NISTIR 7038

A Simulation Analysis of BACnet Local Area Networks

Wong Seok Song
Seung Ho Hong
Steven T. Bushby

National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 7038

A Simulation Analysis of BACnet Local Area Networks

Wong Seok Song
Seung Ho Hong
School of Electrical and Computer Engineering
Hanyang University
Ansan, Korea

Steven T. Bushby
Building and Fire Research Laboratory
Gaithersburg, MD 20899

October 2003

U.S. DEPARTMENT OF COMMERCE
Donald L. Evans, Secretary
TECHNOLOGY ADMINISTRATION
Phillip J. Bond, Under Secretary of Commerce for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Arden L. Bement, Jr., Director

Abstract

BACnet is a standard data communication protocol for building automation and control systems. BACnet defines an object-based model of the information that is exchanged between components of the building automation system and an application layer protocol that is used to access and manipulate this information. It also provides a way to convey the information across a variety of local and wide-area networks that may be interconnected to form an internetwork. In this study, the performance of three BACnet local area networking options is investigated using simulation models developed using ARENA, a tool for simulating discrete event dynamic systems. This study evaluates the delay characteristics of Master-Slave/Token-Passing (MS/TP), Attached Resource Computer Network (ARCNET), and ISO-8802-3 (Ethernet) networks being used to deliver BACnet application services. Analysis of the simulation results was used to identify the network parameters that influence the performance of BACnet application services and to develop recommendations that should be considered when designing and operating BACnet systems.

Key words: ANSI/ASHRAE Standard 135; BACnet; building automation and control; communication protocol; direct digital control; energy management systems; ARENA; discrete event dynamic systems

Table of Contents

1 Introduction

Advanced building automation systems require real-time monitoring and control of building facilities. In order to manage building systems efficiently, a wide variety of building-related information need to be collected, stored, and analyzed. As the demands on building facilities and services have increased, the use of distributed, microprocessor-based control systems has become widespread [1]. Digital communication networks have become a core technology in advanced building automation systems.

In a networked building automation system, many kinds of monitoring, control, maintenance and management data are transmitted through the network. If the network-induced delay of these data exceeds pre-determined limits, building automation systems that require real-time control and operation cannot satisfy their performance and functional requirements. Thus, building automation system designers must understand the performance characteristics of the networks installed in their building.

BACnet (Building Automation and Control networks) is a data communication protocol standard designed specifically for building automation and control systems [2]. BACnet defines an object-based model of the information that is exchanged between components of the building automation system and an application layer protocol that is used to access and manipulate this information. It also provides a way to convey the information across a variety of local and wide-area networks that may be interconnected to form an internetwork.

In this study, simulation models of the three most commonly used BACnet local area networks (LANs) were developed. Those LANs are Master-Slave/Token-Passing (MS/TP), Attached Resource Computer Network (ARCNET), and ISO-8802-3 (commonly referred-to as "Ethernet"). Using the simulation models, the performance characteristics of each of these BACnet LANs was investigated.

This paper consists of five sections. Section 2 briefly describes the features of BACnet. Section 3 presents the simulation models of BACnet LANs developed in this study. Section 4 describes the performance analysis of the BACnet LANs. Finally, conclusions of this study and possible future work are presented in Section 5.

2 A Brief Description of BACnet

Historically, building automation and control systems have used proprietary communication networks. In this kind of closed system, building automation equipment supplied from different manufacturers cannot communicate with each other. Building owners and facility managers were forced to rely on products from a single vendor. Modern building automation and control systems provide a variety of building services such as heating, ventilating, and air-conditioning (HVAC), lighting, fire and life safety systems, security, and vertical transportation. There can be significant safety and operational advantages to integrating these building services through integrated control networks. Closed network systems provide a major barrier to integrated building facilities with the kind of flexibility and expandability that building owners want. In order to solve these problems, the American Society of Heating, Refrigerating, and Air-

Conditioning Engineers (ASHRAE) developed BACnet, the only consensus developed communication protocol standard in the world specifically designed to meet the needs of building automation and control networks.

BACnet defines a set of standard objects whose properties represent the information that is exchanged between components of the building automation system and an application layer protocol that is used to access and manipulate this information. It also provides a way to convey the information across a variety of local and wide-area networks that may be interconnected to form an internetwork.

BACnet has a layered protocol architecture based on a collapsed version of the Open Systems Interconnection (OSI) Basic Reference Model [3]. Layers 1, 2, 3, and 7 of the OSI model are used as shown in Figure 1. The common object model and application layer protocol can be used with any of four LAN technologies or a point-to-point (PTP) protocol suitable for dial-up telephone communications. BACnet also provides wide-area networking capability (not shown in Figure 1) by using Internet Protocols (IP). The network layer provides a way to interconnect any combination of BACnet networks into an internetwork of arbitrary size and complexity. This allows flexibility in configuring various kinds of network systems, and satisfies real-world requirements of building control systems in terms of speed, throughput, and cost [4,5].

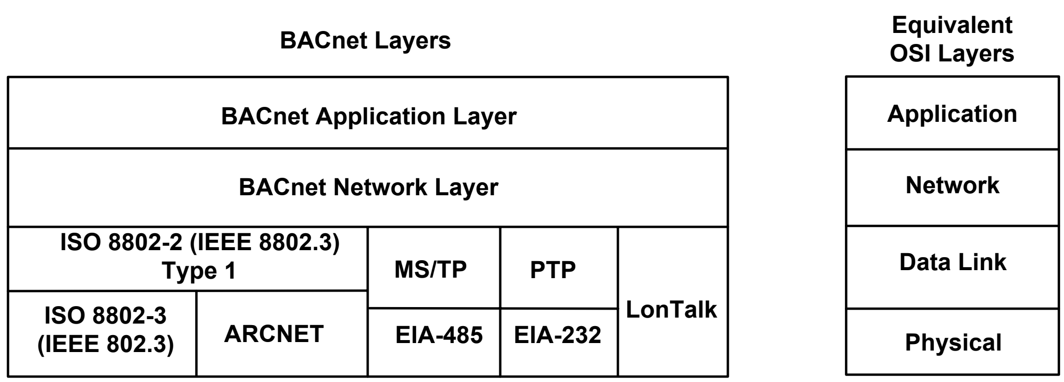


Figure 1. BACnet collapsed protocol architecture.

BACnet is a national standard in the United States and Korea (KS X 6909) [2, 6]. The European Community has adopted it as a pre-standard. A modified form of BACnet has been adopted as a national standard in Japan. Currently, BACnet is proposed as a world standard and being deliberated by International Organization for Standardization (ISO) Technical Committee 205, Building Environment Design [7].

3 Development of BACnet Simulation Models

Building automation systems commonly have a hierarchical structure. A high-speed backbone LAN is used to connect workstations and supervisory controllers. Unitary and application specific controllers typically reside on lower cost, lower speed LANs. BACnet permits such hierarchical structures but does not impose them. Any of the networking options in BACnet can be used alone or combined with others by using routers.

The most commonly used LANs in BACnet systems are Ethernet, ARCNET, and MS/TP. They were selected for this study because of their popularity. Ethernet is now the most widely used LAN technology in the world and is typically used as a high-speed backbone in building automation systems. ARCNET is also a widely known networking technology. In BACnet systems it is typically used over twisted pair networks using EIA-485 [8] signaling. MS/TP is the only networking option that was developed specifically for BACnet. It also uses EIA-485 signaling. The name comes from the fact that it can be configured as a master/slave network, a peer-to-peer token-passing network, or a mixture of the two. MS/TP is the lowest cost LAN technology in BACnet. It is described in more detail in 4.1

Communication networks such as MS/TP, ARCNET and Ethernet can be categorized as a discrete-event dynamic system (DEDS) [9]. In a DEDS, the state of a system is changed whenever an event occurs and events occur at random. Some examples of events that can occur in a communication network system include message generation, message transmission, message reception, and many other protocol specific events such as message collision, token delivery, polling, etc. In this study, the simulation models were developed using ARENA [10], a tool for developing simulation models of various kinds of DEDS systems.

ARENA provides basic templates for the modeling of DEDS systems. Using the basic templates as a starting point, BACnet specific LAN models were developed. Figure 2 shows the structure of the simulation models developed in this study. As shown in the figure, the simulation model has three independent modules; the Common Module, the Application Layer Module and the LAN Protocol Module. Users need not modify the whole simulation model when they make a new model for a specific process. Only the modules corresponding to the specific process need to be modified.

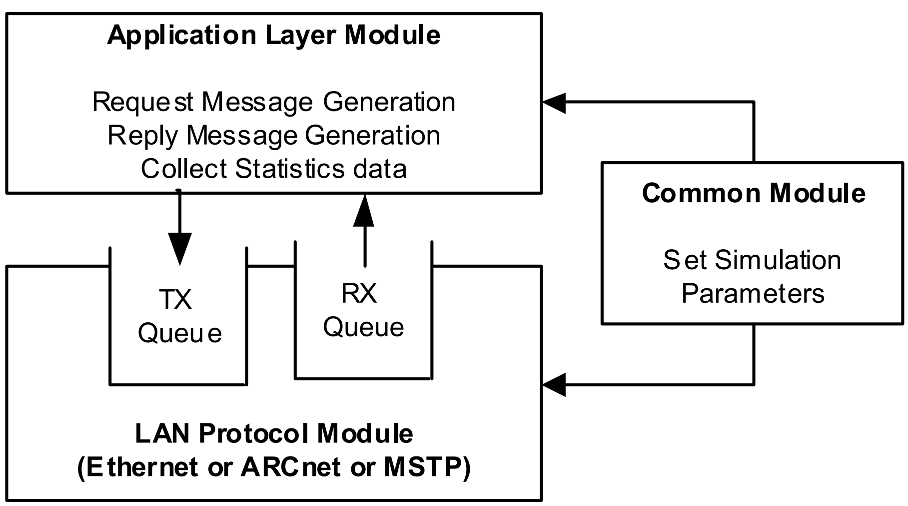


Figure 2. Structure of BACnet simulation models.

Table 1 shows a brief description of the modules developed for modeling BACnet LANs. The Common Module provides an interface for users to set the values of all the simulation parameters. The Application Layer Module generates the request and reply messages of BACnet application services. The messages received by the destination node are used to collect and analyze the statistical information of network-induced delay. Three independent LAN protocol modules were developed, one each for Ethernet, ARCNET and MS/TP.

Table 1. ARENA Modules Developed for Modeling BACnet LANs

Module	Function		Description
Common Module	- Simulation Environment - Ethernet Environment - ARCNET Environment - MSTP Environment		- set the simulation time and the number of replications - set the simulation parameters for Ethernet - set the simulation parameters for ARCNET - set the simulation parameters for MS/TP
Application Layer Module	- Message Generation - Statistical Analysis		- schedule the generation of BACnet messages - collect and analyze statistical information
LAN Protocol Module	Ethernet	- Ethernet Node - Hub	- Ethernet node model - Ethernet hub model
	ARCNET	- ARCNET Node	- ARCNET node model
	MS/TP	- Master Node - Slave Node	- MS/TP master node model - MS/TP slave node model

The Ethernet module models the 10 Mbps CSMA/CD version of the protocol [11]. It consists of an Ethernet node model and hub model that interconnects Ethernet node models. The ARCNET module models a 156.25 Kbps token-passing algorithm based on the ANSI/ATA 878.1 specification [12]. The MS/TP module models 76.8 Kbps token-passing and master/slave algorithms described in the BACnet specification [2]. It consists of a master node model and a slave node model. Using these models, a user can develop a variety of MS/TP network configurations such as single-master, multi-master or all-master systems.

Figure 3 shows a screen capture of the window of an MS/TP protocol simulation model that consists of 5 nodes. The left pane shows basic templates provided by ARENA. The middle pane shows a simulation model for MS/TP. Using the MSTP_ENVIRONMENT dialog box, various network parameters such as data rate, propagation delay, timer values, and message length can be set. The dialog window in the figure shows how the simulation parameters are set. The SIMULATION_ENVIRONMENT module is used to set the simulation related parameters such as simulation time and the number of replications.

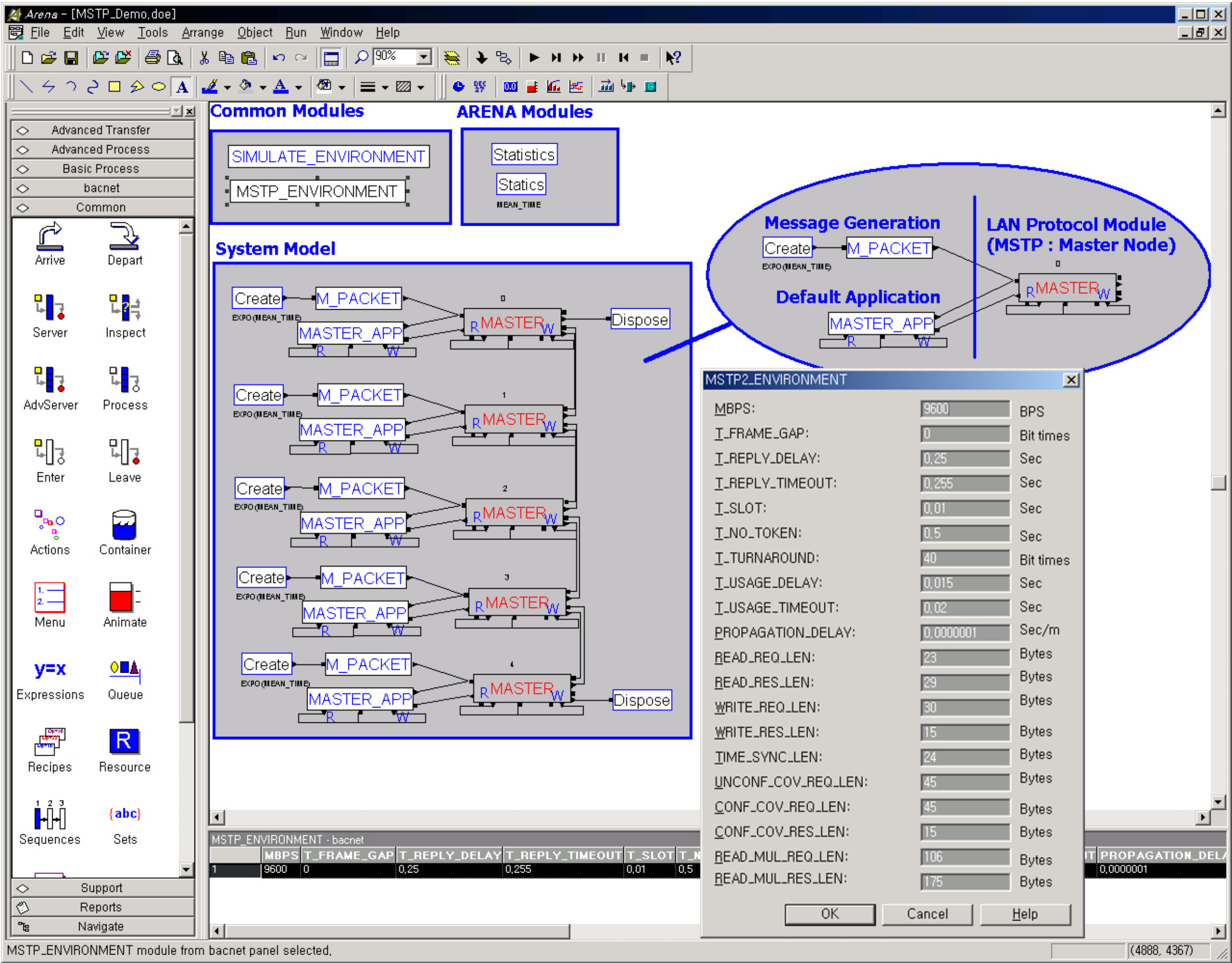


Figure 3. Sample window of the MS/TP simulation model.

In the System Model, the block named MASTER is a master node model. Node address and the value of some network parameters such $N_{max\_info\_frames}$ and *Max_Master* can be set using this model. The block named MASTER_APP is the application layer model of an MS/TP node. This block generates request and reply messages, and calculates statistical information. The block named M_PACKET converts the basic ARENA entity to an MS/TP message. Using the M_PACKET module, a user can generate *ReadProperty*, *WriteProperty*, *ReadPropertyMultiple*, *UnconfirmedCOVNotification*, *ConfirmedCOVNotification* or any other BACnet message. The simulation models for ARCNET and Ethernet have a structure similar to the one shown in Figure 3.

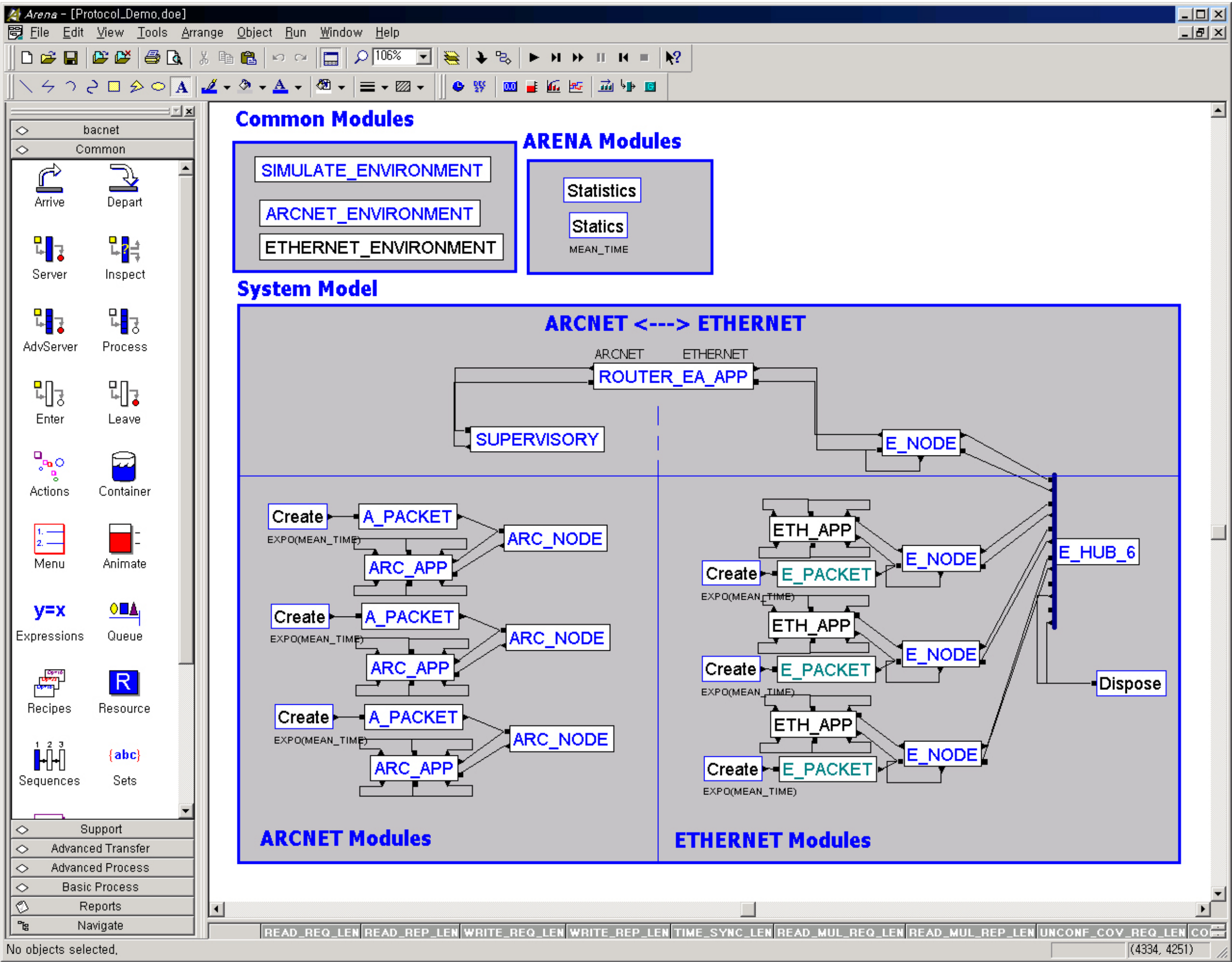


Figure 4. Sample window of the simulation model for integrated network protocols.

