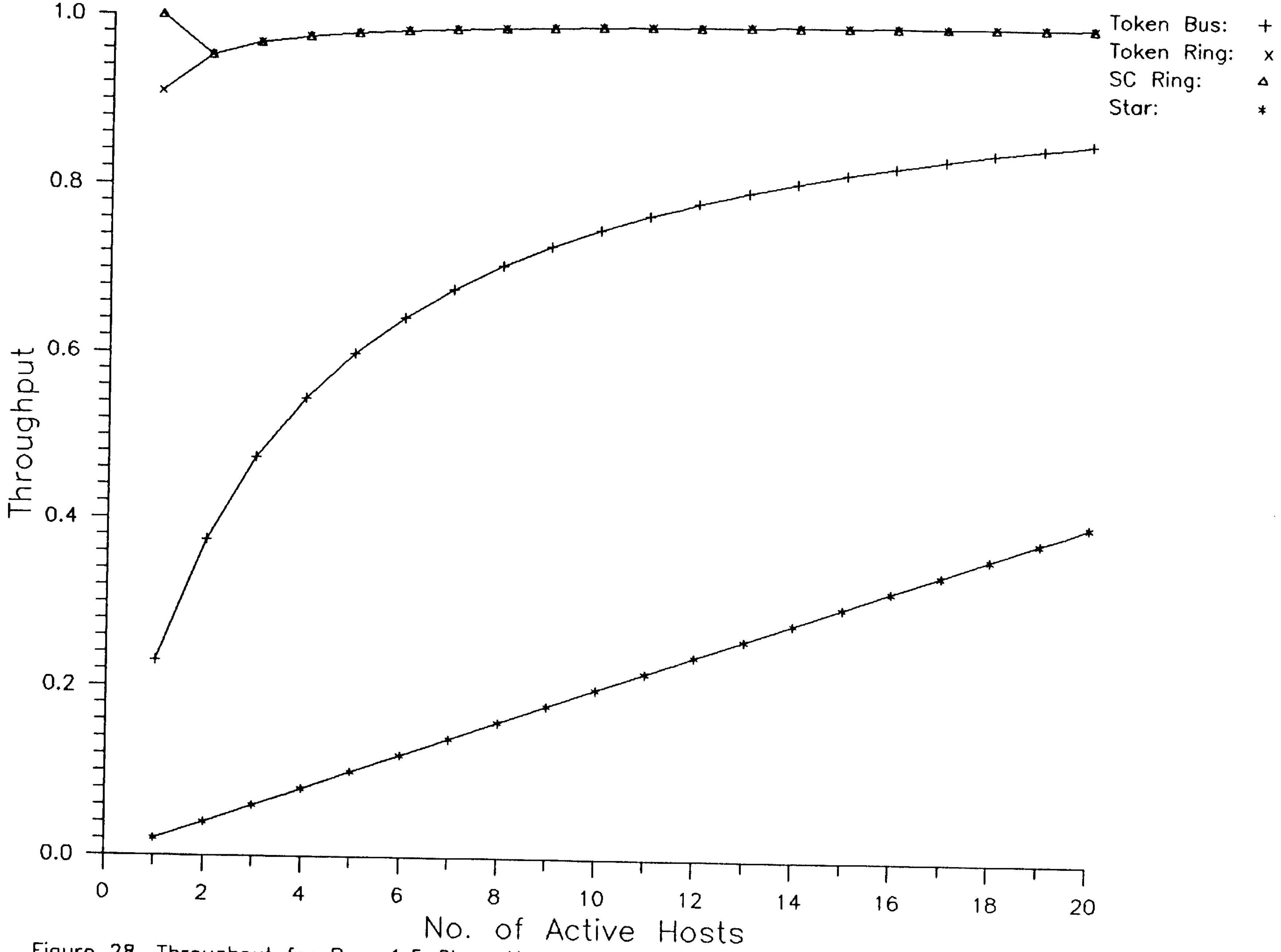
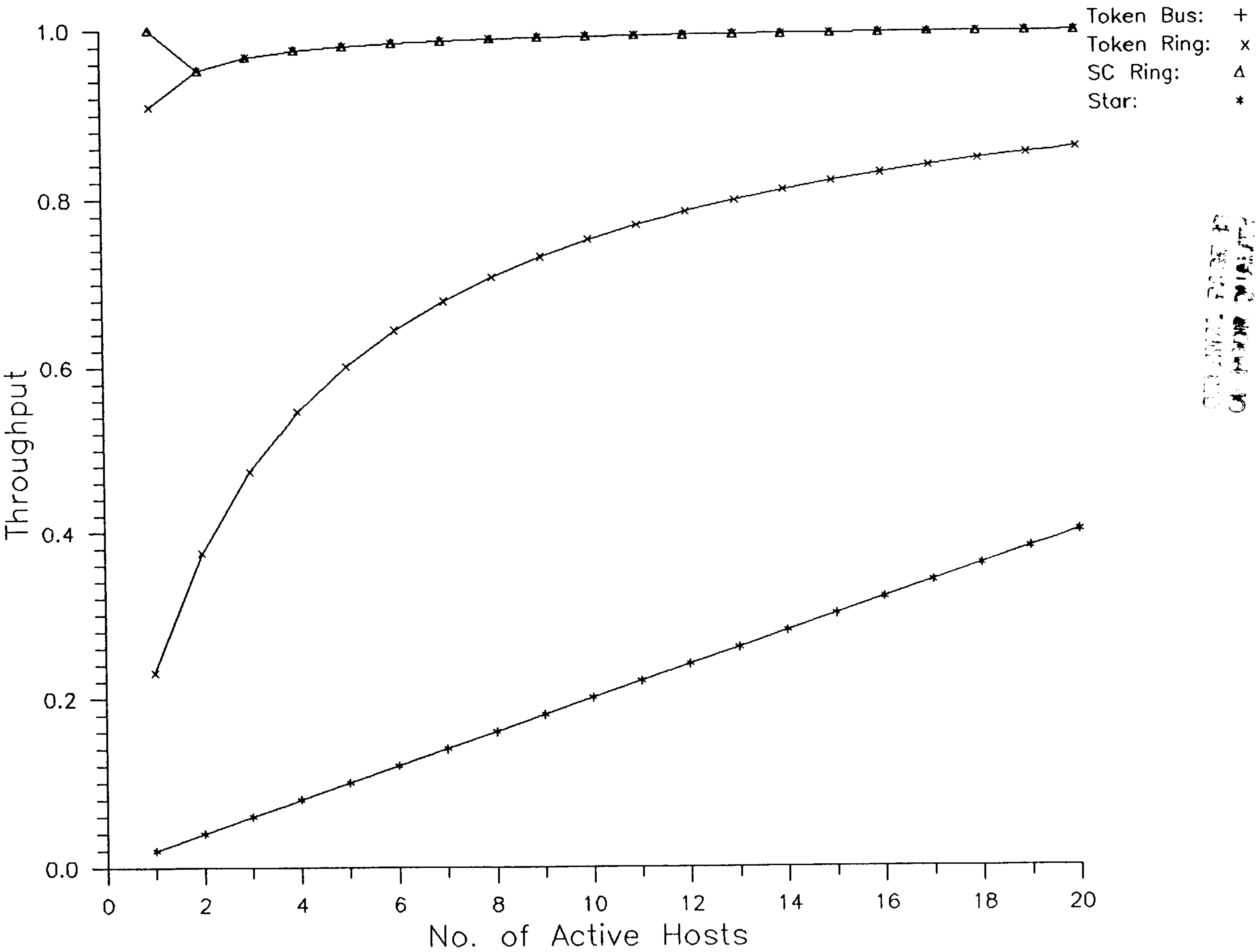


