c.1

DESIGN AND IMPLEMENTATION OF A TOKEN BUS PROTOCOL FOR A POWER LINE LOCAL AREA NETWORK

By

HUA GU

B.A.Sc., South China Institute of Technology, 1982

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(DEPARTMENT OF COMPUTER SCIENCE)

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

April 1988

Abstract

This thesis presents the development and implementation of a token bus protocol for a Power Line Local Area Network (PLLAN) which utilizes intra-building power distribution circuit as the physical transmission medium. This medium provides a low cost means for data communications with a high degree of portability. Due to the characteristics of the power line and the prototype modem, the network would be easily saturated with data and would have a high collision probabilities. The IEEE 802.4 token bus standard is modified to fit the PLLAN and to bring its performance up. A comparative performance of the original protocol and the modified version shows that the latter provides an improvement in network throughput of up to 15 percent and a reduction in the network join-ring delay of up to 20 percent for a wide workload range. The performance figures of the modified version in a power line network of three SUN 3/50 workstations [1] transmitting at 9.6 kilo-bit per second is also presented and analyzed.

[1]Sun workstation is a trademark of Sun Microsystems.

Contents

List of Tables

List of Figures

Acknowledgement

I sincerely thank Dr. Son Vuong, my supervisor, for his invaluable guidance, commitment and supportiveness. I am also grateful to Dr. Samuel Chanson for acting as my thesis second reader. Thanks must also go to Helen See and Steve Ng for their timely input and their time spent on the proofreading of the thesis.

My deepest gratitude to three fellow colleagues, Audivox Ma, Frank Chiu and Morgan Chan for their help with the kernel programmings and the modem development.

Financial support from an NSERC Strategic Grant of Dr. Son Vuong and Dr.X Donaldson is graciously acknowledged.

Without the moral support from my lovely parents and motherland, I would never have overcome the obstacles that each seemed insurmountable then.

Chapter 1

Introduction

Over the recent years, computer Local Area Networks (LANs) have become very popular. Increasing number of computer users are benefiting from this technological advancement to share resources (such as printers, file servers and databases), distribute new softwares, operate diskless workstations, support distributed computations, send electronic mails and conduct computer conferences, amongst other activities. Within industries, LANs are also widely used to give support to auto-control systems.

Today, nearly all LANs require dedicated high speed cables as the underlying communication medium. The cost of cables and their installation is substantial, compared to the cost of computers themselves. This high cost, unfortunately, has prevented a large number of potential users from sharing this fruit of technological advancement and has hindered more potential computer applications from being developed.

For instance, the cost for laying a cable is about 10,000 dollars while a personal computer would sell for only about 1,000 dollars. Aside from the cost, these high speed cables would be under utilized in family and small business environments, since the network loads are usually low to moderate. Under these conditions, the computer users will argue that it is not worth the investment. Similar situations could also occur in industrial settings that employ simple auto-control systems.

In pace with further plummeting computer costs and expanding computer applications, the need to address low-cost LAN is getting stronger. A couple of approaches for low-cost LANs have been proposed in [24,2]. One of these approaches, proposed by the communication research group at UBC, is to employ the electric power distribution circuit as a communication medium [9,25]. This approach not only alleviates the cost problem but also provides a convenient access to the network. Whenever a user wants to use the network, he can simply plug his computer into a standard electric outlet.

This handy network system requires, in addition to computers, three components: a network software package located within each computer, power wires and a special modem which serves as an interface between the power line and the computer.

One of the characteristics of the power lines is that they can only support low-to-medium data rate. At this rate of transmission, the error rate is still perceptibly high. These inherent disadvantages of power lines are due to the fact that they were not originally designed for data transmission. The power line medium does not satisfy the requirements of reliability and efficiency which are assumed by the available medium access control protocols from their physical layers and communication media. The prototype power line modem [1], is currently only half duplex and has a long access delay. With the modem, a power line has a data rate of 9.6 kbps with an error rate of 10^{-4} to 10^{-8}. Further improvement of the modem is possible but would raise the cost substantially and our original goal, the "low-cost" network, will not be achieved.

The potential of power lines and the present limitations for data transmission motivate one to modify the medium access controls protocols such that they can accommodate a medium that is not reliable, nor efficient.

1.1 Thesis Motivation and Objective

In order to make the power line communication medium and the modem usable and reliable, a cooperative Medium Access Control (MAC) protocol is needed. The goal of this thesis is to design and implement a MAC protocol for the power line local area network (PLLAN).

Based on the study of the characteristics of power lines and the modem, a non-contention scheme is deemed to be most suitable. Hence the IEEE 802.4 token bus protocol standard was chosen as the model and was modified for the PLLAN. This modified IEEE 802.4 protocol is capable of cooperating with the modem within the power line environment and the network running this protocol gives a higher throughput than the one that runs the original protocol.

[1] The modems are produced by the Electrical Engineering Department of UBC

Both the original and the modified version of IEEE 802.4 standard were implemented in an experimental network of five SUN 2/50 Workstations. Comparative measurements obtained show that the modified version provides an improvement in network throughput of up to 15 percent and a reduction in network join-ring delay of up to 20 percent over the original protocol.

Most of the modifications proposed could also be applicable to other general local area networks, although the original intention was for applying it to the PLLAN.

1.2 Thesis Outline

The remaining portion of this thesis is outlined as below.

Chapter two studies the characteristics of the power lines and the modem.

Chapter three surveys and compares various medium access control protocols.

Chapter four presents the modifications made to the IEEE 802.4 token bus protocol. First, a brief review of the IEEE 802.4 standard is given. Then based on this review, certain relevant features are discussed and the modifications are explained in detail.

Chapter five serves as a support for the modifications presented in Chapter four. The modified and the original IEEE 802.4 protocols were implemented and compared in an experimental network. Measurements obtained are presented and analyzed.

Chapter six presents the implementation of the modified version on a real power line network environment. The performance of the modified protocol was measured and presented.

Chapter seven wraps up this thesis by giving some general concluding remarks and some suggestions on areas for further research.

Chapter 2

Characteristics of Power Line and Modem

The power line and the modem form the basis for the design of the MAC layer protocol and their behaviours affect the design strongly. The MAC layer protocol runs on top of and mediates the interworking of these physical media. A good understanding of their characteristics can guide the design in the right direction.

For further discussion, the reader can refer to Audivox Ma's thesis [15]. He presents the same concepts and principles based on a different set of parameters.

2.1 Power Lines

Power lines are not originally designed for the purpose of data communications. As such it is considered as a "hostile" environment for digital transmissions. Data can be transferred over it but with an inordinate amount of errors and low data rates. High noise level, strong signal attenuation and narrow bandwidth are the primary factors contributing to the hostile power line communication environment, as suggested by studies performed in the Electrical Engineering Department of University of British Columbia [3,6].

2.1.1 Noise

Noise in power lines is caused, in principle, by any inductive electrical appliances and equipments that could generate inordinate signals into the power line. Furthermore, it varies from time to time and location to location, depending on the number and type of inductive appliances and equipments, that are in use at any given moment. For instance, at night time, the power line has a lower noise level because less equipments are in operation. Equipments such as vacuum cleaner, electrical motors, xerox machines and light dimmers tend to garble the power line communication significantly while fluorescent lights generate less noise. Family and office environments are generally less hostile than factory environments [9].

One of the effects due to power line noise in data transmission is a higher bit error rate. An attempt to give an accurate figure for bit error rate is nearly impossible. It can range from a probability of 10^{-3} in a noisy industrial setting to a probability of 10^{-8} in a residential one. Also, error patterns vary with different environments. Both burst and random bit errors are usually observed in the same power line with burst errors being predominant in industrial areas.

A possible countermeasure approach against the high error rate is to use a software coding technique to recover bit-errors on the receiving end. In this technique, bits in each frame are calculated and modulated into a special code pattern before being sent. This code pattern is much longer than the original frame. On the receiving end, the frame has to be decoded to restore its original bit pattern. This approach is expensive in terms of both CPU time and network bandwidth and is not employed in my implementation due to a low noise level in my experimental environment. A hardware technique called ***bit-interleaving and forward error correction approach***, is proposed in [6] and will be utilized in a new prototype modem. This technique will combat burst errors to a significant degree.

2.1.2 Signal Attenuation

Attenuation can be attributed to a number of factors occurring simultaneously or independently on the power line. Firstly, it can be caused by capacity or inductance circuits which might be used for protecting expensive equipments from power surges. Secondly, when background noise level is high enough to cover the signal level, it has the same effect as signal attenuation. Thirdly, certain types of signals, such as across-phase signals, suffer from much higher attenuation than in-phase signals. Fourthly, the modems connected to the power line consume some of the signal energy and hence cause attenuation.

2.1.3 Bandwidth

Since the power line is designed for alternating currents of 60 Hz, it is not suited for propagating high frequency signals. The quality of a power line local area network is such that high data rates can be achieved but at the expense of short communication distances.

A possible power line local area network configuration for an office environment would be one with a 3.0 V_{rms} (volt root mean square) transmitted signal at a data rate of 9600 kilo-bit per second (kbps) and an error rate of 10^{-6}. It would have to be in-phase and can stretch a distance of 1000 meters, covering a maximum of 16 communication modems. Unlike Ethernet, it does not have any restriction on the minimum distance separating two station nodes.

2.1.4 Topology

The topologies of power lines could be *bus* or *tree*. In this paper, the power lines are considered as a bus topology.

2.2 Modem

Since the modem interacts directly with the MAC layer protocol it has a stronger effect on the design of the MAC layer protocol than the power line.

Prototype modems used in this experiment are built by the Department of Electrical

Engineering at UBC. A frequency shift key (FSK) modulation technique is used to superimpose data onto a 120 kHz radio frequency (RF) carrier to avoid interfering with the 60 Hz power signal. A maximum data rate of 9.6 kilo-bit per second (kbps) is achievable. To reduce the cost and the complexity of the modem, only half-duplex communication is provided.

2.2.1 Transmission and Reception

Each modem works in two modes: transmitting and receiving, and shifts between the two. To avoid having two or more modems transmit simultaneously, each is equipped with an RF carrier detection logic that ensures that it "listens before talking" (*carrier sense*).

Transmission

When a modem connects to the network, it is initialized to the receiving mode and starts listening to the channel. If the channel is quiet and the modem has no data in its buffer, the carrier detection logic automatically turns itself off and the modem stays in the reception mode. Otherwise, the modem switches to the transmission mode if the channel is quiet and there are data in the buffer.

The time needed to detect the presence or absence of a carrier, so-called *collision detect time*, is about 20 milliseconds depending on the noise level and the degree of attenuation. This implies that during this interval, the modems are not capable of detecting whether the channel is quiet or not and a potential collision is allowed to occur.

In this mode, a special code modulated on the RF is broadcast to allow all receiving modems to synchronize with the transmitting modem. This special code also serves as an indication that channel is now occupied. It takes about 10 to 14 milliseconds from the detection of an idle channel to the first user data transmission.

To ensure that all transmitted signals are received completely by the receiving modems, at the end of data transmission the transmitting modem will continue to transmit a RF carrier for about 10 to 20 milliseconds. This is called the *end-of-message extend time.* During this time, the receiving modems are forbidden to transmit.

Reception

If the channel is quiet, the carrier detection logic in a modem is off. Whenever the modem hears a signal on the channel, the carrier detection logic is reactivated and the modem starts to synchronize with the signal. As long as the signal is present, the logic remains active and the modem stays in the receiving mode.

Once the signal disappears, it is imperative that the modem should shut off its carrier detection logic. Failure to do so will cause it to miss a new transmission. The signal for the new transmission will be misunderstood as the continuation of the old transmission and misinterpreted based on the old sychronization.

It takes about 10 milliseconds for the modem to shut off its carrier detection logic. This is called the *carrier detect linger effect.* This implies that if the elapsed time between two successive transmissions is less than the carrier detect linger time, the receiving modems will fail to resynchronize with the new transmission and mistake the new transmission as part of the old one.