The simulation tool enables integrating more than one network protocol into a single model. Figure 4 shows an integrated simulation model where Ethernet and ARCNET networks are interconnected through a router. Both the Ethernet and ARCNET networks have three nodes. The protocol parameters and timer values for ARCNET and Ethernet are set in the ARCNET_ENVIRONMENT and ETHERNET_ENVIRONMENT of the Common Module, respectively.

The left side of the System Model shows an ARCNET module. Its structure is similar to the MS/TP model, consisting of an application layer model and a LAN protocol model, which are represented by the ARC_APP and ARC_NODE blocks, respectively. The A_PACKET block converts the basic ARENA entity to an ARCNET message. The right side of the System Model shows an Ethernet module. The LAN protocol model of Ethernet consists of an E_NODE block and an E_HUB_6 block, representing an Ethernet node and Ethernet hub, respectively. The E_PACKET block converts the basic ARENA entity to an Ethernet message. The upper part of the System Model shows a router model. The router model has both an ARCNET module and an Ethernet module. It enables the message exchange between ARCNET and Ethernet.

4 Performance Analysis of BACnet LANs

In this section, the performance of MS/TP, ARCNET, and Ethernet is analyzed using their simulation models. In this study, we quantify the traffic load of a network as G. The physical meaning of G is defined as a fraction of message transmission time per unit time, excluding the overhead of the network protocol itself. G is expressed as:

$$G = \frac{1}{B}\sum_{i=1}^{N}\frac{L_i}{T_i}$$

where, B is a data transmission rate (bits/s), N is the number of nodes that generate message in the medium, T_i is an average interval of message generation at node i in seconds, and L_i is an average message length in bits generated at node i. G has a value between 0 and 1. G approaches 1 as the traffic load in the network increases. The performance of BACnet LANs is directly affected by changes in the network parameters, B, N, T_i and L_i. In this study, we analyzed the performance of BACnet LANs with respect to the change of these network parameters.

The performance of BACnet LANs is evaluated in terms of *service delay*. Service delay is defined as the elapsed time to complete one transaction of a BACnet service. For a BACnet confirmed service, the service delay is defined from the instant when a request message arrives at the transmitter queue of a client to the instant when a reply message transmitted by its server has completely arrived at the receiver queue of the client. For a BACnet unconfirmed service, the service delay is defined from the instant when a message arrives at the transmitter queue of a sender to the instant when the same message has completely arrived at the receiver queue of a receiver.

The analysis of each protocol is divided into two parts. In the first part, only the delay in medium access is considered. In the second part, the effect of processing time on service delay is considered. The delay in processing the application service request depends upon both the hardware and the software implementation skill.

4.1 MS/TP Networks

4.1.1 Summary of MS/TP Features

The Master-Slave/Token-Passing (MS/TP) protocol was designed to be implemented using a single-chip microprocessor with a universal asynchronous receiver/transmitter (UART). It uses EIA-485 signaling over a twisted-pair line and is the lowest cost LAN option in BACnet. The name reflects the fact that MS/TP networks can be configured as a master/slave network, a peer-to-peer token passing network, or a mixture of the two. MS/TP supports transmission rates of 9.6, 19.2, 38.4 and 76.8 Kbps. In this analysis, we assume the default transmission rate of MS/TP to be 76.8 Kbps because most MS/TP devices are currently implemented with that speed.

MS/TP master nodes maintain a token frame that regulates access to the medium. The token is circulated from one master node to another according to a pre-determined order based on addresses. A master node that holds the token can transmit up to $N_{max\_info\_frames}$ messages to either other masters or to slaves before passing the token. $N_{max\_info\_frames}$ is a network parameter that can be set by the system designer. After receiving the token 50 times, a master node transmits a *Poll_For_Master* frame in order to discover the presence of other master nodes on the network that wish to join the ring. If one is found, it becomes the new successor node in the token ring. If the successor is already the next available address then this step is omitted.

The MS/TP address space is segregated between masters and slaves. There can be at most 128 masters and their addresses are constrained to the range 0 to127. Slaves can have any address in the range 0 to 254. Consequently there can be at most 255 MS/TP devices in a single network. The number of masters and slaves is configurable subject to the limitation of no more than 128 masters.

Slave nodes never hold the token. Slave nodes return a reply only when they receive a request from a master node. A master node that receives a request returns the reply immediately or it may return a *Reply_Postponed* frame, indicating that the actual reply will be returned when it holds the token.

Table 2, summarizes the important parameters that directly affect the performance of MS/TP networks. It also shows the constraints on their values defined in the standard, and the typical values used in the actual MS/TP implementation.

Table 2. Important MS/TP Network Parameters

Parameter	Description	Specified Limits	Typical Value
$N_{max\_info\_frames}$	The maximum number of information frames a node may send before it must pass the token.	User defined (If not writable, its value shall be 1)	8 to 200
T_{frame_gap}	The maximum idle time a transmitting node may allow to elapse between octets of a frame.	20 bit times	0 bit times
$T_{turnaround}$	The Minimum time after the end of the stop bit of the final octet of a received frame before a node may enable its EIA to 485 driver.	40 bit times	40 bit times
T_{usage_delay}	The maximum time a node may wait after reception of the token or a Poll For Master frame before sending the first octet of a frame.	15 msec	40 bit times
T_{reply_delay}	The maximum time a node may wait after reception of a frame that expects a reply before sending the first octet of a reply or Reply Postponed frame.	250 msec	40 bit times

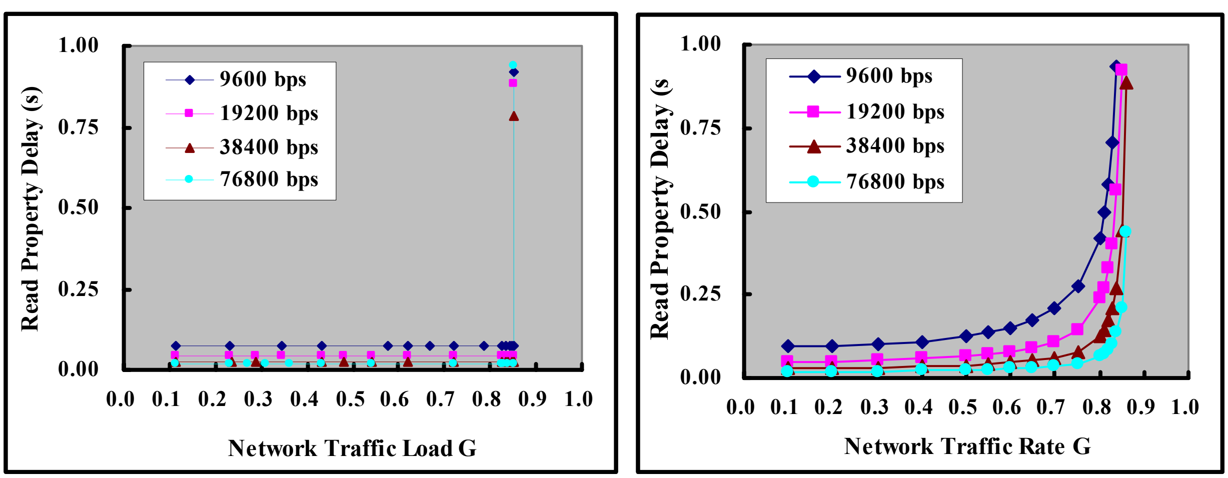
4.1.2 Performance Analysis of Single to Master System

In this section the performance of an MS/TP network with a single to master is analyzed. A single to master system consists of one master and several slave nodes. In this analysis, the application service in the MS/TP frame was assumed to be *ReadProperty*, which is one of the most widely used BACnet services. *ReadProperty* is a confirmed service. A master node generates the request messages. The request messages are inserted into the transmitter queue and transmitted to the corresponding slave nodes. Upon receiving the request message, each slave node sends a reply message to the master node. In this section, we do not consider the processing delay in the application layer, thus T_{reply_delay} in Table 2 was assumed to be negligible. The message length of a *ReadProperty* service request is fixed by the standard. The length of a reply depends upon the property being read. For this analysis it was assumed that a Real value was being returned.

In this simulation analysis, *ReadProperty* service delay was measured with respect to the change of transmission rate and request message generation interval at the master node. Table 3 shows the simulation conditions selected and the corresponding traffic load G. The reply message generation interval at a slave node is 31 (number of slave nodes) times larger than the request message generation interval at the master node. Two different types of message generation interval were considered: periodic and aperiodic. Periodic message generation assumes that the master node generates request messages with a fixed interval. For aperiodic message generation, the message generation interval in the master node is assumed to have a Poisson distribution.

Table 3. Simulation Conditions for a Single to Master MS/TP Network

Data rate (bps)	Message length (bytes) (request/reply)	Number of nodes (master/slave)	Message generation interval at the master node (s)	Traffic load G
9600	23/29	1/31	0.54167 to 0.06438	0.1000 to 0.8414
19200	23/29	1/31	0.01277 to 0.03206	0.1000 to 0.8447
38400	23/29	1/31	0.13542 to 0.01594	0.1000 to 0.8497
76800	23/29	1/31	0.06771 to 0.00794	0.1000 to 0.8530



(a) Periodic traffic (b) Aperiodic traffic

Figure 5. Average service delay for *ReadProperty* service requests (single-master).

Figures 5(a) and 5(b) show the average service delay for *ReadProperty* requests in the single-master MS/TP network when messages are generated periodically and aperiodically, respectively. For periodic traffic, the service delay remains constant as the traffic load is changed. For aperiodic traffic, the service delay increases exponentially as the traffic load increases. In both cases the network resource is eventually saturated.

Single-master MS/TP operation is subject to protocol overhead delays such as $T_{turnaround}$ and T_{frame_gap} (see Table 2). These timers exist to ensure reliable data transmission. In commercial MS/TP implementations a typical value for $T_{turnaround}$ is 40 bit times and T_{frame_gap} is negligibly small. Because the network utilization is subject to these protocol overheads, it is recommended that a designer of single-master MS/TP networks restrict peak traffic load so that $G < 0.8$ when protocol overheads have typical values.

The MS/TP protocol defines the maximum value of T_{frame_gap} as 20 bit times. Figure 6 shows the average service delay for *ReadProperty* requests as a function of T_{frame_gap} when the data rate is 76.8 Kbps. As shown in Figure 6, increasing T_{frame_gap} heavily degrades the performance. It is recommended that the value of T_{frame_gap} should be as small as possible when implementing MS/TP devices.

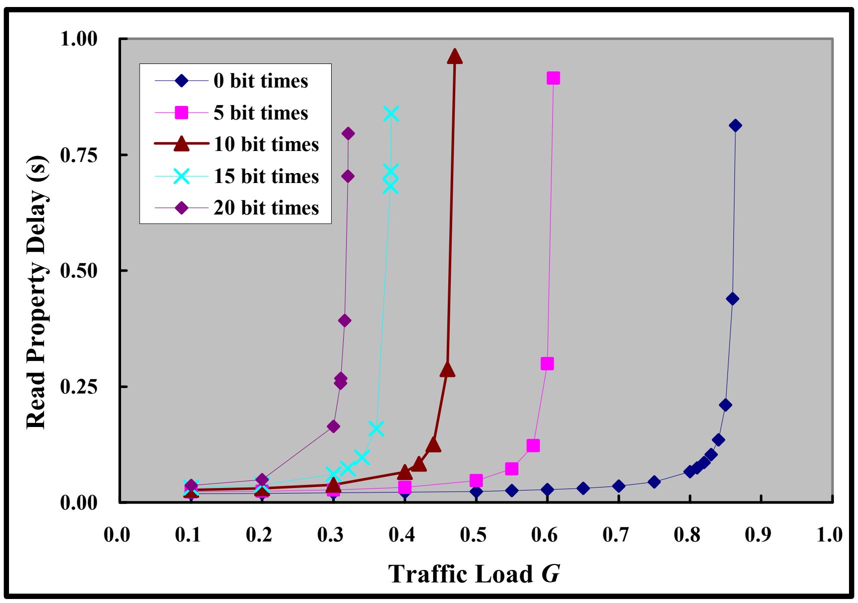


Figure 6. Average service delay for *ReadProperty* service requests as a function of T_{frame_gap}.

In a single-master system, varying the number of slave nodes does not influence the service delay. This is because a slave node does not generate messages by itself. Figure 7 shows simulation results for average service delay of *ReadProperty* requests with respect to the change in the number of slave nodes. The simulation conditions are given in Table 4. Figure 7 verifies that the number of slave nodes does not influence the delay performance in a single-master system.

Table 4. Simulation Conditions for a Variable Number of Slave Nodes (Single-Master)

Number of slave nodes	Message generation interval at the master node (s)	Traffic load *G*
31	0.06771 to 0.00778	0.1 to 0.87
20	0.06771 to 0.00778	0.1 to 0.87
10	0.06771 to 0.00778	0.1 to 0.87
1	0.06771 to 0.00778	0.1 to 0.87

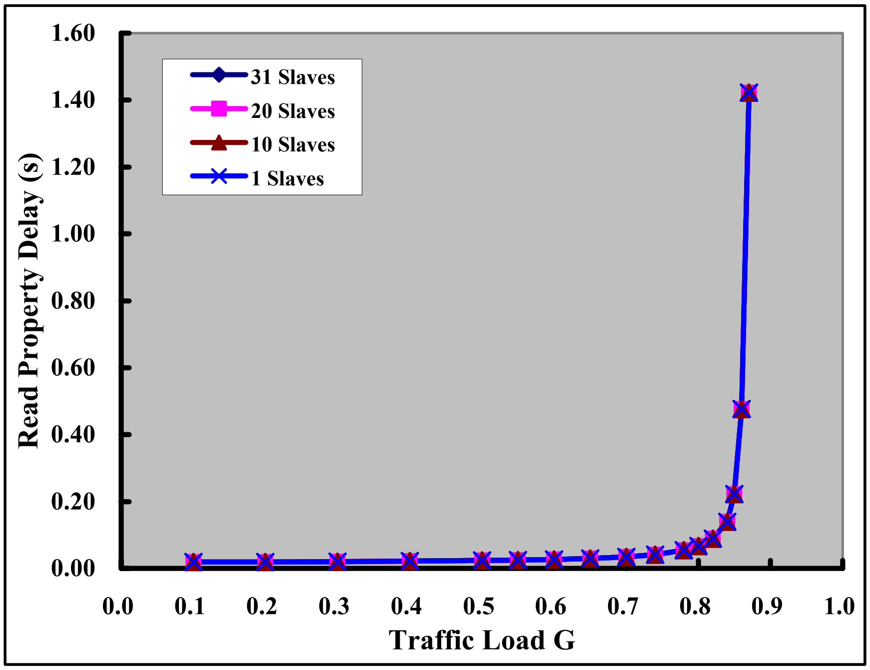


Figure 7. Average service delay for a variable number of slave nodes (single-master).

The average service delay for *ReadPropertyMultiple* requests in a single-master MS/TP network was also investigated. Table 5 shows the simulation conditions. The data rate selected was 76.8 Kbps. The simulation results in Figure 8 show that, as the length of the message is increased, throughput of the network is significantly increased. For a *ReadPropertyMultiple* request with 10 values, the network resource is almost fully utilized. This is because the effect the overhead from $T_{turnaround}$ is reduced as the length of the message is increased. However, increasing the message length also increases the average service delay when the traffic load is low to medium.