Figure 28. Throughput for R = 1.5 Gbps, N = 100, wl = 1.0, th = 5.0 ms.



[illegible] 55. Throughput [illegible] B = 7.5 Gbps, N = 100, [illegible] = 1.0, th = 5.0 ms

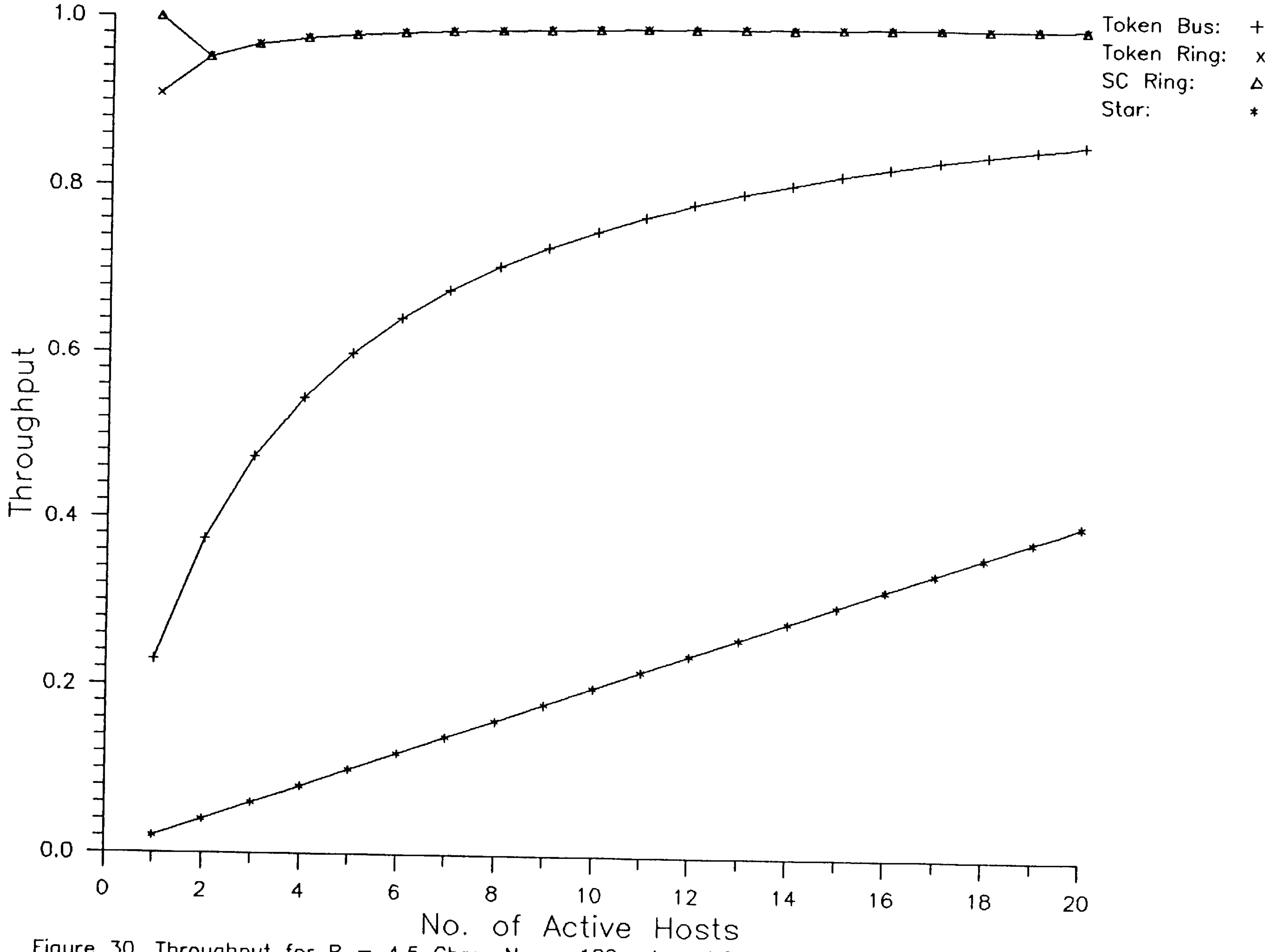
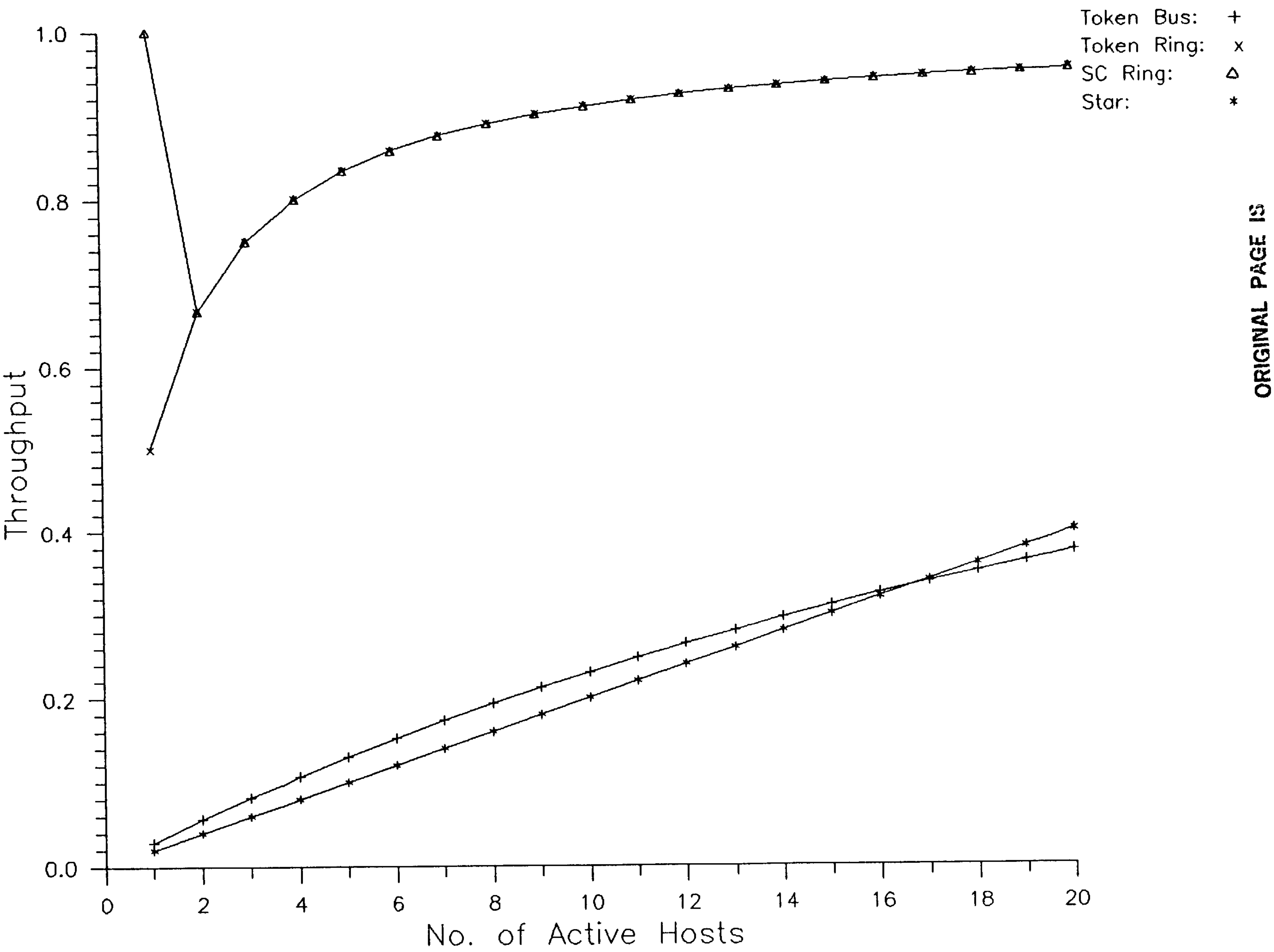