2.2.2 Interface Between Modem and Computer

A modem interfaces with its host via an RS-232 link. Both synchronous and asynchronous data transmission are possible between modem and host. The modem provides clock signals for synchronizing both reception and transmission of data. Out of the 25-pins RS-232 standard, 6 pins are used when asynchronous data is exchanged. They are :

AB (GND) ground reference signal.

BA (TxD) data signal directed to host.

BB (RxD) data signal directed to modem.

CA ($\overline{RTS}$) request to send signal directed to modem.

CB ($\overline{CTS}$) clear to send signal directed to host as a response to an RTS.

CF ($\overline{CD}$) indication of RF carrier signal detected.

Two additional signals are needed when synchronous data are exchanged. They are :

DB ($Txclk$) clock signal directed to modem for synchronizing data on BA.

DD (*Rxclk*) clock signal directed to modem for synchronizing data on BB.

Since the modem is half-duplex, the mode of operation (transmitting or receiving) is controlled by the host via the RTS signal. This signal together with the $\overline{CTS}$ and $\overline{CD}$ signals provided by the modem allow the host to determine the status of the modem and the power line channel. The interaction between these various signals can best be described using the time diagram on page 43-45 of [22].

2.3 Summary

Acceptable error rates of 10^{-3} to 10^{-6} are observed and a fairly decent data rate of 9600 bits per second is achievable, by using the modem and the power lines.

The special properties of the power line modem, carrier sense protocol, synchronization between modems, end-of message extend period and carrier detect linger effect require special attention when selecting and implementing the MAC layer protocol.

Chapter 3

Selection of Suitable MAC Protocol

This chapter focuses on finding a suitable medium access control protocol for the power line local area network. Several factors influence the choice of such a protocol, but two are of the primary importance: the topology of the communication medium and the performance of the network.

Based on these two criteria, various MAC protocols are surveyed and compared. It is then concluded that a non-contentional token bus scheme is the most suitable MAC protocol for the power line local area network.

3.1 Overview

Figure 3.1 shows a classification of the different medium access control protocols considered and how they relate to one another. The survey on protocols will follow the structure of this tree.

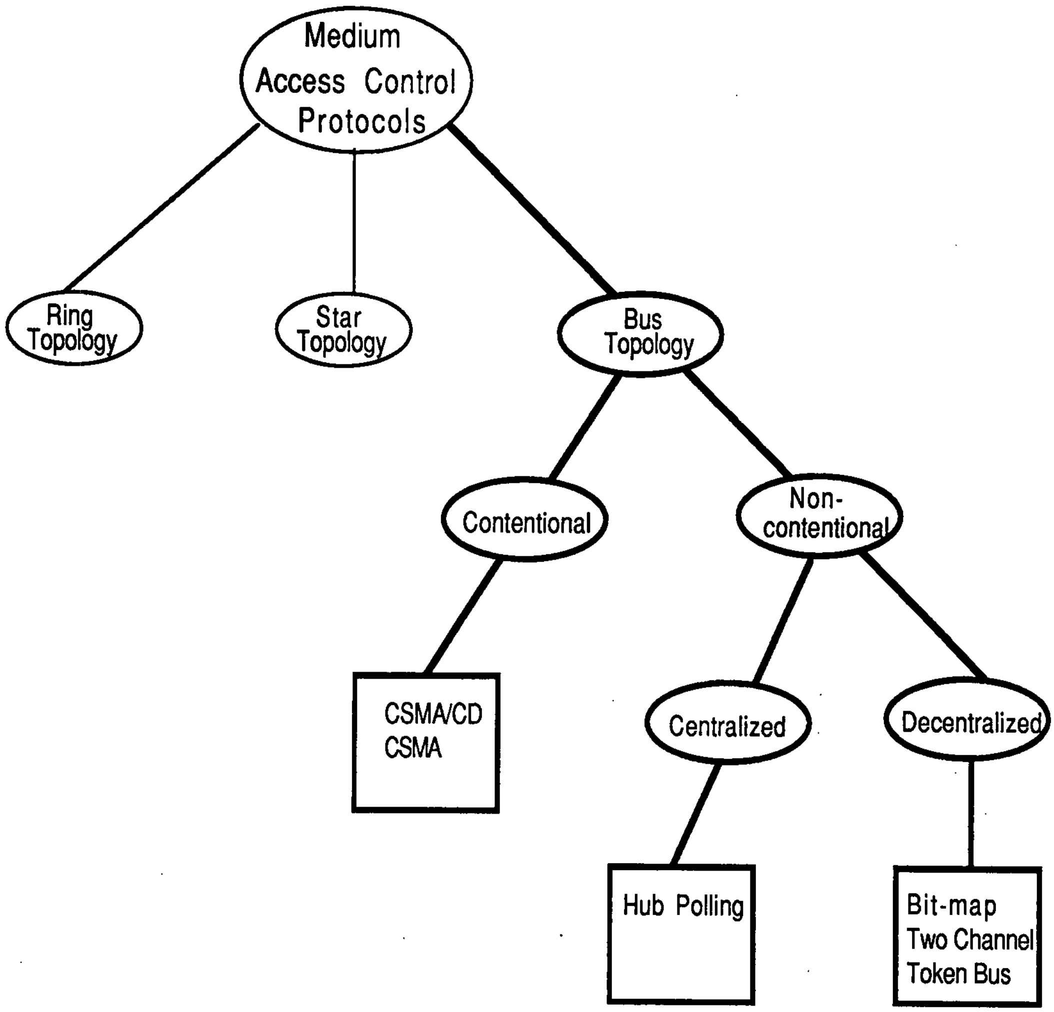

Figure 3.1: Classification of Medium Access Control Protocols

3.1.1 Topology Limitation

Various protocols can be categorized based on communication medium topologies that they run on, such as bus, tree, ring and star. Our survey will narrow down on those which are applicable on bus topology, since it is the only applicable configuration to power lines.

3.1.2 Contention and Non-contention Schemes

Within bus topology, the medium access control schemes can be classified as *contention* or *non-contention.* Contention means that simultaneous accesses to the physical channel are resolved by a probabilistic arbitration scheme. In a non-contention scheme, simultaneous accesses are avoided altogether by coordinating the access right to the channel. Generally speaking, contention schemes are simple and efficient in terms of bandwidth usage. Non-contention schemes tend to be more complicated and involve more overhead for coordinating access rights among various stations. However, under high network activity which implies more contention for the medium, non-contention schemes will out-perform contention schemes.

CSMA/CD and CSMA are the most common *contention* schemes. Their performance relies heavily on the collision probability and this in turns relies on their *contention period.* Contention period is the interval during which two stations may try to access the channel simultaneously without being able to detect each other.

A *non-contention* scheme basically does not have a contention period and has better performance under multi-access conditions than a *contention* scheme. Non-contention protocols differ from each other in the way in which an access right is coordinated among network participants. System based on non-contention protocols can be further divided into *centralized* and *decentralized* systems.

3.1.3 Centralized and Decentralized Systems

A *centralized* system has a primary station that maintains and manages all network activities such as granting channel access rights and removing dead station nodes. It is more efficient in terms of channel utilization and has simpler error recovery procedures. Its main disadvantage is its vulnerability to failures. The failure of one site, the primary site, can cause the failure of the entire network. A minor disadvantage is that during network installation, users have to decide which station plays the role of the primary station. *Hub polling* protocol is an example of this centralized system.

In contrast, *decentralized* systems are more reliable in the sense that the entire network will not fail due to the malfunction of a single station. Such systems need more complex algorithms and involve more housekeeping overhead, since there is no "global view" of the entire network kept by a central station. Each station has to keep track of the network status, in order to maintain network order and to recover from error situations. The *bit-map* protocol, *two-path* protocol and *token bus* protocol are examples of such a system.

3.2 Protocols Considered

3.2.1 CSMA

CSMA (Carrier Sense Multiple Access) is a common MAC scheme for bus topology media. It is also referred to as a *listen before talk* protocol. Besides, it can be used with a half-duplex modem.

Basic Concepts and Principles

A station wishing to transmit first listens to the medium to determine if another transmission is in progress. If the medium is idle, the station transmits. Otherwise, it backs off for some period of time based on an algorithm and tries again.

A collision can occur when more than one station begins transmitting within a short time called *contention period.* The contention period is the time interval during which simultaneous transmissions can occur. For a CSMA scheme, usually, this period is the propagation time between two furthest separated stations. If a station begins to transmit, and there is no collision during the time it takes for the leading edge of the packet to propagate to the farthest station, then the station has seized the channel and the remainder of the packet will be transmitted without collision.

After transmitting, a station waits for a certain amount of time for an acknowledgement which takes into account the maximum round-trip propagation delay and the fact that the acknowledging station must also contend for the channel in order to respond. This scheme does not have the capacity for collision detection and thus results in timeout events and retransmissions. The overhead involved is fairly significant.

Whenever two packets collide, the medium remains unusable for the duration of transmission of both the damaged packets. For packets that are long, the amount of wasted capacity can be considerable.

Performance

Being a contention scheme, the performance of the CSMA scheme is affected strongly by the collision probabilities and the probabilities in turn depend on the contention period. For 1km cable, the contention period is 5 microseconds. While for the power line local area network, the contention period is the ***contention detect time*** of the modem which is 4 to 10 milliseconds.

To determine the suitability of the CSMA scheme for the power line local area network, the collision probability P (with the contention period of 4 milliseconds) and the collision probability P' (with the contention period of 5 microseconds) are briefly examined. A ***"pure"*** CSMA [1] scheme is employed for the examination.

Assuming that the rates of accessing the physical medium follows the Poisson dis-

[1] "Pure CSMA scheme" means those that do not use any collision resolution algorithms such as p-persistent.

tribution, the probability $P_n(t)$ of having exactly n accesses during a contention period of t seconds is given by

$$P_n(t) = \frac{(t\lambda)^n}{n!} e^{-t\lambda}$$

where λ is the rate of the medium access. A collision happens when two or more accesses occur during the interval t. The probability $P_{n>1}(t)$ is given by

$$\begin{aligned} P_{n>1}(t) &= 1 - P_0(t) - P_1(t) \\ &= 1 - (1 + t\lambda)e^{-t\lambda} \end{aligned}$$

Table 2.1 gives the values of $P_{n>1}(t)$ and $P'_{n>1}(t)$ with different λ values.

It can be seen that P is dramatically different from P'. For instance, with $\lambda = 100$, P' is 0.0, but P reaches 0.9084. With $\lambda = 1000$, P has reached 1, while P' is just 0.000012. Even for small λ values such as 20, the collision probability for P is relatively high. About 20 percent of the frames transmitted may be garbled due to collisions. Furthermore, these collisions cannot be detected and thus lead to timeout events. The examination strongly suggests that this scheme will perform poorly in the power line local area network environment.

For reducing collision probabilities, networks with light workloads are preferred. However, this does not seem to be the case for the power line local area network. The limited data rates provided by the power line medium would lead to easy saturation of the channel. The latter can be illustrated by a network that is modelled using a M/M/1

λ (access/sec)	$P_{n>1}(t)$	$P'_{n>1}(t)$
10	0.0615	0.0
20	0.1912	0.0
30	0.3373	0.0
40	0.475	0.0
50	0.594	0.0
60	0.6916	0.0
70	0.7689	0.0
80	0.8288	0.0
90	0.8743	0.0
100	0.9084	0.0
1000	1.0	0.000012

Table 3.1: Relationship between λ value and collision prob.

queuing system. Let us assume that transmission requests at a station are generated at λ frames/second and the service rate is μ frames/second. For a data rate of 14.4 kbps and a frame size of 128 bytes, the value of μ would be 14.0625 frames/second. The mean number of transmission requests in the system is given by

$$N = \frac{\rho}{1-\rho}$$

where $\rho = \frac{\lambda}{\mu}$. To achieve a condition that the value of N will not grow unbounded, the condition $\lambda < \mu$ must be satisfied. With a small value of μ, the network is saturated rather easily. High network utilization is a direct consequence of this easy saturation property. Study [12] shows that the performance of a CSMA/CD network degrades considerably if network utilization is higher than 40%.