Table 5. Simulation Conditions for *ReadPropertyMultiple* Requests (Single-Master)

Service	Message length (bytes) (request/reply)	Message generation interval at the master node (s)	Traffic load *G*
ReadProperty	23/29	0.06771 to 0.00778	0.10000 to 0.87000
ReadPropertyMultiple 1 value	25/31	0.07292 to 0.00829	0.10000 to 0.88000
ReadPropertyMultiple 10 values	106/175	0.36589 to 0.03772	0.10000 to 0.97000

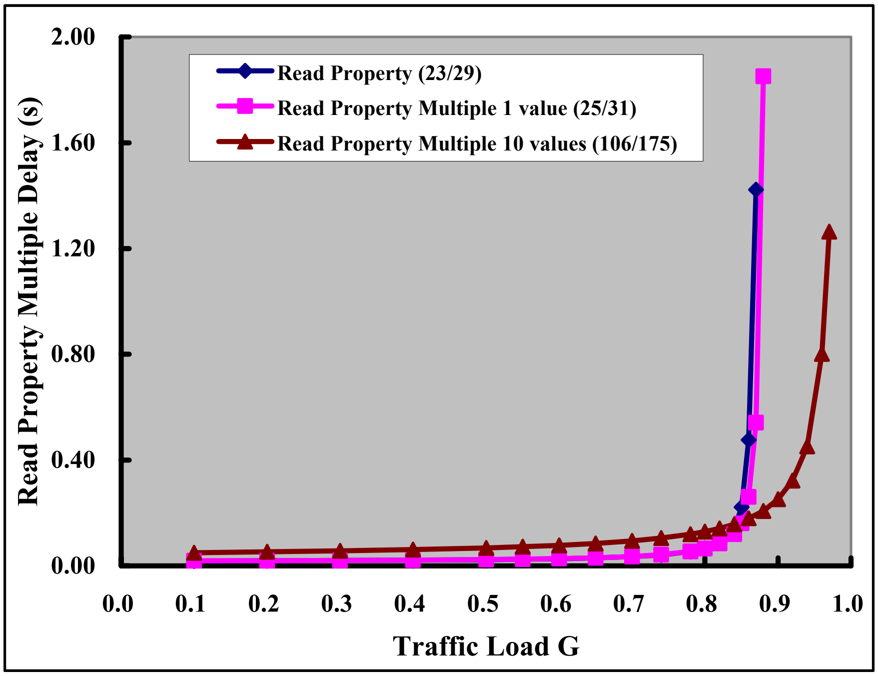


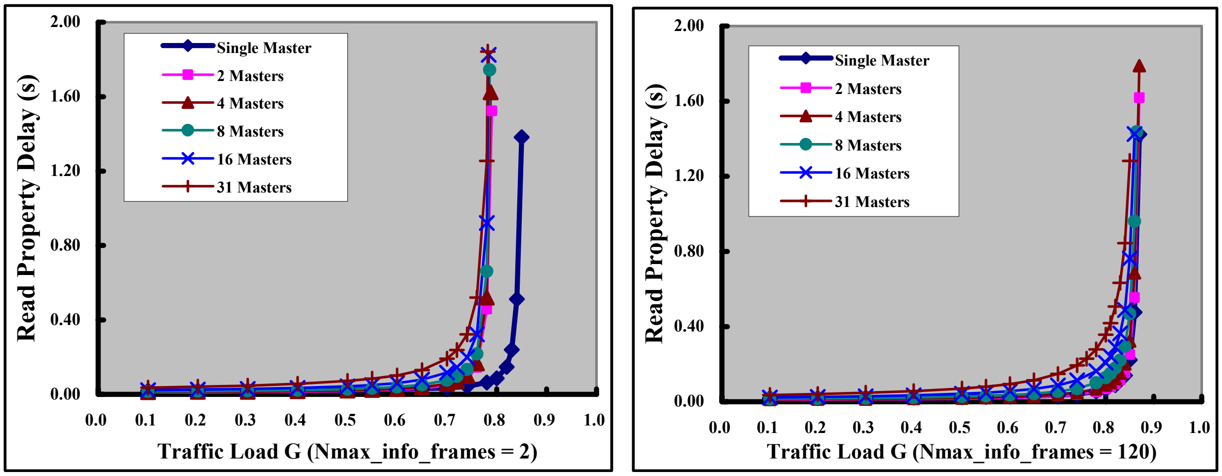
Figure 8. Average Service Delay for *ReadPropertyMultiple* service requests (single-master). The numbers in parenthesis indicate the message length in octets for the request and reply.

4.1.3 Performance Analysis of Multi-Master Systems

In this section the effect of incrementing of the number of master nodes in an MS/TP network on the average service delay for *ReadProperty* requests is analyzed. Table 6 shows the simulation conditions for the multi-master system. In this simulation analysis, the number of master nodes is increased from 1 to 31. The transmission speed was set to 76.8 Kbps and the message generation interval in the master nodes was assumed to have a Poisson distribution. T_{reply_delay} was assumed to be negligible. In this analysis, network traffic load, G, is adjusted by changing the average message generation interval at the master nodes. $N_{max\_info\_frames}$ was set to a small value (2) and a large value (120) and the results were compared.

Table 6. Simulation Conditions for Multi-Master MS/TP Networks

Number of Nodes (master/slave)	N_{max}	Message Generation interval at the master node(s)	Traffic load *G*
1/31	2/120	0.06771 to 0.00778	0.10000 to 0.87000
2/30	2/120	0.13542 to 0.01557	0.10000 to 0.87000
4/28	2/120	0.27083 to 0.03113	0.10000 to 0.87000
8/24	2/120	0.54167 to 0.06262	0.10000 to 0.86500
16/16	2/120	1.08330 to 0.12597	0.10000 to 0.86000
31/1	2/120	2.09890 to 0.24694	0.10000 to 0.85000



(a) $N_{max\_info\_frames} = 2$ (b) $N_{max\_info\_frames} = 120$

Figure 9. Average service delay for *ReadProperty* service requests (multi-master).

Figure 9 shows the simulation results for a multi-master MS/TP system with a varying number of master nodes. Figure 9(a) and Figure 9(b) show the results when $N_{max\_info\_frames}$ is 2 and 120, respectively. As shown in Figure 9, service delay increases as the number of master nodes in the MS/TP network is increased. This is due to the effect of T_{usage_delay} (see Table 2). As the number of master nodes is increased, token-passing delay due to T_{usage_delay} is increased. In these simulations the master nodes were assigned consecutive addresses. The service delay would be expected to increase more dramatically if there were gaps in the addresses because of the resulting increase in the frequency of *Poll For Master* frames. By using consecutive addresses only the master with the highest address needs to poll for new masters attempting to enter the ring.

Figure 9 shows that, if one considers service delay only, the performance of a single-master system is slightly better than that of multi-master system. This is because of the reduced token management overhead. However, there are important application implications to a single-master system because the slave nodes cannot initiate messages. This means that the dynamic discovery features of BACnet (*Who-Is* and *I-Am*, *Who-Has* and *I-Have*) do not work and slaves cannot spontaneously transmit an alarm or change-of-value notification. There is a proposed addendum to BACnet that would provide a mechanism for the master node to serve as a proxy to the slaves to overcome the dynamic discovery limitation.

Slave nodes are somewhat easier and cheaper to implement because the MS/TP state machine is much simpler. When combined with the potential for reduced service delay, there can be significant benefits to an MS/TP network with a mixture of masters and slaves.

Figure 9 shows that the average service delay is affected by the change in $N_{max\_info\_frames}$. The effect of $N_{max\_info\_frames}$ on the network performance was investigated more fully. Table 7 shows the simulation conditions that were used. In these simulations the number of masters was two. The message generation interval was assumed to have a Poisson distribution and transmission speed was 76.8 Kbps. The traffic load was adjusted by changing the average message generation interval of master nodes.

Table 7. Simulation Condition for Investigating N_{max}

Number of nodes (master/slave)	N_{max}	Message generation interval at the master node (s)	Traffic load G
2/30	1	0.13542 to 0.01868	0.10000 to 0.72500
2/30	5	0.13542 to 0.01612	0.10000 to 0.84000
2/30	10	0.13542 to 0.01584	0.10000 to 0.85500
2/30	120	0.13542 to 0.01557	0.10000 to 0.87000
2/30	255	0.13542 to 0.01557	0.10000 to 0.87000

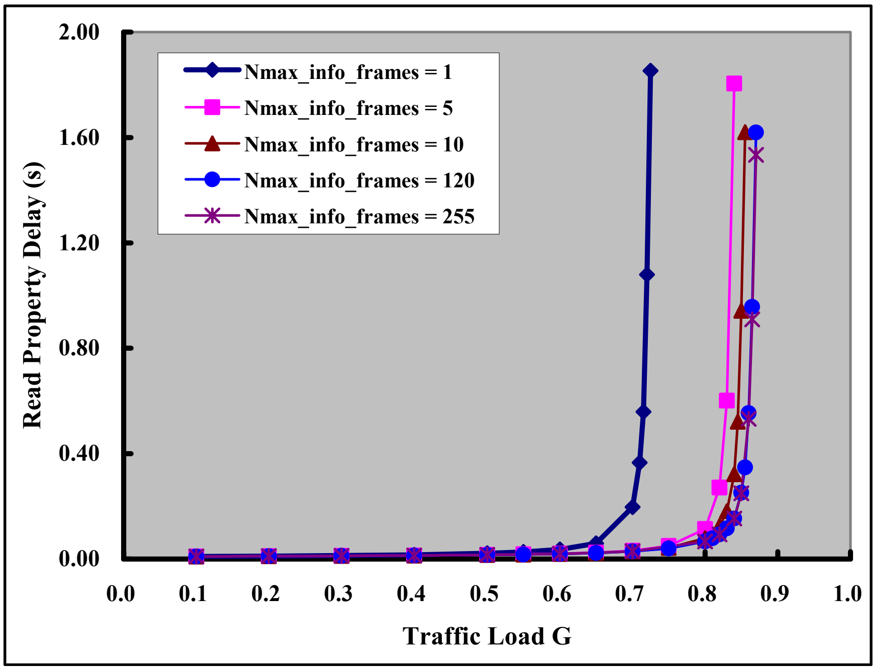


Figure 10. Average service delay for *ReadProperty* service requests with varying N_{max}.

Figure 10 shows the impact of $N_{max\_info\_frames}$ on average service delay. As $N_{max\_info\_frames}$ becomes smaller, master nodes have to exchange the token more frequently, and the relative overhead for token passing becomes larger. Master nodes also have to execute the *Poll For Master* cycle more frequently. Because $N_{max\_info\_frames}$ is a network configuration parameter, the network designer should select a sufficiently large value of $N_{max\_info\_frames}$ in order to reduce the average network-induced service delay. These results indicate that a value of five may be sufficient. Although higher values can reduce the average delay even more, this must be balanced against the possibility that an individual critical message, such as an alarm, might be delayed while waiting for other masters to transmit multiple messages. The marginal increase in performance for $N_{max\_info\_frames} > 5$ is because the message queue length seldom exceeded five.

Most of the MS/TP networks currently used in real buildings are all-master systems. The average service delay for an all-master system was measured and compared with the results for a single-master system. The operation scenario and simulation conditions were exactly the same as that of the single-master system given in Table 3, except that all the nodes are masters. Among them, one master node was responsible for executing all of the service requests. In this analysis, $N_{max\_info\_frames}$ was set to 120. Figure 11 shows the simulation results. As expected, a comparison with the performance of a single-master system given in Figure 5(b) shows that the performance of an all-master system is worse: ReadProperty delay has increased and saturation occurs at a lower value of *G*.

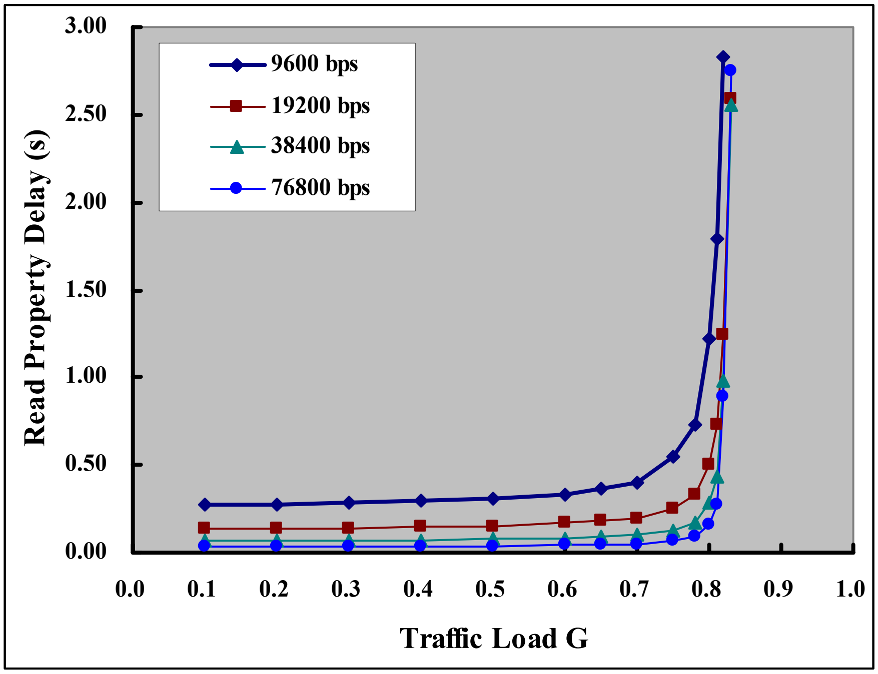


Figure 11. Average service delay for *ReadProperty* service requests in an all-master system.

Figure 12 shows the effect of T_{usage_delay} on service delay in an all-master system when the transmission rate is set to 76.8 Kbps. Figure 12 indicates that a network designer of multi-master MS/TP networks must restrict network traffic load *G* according to the value of T_{usage_delay} of the MS/TP device.

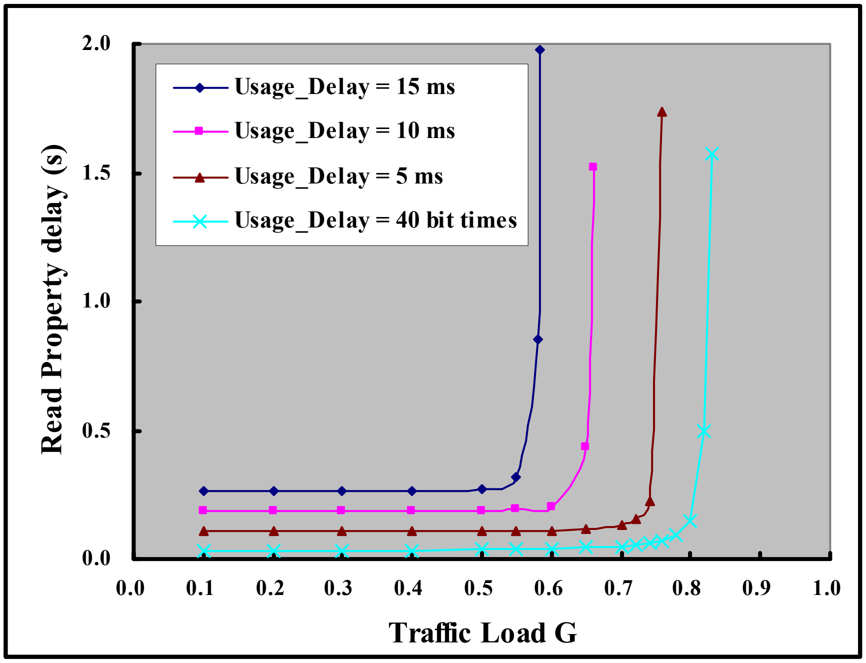


Figure 12. Effect of T_{usage_delay} on average service delay.

4.1.4 Performance Analysis of BACnet Services in MS/TP Networks

This section evaluates the performance of BACnet services over MS/TP networks. Four representative BACnet services, *ReadProperty*, *ReadPropertyMultiple*, *UnconfirmedCOV-Notification* and *ConfirmedCOVNotification*, were considered, and their average service delays were measured. In this study, the MS/TP network consists of all master nodes because most of the MS/TP networks currently operated in real buildings are all-master systems.

In the *ReadProperty* service case, one node acts as a central controller. The controller node sends a *ReadProperty* request message to all the other nodes. Upon receiving the request message, each node immediately returns a reply message. In this analysis, $N_{max\_info\_frames}$ was set to 120. Table 8 shows the simulation conditions used for this case. Figure 13 shows the resulting average service delays for *ReadProperty* service requests with respect to the change in the number of nodes. As shown in Figure 13, average service delay is sensitive to the number of nodes. This is because the overhead of token circulation is increased as the number of nodes in the medium is increased.

Table 8. Simulation Conditions for *ReadProperty* Service Requests in an All-Master MS/TP Network

Message length(bytes) (request/reply)	Number of nodes	Message generation interval at the controller node (s)	Traffic load *G*
23 / 29	5	0.06771 to 0.00787	0.10000 to 0.86000
	10	0.06771 to 0.00792	0.10000 to 0.85500
	20	0.06771 to 0.00801	0.10000 to 0.84500
	30	0.06771 to 0.00816	0.10000 to 0.83000
	60	0.06771 to 0.00857	0.10000 to 0.79000
	90	0.06771 to 0.00897	0.10000 to 0.75500

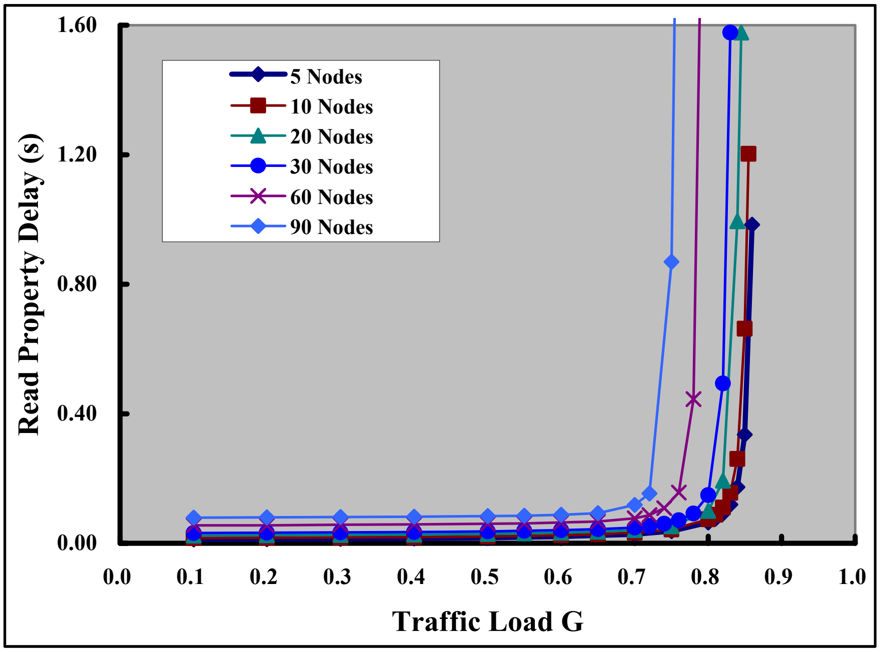


Figure 13. Average service delay for *ReadProperty* requests in an all-master MS/TP network.

Table 9 shows the simulation conditions for *ReadPropertyMultiple* service requests with ten Real values. The simulation results are shown in Figure 14. Compared to the *ReadProperty* service in Figure 13, network resource for the *ReadPropertyMultiple* service is saturated at higher traffic load, thus the throughput performance is increased. This is because the effect of token circulation overhead is reduced in the *ReadPropertyMultiple* service case. However, the service delay is slightly higher for the *ReadPropertyMultiple* service because of the affect of increased message length on transmission time.