Figure 30. Throughput for R = 4.5 Gbps, N = 100, wl = 1.0, th = 5.0 ms.


Token Bus: +
Token Ring: x
SC Ring: Δ
Star: *
Throughput
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
16
18
20
No. of Active Hosts

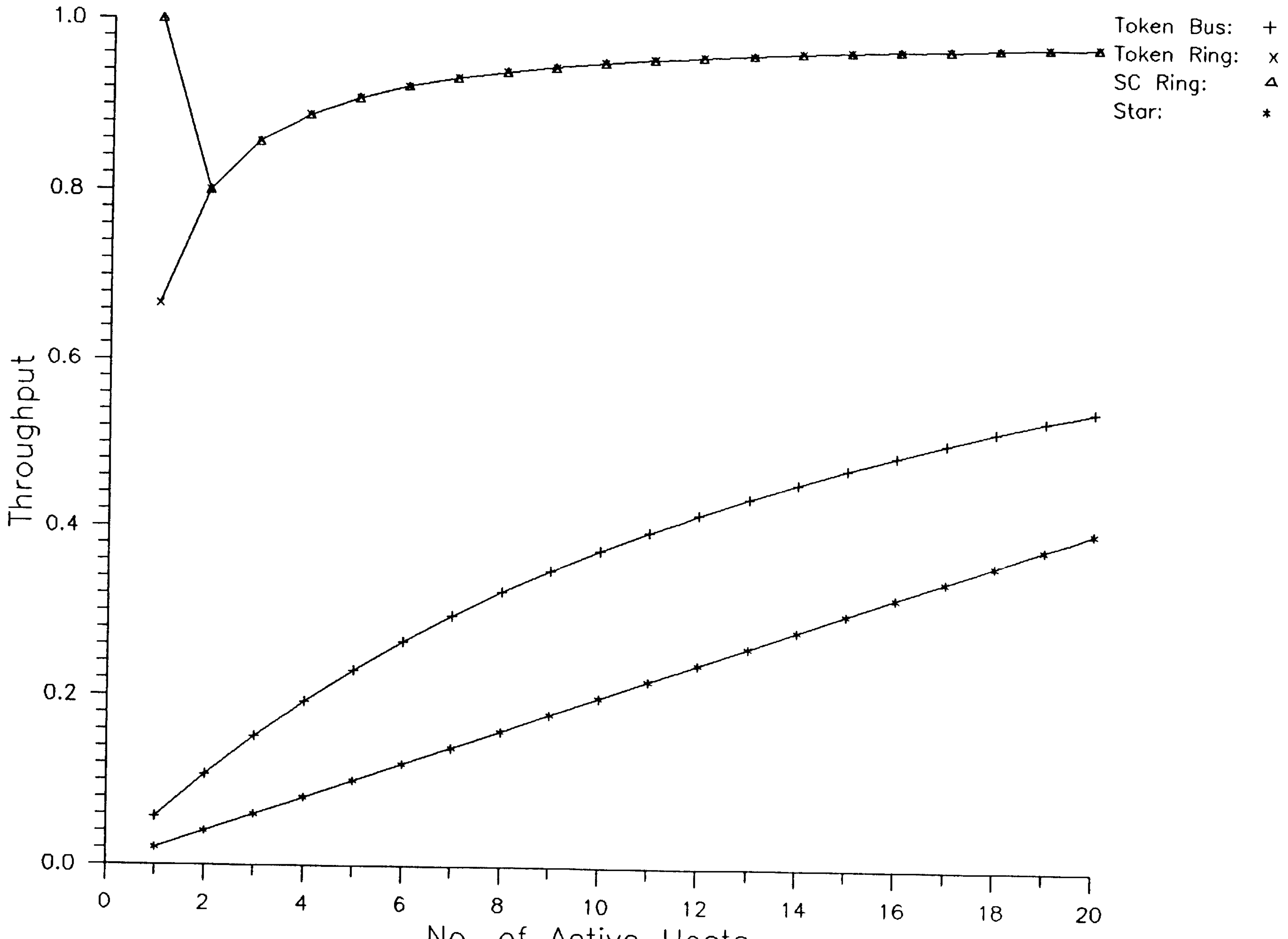


Figure 32. Throughput for R = 2.5 Gbps, N = 100, wl = 1.0, th = 1.0 ms.