Moreover, the high collision probabilities would result in more retransmissions, hence lead to the heavy workload conditions. In turn, the heavy workload conditions will feedback to increase the collision probabilities.

Summary

In conclusion, the CSMA scheme is inappropriate for power line communication because

- it has a fairly long contention period of 4 to 10 milliseconds,
- the power line local area network is easily saturated due to its low-to-medium data rate,
- these two factors are recurring and effecting mutually.

In a sense, these are also common problems for general contention schemes.

3.2.2 CSMA/CD

The most common contention scheme for bus topology media is the CSMA/CD (Carrier Sense Multiple Access with Collision Detection), also referred to as the *listen while talk* protocol. The original baseband version of this scheme was developed and patented by Xerox [17] as part of its Ethernet local area network [16]. The original broadband version was developed and patented by MITRE [11] as part of its MITREnet local area network [10].

The CSMA/CD scheme is a refinement of the CSMA scheme. It attempts to overcome one glaring inefficiency of the CSMA scheme: each time a station transmits, it first listens to the medium. If a collision is detected during the transmission, the station immediately ceases the transmission and transmits a brief jamming signal to ensure that all stations know about the occurrence of the collision. After that, the station waits for a random amount of time based on an algorithm and attempts to transmit again.

Since the power line local area network uses a half-duplex modem, it cannot adopt this scheme.

3.2.3 Bit-map

Bit-map protocols are those which control the access right among stations by means of bit-slots. Protocols proposed by Scholl [21] and Chlamtac [7] are variations of bit-map protocols. Medium access rights are granted to each station in a round robin fashion. A station can indicate its intention of transmission by inserting a set bit into the bit slot when it comes to its turn to access the channel. Data can then follow. A station with no data can just let its bit slot go idle, so the next station will gain the medium access right after an 1-bit delay. This kind of protocol requires an establishment of tight synchronization among stations. A single out-of-sync station will misbehave and ruin the harmony of the network.

These non-contention protocols are not suitable for the power line local area network for the two reasons. First, synchronization between modems is by itself a problem under a noisy environment. Second, every station has to know not only the number of stations in the network but also their positions in the network.

3.2.4 Two Channel

Sterling [23] proposed a scheme to avoid the contention by separating the data transfer function from the medium access control function by using two channels. The channel used for medium access control is called the $C - path$ while the channel used for data exchange is called the $D - path$. Error recovery relies on the ability to detect collisions while transmitting. The scheme is simple and eliminates the need for a station to wait vacuously for a turn to transmit.

This protocol is again not usable for the power line local area network for the following two reasons. First, a separate C- and D- path cannot be provided by the modem. Secondly, the lack of a full-duplex communication channel would make the proposed error recovery impossible to implement.

3.2.5 Hub Polling

The network using *hub polling* scheme consists of one primary station and any number of secondary stations. The scheme puts all stations into a *logical ring* and access rights are regulated by circulating a predefined bit pattern called a *poll*. The primary station keeps track of the network global information in order to maintain and manage the network. Its tasks include inviting new stations to join the ring, removing the idle stations, restoring the "broken" logical ring and so on.

The main advantage of the hub polling scheme comes from its centralized property. It is simple and has better global control over the entire network. Hence, it could make better use of the channel bandwidth, provide an easier error recovery and provide good performance under multi-access conditions. Also, it is possible to determine the maximum time for a station to gain its turn for transmission.

The main disadvantage of the scheme also comes from its centralized property. It is vulnerable at the poll master's site because the operation of the entire network depends on the robustness of the poll master. When the network is being installed, the users have to explicitly pick a station to be the primary.

The power line local area network is expected to target a family or small business environments where users could be fairly unfamiliar with protocols and maintenance of such. Hub-polling requires user intervention now and then to keep the network going and hence is not acceptable as a power line local area MAC protocol.

3.2.6 Token Bus

The token bus scheme was chosen for the power line local area network for the following reasons.

- It is a non-contention scheme. Thus it can basically avoid the problems caused by having a long contention period of 4 to 10 milliseconds.
- It has tolerable performance under heavy network load conditions.
- It can support time-critical applications.
- It does not suffer from some of the disadvantages that centralized system have. The malfunction of an individual station will not cause the collapse of an entire network. Users do not need to pinpoint a primary station to act as the central node.

There is a number of disadvantages in using the token bus scheme and these stem are mainly from the fact that it is decentralized. No "global view" of the entire network is kept by a central station. Each station is responsible for maintaining their own view of the network status. Some non-data traffics are generated on the network for maintaining the virtual ring. For these reasons, the protocol is more complex than most others.

Naturally, improving the efficiency of the protocol becomes a motivation of this thesis.

3.3 CSMA/CD and Token Bus Schemes

This section focuses on two protocol schemes that have been widely used on bus topology media. Hopefully, it can also serves as a further support to the protocol selection made in the previous section. This section compares and contrasts the two schemes in finer detail to determine how they behave under various network conditions, types of services required and workloads on the network. Types of services include *time critical* and *non-time critical* services. Network workloads are informally classified as being “light” or “heavy”.

The CSMA/CD scheme has found a wide acceptance as a protocol for local area networks because of its simplicity, efficiency and relatively good performance. In fact, it often runs on a high speed coaxial cable which provides a very high bandwidth. It performs fairly well under light-to-medium network loads. What is undesirable about it is its non-deterministic property and its poor performance under high network loads. The performance degrades exponentially as the network load goes up. After exceeding the saturation point, the network performance becomes totally unacceptable. On the occasional burst of a heavy load, the network may thrash for a while before being normal again.

The former precludes it from supporting time critical services. The latter makes it unsuitable for a network which has an expected heavy workload.

As computers and computer services flourish, more new network applications may require time-critical services and the network are likely to carry heavier loads. For example, the concept of using the same network to support multiple services such as voice, video, computer data, etc., has become more attractive than ever before. (In the public domain, standards are being drawn for ISDN.) Video transmissions require a lot of bandwidth and time-critical delivery service in order to avoid the lost of video packets which may lead to degradation in video quality. In an automated manufacturing environment, data packets for machine control purposes must be delivered in time to avoid undesirable or sometimes disastrous effects. Also, in a distributed computing environment, the performance of the system may very much rely on tight synchronization of processes which communicate with each other through a local area network. Such strict delay constraints and heavy workload conditions required by these applications cannot be adequately met by the CSMA/CD scheme which is originally designed for conventional computer communications.

The channel access *efficiency* of the CSMA/CD scheme is a tradeoff against the *inefficiency* of its channel utilization. In order to maintain a reasonable performance, utilization of the channel has to be kept at 30 percent or lower [23]. Otherwise throughput would drop down due to excessive collisions. In a sense, the other 70 percent of the network bandwidth could be considered as an overhead for this scheme. At least in theory, such an overhead is unreasonable and fatal.

On the contrary, the IEEE 802.4 token bus scheme accommodates both synchronous (time-critical) traffics and heavy workload conditions. It enables the token bus scheme to meet the requirements demanded by the new network applications, while its complexity and inefficiency can be offset by the more powerful hardware.

Could it be expected that the token bus scheme would play a more important role than the CSMA/CD scheme in the future of computer communication industry? It is hard to say.

Chapter 4

Modifications to IEEE 802.4 Protocol

Due to the low data rate of power line medium, the long access delay of the power line modem and the potential inefficiency of the token bus scheme, it is desirable to improve the network performance by modifying the IEEE 802.4 token bus protocol.

Section 1 reviews the concepts and principles of the IEEE 802.4 protocol. Section 2 suggests a revised frame format which is shorter than the one used in the original protocol. Section 3 presents an *adaptive* scheduling for soliciting new stations to join the virtual ring which can reduce the number of redundant soliciting actions and fruitless token passings. Section 4 describes a so-called *skip* operation which can further reduce the overhead caused by idle stations. Section 5 proposes a slight change to *leaving-ring* procedure such that a potential high overhead for rebuilding the virtual ring can be avoided.

Although our original motivation for modifying the protocol was for the power line local area network, I believe that the modified IEEE 802.4 protocol could be also applicable to other general local area networks.

For simplicity, the modified protocol does not implement the priority classes specified in the original IEEE 802.4 protocol.

4.1 Review of the IEEE 802.4 Protocol

4.1.1 Principle of the Standard

Token bus schemes adopt the concept of *ring* from *token ring* schemes and apply it to local area networks which employ a bus topological communication medium. With such a scheme, a special control frame known as the *token* regulates the right to access the bus among all the stations. The station receiving the token is granted control of the medium for a period of time. When the station has finished its transmission or its token holding time has expired, it passes the token to another station on the bus. The sequence in which the stations receive the token forms a *virtual ring* [1]. The physical order of the stations on the bus is irrelevant and independent of their logical order in the virtual ring.

A well-known disadvantage of both the token ring and the token bus systems is the overhead involved. Under lightly loaded conditions, a station may have to wait through many fruitless token passes for a turn and the network may have to waste a lot

[1] In this paper, the terms of virtual ring and ring are used interchangeably.

of bandwidth in passing the token around before user data are transmitted. The IEEE 802.4 protocol describes the concept of *dynamic* virtual ring to reduce the overhead. In here, many stations are connected to the bus and at any moment only those which currently have or will soon have data to send are included in the virtual ring. Thus, the size of the virtual ring is self-adjusting, depending on the number of stations that want to send messages at any moment. The overhead caused by fruitless token passings can therefore be reduced.

An important feature and advantage of the IEEE 802.4 token bus protocol is that it provides services for priority classes.

4.1.2 Primary Operations in the Protocol

There are altogether eighteen states(including substates) and sixty transactions that are considered in the IEEE 802.4 protocol. Depending on the situation at hand, the protocol requires a station to perform appropriate operations. Generally, some operations are executed more frequently than others. The operations performed by a station can roughly be grouped into the following categories:

1. Transmit data frames on a request from the Logical Link Control(LLC) Layer.

2. Pass the token to the successor.

3. Join or leave the virtual ring.

4. Solicit new stations which are currently out of the virtual ring to join the virtual ring.
5. Initialize or reset the virtual ring.
6. Restore the lost token.
7. Eliminate extra tokens.
8. Report a faulty transmitter or a faulty receiver.
9. Relink the virtual ring after it is broken.
10. Remove a dead station.

To complete a transmission at a user request, a station has to go through the following steps: it has to join the ring, get the token, send messages out, pass the token, and finally leave the ring. In the meantime, another station in the ring has to perform a procedure called *soliciting new station* to grant an opportunity for the former to join. A station can only join the ring passively after it has heard an invitation.

On analysis, operations 1 to 4 are executed more frequently than others. The fifth operation is executed only if the network has just started or is subject to reinitialization. The other operations are for error recovery and are seldom executed as long as the network is running properly. Hence the efforts for modifications are concentrated on the first four operations to achieve greater improvements.

4.2 Reduction of the Frame Format Size

The power line local area network is aimed to be used in a family or small business environments. In such an environment, the total number of stations in the network will not be large. Coupled with the fact that the modem cannot handle too many stations, one byte address space should be sufficient.

The frame format used in IEEE 802.4 protocol is :

SD	FC	DA	SA	DATA	FCS	ED

- SD (start delimiter) of 1 octet.
- DA (destination address) of 2 or 6 octets.
- SA (source address) of 2 or 6 octets.
- FC (frame control) of 1 octet.
- DATA (only used in *set_successor* frame) of 2 or 6 octets.
- FCS (frame check sequence) of 4 octets.
- ED (end delimiter) of 1 octet.

Such a format size with 10 to 24 octets can be reduced to 7 or 8 octets as follows.

FLG	DA	SA	FC	DATA	FCS	FLG

- FLG (frame flag) of 1 octet.

- FC of 1 octet.
- DA of 1 octet.
- SA of 1 octet.
- FCS of 2 octets.
- DATA (only used in *set_successor* frame) of 1 octet.
- FLG (frame flag) of 1 octet.

By using this shorter frame format, the probability of frames garbled by noise in a power line channel is reduced. The network traffic caused by the MAC control frames can be cut down by about 30 percent. These two consequences are significant, considering the low data rates and high error rates of the power line media.