Table 9. Simulation Conditions for *ReadPropertyMultiple* Service Requests in an All-Master MS/TP Network

Message length(bytes) (request/reply)	Number of nodes	Message generation interval at the controller node(s)	Traffic load *G*
106 / 175	5	0.36589 to 0.03792	0.10000 to 0.96500
	10	0.36589 to 0.03811	0.10000 to 0.96000
	20	0.36589 to 0.03851	0.10000 to 0.95000
	30	0.36589 to 0.03851	0.10000 to 0.95000
	60	0.36589 to 0.03913	0.10000 to 0.93500
	90	0.36589 to 0.03977	0.10000 to 0.92000

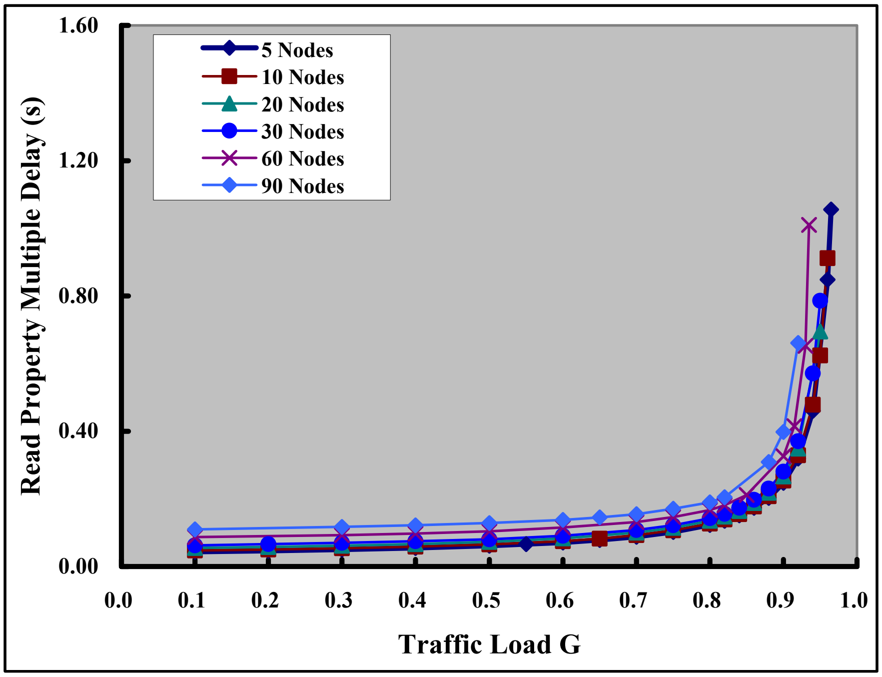


Figure 14. Average service delay for *ReadPropertyMultiple* service requests in an all-master MS/TP network.

Table 10 shows the simulation conditions for *UnconfirmedCOVNotification* service requests. One node is designated as a central controller node. All the other nodes transmit COV notification messages to the central controller node when a COV occurs. The central controller node does not transmit a reply message. Figure 15 shows the simulation results. The average service delay for *UnconfirmedCOVNotification* requests is also affected by the number of master nodes because of the token circulation overhead.

Table 10. Simulation Conditions for *UnconfirmedCOVNotification* Service Requests

Message length(bytes)	Number of nodes	Message generation interval at a node(s)	Traffic load G
45	5	0.23438 to 0.02548	0.10000 to 0.92000
	10	0.52734 to 0.05795	0.10000 to 0.91000
	20	1.11320 to 0.12234	0.10000 to 0.91000
	30	1.69920 to 0.18880	0.10000 to 0.90000
	60	3.45700 to 0.39284	0.10000 to 0.88000
	90	5.21480 to 0.61351	0.10000 to 0.85000

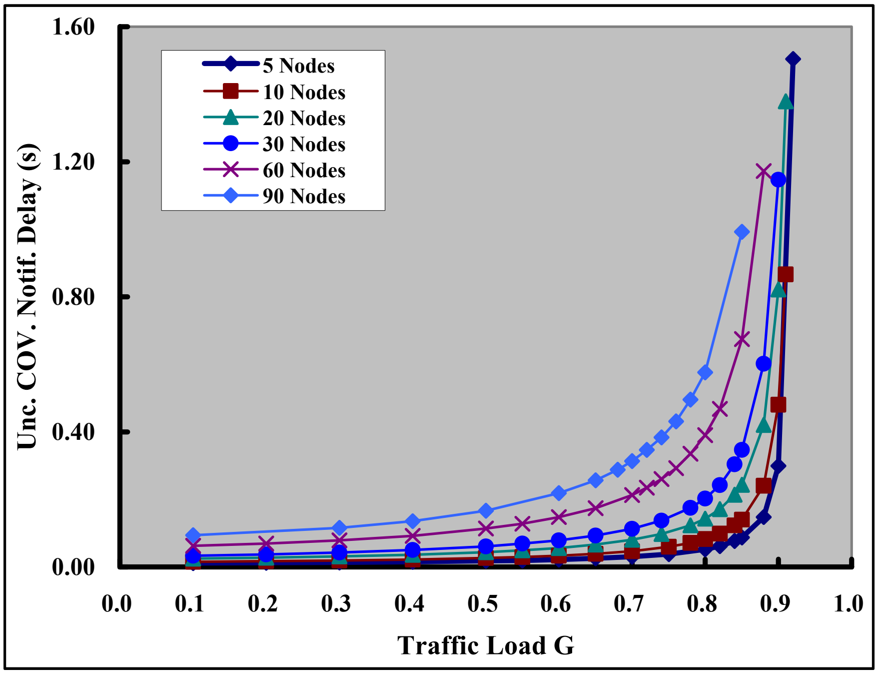


Figure 15. Average service delay for *UnconfirmedCOVNotification* requests in an all-master MS/TP network.

Table 11 shows the simulation conditions for *ConfirmedCOVNotification* service requests. Like the UnconfirmedCOVNotification case, a central controller node receives all of the COV notifications, which are transmitted by the other nodes when a COV occurs. Upon receiving the COV notification message, the central controller node immediately transmits a reply message. Figure 16 shows the simulation results. Comparison with Figure 15 shows that, in an all-master MS/TP network, the difference in service delay between the unconfirmed service and the confirmed service is not significant. The only additional delay for the confirmed service is the transmission delay of a reply message. This is because, in MS/TP networks, the reply is transmitted immediately instead of waiting for the next time the responding node has the token.

Table 11. Simulation Conditions for *ConfirmedCOVNotification* Service Requests

Message length(bytes) request/reply	Number of nodes	Message generation interval(s)	Traffic load *G*
45 / 15	5	0.31250 to 0.03571	0.100000 to 0.87500
	10	0.70313 to 0.08036	0.100000 to 0.87500
	20	1.48430 to 0.17160	0.100000 to 0.86500
	30	2.26560 to 0.26344	0.100000 to 0.86000
	60	4.60930 to 0.55202	0.100000 to 0.83500
	90	6.95310 to 0.84794	0.100000 to 0.82000

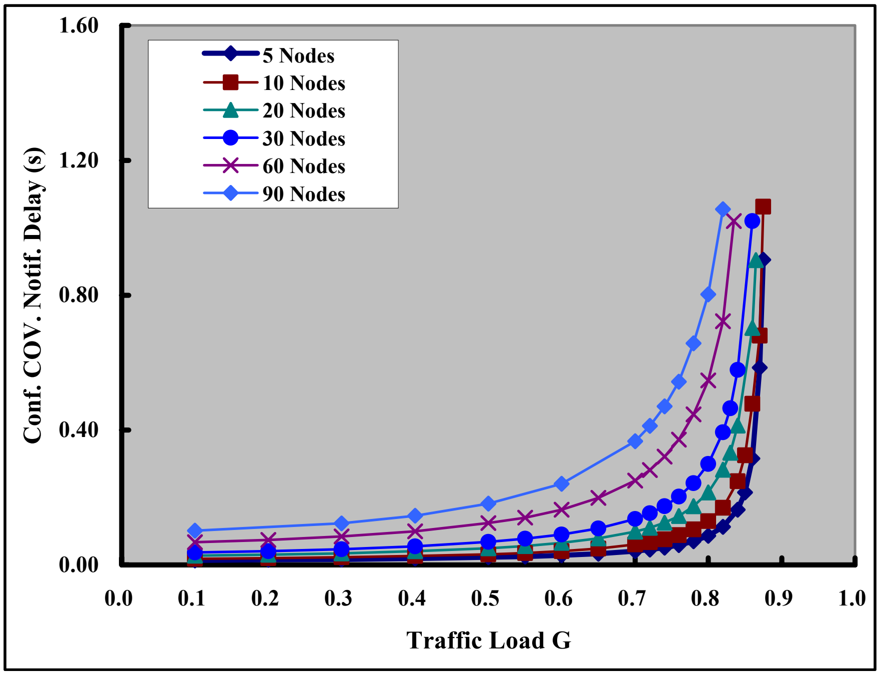


Figure 16. Service delay for *ConfirmedCOVNotification* requests in an all-master MS/TP network.

4.1.5 Effect of processing time on the service delay

In this section, we investigate the effect of processing time on the service delay. In the simulation analysis, the processing time for BACnet application services in the application and user layers are included in the service delay. The MS/TP network is assumed to be made up entirely of master nodes because most of the MS/TP networks currently operated in real buildings are all-master systems. Figure 17 shows the configuration of the MS/TP network considered in this analysis. The MS/TP network is assumed to consist of a BACnet Building Controller (B-BC), BACnet Advanced Application Controllers (B-AACs), and BACnet Application Specific Controllers (B-ASCs). The MS/TP network traffic results from four BACnet application services; *ReadProperty*, *ReadPropertyMultiple*, *UnconfirmedCOV-Notification*, and *ConfirmedCOVNotification*.

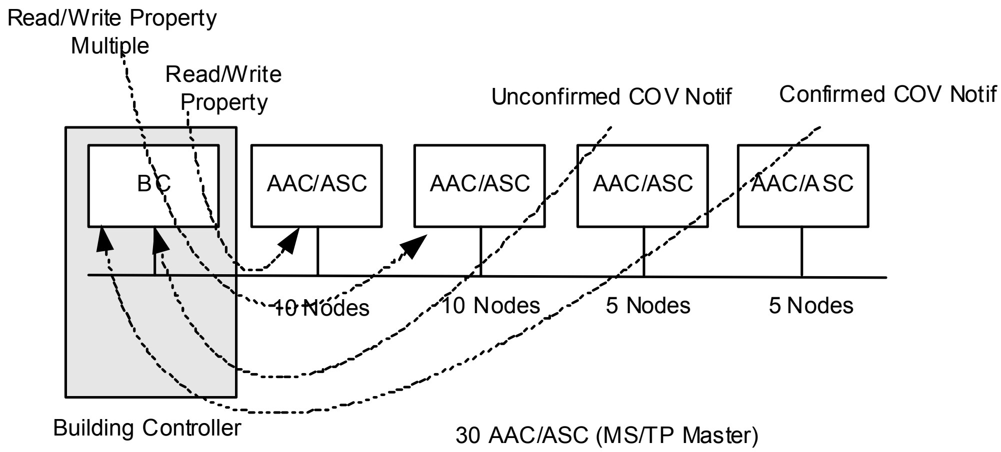


Figure 17. Configuration of MS/TP network simulation with processing time.

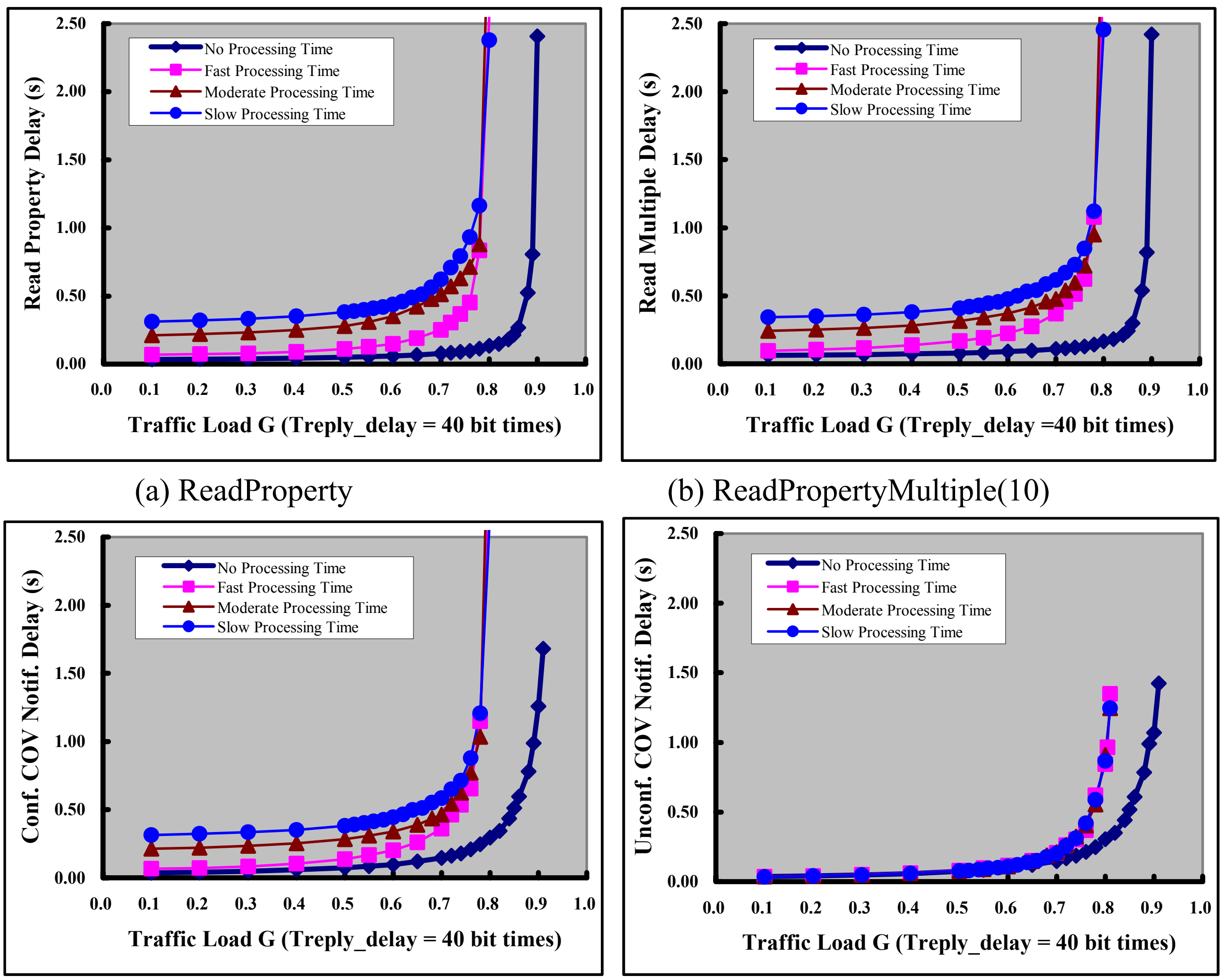
The network consists of 31 nodes (one B-BC and 30 B-AACs/B-ASCs). Ten B-AAC/B-ASC nodes execute *ReadProperty and WriteProperty* service requests from the B-BC. Another ten nodes execute *ReadPropertyMultiple and WritePropertyMultiple* service requests from the B-BC that read or write ten Real values. Five B-AAC/B-ASC nodes initiate *UnconfirmedCOV-Notification* service requests directed to the B-BC. Another five B-AAC/B-ASC nodes initiate *ConfirmedCOVNotification* service requests directed to the B-BC. The network speed is assumed to be 76.8 Kbps. The length of the message is determined from the corresponding application service. The message generation interval is determined such that the traffic load of *ReadProperty/WriteProperty*, *ReadPropertyMultiple/WritePropertyMultiple*, *UnconfirmedCOV-Notification* and *ConfirmedCOVNotification* are 1/3, 1/3, 1/6 and 1/6 of the total traffic load G, respectively. The message generation interval has a Poisson distribution. Table 12 shows the simulation conditions.

Table 12. Simulation of MS/TP with Processing Time

Message Type	Message length (byte) (request/reply)	Message generation interval (s)	Traffic load G
ReadProperty	23 / 29	0.20313 to 0.02232	0.1000 to 0.9100
ReadPropertyMultiple(10)	106 / 175	1.09760 to 0.12062	0.1000 to 0.9100
ConfirmedCOVNotification	45 / 15	2.34370 to 0.25755	0.1000 to 0.9100
UnconfirmedCOVNotification	45	1.75780 to 0.19317	0.1000 to 0.9100

In this simulation analysis, the MS/TP network parameters are determined as follows: $\mathbf{N}_{max\_info\_frames}$ = 120, $\mathbf{T}_{frame\_gap}$ = 0 bit time, $\mathbf{T}_{turnaround}$ = 40 bit times, $\mathbf{T}_{usage\_delay}$ = 40 bit times,

$\mathbf{T}_{reply\_delay}$ = 40 bit times. We considered the following four cases of processing time for application services; (i) 0 ms (no processing time), (ii) 1 ms to 20 ms (fast processing time), (iii) 100 ms to 200 ms (moderate processing time), and (iv) 200 ms to 300 ms (slow processing time). The processing time is assumed to have a uniform distribution within the given range. The processing time includes $\mathbf{T}_{reply\_delay}$, which is defined as the maximum time a node may wait after reception of a frame that expects a reply before sending the first octet of a reply or Reply Postponed frame.