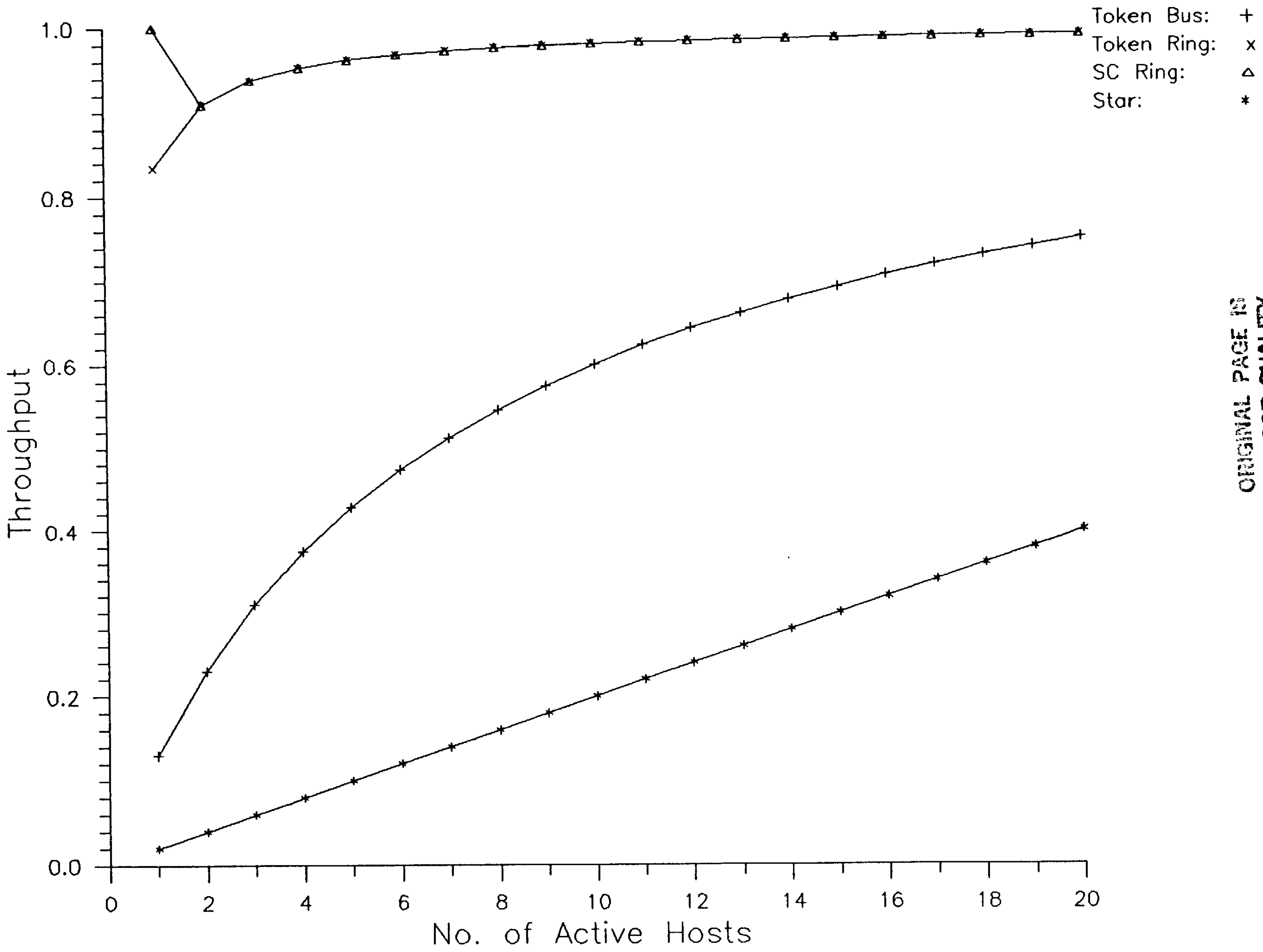


Figure 33: Throughput for B = 2.5 Gbps, N = 100, wl = 