The DA field is moved right up to the front immediately after the FLG because in the current implementation the Ethernet hardware chip takes the first field of an incoming frame as the DA field. Also, the Ethernet hardware chip automatically generates the FLG and attaches it to a frame when the frame is sent out.

4.3 Adaptive Soliciting Strategy

Subsection 1 explains how and why actions to solicit new stations to join the virtual ring is time-consuming and expensive in terms of network bandwidth. Subsection 2 studies how and why the number of redundant soliciting actions and fruitless token passings are allowed to take place in the original protocol. Subsection 3 presents the new adaptive soliciting strategy.

4.3.1 Overhead Involved in Soliciting Actions

To solicit new stations, the station holding the token issues a *solicit_successor* frame. It could hear one of the following responses and has corresponding responses for it.

- No station wants to join the ring. This situation cannot be detected until a timer at the soliciting station expires. The station then passes the token to its successor. The value of the timer is usually longer than the time it takes to hear a response from stations that want to join.

- Only one station joins the ring. In this case, at least two additional channel accesses must be performed by two stations to allow exchanges of two frames namely, *set_successor* and *token.*

  A successful soliciting procedure is described as below. There are three stations "A", "B" and "C" . A and B are currently in the ring, and A is holding the token. C is the joining station. First, A broadcasts a *solicit_successor* frame after its data transmission. Upon receiving the frame, C accesses the channel and responds to A with a *set_successor* frame. When A hears the response, it accesses the channel and passes the token to C. C then holds the token and accesses the channel to start its transmission.

- A contention might have occurred when more than one station wants to join at the same time. A contention resolution procedure is called for in this case. The soliciting station issues a *resolve_contention* frame and the contending stations will retry after a certain delay. This delay depends on the value of their addresses. This procedure may be repeated a number of times [2]. Whoever among the contending stations responds to the soliciting station first (or last) without collision, the soliciting station will pass the token to that station and the contention is thus resolved. In the case where none of the contending stations is successful, the solicitation will be performed again on the next token round.

The scheduling strategy of soliciting actions can, therefore, impose a strong influence on the amount of overhead involved in the protocol. An efficient scheduling strategy is always desirable.

4.3.2 Original Soliciting Strategy

The original soliciting strategy employs a timer (ring_maintenance timer) and a counter (inter_solicit_count) to schedule the soliciting actions in inviting new stations to join the ring.

[2]The number of repeats is network specific.

- *inter_solicit_count:* This counter kept at each station prevents all stations from soliciting at the same token rotation round. The idea is to have a station solicit occasionally only and have a small number of stations do solicitation at each token rotation round. The work of solicitation is then spread out among all the stations and within several rounds. After a station finishes its data transmission, it checks the value of its *inter_solicit_count.* If the value is zero, a *solicit_successor* frame is sent out; otherwise the count is decremented by 1 and the token is passed to the next station. To avoid all stations having the same *inter_solicit_count* values, the initial values are chosen randomly in each station within the range $(0, n)$ [3], and is reinitialized randomly each time it reaches zero.

- *ring_maintenance timer:* This timer kept in each station is used to implement the maximum token rotation time denoted as Token Rotation Time (TRT). The TRT ensures that every station can have an access to the bus within a finite amount of time. The value of TRT is predefined by a network manager and is loaded into the *ring_maintenance* timer every time a station finishes its transmission. Whenever a station receives a token and the timer has expired, its soliciting action is deferred until the next token rotation round. The timer is reset at the time the station releases the token.

[3] The upper bound n could be network dependent and is set to 10 in our implementation.

Hence, before passing the token, a station does solicitation only if the following boolean expression is true:

(inter_solicit_count = 0) and NOT(ring_maintenance timer expires) (3.1)

The scheduling strategy based on the *inter_solicit_count*, however, has undesirable effects:

- Redundant soliciting actions are possible. In the case where all stations have joined the virtual ring, any attempts to perform soliciting actions would be meaningless and wasteful. This may happen in a station if the station's *ring_maintenance* timer has not expired (due to some stations not using up their Token Holding Time (THT)) and its *inter_solicit_count* reaches zero.

- Fruitless token passings may occur. In the case where stations in the ring are temporarily idle or have little to send, and there are other active stations waiting to join the ring (which are able to contribute to network throughput), the protocol should be adaptive enough such that more soliciting actions are performed instead of fruitless token passings. However, the stations in the ring refrain from doing soliciting actions as long as their *inter_solicit_counts* do not reach zero.

- The parameter n, the upper bound of the inter_solicit_count, is hard to define. A large n might lead to rare executions of soliciting actions while a small n might result in too frequent soliciting actions.

4.3.3 New Strategy

The new strategy proposed does not use the *inter_solicit_count*. It schedules soliciting actions adaptively, depending on changing workload of the ring.

Before presenting the strategy, the definition of *busy* station and *free* station, as well as *busy* ring and *free* ring are in order. A station is *free* if it can send off all of its accumulated frames within one THT. Otherwise, it is *busy*. A station views the ring to be *busy* if the station's *ring_maintenance timer* expires after the token has completed a round trip. Otherwise the ring is *free*.

It is desirable that more soliciting actions be performed when the ring is free and fewer when the ring is busy. However, in a decentralized environment each individual station has knowledge of its local status only, while the outer network is viewed as a black box. How can an individual station know when the ring is busy or free?

I believe that it is possible for an individual station to sense the workload conditions of the ring by floating the token holding time while keeping the token rotation time constant.

Two timers, the *ring_maintenance* timer and the so-called *token_holding* timer both defined in the IEEE 802.4 protocol, play important roles in the new soliciting strategy. The former timer in a station is reset to the values of the TRT every time the station passes the token and the latter timer is reset to the value of THT every time the station receives the token. When the token completes one round, the value in the *ring_maintenance* timer (before reset) represents the remaining token rotation time (R_TRT). Hence,

$R\_TRT = TRT - A\_TRT$

where A_TRT (Actual TRT) denotes the time interval during which the token has actually completed one trip around the ring. According to my understanding of the IEEE 802.4 standard, in the case where networks do not include many stations (say 100 or less) and use a high speed cable (say 10 Mbytes), TRT can be given by:

$TRT = N * THT + c$ (1)

where N is the total number of stations connected to the bus, and c is a constant representing the overhead involved, such as modem access delays and token passings.

The new strategy makes use of the ring_maintenance timer to allow the maximum token holding time (THT) to be dynamically expandable (or so_called floating in this thesis).

If $R\_TRT > 0$ (the *ring_maintenance* timer has not yet expired) when a station receives the token , the station will view the ring as being free. This can happen if not all stations have joined the ring, or some stations have not used up their THT (Eq.1). In either case, the network has not been fully utilized and there is some free bandwidth left in the current token round. Therefore, as long as the station is busy, it is allowed to expand its THT to $K * THT$. The scaling factor "K" is the parameter which could be set by the network manager, for instance, $K = 2$ in our implementation. On the other hand, if a station is free, it can perform a soliciting action rather than wait until its *inter_solicit_count* reaches zero as in the original scheme.

If R_TRT = 0, the station receiving the token knows that the ring is busy. This can happen if all stations have joined the ring and used up their THT or some previous stations have expanded their THT. In this case, the station should not perform a soliciting action at this time. However, a worst scenario may occur wherein some stations outside the ring are waiting to join.

This is an unavoidable situation when limited resources are being demanded by too many consumers. Any solution can always satisfy only part of the consumers while sacrificing others. The new strategy aims at providing a high network throughput and hence is biased toward those stations already in the ring. The new strategy keeps others waiting since allowing them in would only overload the ring which is already fully utilized.

The new strategy improves the protocol performance in two ways. First, it attempts to maintain the ring always at maximum capacity without overloading it by scheduling more soliciting actions when the ring is free and fewer when it is busy. Secondly, it increases the channel utilization U = DTT / T by expanding THT to a certain extent. Let DTT be the time spent for data transmission during given time T and denote the actual time spent for data transmission as A_THT (Actual Token Holding Time), then channel utilization can be defined as follows,

$$U = DTT/T = \frac{\sum_{j=0}^{J} A\_THT_j}{T} \qquad (2)$$

$$T = \sum_{j=0}^{J} A\_THT_j + J * TTP + \sum_{i=0}^{I} SNS_i \qquad (3)$$

In equation (2), A_THT_j is the A_THT spent in the j^{th} transmission and J is the total number of transmissions during T. In equation (3), TTP is the average time caused by one token passing. SNS_i is the time spent for the i^{th} soliciting action and I is the total number of soliciting actions during T.

Since the value of T remains constant, when $\sum_{j=0}^{J} A\_THT_j$ increases, $j * TTP + \sum_{i=0}^{I} SNS_i$ decreases (Eq.3). This corresponds to the situation when the ring is busy. The stations in the ring expand their token holding time leading to fewer soliciting actions and the same or fewer token passings. Therefore, a larger U is achieved (Eq.2). In the case where the modem has a long access delay and each soliciting action is expensive, the benefit can be especially significant. The overhead, in terms of CPU time and memory space involved in this strategy, is negligible.

A drawback of the new soliciting strategy is a longer delay for stations which wish to join the ring. To avoid starvation of stations outside the ring, the delay can be bounded by using a counter technique which is similar to the *inter_solicit_count* in the IEEE 802.4 protocol. The counter is loaded with a predetermined constant and is decremented by one every time the station receives the token. Whenever the counter reaches zero and the *ring_maintenance* timer has not expired, the station is obliged to perform a soliciting action even if it is busy. This will ensure an upper bound on the time that a station has to wait to join the ring.

In the case when priority classes [4] are implemented, this strategy can still be applied with a carefully chosen value of K. K is the coefficient in the expression $K * THT$, defining the expanding factor for the THT.

4.4 Skip Idle Stations

As mentioned previously, some stations in the ring may be temporarily idle. For instance, they may be waiting for data frames to arrive from the upper LLC sublayer. These idle stations nonetheless still participate in token passings in the ring and cause some overhead which reduce the network bandwidth utilization. Methods for dealing with this overhead have been suggested in the literature [23,26]. They are generally not very efficient or require additional resources. In [23], an extra channel is used to manage additional resources. Whereas in [26] a static skip operation is introduced which allow all idle stations to register to be skipped for the next round. Our proposed scheme can be viewed as an extension of [23]. A dynamic skip operation is used which allows idle stations to be skipped at any round instead of every other round as in [23].

[4] The IEEE 802.4 standard provides priority classes option.

4.4.1 Identification of Idle Stations

An idle station can be easily identified with three flags maintained by each station as defined in the IEEE 802.4 standard: *in_ring*, *in_ring_desire* and *any_pending*.

- *in_ring* indicates whether the station is in the ring or not. It is set and cleared by the MAC layer.
- *in_ring_desired* indicates whether the station wants to be in the ring or not. It is set and cleared by the LLC layer.
- *any_pending* indicates whether the station has any frame pending to be sent off or not. It is set by LLC layer and cleared by MAC layer.

After finishing data transmission, a station marks itself as idle if the boolean expression (in_ring & in_ring_desired & NOT any_pending) is true.

4.4.2 Skip Algorithm

To allow idle stations in the ring to be skipped, a temporary link between the predecessor and the successor of each idle station has to be established. Each station needs two additional variables, *t_predecessor* and *t_successor*, and three more frame types, *skip*, *token_2* and *token_3*. *T_predecessor* and *t_successor* store the addresses of a temporary predecessor and a temporary successor, respectively. The skip operation works in two phases as follows:

- *skip preparation:* When a station S1 marks itself as idle, it sends a *skip* frame to its predecessor S0 and a *token_2* frame to its successor S2 to prepare for the skipping of S1 in the next round. Upon receiving the skip frame, S0 knows that S1 is going to be skipped and S2 will become S0's temporary successor in the next token round. Similarly, when S2 receives token_2, it knows that S0 will be its temporary predecessor in the next token round.

- *skipping:* In the next token round, S1 is skipped by S0 when passes the token_3 frame to S2.

On the third token round, the temporary link is automatically terminated. S1 becomes active again and it receives the token from S0 (or t_predecessor of S0 if S0 is being skipped). It can repeat the skip operation if it is still idle. This scheme allows a station to be skipped one token round at a time. Figure 4.1 depicts the two phases of the skip operations.