(a) ReadProperty (b) ReadPropertyMultiple(10)

(c) ConfirmedCOVNotification (d) UnconfirmedCOVNotification

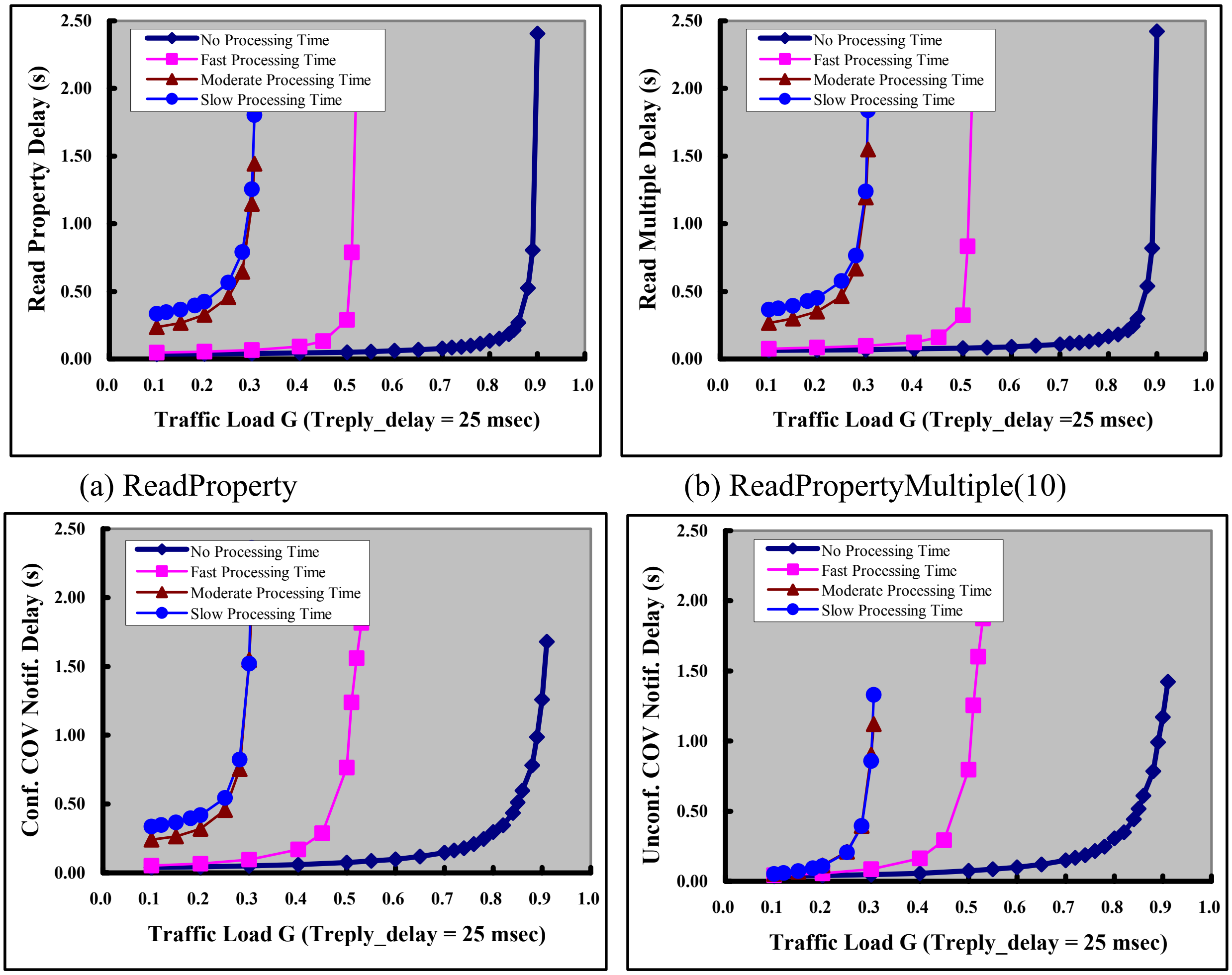
Figure 18. Average service delay of MS/TP with processing time ($\mathbf{T}_{reply\_delay}$= 40 bit times).

Figure 18 shows the simulation results for *ReadProperty*, *ReadPropertyMultiple (10)*, *UnconfirmedCOVNotification* and *ConfirmedCOVNotification* with respect to the change of traffic load and processing time. As we have already examined in section 4.1.4, the service delays increase exponentially as the traffic load is increased. Note that, the delay of confirmed services, *ReadProperty*, *ReadPropertyMultiple (10)*, and *ConfirmedCOVNotification* are almost identical and they are saturated at the same traffic load. This is because these services are sharing the same transmitter queue in the B-BC, and their queuing delays are almost identical. The only difference in these service delays is message transmission time, which is much smaller than the queuing delay and processing time. Even though the processing time varies from fast to slow, network resource is saturated at the same traffic load.

The simulation results in Figure 18 show that, as the processing time for BACnet application services is increased, the service delays of MS/TP networks increase. This is because MS/TP is operated on a request/reply mechanism. When a request that expects a reply is sent to an MS/TP node, the sender waits for the reply to be returned before passing the token. If the processing time of the BACnet application service is increased, an MS/TP node will hold the token longer. This causes an increase in token rotation time. More messages will build up in the transmitter queue of the nodes as the token rotation time is increased. The increase in queuing delay causes a corresponding increase in service delay.

When processing delay is considered, the service delay for *ConfirmedCOVNotification* is longer than the delay for *UnconfirmedCOVNotification*. However, Figure 18(d) shows that processing time also affects the service delay for the unconfirmed service. This is because the four application services used in this simulation analysis share the resource of one MS/TP network. The increased token rotation time caused by the processing time of the confirmed services also increases the queuing delay for the unconfirmed service.

When the processing delay time exceeds $\mathbf{T}_{\text{reply\_delay}}$ the responding node returns a *Reply Postponed* frame, indicating that the actual reply will be returned later. In order to investigate the effect of $\mathbf{T}_{\text{reply\_delay}}$ on the service delay, we compared two values for $\mathbf{T}_{\text{reply\_delay}}$, 40 bit times (the minimum possible) and 25 msec. Figure 19 shows the simulation results. As shown in Figure 19, the increase in $\mathbf{T}_{\text{reply\_delay}}$ severely degrades the performance of service delay, especially when the processing time exceeds $\mathbf{T}_{\text{reply\_delay}}$. This is because the larger value of $\mathbf{T}_{\text{reply\_delay}}$ has a greater impact on the token circulation time when the processing time is large. In an MS/TP local network, the processing time for application services significantly affects the service delay. The system designers must carefully consider processing time and $\mathbf{T}_{\text{reply\_delay}}$ when using MS/TP networks.



(a) ReadProperty

(b) ReadPropertyMultiple(10)

(c) ConfirmedCOVNotification

(d) UnconfirmedCOVNotification

Figure 19. Average service delay of MS/TP with processing time ($\mathbf{T}_{reply\_delay}$= 25 ms).

4.2 ARCNET Networks

4.2.1 Summary of ARCNET Features

ARCNET [12] is a token passing protocol that supports a range of data transmission rates (156.25 kbps to 2.5 Mbps) and a variety of network media including twisted pair, coaxial cable and fiber optic cable. In BACnet systems it is more common to use 156.25 Kbps because it makes use of low cost twisted pair wiring. ARCNET provides faster transmission speeds and more media options than MS/TP. Unlike MS/TP, ARCNET permits a node to transmit only one message when it receives the token even if there is more than one message in the transmitter queue. Upon receipt of a confirmed request, an ARCNET node must wait for the token before transmitting a reply.

4.2.2 Transmission Delay in ARCNET Networks

In this section the performance of ARCNET is evaluated in terms of *transmission delay*. In this study, transmission delay is defined as the time interval from the instant when a message arrives at the transmitter queue of a source node to the instant when the same message has completely arrived at the receiver queue of the destination node. The performance of ARCNET is measured with respect to the change of traffic load G . The traffic load is adjusted by changing the number of nodes, message length and the message generation interval at each node.

Table 13 summarizes the simulation conditions for ARCNET. In this simulation analysis, a transmission rate of 156.25 Kbps was considered. The magnitude of the transmission delay is a function of the data transmission speed but the network saturation results are applicable to other speeds because the analysis is in terms of the normalized traffic load, G. The maximum allowable message length in the ARCNET packet is 517 bytes. The message generation in each node was assumed to have Poisson distribution.

Table 13. Simulation Conditions for ARCNET

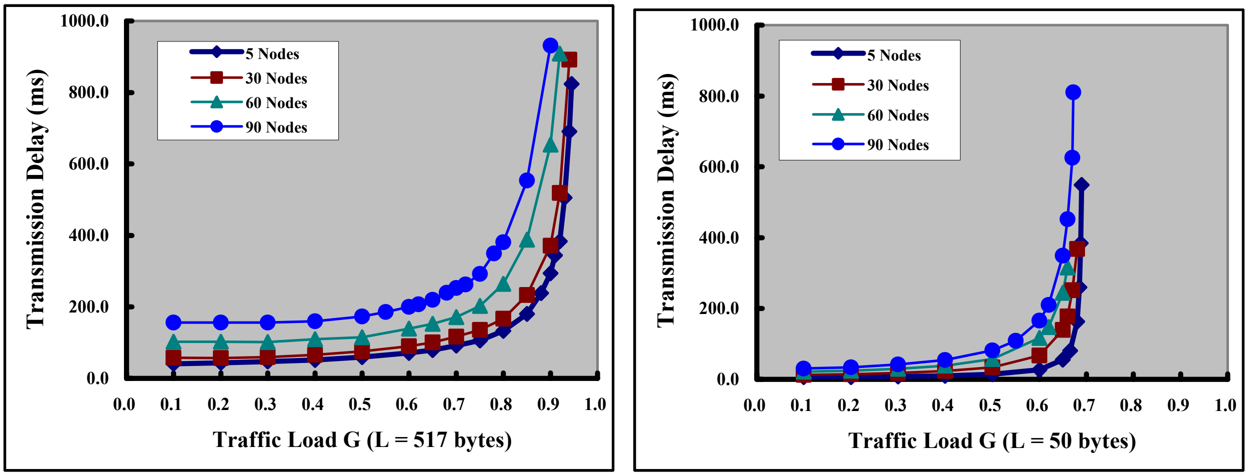
Message length (bytes)	Number of nodes	Message generation interval at a node (s)	Traffic load G
517	5	1.81820 to 0.19241	0.10000 to 0.9450
	30	10.90900 to 1.1605	0.10000 to 0.9400
	60	21.81800 to 2.3716	0.10000 to 0.9200
	90	32.72800 to 3.6364	0.10000 to 0.9000
200	5	0.70240 to 0.07804	0.10000 to 0.9000
	30	4.21440 to 0.47353	0.10000 to 0.8900
	60	8.42880 to 0.96882	0.10000 to 0.8700
	90	12.64300 to 2.94360	0.10000 to 0.8600
50	5	0.17440 to 0.02528	0.10000 to 0.6900
	30	1.04640 to 0.15618	0.10000 to 0.6700
	60	2.09280 to 0.31709	0.10000 to 0.6600
	90	3.13920 to 0.50632	0.10000 to 0.6200

Figures 20 – 22 show the simulation results. Figure 20 shows the transmission delay with respect to the change of the number of nodes when message lengths are 517 bytes and 50 bytes. Figure 20 indicates that transmission delay increases as the number of nodes in the medium increases. The figure also shows that, as the message length is increased, the network resource is saturated at higher value of offered traffic. This is because the effect of the protocol overhead is reduced as message size increases.

Figure 21 shows the transmission delay with respect to the change of message length when the number of nodes is 5 and 90. The figure shows that the transmission delay is less for shorter messages when the traffic load is low. However, the network resource experiences saturation at lower offered traffic for shorter messages

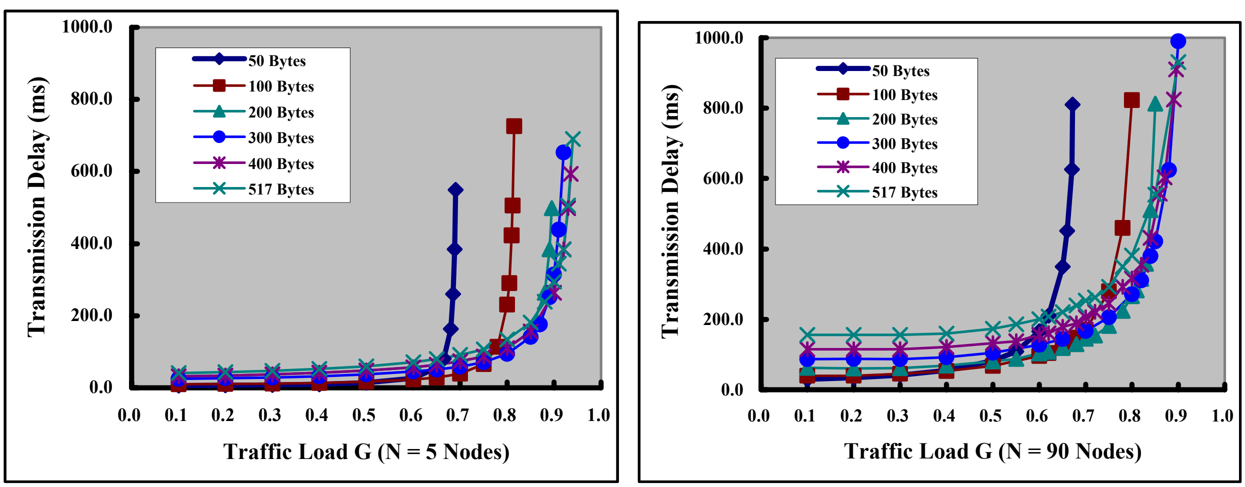
Figure 22 shows a 3-D graph of message transmission delay. The surfaces in Figure 22 represents traffic loads G =0.1, 0.2, 0.3, 0.4, 0.5, with G = 0.1 at the bottom. As shown in Figure

22, at the same offered traffic, transmission delay is more severely affected by the increment of the number of nodes rather than the increment of message length. This is because of the increase in token delivery overhead as the number of nodes increases.



(a) $L = 517$ bytes (b) $L = 50$ bytes

Figure 20. Transmission Delay in ARCNET Networks with a Change in the Number of Nodes



(a) $N = 5$ (b) $N = 90$

Figure 21. Transmission Delay in ARCNET Networks with a Change in Message Length

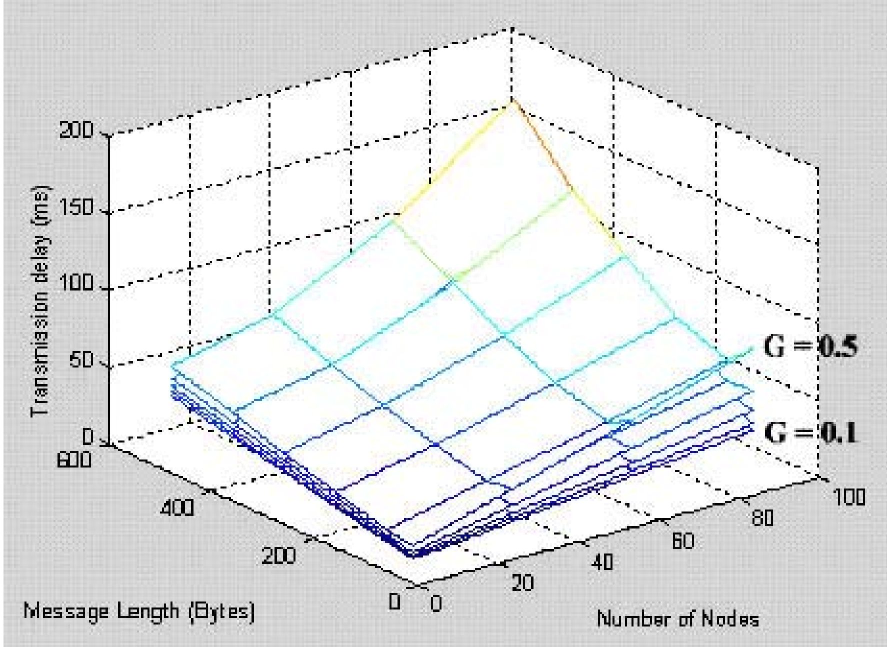


Figure 22. Transmission Delay in ARCNET Networks.

4.2.3 Performance Analysis of BACnet Services in ARCNET Networks

This section evaluates the performance of BACnet services over ARCNET networks. In a manner similar to the analysis described in Section 4.1.4 for MS/TP networks, four representative BACnet services, *ReadProperty*, *ReadPropertyMultiple*, *UnconfirmedCOV-Notification* and *ConfirmedCOVNotification*, were considered, and their average service delays were measured.

In the *ReadProperty* service case, one node acts as a central controller. Upon capturing the token, the controller node sends a request message to one of the other nodes on the network. A node that receives a request message returns a reply message to the controller node when it captures the token. Table 14 shows the simulation conditions for this case and the simulation results are shown in Figure 23.

Due to the effect of token circulation overhead, service delay is sensitive to the number of nodes in the network. By comparing the ARCNET results in Figure 23 with the MS/TP results in Figure 13 it can be seen that the network resource in ARCNET networks saturates at a much lower traffic load G. This indicates that the effect of token circulation overhead in ARCNET networks is greater than in MS/TP networks. This is because ARCNET nodes must always wait for the token before transmitting a reply and only a single message can be transmitted when holding the token. When the number of nodes in the network is 30, the network becomes saturated even when G is less than 0.2.

Table 14. Simulation Conditions for *ReadProperty* Requests

Message length(bytes) (request/reply)	Number of nodes	Message generation interval at the controller node (s)	Traffic load G
22 / 28	5	0.06912 to 0.00789	0.05000 to 0.43800
	10	0.06912 to 0.01016	0.05000 to 0.34000
	20	0.06912 to 0.01464	0.05000 to 0.23600
	30	0.06912 to 0.01920	0.05000 to 0.18000

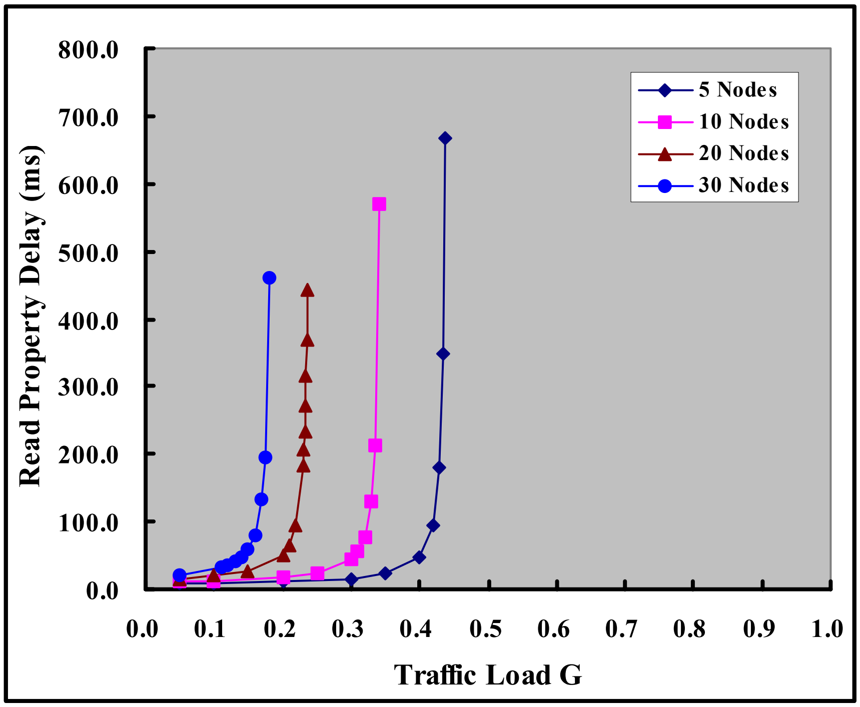


Figure 23. Average Service Delay for *ReadProperty* Requests.