1.0, th = 2.5 ms

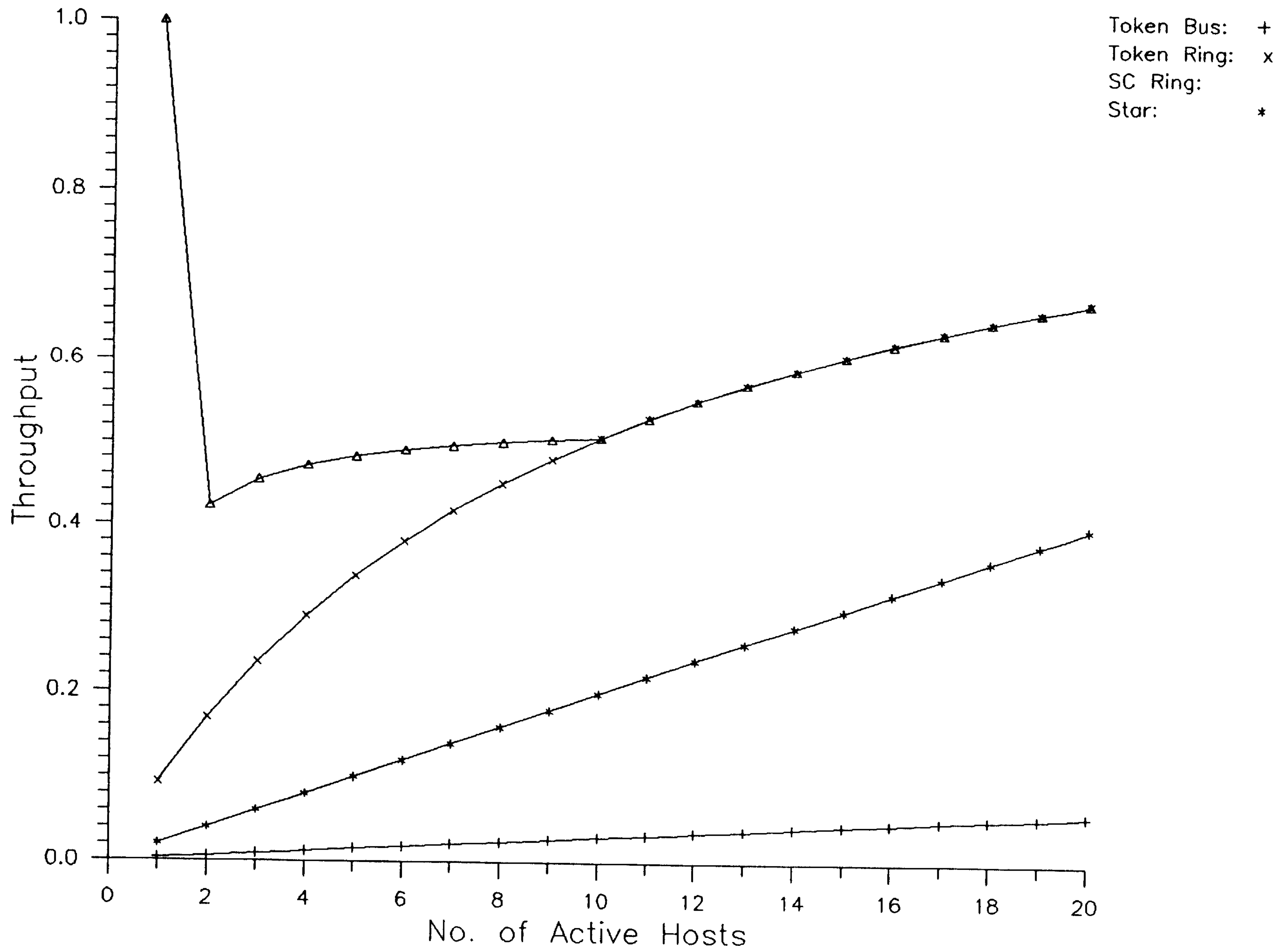
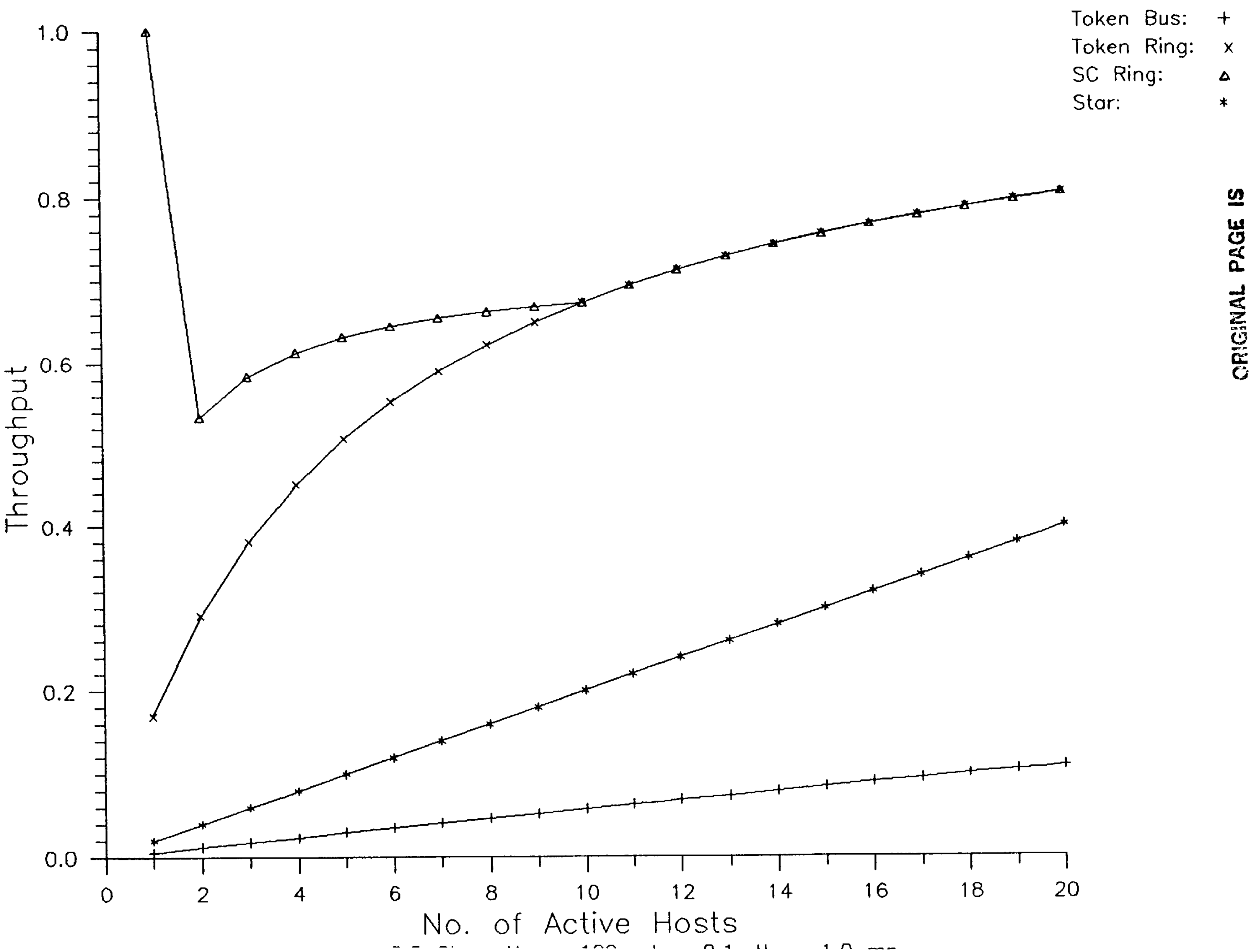