The frame format of the IEEE 802.4 protocol is used for the three frames, *skip*, *token_2* and *token_3*:

SD	FC	DA	SA	DATA	FCS	ED

For *skip* frame, the SA (Source Address) field is filled with the idle station's address, e.g. S1. The DA (Destination Address) is set to the predecessor's address, e.g. S0 or S0's t_predecessor if S0 is itself an idle station to be skipped. The FC (Frame Control)

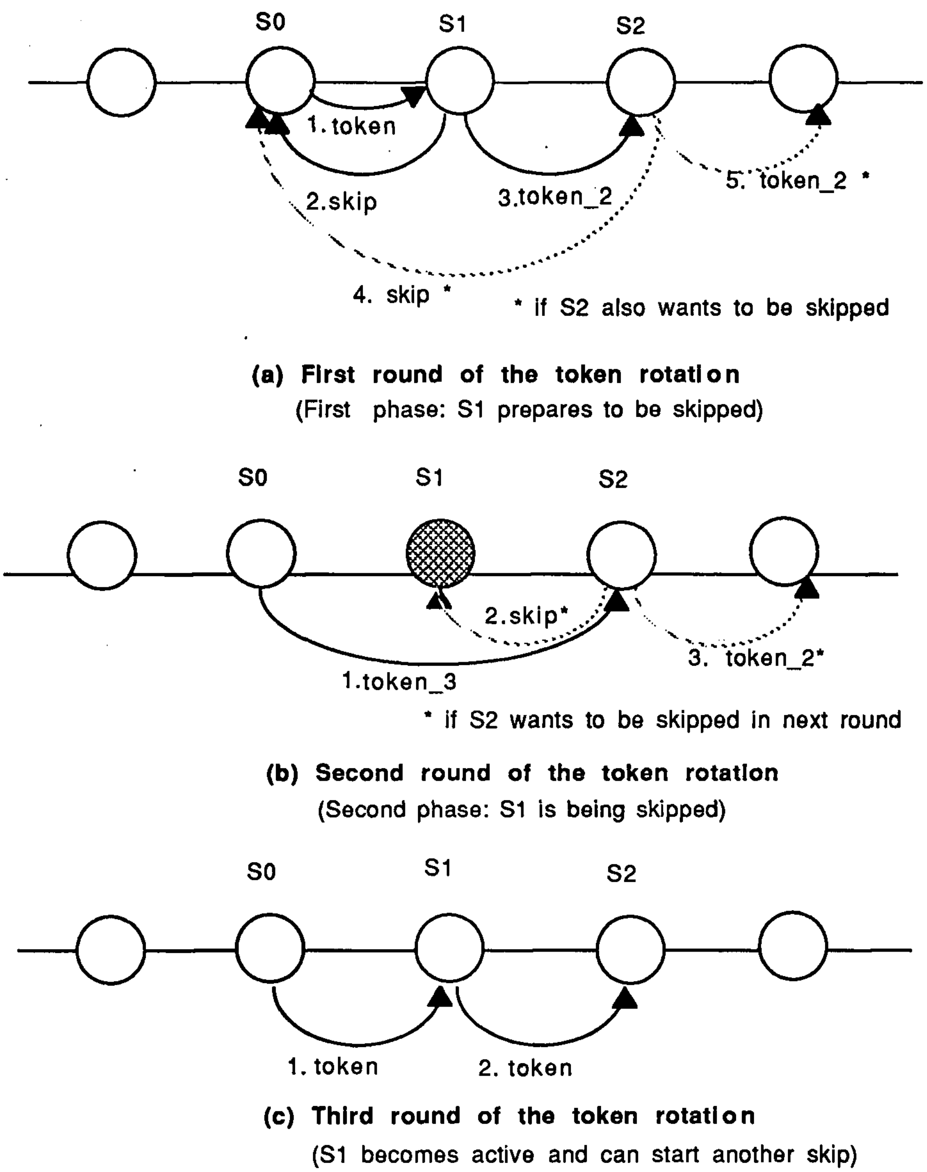

Figure 4.1: Skip Operation

field is filled with the frame type, skip. The DATA field is filled with the address of idle station's successor, e.g. S2.

The *token_2* frame is simply a substitution of the *token* frame and is sent from an idle station, S1, to its successor, S2. The FC field indicates the frame type *token_2*. The SA field is filled with the predecessor's address, S0, or S0's t_predecessor's address if S0 is also being skipped. The receiving station S2 gets its t_predecessor's address from this field. DA field contains the successor's address, S2. For the case where S2 is also idle and wants to be skipped in the next round, S2 sends a *skip* frame to its *t_predecessor* S0 (or S0's predecessor if S0 is also idle) upon receiving *token_2* from S1.

The *token_3* frame is also a substitution of the token frame and is sent by a t_predecessor (S0) to its t_successor (S2) in the second phase of the skip operation. The SA field contains the source station's address (S0). The DA field contains the temporary successor's address (S2). The FC field indicates the frame type, token_3. Upon receiving this frame, a station (S2) knows that its predecessor (S1) has been skipped in this round and will become active again in the next round. Therefore, if S2 becomes idle and wants to be skipped in the next round, it should send a *skip* frame to S1 rather than to its t_predecessor (S0). The format of *token_3* is the same as *token_2*.

The skip operation is implemented as a module which can be added to or removed from the entire protocol implementation. Thus the skip operation can be implemented in the protocol as an option.

Significant savings may result if several idle stations are skipped in a round. The overhead for a token relaying depends mainly on the hardware and software time for handling the token reception and the channel access delay for sending the token. The token propagation time is negligible in comparison. The number of frames involved in the protocol using the skip operation is the same as the one required in the original protocol without the skip operation. However, in terms of time and channel utilization, the former is more efficient, because it performs at least one less token relay during every skip operation. In fact, sending two consecutive frames, *skip* and *token_2*, is more efficient than relaying (receiving and passing) two tokens at once. It saves the processing of a token reception and an extra channel access.

Frames *token_2* and *token_3* are simply substitutions of the *token* frame. They do not incur any extra overhead. Since the token bus is a broadcast medium, a *skip* frame and a *token_2* frame can be further combined into a single frame to save an additional frame transmission. The overhead in terms of CPU time is negligible. The memory overhead for using the skip operation is two integers and three constants.

4.4.3 Future Study

Future study can be made to allow a station to skip n consecutive rounds ($n > 1$). In the current implementation, an idle station is allowed to skip only one token round at a time. It will be revisited by the token immediately on the next round. With a one-round constraint, the scheme is simple, the virtual ring is easy to maintain, and there is a minimal overhead. A far more complex algorithm with far more overhead would be required for a station to skip more than one consecutive round.

The complexity arises mainly because the ring configuration may change while the station is being skipped. The station that used to be another station's predecessor may no longer be so. It may be skipped, it may leave the ring altogether, or a new station may be inserted in between. The same is true for a station's successor. The algorithm that promises a solution must be able to somehow communicate all the changes that have occurred to the skipped station such that it can be reactive properly. This seems like a contradictory situation since the skipped station is refraining from receiving the token.

4.5 Leave-Ring Operation

The leave-ring operation in the original protocol may cause certain potential overhead. In the worst, if three or more adjacent stations leave the ring on the same token rotation round, the ring cannot be maintained properly. The reason for this is that leaving stations inform only their predecessors when they leave. They do not inform their successors.

A station performs a leave-ring operation by sending out a *set_successor* frame to its predecessor. Upon receiving this frame, the predecessor station knows that its successor is leaving and hence will pass the next token to the leaving station's successor on the next token round. On the other hand, the successor knows nothing about its predecessor dropping out. It will only be made aware when it receives a token from a different station in the next token round.

In the case where two adjacent stations are leaving within the same token round, the first one will inform its predecessor properly. The second one would try to inform its predecessor when instead it should be informing its predecessor's predecessor. Failing to do the latter would cause the first station's predecessor to send its next token to the wrong target.

An example may help to clarify the situation. Let S0, S1, S2, S3 and S4 be five adjacent stations in the ring. The scenario arises when S1 and S2 leave the ring on the same token round. S1 performs the leave operation by telling S0 about it. Upon receiving the *set_successor* frame, S0 will send its next token to S2. Unfortunately, S2 is also leaving the ring in the same token round. Not knowing what happened to S1, S2 will carry out its leave operation by informing S1 of it. In the next token round, S0 will send its token to S2 instead of S3. S3 would sit and wait futilely for a token. The protocol specifies that under this circumstance, S0 will attempt to pass the token n number of times after which it will issue a *who_follows* frame [5] to try to link up with

[5]Only can the station whose predecessor's address matches the address in the DA field of the *who_follows* frame respond S0.

S2's successor. In this case, S3 is in the ring and it responds properly.

The situation worsens when more than two consecutive stations leave the ring in the same token round. In this case, following the previous example, S3 is no longer in the ring and will not respond to the *who_follows* frame. S0 will continue to send out *who_follows* frames until it gives up. At this point it will broadcast a *solicit_any* frame to rebuild the ring. Hence, each time three or more consecutive stations leave the ring in the same token round, n *token* frames and n *who_follows* frames have to be transmitted each accompanied with a timeout event. In addition, a ring reinitialization procedure will be called for.

A very slight modification to the protocol can solve the problem. Each time a station leaves the ring, it alters the SA field value of the token frame to the address of its predecessor. By doing this, the successor, on receiving, the token can update its predecessor properly. Hence whenever a station leaves the ring, it informs not only its predecessor but also its successor. Thus every station in the ring will always obtain up-to-date information on who its neighbors are and will be able to pass the token to the correct station on each round. A advantage of this modification is that it costs absolutely negligible overhead.

Chapter 5

Experimental Results and Analysis

This chapter presents the results and analysis gathered on an experimental network of five SUN workstations where both the original IEEE 802.4 protocol and the modified version were implemented. Much research on the performance of token bus protocols have been conducted [20,19,14,5,4,13,18,8]. Here, the emphasis is on the comparison of the relative performance of two token bus protocol implementations instead of the absolute performance of either scheme. The measurements obtained from the experiment show that the modified version provides a network throughput improvement of up to 15 percent and a reduction in the network join-ring delay of up to 40 percent, while incurring a negligible overhead.

Section 1 of this chapter describes the experimental network setup. Section 2 explains how the network load is generated. Section 3 discusses the parameters, which when varied will affect the performance of the protocols. Finally, Section 4 presents the measurement results and their analysis.

5.1 Experiment Set Up

The set up of the experimental network is depicted in Figure 5.1 There are Six SUN 2/50s connected to an Ethernet with five of them acting as host stations communicating with each other over the Ethernet. The other station functions as the monitor station and collects statistics for performance evaluation.

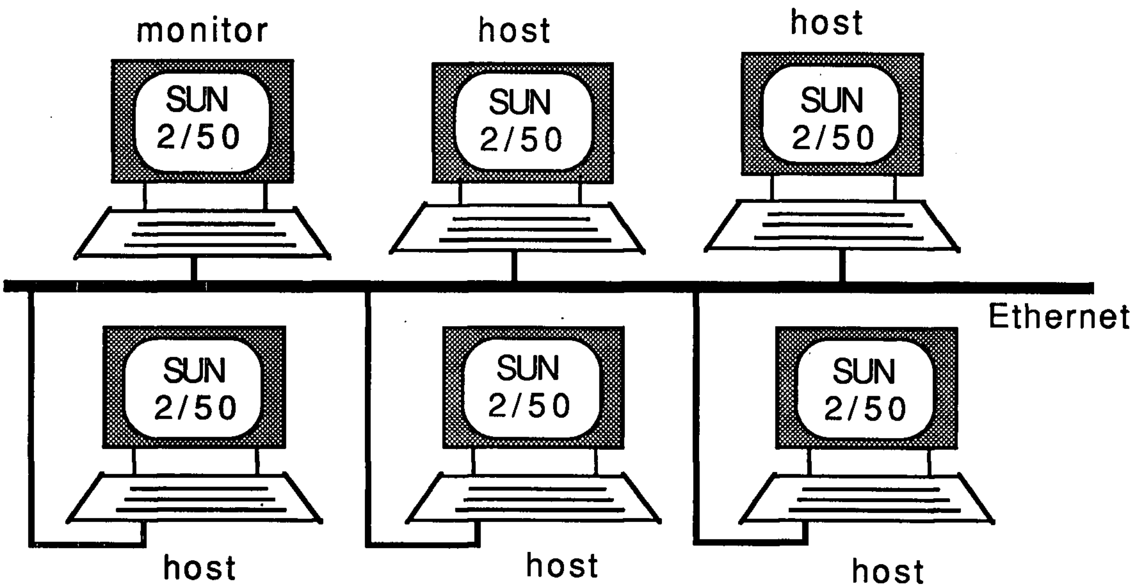

Figure 5.1: Experimental Network Set Up

The network system configuration in each node is given in Figure 5.2

The *token bus protocol* is the protocol of interest and it sits on top of TCP/IP [1]. TCP/IP stands for Transmission Control Protocol and Internet Protocol. These two form an end-to-end datagram protocol implemented in the UNIX [2] 4.2 BSD kernel.