Table 15 shows the simulation conditions for *ReadPropertyMultiple* service requests with ten Real values. As before, one node makes all of the read requests. The simulation results are shown in Figure 24. Comparison with the *ReadProperty* service delay in Figure 23 indicates that using the *ReadPropertyMultiple* service increases the throughput of network because of the reduced effect of token circulation overhead. The performance improvement for ARCNET is much more significant than it was for MS/TP networks (compare Figures 23 and 24 with Figures 13 and 14). Thus, when using ARCNET networks, it is particularly desirable to use the *ReadPropertyMultiple* service rather than several repetitions of the *ReadProperty* service.

Table 15. Simulation Conditions for *ReadPropertyMultiple* Service Requests

Message length(bytes) (request/reply)	Number of nodes	Message generation interval at the controller node(s)	Traffic load *G*
105 / 174	5	0.39155 to 0.02417	0.05000 to 0.81000
	10	0.39155 to 0.02646	0.05000 to 0.74000
	20	0.39155 to 0.03108	0.05000 to 0.63000
	30	0.39155 to 0.03560	0.05000 to 0.55000

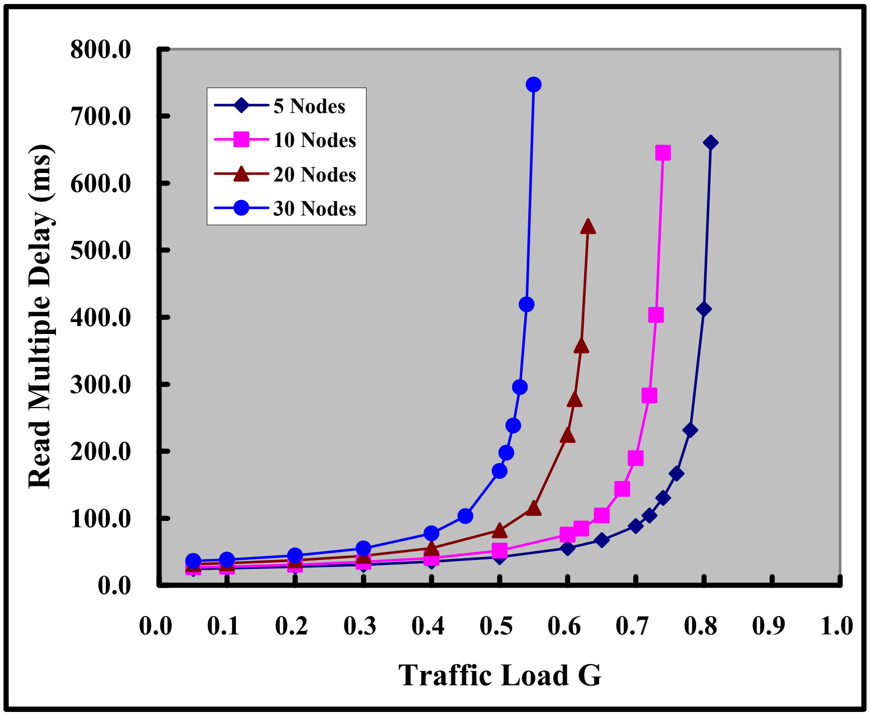


Figure 24. Average Service Delay for *ReadPropertyMultiple* Service Requests.

Table 16 shows the simulation conditions for *UnconfirmedCOVNotification* service requests. One node is designated as a central controller node. All the other nodes transmit COV notification messages to the central controller node when a COV occurs. This service is completed in one token circulation because unconfirmed service does not require a reply.

Figure 25 shows the simulation results. The average service delay for *UnconfirmedCOV-Notification* requests is identical to the transmission delay. Because unconfirmed services require only one token circulation, their performance is less affected by the change in the number of nodes. The average service delay is increased abruptly when G exceeds 0.6 and the network resource for *UnconfirmedCOVNotification* service becomes saturated.

Table 16. Simulation Conditions for *UnconfirmedCOVNotification* Service Requests

Message length (bytes)	Number of nodes	Message generation interval at a node(s)	Traffic load G
44	5	0.24525 to 0.01901	0.05000 to 0.64500
	10	0.55181 to 0.04245	0.05000 to 0.65000
	20	1.16490 to 0.08961	0.05000 to 0.65000
	30	1.77800 to 0.13677	0.05000 to 0.65000

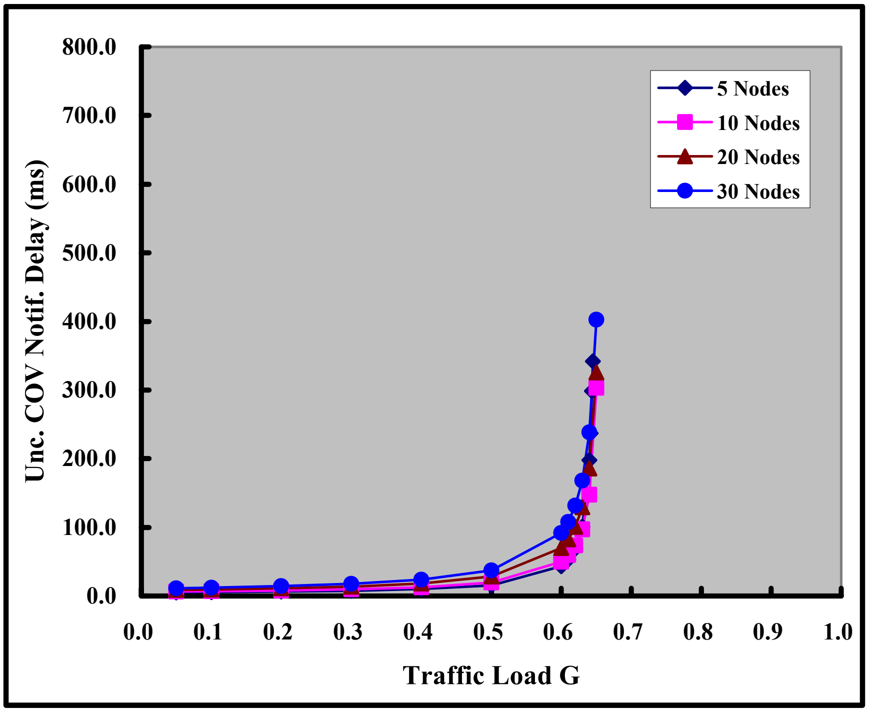


Figure 25. Average Service Delay for *UnconfirmedCOV-Notification* Requests

Table 17 shows the simulation conditions for *ConfirmedCOVNotification* service requests. Like the *UnconfirmedCOVNotification* case, a central controller node receives all of the COV notifications, which are transmitted by the other nodes when a COV occurs. Upon receiving the COV notification message, the central controller node transmits a reply message. Figure 26 shows the average service delay for *ConfirmedCOVNotification*.

For MS/TP networks, the difference in average service delay between the *ConfirmedCOV-Notification* and *UnconfirmedCOVNotification* was found to be negligible (see Figures 15 and 16). However, a comparison of Figure 25 and Figure 26 shows a significant difference in performance between *ConfirmedCOVNotification* and *UnconfirmedCOVNotification* in ARCNET networks. This is because ARCNET nodes must wait for the token before transmitting a reply. In addition, because the controller node can transmit only one reply message when it captures the token, the reply messages are built up in the transmitter queue, and the service delay increases rapidly. Since most BACnet services are confirmed, traffic on ARCNET networks needs to be restricted to $0.15 < G < 0.45$ depending on the number of nodes in the network

Table 17. Simulation Conditions for *ConfirmedCOVNotification* Service Requests

Message length (bytes) request/reply	Number of nodes	Message generation interval at a node (s)	Traffic load G
44 / 14	5	0.32154 to 0.03406	0.05000 to 0.47000
	10	0.72346 to 0.09672	0.05000 to 0.37400
	20	1.52730 to 0.28817	0.05000 to 0.26500
	30	2.33110 to 0.57701	0.05000 to 0.20200

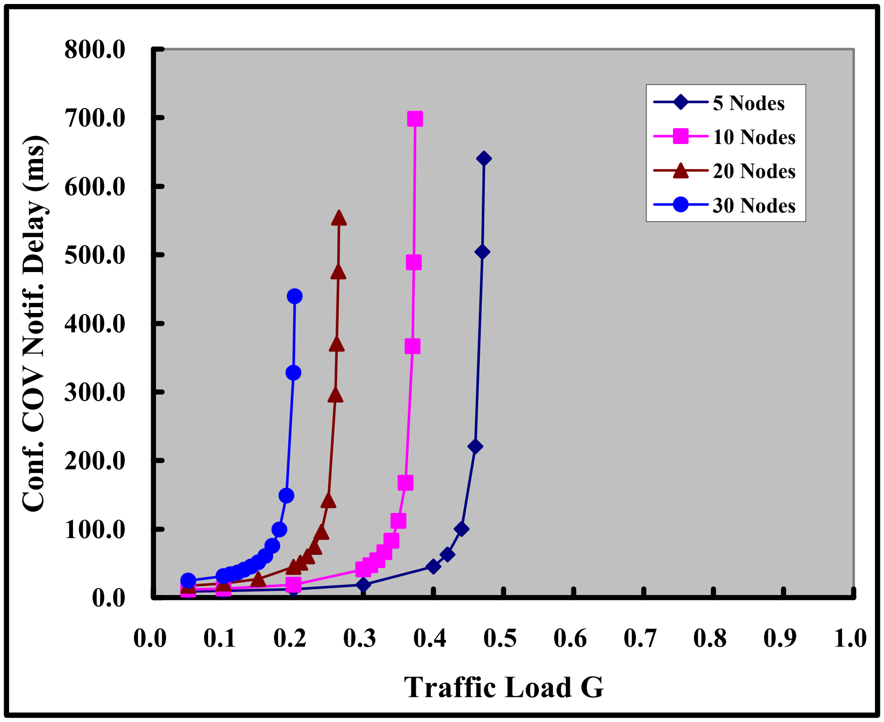


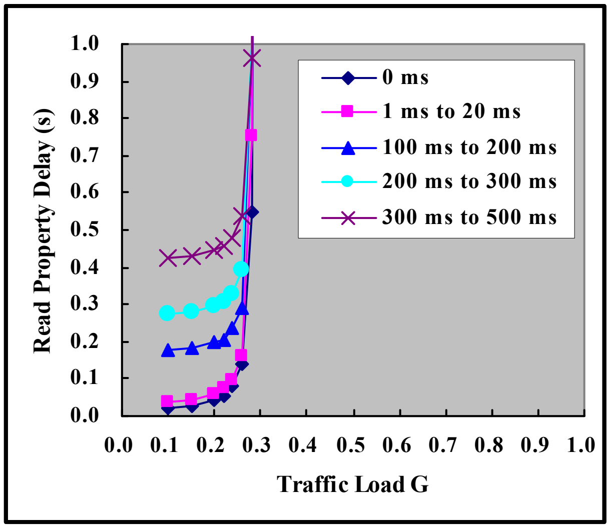
Figure 26. Average Service delay for *ConfirmedCOVNotification* requests.

4.2.4 Effect of processing time on the service delay

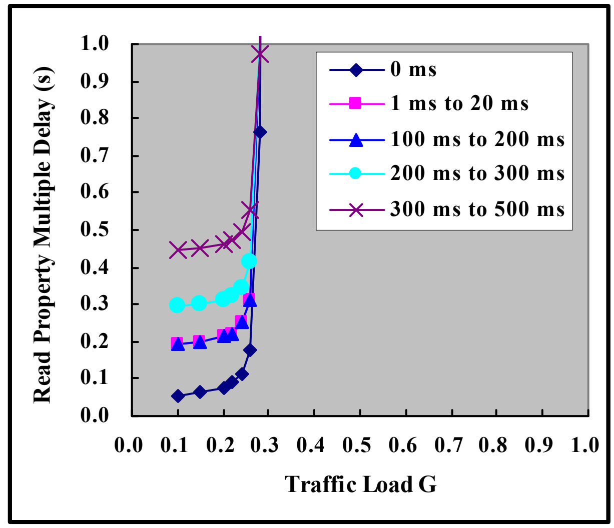
This section presents the effect of processing delay on the performance of ARCNET service delay. Similar to the previous analysis of MS/TP networks, the following four BACnet application services, *ReadProperty, ReadPropertyMultiple, UnconfirmedCOVNotification* and *ConfirmedCOVNotification*, are executed. The configuration of the network system and BACnet application services are the same as those for MS/TP given in Figure 17 except that MS/TP master nodes are replaced by ARCNET nodes. The data rate for the ARCNET network is assumed to be 156.25 Kbps. The simulation conditions are exactly same as those for MS/TP except that the ARCNET frame overhead is applied in message length. Table 18 shows the simulation conditions used.

Table 18. Simulation of ARCNET with Processing Time

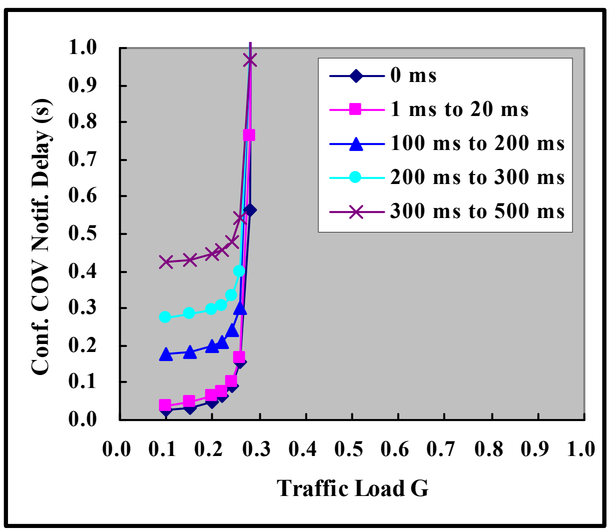
Message Type	Message Length (bytes) (request/reply)	Message Generation Interval (s)	Traffic Load *G*
ReadProperty	22 / 28	0.10368 to 0.01296	0.0333 to 0.2666
ReadPropertyMultiple (10)	105 / 174	0.58732 to 0.07342	0.0333 to 0.2666
ConfirmedCOVNotification	44 / 14	1.2057 to 0.15072	0.0166 to 0.1333
UnconfirmedCOVNotification	44	0.91968 to 0.11496	0.0166 to 0.1333



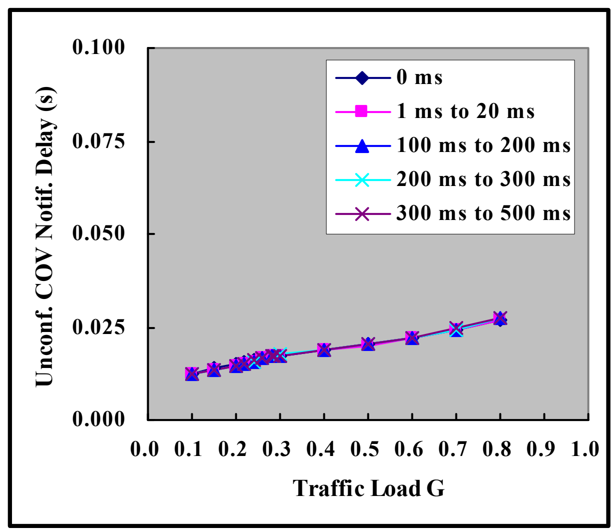
(a) ReadProperty



(b) ReadPropertyMultiple (10)



(c) ConfirmedCOVNotification



(d) UnconfirmedCOVNotification

Figure 27. Average service delay of ARCNET with processing time

Figure 27 shows the average service delay for *ReadProperty*, *ReadPropertyMultiple (10)*, *UnconfirmedCOVNotification,* and *ConfirmedCOVNotification* services in the ARCNET network. As we have already examined in section 4.2.3, ARCNET provides low efficiency when it delivers BACnet messages that require a confirmed service. This is because the nodes can transmit only one message at a time when they capture the token. It may require several token transactions to execute a confirmed service. In this analysis, confirmed services are sharing the transmission queue of the B-BC. As we have already seen in the MS/TP network analysis (section 4.1.5), the delay for confirmed services, *ReadProperty*, *ReadPropertyMultiple (10)*, and *ConfirmedCOVNotification* are almost identical and they are saturated at the same traffic load.

It is interesting to note that the processing time for application services does not affect the network-induced delay in ARCNET networks. Figures 26 (a), (b) and (c) show that the service delay increases linearly as the processing time is increased, i.e., the service delay is increased as much as the processing time is increased. This is because the ARCNET node does not wait for the reply to be returned before passing the token. The responding node sends the reply when it captures the token. This is different from the MS/TP case (see Figure 18) where the token rotation time can increase because nodes are waiting for a reply before passing the token. Figure 27 (c) indicates that the service delay for unconfirmed services is not affected by the change of processing time. Figure 27 also shows that the change of processing time does not influence the point at which the network becomes saturated.

In order to confirm these results, we measured the average token rotation time for ARCNET with respect to the change of processing times. As shown in Figure 28, token circulation time was not affected by the change of processing time. Its value is also quite small compared to the processing time. Comparing Figure 28 with Figure 27 (d) confirms that the token rotation time is the dominant influence in service delay for unconfirmed services.