Figure 34. Throughput for R = 2.5 Gbps, N = 100, wl = 0.1, th = 0.5 ms.



ORIGINAL PAGE IS
OF POOR QUALITY

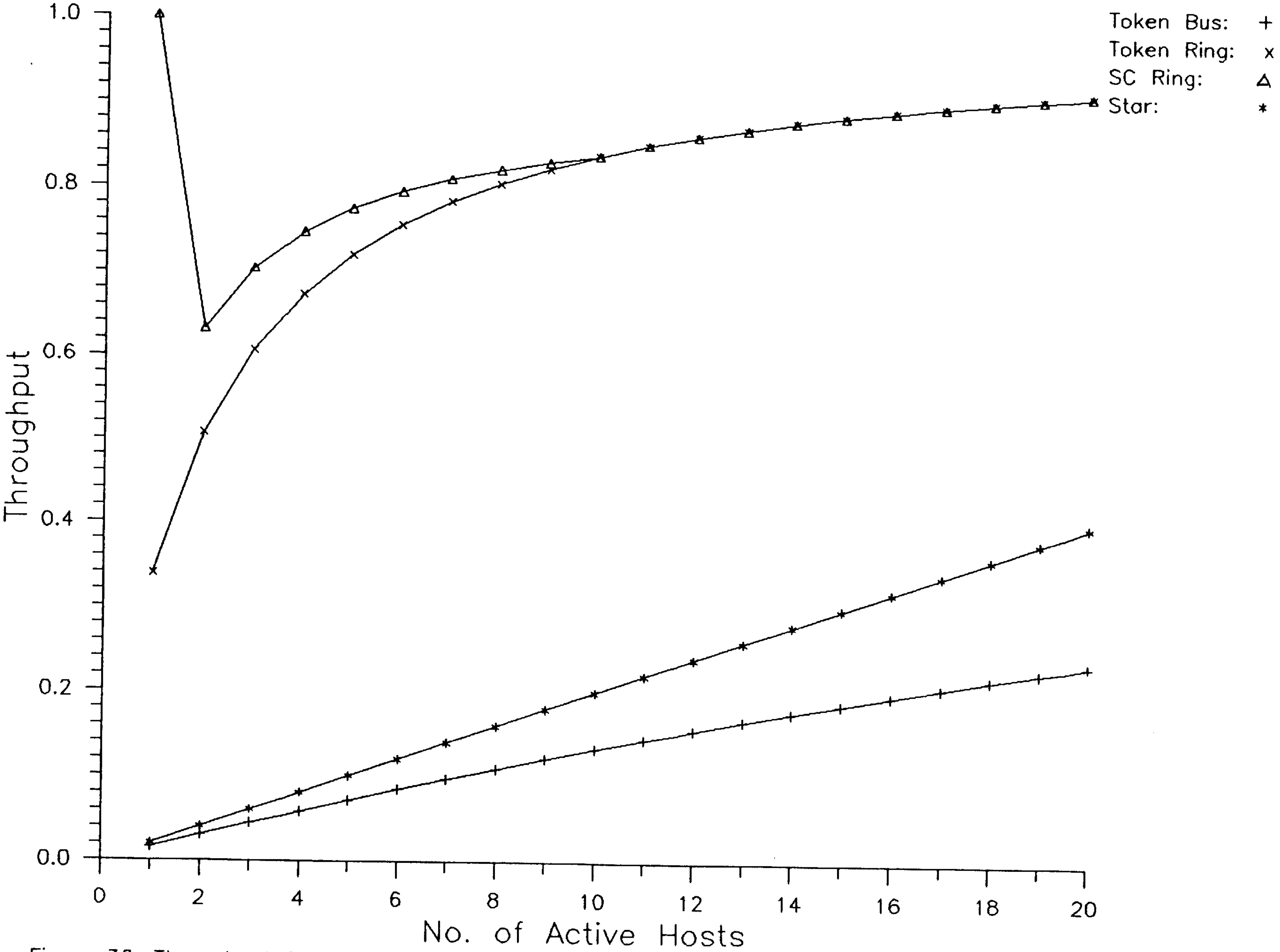


Figure 36. Throughput for R = 2.5 