[1]TCP/IP is American Department of Defence standards.

[2]UNIX is a trademark of A T & T Bell laboratories.

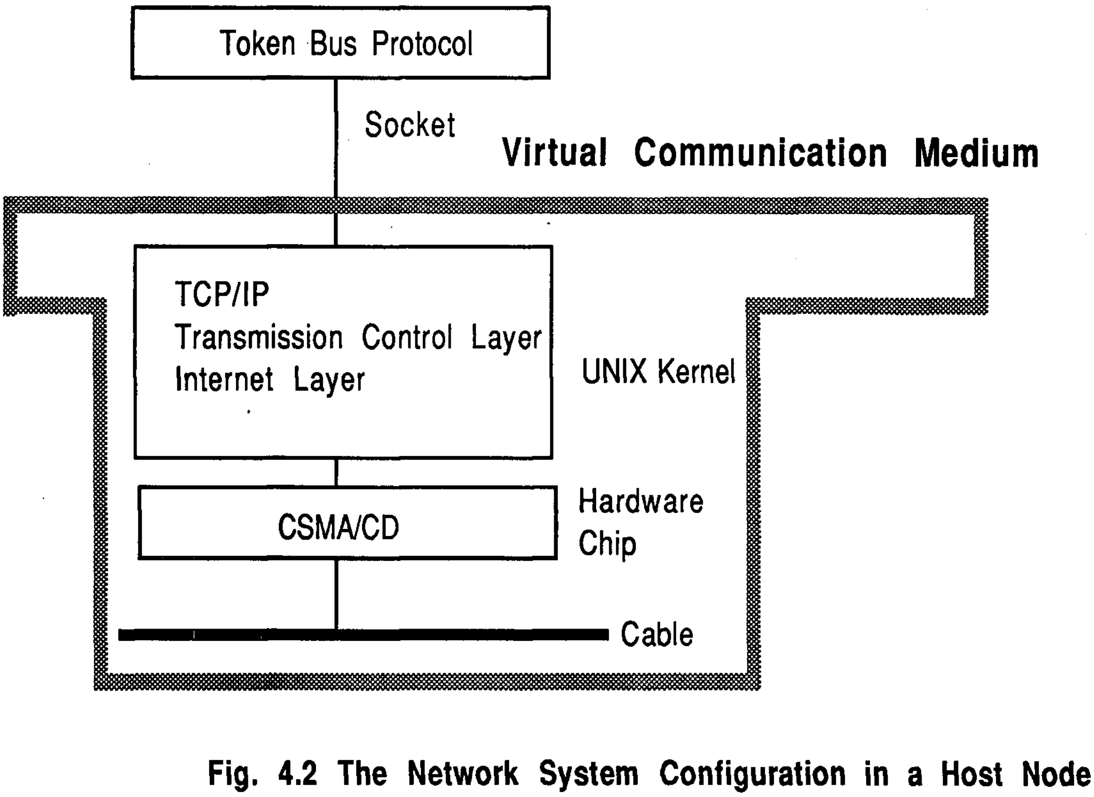

Figure 5.2: Network System Configuration in a Host Node

Beneath TCP/IP is the CSMA/CD (Carrier Sense Multiple Access with Collision Detection) medium access control protocol and it is implemented as a hardware chip in the transceiver that is directly connected to the Ethernet cable.

The UNIX operating system and the SUN 2/50s, together with the cable, provide users with a practical network environment which includes a communication channel and a set of network protocols up to the transport layer with respect to ISO reference model. In our experimental network, the token bus protocols run on top of another rather complete network system which includes a MAC protocol (CSMA/CD). Obviously, such a network configuration would not be suitable for measuring the performance of token bus protocols. The results obtained from such a network configuration cannot reflect the true performance of the protocols because the token bus protocol is not sitting inside the medium access control layer.

As stated earlier, what is of concern here is the relative performance of two token bus protocols. In view of this, the above network model should be acceptable. The network system provided by UNIX and SUN 2/50 can be viewed as a reliable ***virtual communication channel*** on top of which the token bus protocols sit. Since the two token bus protocols are examined under the same environment and are affected by that environment in the same way, difference in the results can be attributed to difference in the protocols themselves.

In this network model, a feature of the token bus protocol that cannot be simulated is collision detection in the channel because the CSMA/CD within the virtual communication medium has already handled the collision. However, this deficiency is congruous with the power line network. The modem used is only half duplex and it does not have the capability to detect collisions either.

The token bus protocol now sits on top of a virtual communication medium and it accesses the medium through a UNIX feature, socket. In this implementation, a socket is created with an *INET (internetwork)* domain, a *SOCK_DGRAM (datagram)* socket form and an *UDP (datagram)* protocol. This configuration allows the socket to simulate a communication medium by making it an unreliable datagram server. It does not guarantee delivery of packets. Neither does it guarantee the order in which the packets are received. Frames may be lost or duplicated. Furthermore, it is possible to do broadcasting through this virtual communication medium. Further information on sockets can be obtained in [1].

5.2 Traffic Generation

The network traffic is contributed by each host station randomly. Each host station has a *ready queue* where frames from the LLC layer are queued up. A special procedure called *traffic_generator* in each station is responsible for generating frames and inserting them into the ready queue, simulating the function of the station's LLC layer. The frames from this procedure can be considered as coming from the LLC layer. When a host station receives the token and its ready queue is not empty, it picks up the frames off the ready queue and sends them out to the channel.

Whenever the traffic_generator procedure is executed, it randomly performs one of the following activities, specified in the IEEE 802.4 protocol: (i) with a probability of 25 percent, the procedure performs a MNF (More New Frames) activity which involves generation of new frames and their insertion into the ready queue, (ii) with a probability of 50 percent, the procedure performs a NNF (No New Frames) activity in which no new frames are generated, and (iii) with a probability of 25 percent, the procedure performs a NRD (Not Ready) activity which means that the station has no data frames at the moment but new frames will be available soon.

Each frame generated by the traffic_generator is 16 bytes long. Whenever a frame is sent out, the CSMA/CD controller chip automatically appends a FCS (Frame Check Sum) and the FLGs (the frame delimiter flags) to the frame, bringing the total frame size to 20 bytes.

5.3 Other Parameters

There are a number of parameters which could affect the performance measurement of the token bus protocols. The ones considered are as follows.

- N (Number of Host Stations). This is set to 5. With a larger number of N ($5 < N < 100$), an improvement in the performance of the modified version is expected. In this set up, only six machines are available and one of them serves as the monitor station.

- FET (Frequency of Execution of the Traffic_Generator). The traffic_generator procedure is executed once every 800 milliseconds by having the system routine *timeout* invoke it. Whenever the *timeout* routine runs, the process running the token bus protocol is interrupted and control of execution is transferred to the traffic_generator procedure. If the timeout interval is short, the process running the token bus protocol would be interrupted too frequently causing undesirable effects. By trial and error, it was determined that 800 milliseconds is a suitable timeout interval.

- LMI (Length of a Measurement Interval). This parameter is set at 1 hour. After many trials and observations, it was found that it takes about half an hour for the measured system to reach a "steady state". Hence, the LMI is set at 1 hour. A shorter LMI may fail to produce an accurate result while a longer LMI is unnecessary.

- NNF (Number of New Frames). NNF stands for the number of new frames generated by traffic_generator each time. The NNF value controls the load on the network which can be varied from light, to medium, to heavy. The NNF values were set to the following sequence: 160, 200, 230, 260, 300, 330, 360.

These values represent a range from light to heavy workload.

- UBI (Upper Bound of the Inter_solicit_count). This parameter is used only in the original version. The UBI is given a value of 10. In the original protocol, a larger value of UBI would lead to fewer soliciting actions being performed while a smaller value would lead to more. A larger UBI is expected to favour the performance of the modified version. The performance of the protocols are not very sensitive to the value of UBI.

- THT (Token Holding Time). This is set at 40 milliseconds. A shorter THT would favour the performance of the modified version. The performance of both protocols is sensitive to THT.

- TRT (Token Rotation Time). The measurements were taken with a TRT of $(N+1) * THT$ milliseconds, where N is total number of stations. In this case it is set at 5. One additional THT in TRT provides some allowance for the overhead such as token passings and possible contentions during soliciting actions.

Based on numerous observations, the network system used for measurement is not always stable. The following factors could possibly be contributing to the instability.

- It is highly probable that the variation in the result could be caused by the varying load on the cable and the file server. The stations used for the host station and for the monitor are diskless stations which are linked up to a file server. Access to the file server is through the virtual communication channel. The file server services not only these six stations but other computers and computer equipments as well.

- *The random number sequence.* Instead of using a fixed sequence of random number for measuring both protocols, a real random number sequence was used. The latter could have caused the network load conditions to be somewhat different as well, hence contributing to measurement instability.

5.4 Analysis of Results

5.4.1 Throughput

Figure 5.3 shows the network throughput versus the network load for the two token bus protocols.

The throughput is defined as $dataframes/second$. Data frames are those frames which come from the LLC layer (that is, generated by the traffic_generator procedure), excluding all the medium access control layer frames such as *token*, *solicit_successor* and *set_successor* frames. The workload is represented by the number of new frames (NNF) [3]. From the plot in Figure 5.3, the following observations are worth noting.

[3]Please reference the section "Other Parameters" and the section "Traffic Generation" for the meaning of NNF.

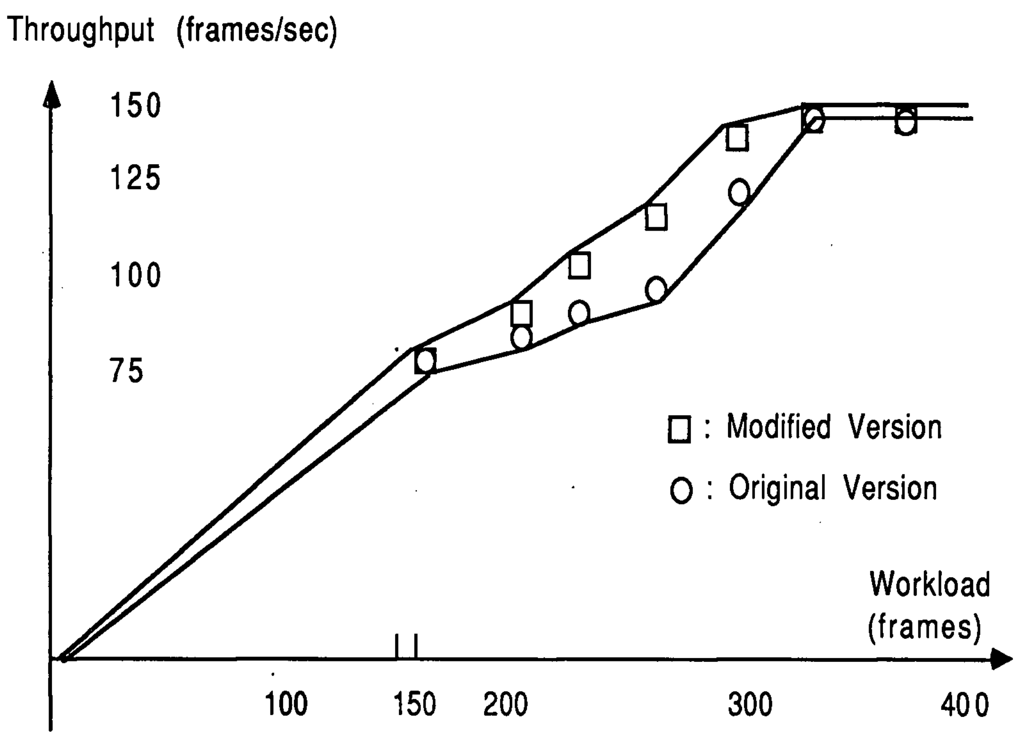

Figure 5.3: Network Throughput vs Workload

- When the NNF is under 160, the throughput curves of the two protocols are fairly close. This indicates that under light load conditions, both protocols are capable of sending out all the frames that were pending in each station. This behaviour is expected.