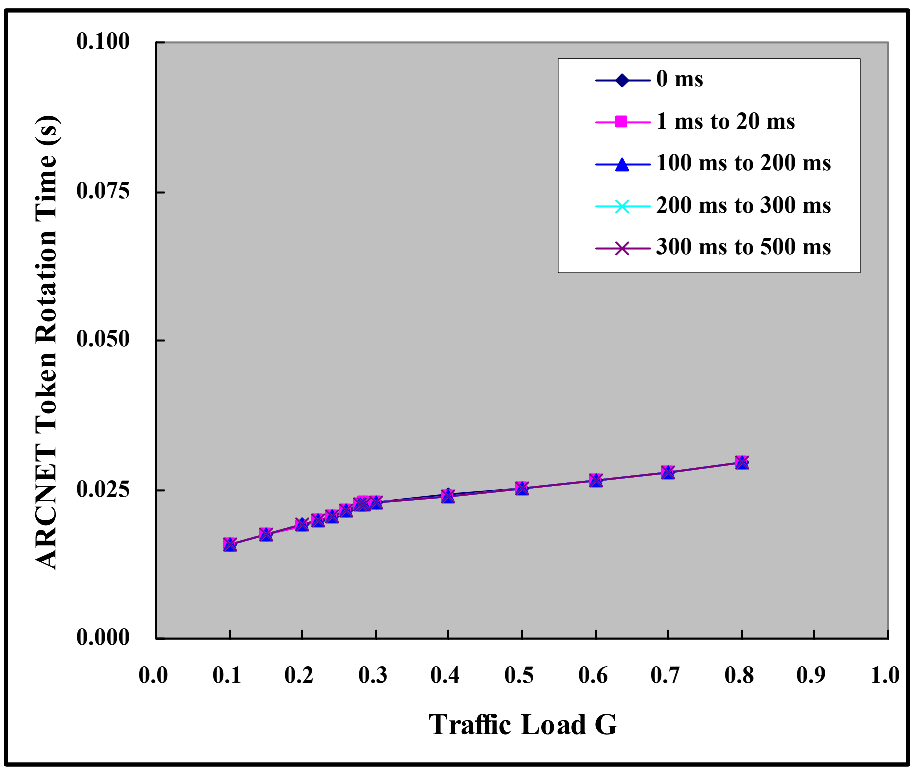


Figure 28. Average token rotation time in the ARCNET

4.3 Ethernet Networks

4.3.1 Summary of Ethernet Features

Ethernet [13] is the most widely used LAN technology in the world. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD) [11]. On a CSMA/CD network, nodes monitor the network to determine if it is busy. A node wishing to send data waits for an idle condition then transmits its message. A collision can occur when two nodes transmit at the same time, thus nodes must monitor the network when they transmit. When a collision happens both nodes stop transmitting frames and transmit a jamming signal. This informs all nodes on the network that a collision has occurred. Each of the nodes then waits a random period (back off) before attempting a retransmission. Nodes thus contend for the network and are not guaranteed access to it. Collisions generally slow down the network. Each node on the network must be able to detect collisions and must be capable of transmitting and receiving simultaneously. Ethernet transmission rates are 10 Mbps, 100 Mbps, or 1 Gbps. A variety of media can be used including coaxial cable, twisted-pair, and fiber optics. BACnet allows any of these options and permits them to be combined.

4.3.2 Performance Analysis of Ethernet Networks

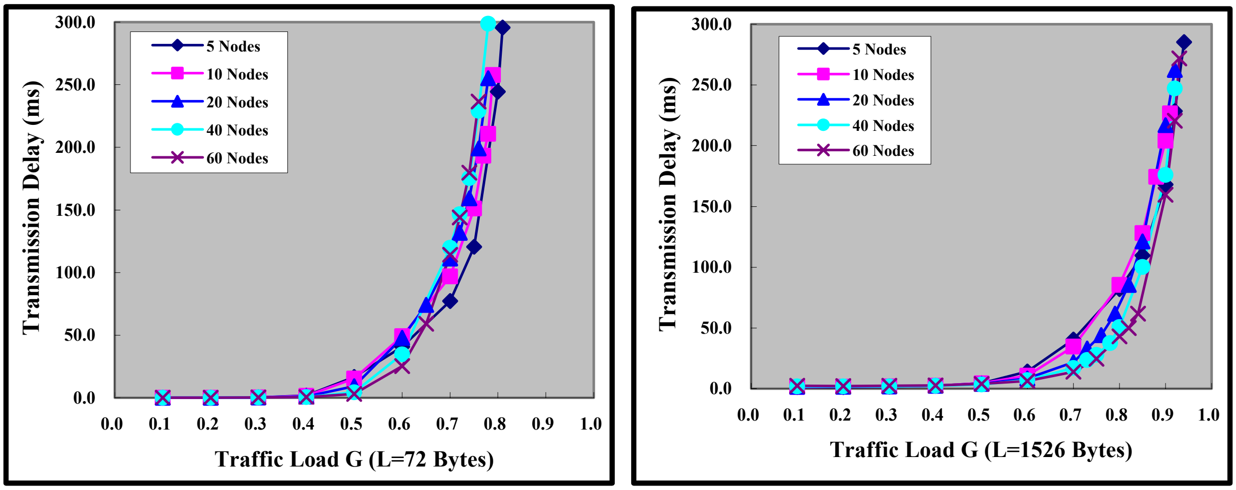
In this section the performance of Ethernet networks is evaluated in terms of transmission delay for a varying traffic load G. The traffic load is adjusted by the changing of the number of nodes, message length and message generation interval. Table 19 shows a part of the simulation conditions for Ethernet. The data rate used was 10 Mbps. The message generation interval in each node was assumed to have a Poisson distribution.

Table 19. Simulation Conditions for Ethernet

Number of nodes	Message length (bytes)	Message generation interval at a node (ms)	Traffic load G
5	64	2.5600 to 0.3200	0.1000 to 0.8000
	100	4.0000 to 0.4651	0.1000 to 0.8600
	200	8.0000 to 0.9091	0.1000 to 0.8800
	500	20.0000 to 2.1739	0.1000 to 0.9200
	1000	40.0000 to 4.2100	0.1000 to 0.9400
	1518	60.7200 to 6.3916	0.1000 to 0.9500
60	64	30.7200 to 4.0960	0.1000 to 0.7500
	100	48.0000 to 6.0000	0.1000 to 0.8000
	200	96.0000 to 11.4286	0.1000 to 0.8400
	500	240.0000 to 27.5862	0.1000 to 0.8700
	1000	480.0000 to 54.5454	0.1000 to 0.8800
	1518	728.6400 to 80.9600	0.1000 to 0.9000

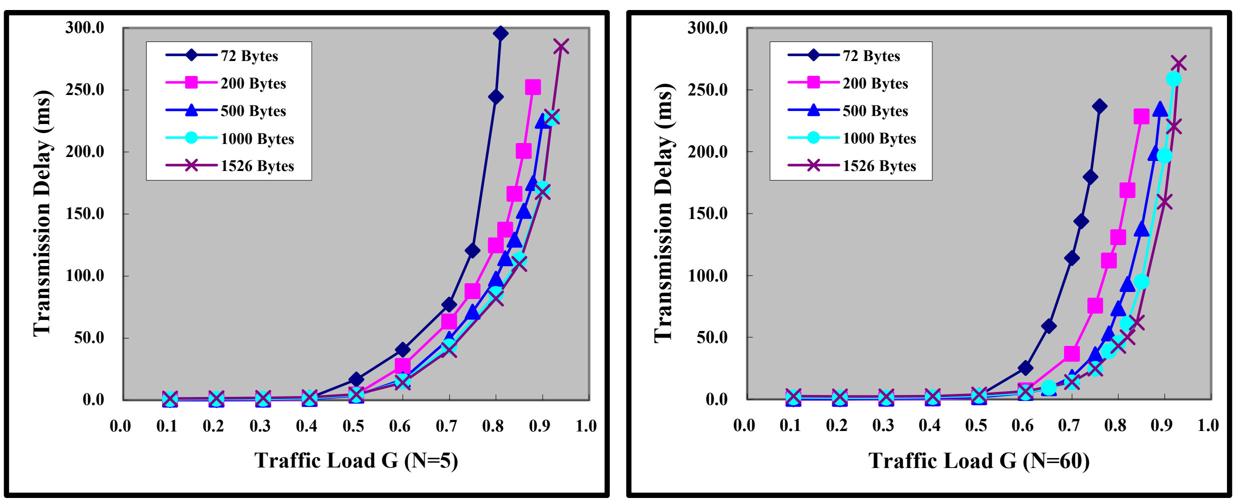
Figures 29 – 31 show the simulation results. Figure 29 indicates that the number of nodes does not affect transmission delay. This is different from the ARCNET case (see Figure 20) where transmission delay was proportional to the number of nodes. This difference is because Ethernet does not experience token overhead at each node. Figure 29 shows that, as the message length increases, the network resource becomes saturated at higher value of traffic load. This phenomenon can also be seen in Figure 30.

Figure 31 shows a 3-D graph of message transmission delay for Ethernet networks. The surfaces in Figure 31 represents traffic loads $G = 0.1$, 0.2, 0.3, 0.4, 0.5,0.6, with G = 0.1 at the bottom. The figure shows that transmission delay increases as the traffic load is increased. Compared to the ARCNET networks (see Figure 22) the delay characteristics of Ethernet are more randomized.



(a) $L = 72$ bytes (b) $L = 1526$ bytes

Figure 29. Transmission delay of Ethernet with the change of the number of nodes.



(a) $N = 5$ (b) $N = 60$

Figure 30. Transmission delay of Ethernet with the change of message length

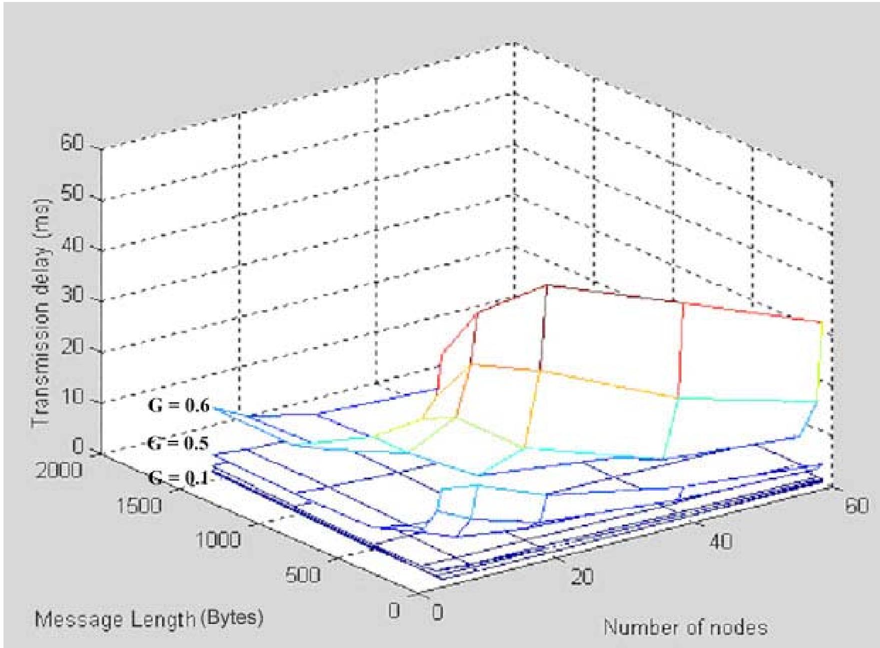


Figure 31. Transmission delay in Ethernet Networks.

4.3.3 Performance Analysis of BACnet Services in Ethernet Networks

This section evaluates the performance of BACnet services over Ethernet networks. The same four representative BACnet services used with the other network technologies were simulated, *ReadProperty*, *ReadPropertyMultiple*, *UnconfirmedCOVNotification* and *ConfirmedCOV-Notification*. In addition, the average service delay of the *AtomicWriteFile* service was measured because Ethernet is often used as a backbone network.

In the *ReadProperty* service simulation, one node acts as a central controller node. The node sends a request message to the other nodes in the network whenever it is ready in the transmitter queue. The message may experience a collision before being delivered to its destination node. A node that receives a request message returns a reply message. It may also experience collision.

Table 20 shows the simulation conditions for this case and the results are shown in Figure 32. In Ethernet networks, the number of message collisions increases as the traffic load is increased. Figure 32 shows that the service delay suddenly begins to increase when traffic load G crosses over 0.3. Comparison with the results in Figure 23 shows that ARCNET networks saturate at a lower value of G than Ethernet networks when the number of nodes increases. This is because token overhead increases with the number of nodes in ARCNET but collisions are a function of message generation rate instead of the number of nodes.

Table 20. Simulation Condition for *ReadProperty* Service Requests in Ethernet Networks

Message length(bytes) (request/reply)	Number of nodes	Message generation interval at the controller node(s)	Traffic load G
72/72	5	0.00115 to 0.00019	0.1000 to 0.6000
	10	0.00115 to 0.00029	0.1000 to 0.4000
	20	0.00115 to 0.00032	0.1000 to 0.3600
	40	0.00115 to 0.00033	0.1000 to 0.3500

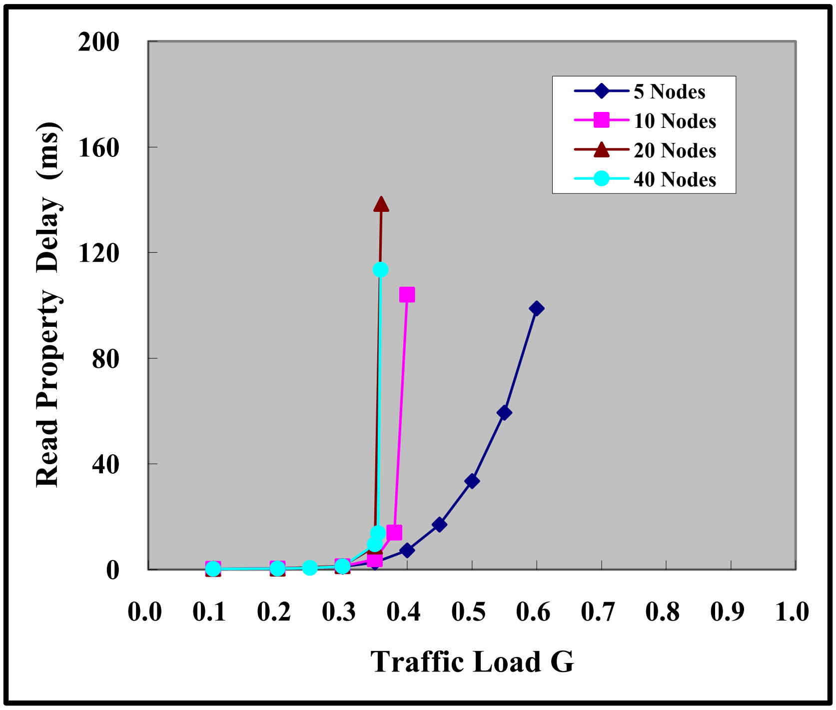


Figure 32. Average Service delay for *ReadProperty* Service Requests.

Table 21 shows the simulation conditions for *ReadPropertyMultiple* service requests with 10 Real values. The simulation results are shown in Figure 33. Compared to the *ReadProperty* service in Figure 32 the, *ReadPropertyMultiple* service increases the throughput of the network system just as is did or the other network technologies. Comparison with the results from ARCNET in Figure 24 shows that the average service delay for Ethernet networks is smaller at light traffic loads but that Ethernet is saturated at a lower value of *G*. This illustrates that CMSA/CD is faster at low traffic loads because it does not have token management overhead. As the traffic load increases collisions cause the network to reach saturation earlier than with token passing networks.

Table 21. Simulation Conditions for *ReadPropertyMultiple* Service Requests

Message length(bytes) (request/reply)	Number of nodes	Message generation interval at the controller node(s)	Traffic load *G*
125 / 194	5	0.00255 to 0.00034	0.1000 to 0.7500
	10	0.00255 to 0.00046	0.1000 to 0.5600
	20	0.00255 to 0.00051	0.1000 to 0.5000
	40	0.00255 to 0.00052	0.1000 to 0.4900

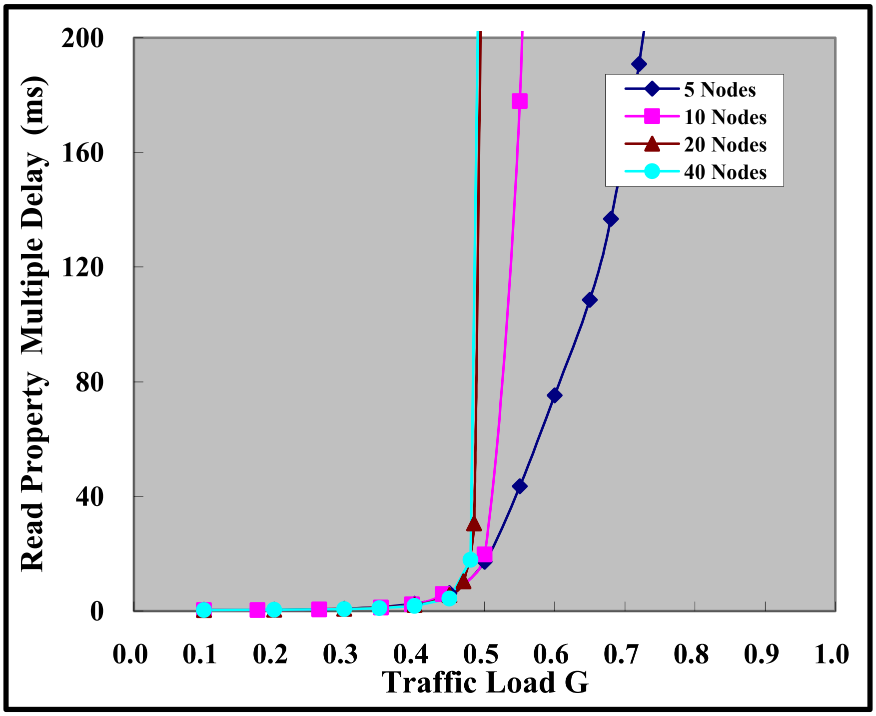


Figure 33. Average Service Delay for *ReadPropertyMultiple* Service Requests in Ethernet Networks.

Table 22 shows the simulation conditions for *UnconfirmedCOVNotification* service requests. Similar to the MS/TP and ARCNET networks, all the nodes in the medium transmit COV notification messages to a central controller node when a COV occurs. Figure 34 shows the simulation results. Service delay begins to increase when *G* is greater than 0.4.