Gbps, N = 100, wl = 0.1, th = 2.5 ms.

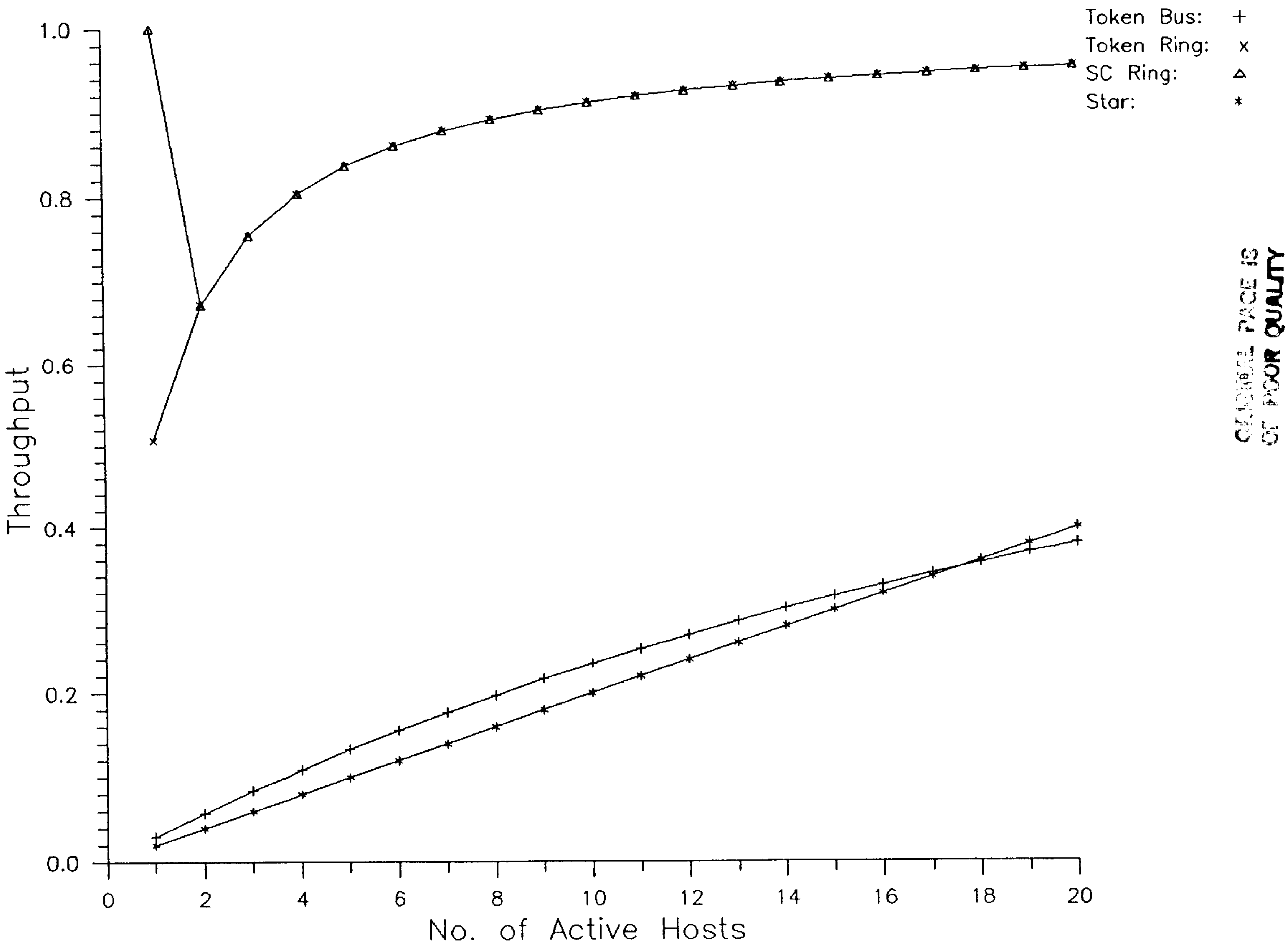


Figure 37. Throughput for R = 2.5 Gbps, N = 100, wl = 0.1, th = 5.0 ms.

ORIGINAL PAGE IS OF POOR QUALITY