- When the NNF is more than 330, the gap between the two curves narrows and both curves tend to flatten out. This indicates that fast frame arrivals force the five stations to stay in the ring and use up their THT oftentimes. Saturated behaviour is also expected for heavy loaded conditions.

  As indicated in the previous chapter, the modified version is expected to have higher throughput in two ways: (i) by performing more soliciting actions under light load conditions and expanding the THT in some stations under heavy load conditions; and (ii) by skipping idle stations. When the network is saturated, all the stations are in the ring and none of them is idle. Both protocol versions should achieve the same throughput.

  However, the curve of the modified version shows a little higher throughput. The reason for this has to do with the new soliciting strategy. Some stations still have some (rare) chances to expand their THT and hence increase the efficiency of channel utilization, leading to a slightly higher throughput.

- The most interesting phenomenon is noted in the interval from 160 to 330. This range represents normal operating interval in which the advantages of the modified version can be fully observed. The modified version provides a higher throughput of up to 15 percent than the original one. As the workload increases, some stations start to expand their THTs so as to increase the usage of the channel. Also more soliciting actions are scheduled when there is free network bandwidth, leading to further improvement. Finally, reduction in overhead may be achieved by skipping the idle stations.

For larger networks with more stations and/or longer channel access delay, a more substantial improvement is naturally expected.

5.4.2 Network Join-Ring Delay

The network join-ring delay (NJRD) is another performance measure of interest. The NJRD is defined by

$$NJRD = \frac{\sum_{j=1}^{N} \sum_{i=0}^{M_j} jrd_{i,j}}{N}$$

where N = 5 is the total number of stations and M_j is the total number of join-ring actions of station j during the measurement interval T. $jrd_{i,j}$ denotes the join-ring delay of station j in the i^{th} join-ring action. It is the delay from the time a station starts a join action to the time it receives a token or gives up the join action. This delay depends on two factors: (i) the frequency of soliciting actions and (ii) the soliciting window, That is, the range of addresses of stations which are solicited to join the ring in a soliciting action. According to the IEEE 802.4 standard the soliciting window for each soliciting action spans from the station holding the token to its successor in the virtual ring. This is implies that the more stations are in the ring, the narrower the soliciting window will be and consequently the longer the join-ring delay would be.

The measurement of the NJRD is more interesting than the individual join-ring actions because the NJRD also takes into consideration of the total number of join-ring actions, which gives a clearer picture of the overall network join-ring delay performance.

Figure 5.4 shows the plots of NJRD versus network load for the original protocol and the modified protocol. Generally, the modified protocol provides a better NJRD performance than the original protocol except for the range of excessively heavy workloads.

The NJRD behaviour is as expected. For the case of light workload, say less than 160, both protocols have a very small NJRD. Oftentimes, the ring includes only one station which performs frequent soliciting actions with the full-size soliciting window. A few stations outside the ring can join the ring quickly with virtually no waiting or contention delay. As the workload increases from light to heavy, the ring tends to be busy with more stations in the ring. Therefore, fewer soliciting actions are performed with narrower soliciting windows. Thus, the NJRD gets much larger because the ring-join delay jrd gets larger and there are a reasonable number of ring-join actions due to stations leaving and rejoining the ring. However, as the workload becomes quite

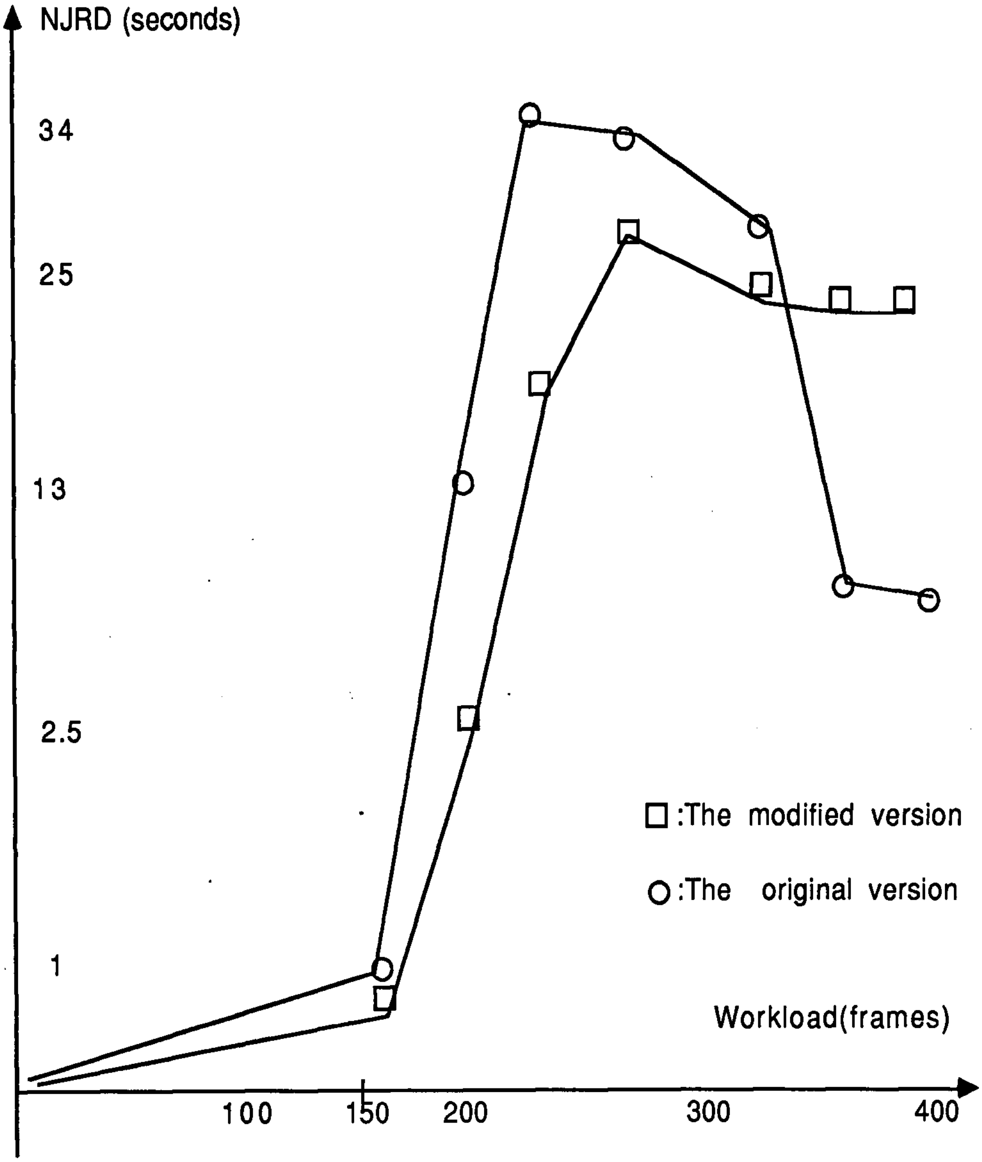

Figure 5.4: Network Join Ring Delay vs Workload

heavy, say around 250, the NJRD begins to drop as the number of ring-join activities decrease. As the workload increases further, the NJRD eventually drops to a constant level. This corresponds to the case where every station joins the ring remains busy throughout the time interval T.

Chapter 6

Implementation on the PLLAN

This chapter presents the performance measurements obtained from the power line local area network. Section 1 describes the set up of the power line local area network. Section 2 evaluates the performance of the modified IEEE 802.4 protocol. Some features specific to this layer such as response time and join-ring time are investigated. Section 3 presents the network performance at a user level by means of file transfers over the network.

6.1 Environment Set Up

The PLLAN is set up as depicted in Figure 6.1. Three SUN 3/50 Workstations (due to the fact that only three modems were available) are interconnected to each other by power distribution circuits via the use of power line modems. The Protocol Tester (PT) serves as a monitor. The modified token bus protocol is placed in the UNIX 4.2 BSD kernel and is able to handle interruption signals caused by incoming

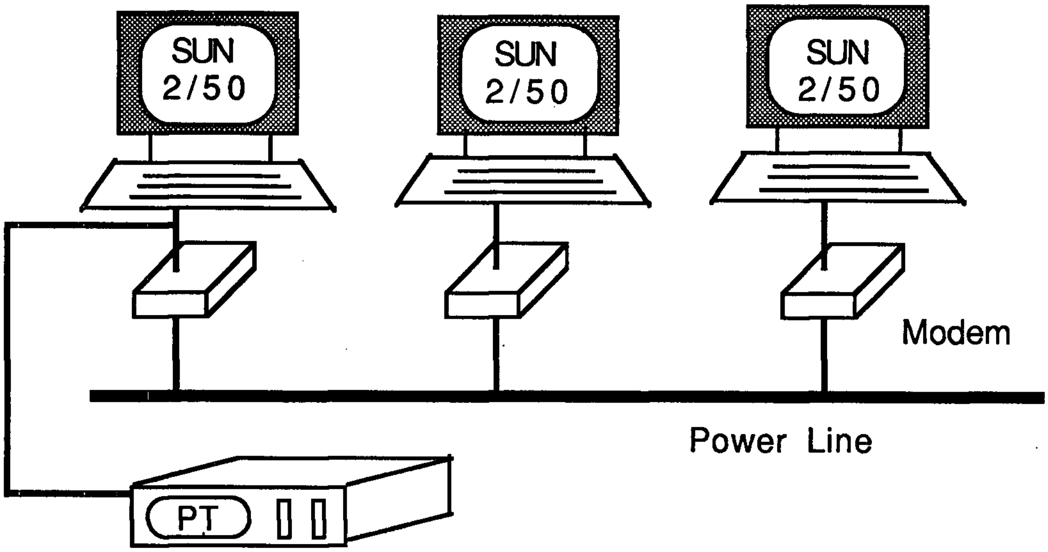

Figure 6.1: Power Line Network Set Up

frames efficiently.

The PT is a product of IDACOM company and is a sophisticated multiprocessor-based tool used for passively monitoring the traffic on the medium. It is attached onto an RS-232C line between a SUN Workstation and its modem. The PT also has the ability to capture and time-stamp communication scenarios for future playback and analysis. With the aid of this tool, several software probes can be eliminated from the software under measurement and a more accurate measurement can be obtained.

The measurement only focuses on the the modified token bus protocol, since a comparison of two protocols requires too much work.

6.2 Performance of the Modified Protocol

The performance of the token bus protocol depends on a number of parameters. Deciding a value for each of these parameters requires some knowledge of the end user application. Some "basic" parameters which are relatively application-independent were measured. One of the basic parameters we have controlled during these measurements is the channel idle time.

The channel idle time is the longest period of time in which a station allows the channel to be idle before further action. It should be kept minimal to make efficient use of the channel bandwidth. This time can be determined through measurements since it depends mainly on the speed of both the communication software and hardware. Measuring how fast a station can respond can give a lower bound for this parameter. Many other parameters such as the time to detect the death of a station and to retransmit will depend largely on the channel idle time.