Table 22. Simulation Condition for *UnconfirmedCOVNotification* Service Requests

Message length (bytes)	Number of nodes	Message generation interval at a node (s)	Traffic load *G*
72	5	0.00230 to 0.00033	0.1000 to 0.7000
	10	0.00518 to 0.00074	0.1000 to 0.7000
	20	0.01094 to 0.00156	0.1000 to 0.7000
	40	0.01670 to 0.00246	0.1000 to 0.6800

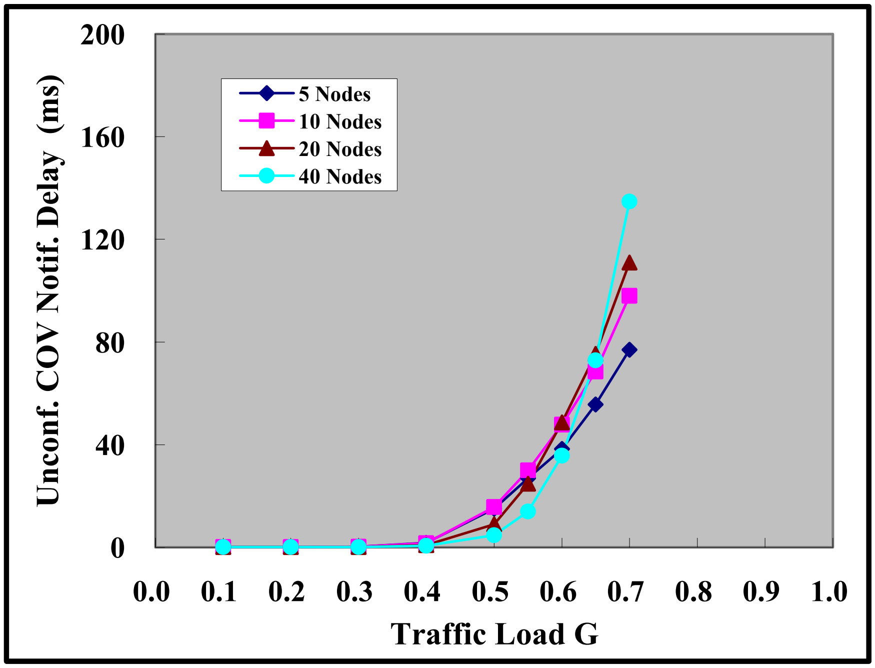


Figure 34. Average Service Delay for *UnconfirmedCOV-Notification* Service Requests in Ethernet Networks.

Table 23 shows the simulation conditions for the *ConfirmedCOVNotification* service and the simulation results are given in Figure 35. Like the *UnconfirmedCOVNotification* case, a central controller node receives all of the COV notifications, which are transmitted by the other nodes when a COV occurs. Upon receiving the COV notification message, the central controller node transmits a reply message.

Compared with the unconfirmed service in Figure 34, the service delay for the confirmed service is larger, and it increases more abruptly. Comparison with the results from ARCNET in Figure 26 shows, once again, that the network resource for *ConfirmedCOVNotification* service in Ethernet is saturated at higher values of G when the number of nodes increases.

Table 23. Simulation Conditions for *ConfirmedCOVNotification* Service Requests

Message length(bytes) request/reply	Number of nodes	Message generation interval at a node (s)	Traffic load G
72/72	5	0.00461 to 0.00071	0.1000 to 0.6500
	10	0.01037 to 0.00173	0.1000 to 0.6000
	20	0.02189 to 0.00398	0.1000 to 0.5500
	40	0.03341 to 0.00630	0.1000 to 0.5300

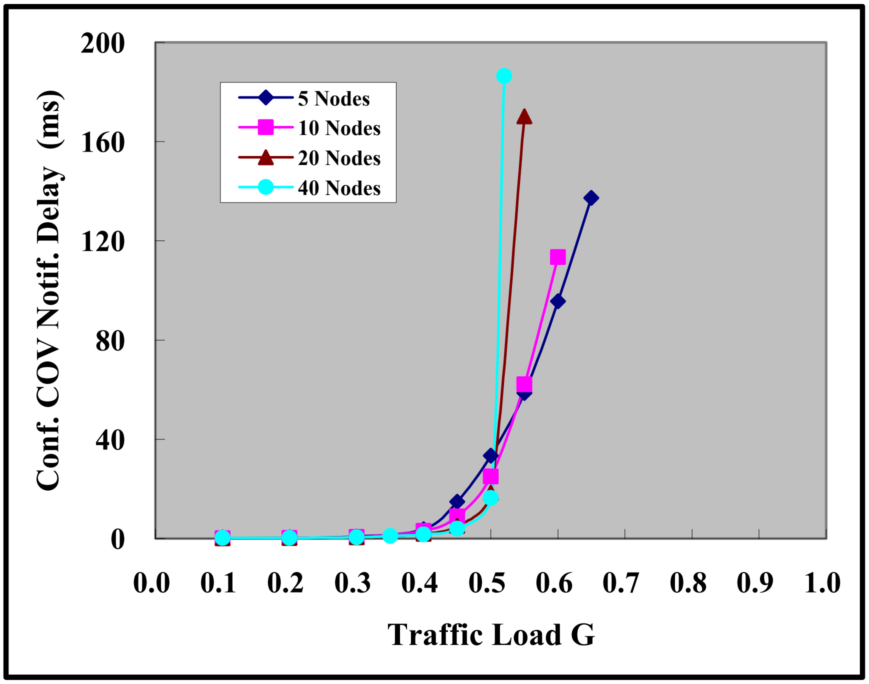


Figure 35. Average Service Delay for *ConfirmedCOVNotification* Service Requests in Ethernet Networks.

Most building automation and control system architectures use Ethernet as a backbone network. In this section, performance of the service delay for *AtomicWriteFile* service on Ethernet networks is investigated. *AtomicWriteFile* was chosen as a way to represent the impact of large message sizes.

Table 24 shows the simulation conditions. In this simulation, the message length for a file is assumed to be the maximum length of an Ethernet packet. Figure 36 shows the simulation result. Figure 36 shows that the service delay increases exponentially as the traffic load is increased. The *AtomicWriteFile* service delay is not significantly affected by a change in the number of nodes.

Table 24. Simulation Conditions for *AtomicWriteFile* Service Requests

Number of nodes	Message length (bytes) request/reply	Message generation interval at a node (s)	Traffic load *G*
5	1526 / 72	0.051136 to 0.006641	0.1000 to 0.7700
10	1526 / 72	0.115056 to 0.015341	0.1000 to 0.7500
20	1526 / 72	0.242896 to 0.032386	0.1000 to 0.7500
30	1526 / 72	0.370736 to 0.049432	0.1000 to 0.7500

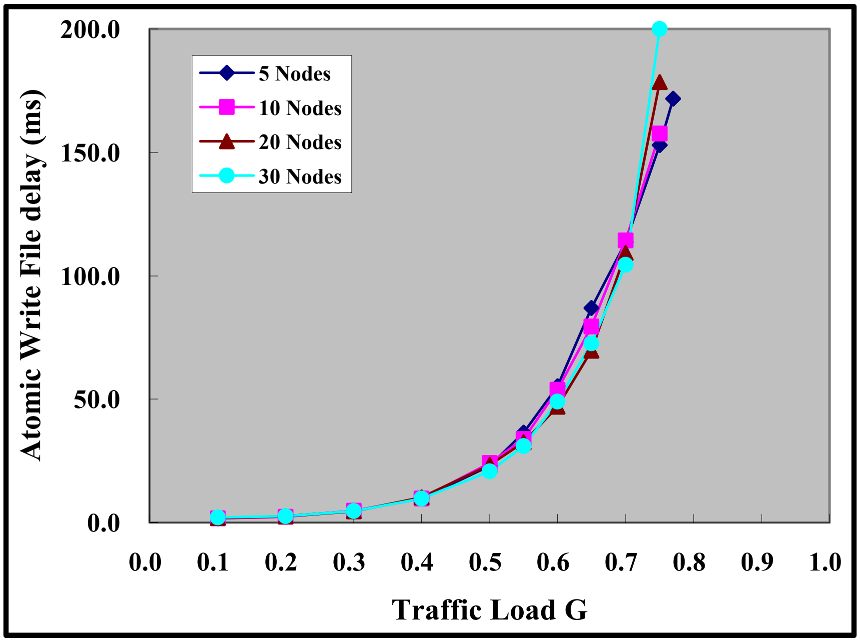


Figure 36. Average Service delay for *AtomicWriteFile Service Requests* in Ethernet Networks.

4.3.4 Effect of processing time on the service delay

In this section the *AtomicWriteFile* service is used to study the effect of processing delay on the performance of Ethernet service delay. The number of nodes in the medium is assumed to be 60. Fifty-nine nodes transmit *AtomicWriteFile* request messages to the remaining node that sends a reply message whenever it receives a request. The data rate of the Ethernet network is assumed to be 10 Mbps. The *AtomicWriteFile* packet length is assumed to be the maximum allowable length in the Ethernet. The message generation interval is assumed to have a Poisson distribution and determines the traffic load. The simulation conditions are summarized in Table 25.

Table 25. Simulation of Ethernet with Processing Time

Message Type	Message length (bytes) (request/reply)	Message generation interval (s)	Traffic load
AtomicWriteFile	1526 / 72	0.79398 to 0.12676	0.0949 to 0.5949

The following five cases of processing delay are considered; 0 ms (no processing time), 1 ms to 20 ms (fast processing time), 100 ms to 200 ms (moderate processing time), 200 ms to 300 ms (slow processing time), and 300 ms to 500 ms (very slow processing time). The processing time is also assumed to have a uniform distribution within the range.

Figure 37 shows the simulation results of average service delay with the changes of traffic load and processing time. Service delay is exponentially increased after the traffic load exceeds 0.5. Like the ARCNET case, processing time linearly contributes to the increment of service delay. Service delay in the Ethernet is increased as much as the processing time is increased. Each node transmits its messages based on the CSMA/CD mechanism. Unlike an MS/TP node, an Ethernet node that receives a confirmed service request does not occupy the network medium during the processing time.

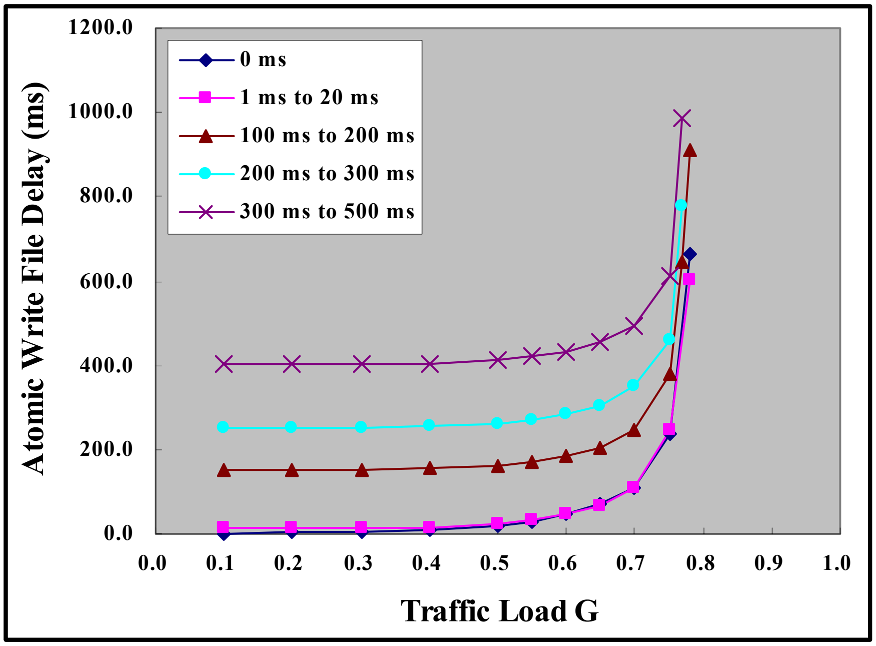


Figure 37. Average service delay of Ethernet with processing time

5 Conclusions

A building automation system cannot satisfy the requirement of real-time operation if the network-induced delay exceeds the application requirements. This study examined the delay characteristics of three popular BACnet LANs, MS/TP, ARCNET and Ethernet. Simulations were made using a selection of BACnet messages that represent confirmed and unconfirmed services, and traffic load that varies from low to high.

MS/TP provides simple and low cost means of communication. This study identifies some network parameters in the MS/TP protocol that influence the performance of BACnet application services. Because large values for T_{frame_gap} or T_{usage_delay} heavily degrade performance, it is recommended that the values of T_{frame_gap} and T_{usage_delay} should be as small as possible when implementing MS/TP devices. As the length of the message increases, the network utilization also increases because the effect of the protocol overhead from $T_{turnaround}$ is reduced. Network utilization is reduced as the number of master nodes in the MS/TP network is increased. This is because of the effect of the overhead from T_{usage_delay} . As $N_{max\_info\_frames}$ become smaller, the relative overhead for token passing becomes larger. Because $N_{max\_info\_frames}$ is a network configuration parameter, the network designer should select a sufficiently large value of $N_{max\_info\_frames}$ in order to reduce the network-induced service delay. The results of these simulations suggest that a value of $N_{max\_info\_frames} \approx 5$ would be appropriate.

In a single-master MS/TP network, it is recommended that peak traffic load be restricted so that $G < 0.8$. From the perspective of service delay, the performance of a single master system can be better than that of a multi-master system. However, a single-master system has more limited application functionality because a slave device cannot initiate messages. If master and slave nodes are combined in one network, the nodes that require the ability to initiate messages must be master nodes, but all the other nodes should be slaves.

Most of the MS/TP networks currently operated in real buildings are all-master systems. Using the *ReadPropertyMultiple* service to retrieve multiple data values instead of repeated use of the *ReadProperty* service significantly increases throughput performance but slightly increases the service delay. In all-master MS/TP networks, the difference in service delay between the *UnconfirmedCOVNotification* and *ConfirmedCOVNotification* was found to be negligible. The service delay of MS/TP networks increases as the processing time of the BACnet application service increases. In MS/TP networks, processing time and the value of $\mathbf{T}_{reply\_delay}$ significantly affect service delay.

Even though MS/TP networks are relatively slow, they are quite efficient for BACnet application services. This is because an immediate reply to confirmed services and the ability to transmit more than one message when holding the token are features well suited to the client/server communication nature of building automation systems.

ARCNET provides faster communication speeds than MS/TP. The transmission delay for ARCNET increases as the number of nodes increases and as the message length increases. The transmission delay is more severely affected by increasing the number of nodes than by increasing message length. This is a result of the impact of token management overhead.

For *ReadProperty* service requests, the network resource in ARCNET networks saturates at a much lower traffic load than in MS/TP. With ARCNET, using the ReadPropertyMultiple service rather than several repetitions of the ReadProperty service can significantly increase network utilization. The delay for *UnconfirmedCOVNotification* in ARCNET is less affected by the token overhead. The delay of *ConfirmedCOVNotification* service, in ARCNET, is significantly affected by the token overhead, and the performance is degraded compared to the *UnconfirmedCOV-Notification* service. In ARCNET networks, the service delay for confirmed services increases linearly as the processing time is increased, i.e., the service delay is increased as much as the processing time is increased. The delay for unconfirmed services in ARCNET, however, is not affected by the change of processing time.

ARCNET is significantly faster than MS/TP but, because the effect of token circulation overhead is greater for ARCNET, the network performance degrades at lower traffic levels than MS/TP. Both MS/TP and ARCNET are suitable for a LAN that requires real-time communication because they are operated on token-passing discipline.

On the other hand, Ethernet is suitable for backbone LAN of building automation system. Ethernet supports sufficiently high data transmission rate for building automation application.
Compared to the case of ARCNET, the transmission delay in Ethernet is less affected by the change of the number of nodes. As the message length increases, the network utilization in Ethernet is increased. Compared to the ARCNET networks the delay characteristics of Ethernet are more randomized with respect to the traffic change.

Compared to ARCNET, the network resource for *ReadProperty* service in Ethernet is saturated at higher traffic load especially when the number of nodes in the network is larger. *ReadPropertyMultiple* service in Ethernet increases the throughput of the network system just as it did or the other network technologies. The network resource for *UnconfirmedCOVNotification* and *ConfirmedCOVNotification* services in Ethernet begins to saturate as $G > 0.4$. The rate of the increase in service delay for *ConfirmedCOVNotification* is higher than that for *UnconfirmedCOVNotification*. Like ARCNET, processing time linearly contributes to the increment of service delay. Service delay in Ethernet is increased as much as the processing time is increased.

Although Ethernet is very efficient at low traffic load, protocol overhead caused by contention is increased as the traffic load is increased. At a heavy traffic load, it may not be able to guarantee the real-time requirements.

The simulation results obtained from this study can provide some guidelines for designing BACnet networks used in building automation systems. In particular, the results provide insight into how to optimize the performance of MS/TP networks, characteristics that can be used to help select the appropriate LAN, and operating constraints that must be met to remain within acceptable service delay limits.

References

[1] Newman, H. M., *Direct Digital Control of Building Systems: Theory and Practice*, John Wiley & Sons, Inc., New York, 1994.

[2] ANSI/ASHRAE Standard 135-2001, *BACnet: A Data Communication Protocol for Building Automation and Control Networks*, American Society of Heating, Refrigeration, and Air-Conditioning Engineers Inc. Atlanta, GA.

[3] ISO 7498, *Information processing systems – Open Systems Interconnection – Basic Reference Model.*

[4] Bushby, S.T., *BACnet^{TM}: a standard communication infrastructure for intelligent buildings,* Automation in Construction, Vol. 6 No. 5-6, 1997, p. 529-540.

[5] http://www.bacnet.org

[6] KS X 6909 *Building Automation and Control Network (BACnet),* Korean Standards Association, 1999.

[7] ISO DIS 16484-5 Building automation and control systems – Part 5 Data communication protocol

[8] EIA-485, *Standard for Electrical Characteristics of Generators and Receivers for use in Balanced Digital Multipoint Systems.*

[9] Casandras, C. G. and Lafortune, S., *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, 1999.

[10] Kelton, W. D., Sadowski, R. P. and Sadowski, D. A., *Simulation with ARENA*, McGraw Hill College Div, July 2001.

[11] ANSI/IEEE Standard 802.3, Local Area Networks – Carrier Sense Multiple Access With Collision Detection(CSMA/CD) Access Method and Physical Layer Specifications, IEEE, Piscataway, NJ , 1998.

[12] ANSI/ATA 878.1 standard, *Local Area Network: Token Bus (2.5 MBPS)*, ARCNET Trade Association, 1992.

[13] Buchanan, B., *Handbook of Data Communication and Networks*, pp. 497, Kluwer Academic Publishers, Boston, 1999