By using the PT, the response time is measured as the difference between the time a token frame is detected and the time a frame sent by the token receiver is detected. The mean response time over 20 samples is found to be 45.3 milliseconds, under the condition that there is only one user on each SUN. This time depends solely on the speed of the hardware and the MAC software. The response time can be broken down into the time for asserting the channel (which is about 15 milliseconds), data transmission time, propagation time, end-of-message extend time imposed by the modem and software processing time. Since the power lines cover only a short distance and a token frame has only seven bytes, the data transmission time and the propagation time should be negligible. The end-of-message extend time of the modem is about about 30 to 35 milliseconds respectively. This implies that only a few milliseconds are used by the software to process the send and reception sequence.

The response time is obtained by measurements between two SUN 3/50's each of which supports multiple-users and allows multi-processing. At the time of measurement, all the workstations were very lightly loaded. In fact, only one user was logged onto each workstation. The response time is expected to increase as the load of the workstation increases. Based on the 45.3 milliseconds response time obtained, the channel idle time was set to 90 milliseconds to accommodate software delay incurred by heavy usage of the workstation or the network.

The ring joining process is a important feature of the protocol. This process involves interactive actions between the stations. A time cost estimate of this process can provide us with a clearer picture about the performance of the protocol.

The ring joining time is measured as the elapsed time between the detection of a *solicit_successor* frame and the detection of a data frame sent by the station which joined the ring. The case where one station joins the ring takes about 155.7 milliseconds. The four steps which constitute in a ring joining process are as follows.

- The station holding the token issues a solicit_successor frame and then starts a timer. If nothing is heard after the timer expires, the station passes on the token to its successor. In the case where the station is a the sole station in the ring and it does not have data to send, it repeats the soliciting action n times before leaving the ring. Otherwise, it will go back and transmit.

- The station that wants to join the ring and whose address fits the range of the soliciting window responds to the soliciting station with a *set_successor* frame. Then it returns to listen to the channel.

- When receiving the *set_successor* frame, it stops the timer and passes the token to the joining station and starts a new timer. If it hears any frame in the channel, sent by the joining station, it assumes that the joining station has already joined the ring successfully, and it stops the timer and goes back to the idle state. Otherwise, when the timer expires, it sends the token frame again. If it fails to pass the token n times, it passes the token to its original successor. In the case where the station is the sole station, it will either leave the ring or continue its transmission.

- When the joining station receives the token addressed to itself, it knows it has joined the ring successfully and can use the channel. Otherwise, if any other frame is received, it implies another station is using the token and the invitation process is over. The station goes back to the idle state.

As in the case of response time, the frame transmission and the propagation times are negligible. The modems have a time expensive of approximately 135 milliseconds. We can see that the software costs are negligible.

6.3 Overall Performance

It is interesting to assess the overall performance of the PLLAN at a user level. A link level file transfer is performed so that the whole network, MAC, LLC sublayers and the physical channel can be exercised together.

A ASCII text file is transferred between two UNIX processes (client and server) on different machine using the connection-oriented services. The client process first establishes a link with the server process. The name of the file is then transmitted to the server, the server then sends the number of frames expected followed by the content of the file. User level delay of the file transfer is measured using the "/bin/time "utility on UNIX.

The frame size is 128 bytes of which 112 bytes are for data usage and 16 bytes are for the header. The token holding time is set to 0.5 second. The file transferred has 311 lines, 6787 characters. The modem operates at 9,600 bps and has a total delay of 40 to 45 milliseconds. Note that the user level delay includes the link establishment and tear down times as well as the file transfer time.

It takes about 16.4 seconds for the file transfer to be completed. The time for the data transmission takes about 2 seconds in total. In a similar study done by Audivox Ma [15], a Hub Polling Scheme was used. The modem used operated at 14,000 bps and had a delay of 20 to 25 milliseconds. The file transferred was 21,000 bytes. The frame size was also 128 bytes and the token holding time was set to 1 second.

It took about 54.8 seconds for the file transfer to be completed. The data transmission also took about 2 seconds.

Chapter 7

Conclusions

7.1 Summary

This thesis describes the design and implementation of an improved IEEE 802.4 token bus protocol for a low-cost and portable local area network which utilizes the power distribution circuit as the communication medium and the prototype power-line modem as an interfaces between a computer and the power line.

The network is easily saturated with data due to the limited data rate of 9.6 Kbytes psd provided by the power line medium. Moreover, the prototype power-line modem cannot detect collisions due to its half-duplex nature and it has a long contention period of 4 to 10 milliseconds, resulting in high collision probabilities. Contention protocols such as CSMA and CSMA/CD, therefore, are not suitable for this power line network.

The major rationales behind the adoption of the IEEE 802.4 token bus protocol are its non-contention property and tolerable performance under heavy network load conditions, besides its effective support of time-critical services and its reliability as a decentralized system.

The main disadvantage of the protocol is the overhead involved in maintaining the virtual ring. Such a overhead can be reduced considerably by making the following modifications to the protocol. (i) The new soliciting strategy for inviting new stations to join the ring attempts to keep the network functioning at maximum capacity without overloading it, by scheduling more soliciting actions when the ring is free and fewer when it is busy. (ii) The skip operation, that is introduced, further cuts down the overhead caused by the idle stations. (iii) A potential overhead caused by the leave-ring procedure was spotted and a slight change was proposed that can avoid the overhead. (iv) A shorter frame format with one byte address space is suggested, taking into consideration the fact that the power line networks tend to be small in size. Thus the traffic generated by the MAC layer can be reduced by up to 30 percent. The first three of the above modifications could also be applied to other general local area networks if the token rotation time of the networks can be defined by $TRT = N * THT + c$. Where N is the total number of stations connected to the bus, and c is a constant presenting the overhead involved such as modem access delay and token passing.

Both the original IEEE 802.4 protocol and the modified version were implemented in an experimental network of five SUN 2/50 Workstations. The measurements obtained from the experiment show that the modified version provides a network throughput improvement of up to 15 percent and a reduction in the network join-ring delay of up to 20 percent.

The modified version was implemented on a real power line area network environment where 3 SUN 3/50 Workstations were transmitting data through the modems and over the power line. A typical response time of 45.3 milliseconds is measured at the MAC layer. This indicates that the access (MAC) communication software takes only a few milliseconds and represents a small overhead over the 40 to 45 millisecond spent by the modems. The overall (user-level) delay for transferring a 6787 byte file between two stations in the network is approximately 16.4 seconds. The measurements support the practicality of the idea of using power lines as a data transmission medium.

The modified protocol program has about 1700 lines. Other suporting programs for implementation and performance evaluation have about 2100 lines.

7.2 Suggested Applications

Using intra-building electric power lines as a transmission medium for interconnecting computers provides a number of advantages, including low-cost and portability. Such a network is very suitable for a family or small business environments or non-critical auto-control systems. Also due to the readiness of this medium, it is an ideal candidate for providing an alternate route in a hybrid local area network to improve reliability of a communication system.

Considering its merits, the modified IEEE 802.4 protocol can be used for local area networks which are required to support time-critical services and handle heavy workloads.

7.3 Future Works

In the skip operation, an idle station is allowed to be skipped by only one token rotation round each time. It will be revisited again by the token in the next round. A possible related topic for future work may be to extend the current "single round skip" operation to "multiple round skipping". Such an extension potentially provides further improvement in channel utilization at the cost of having a more complex algorithm and added processing.

Priority classes were not considered in this work, the incorporation of priority classes in the proposed modified protocol could be another interesting topic for future research.

Finally, from the experience of using the prototype modems, it is found that the carrier sense protocol which is internally wired up in the modem causes three problems: (i) It is redundant and incurs extra overhead when a non-contention MAC sublayer is used. (ii) If a modem fails to release the physical channel, the whole system can be totally overtaken. (iii) A rather long channel access time is a direct consequence of executing this protocol. Removing this feature from the modem logic when using a non-contention MAC protocol would improve the overall performance of the network.

Bibliography

[1] AT&T. *UNIX System V - Network Programmer's Guide*, pages 16–49. Prentice-Hall, Inc., 1987.

[2] R. Bermbach, H. Geisse, and A. Kappas. Datalane - a low cost local area network. In *IFIP TC 6 In-Depth symposium on Network in Office Automation*, 1984.

[3] Morgan Chan. *Communication Channel Characteristics and Behaviour of Intra-building Power Line Distribution Circuits.* University of British Columbia, 1985.

[4] Ben Chang and Tuncay Saydam. Mathematical properties of conditional rotation times for bus/ring local area networks. In *IEEE Infocom'87*, pages 155–160, 1987.

[5] J. Y. Chien. Performance analysis of 802.4 token bus medium access control protocol. In *Workshop on Analytic and Simulation Modeling of IEEE 802.4 Token Bus LAN, NBS*, 1985.

[6] Frank Chiu. *Data Communications Using Coherent Minimum Frequency Shift Keying on Intrabuilding Polyphase Power Line Network.* University of British Columbia, 1985.

[7] I. Chlamtac, W. R. Frank, and D. Levin. Bram: the broadcast recognization access method. *IEEE Transaction Communications*, COM-27:1183–1190, 1979.

[8] M. Alex Colvin and Alfred C. Weaver. Performance of single access classes on the ieee 802.4 token bus. *IEEE Transaction on Communications*, COM-34(12):1253–1256, 1986.

[9] P. K. Van Der Gracht and R. W. Donaldson. Communication using pseduonoise modulation on electric power distribution. *IEEE Transactions on Communications*, COM-33(9), 1985.

[10] G. T. Hopkins. Multiple communications on the mitrenet. In *Local Area Communications Network Symposium*, 1979.

[11] G. T. Hopkins and P. E. Wagner. Multiple access digital communications system. In *U.S. Patent 4,210,780*, 1980.

[12] McDowell J. Broadband or baseband: csma/cd or token - what's the difference? In *LOCALNET84*, pages 758–762, 1984.

[13] Anura P. Jayasumana. Performance analysis of a token bus priority scheme. In *IEEE Infocom'87*, pages 46–54, 1987.

[14] B. A. Loyer and Kolton D. Performance issues of ieee 802.4 token bus lan's. In *Workshop on Analytic and Simulation Modeling of IEEE 802.4 Token Bus LAN, NBS*, 1985.

[15] Audivox Hing Ting Ma. *A Power Line Local Area Network: Design and Implementation.* University of British Columbia, 1987.

[16] R. M. Metcalfe and D. R. Boggs. Ethernet: distributed packet switching for local computer network. In *Communications of the ACM*, 1976.

[17] R. M. Metcalfe, D. R. Boggs, C. P. Thacker, and Lampson B. W. Multipoint data communication system with collision detection. In *U.S. Patent 4,063,220*, 1977.

[18] K.H. Muralidhar and J.R. Primentel. Performability analysis of the token bus protocol. In *IEEE Infocom'87*, pages 55–63, 1987.

[19] A. Nakassis. On the stability of a token passing network. In *Workshop on Analytic and Simulation Modeling of IEEE 802.4 Token Bus LAN, NBS*, 1985.

[20] S. R. Sachs, K. Kan, and J.A. Silvester. Performance analysis of a token bus protocol and comparison with other lan protocols. In *The* 10^{th} *Conference on Local Computer Networks*, 1985.

[21] M. Scholl. *Multiplexing Techniques for Data Transmission Over Packet Switched Radio Systems.* Computer Science Dept., UCLA, 1976.

[22] William Stallings. *Data and Computer Communications.* Macmillian Publishing Company, New York, 1985.

[23] T. L. Sterling, Williams R. D., and Kirtley Jr. Comment: an intrabuilding data link. In *IFIP TC 6 International In-Depth symposium on local networks*, 1982.

[24] Flatman A. V. Cheapernet - a low cost open system lan. In *LOCALNET84*, 1984.

[25] S. T. Vuong, A. Ma, and H. Gu. Design and implementation of the access and link protocols for a low-cost lan using electric power lines. In *GLOBECOM*, pages 1415–1421, 1987.

[26] Alfred C. Weaver and David W. Butler. A fault-tolerant network protocol for real-time communication. *IEEE Transactions on Industrial Electronics*, IE-33(3):207–211, 1986.