Micro Network Protocol and Hardware Design for Distributed Data Acquisition

Tan Loc, PHAM

B. Eng., McGill University, 1992

Department of Electrical Engineering

McGill University, Montreal

March 1996

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirement of the degree of
Master of Engineering

ISBN 0-612-12132-1

Acknowlegement

I would like to express gratitude, first and foremost, to Prof. V. Hayward for the long hours he dedicated to this project during the course of its development. I am deeply thankful and indebted for his kind support and guidance.

I would like to thank my family, who were always caring and supportive throughout the time of my studies. I would like to express my gratitude especially to my Dad and Mom for being my moral backbone and teaching me to be strong in the face of adversity. I would also like to thank all my best friends for their precious suggestions and their constant encouragements, and in particular, Don Bui for giving the extra time in the proof-reading.

The financial support for this work was provided by the Canadian National Center of Excellence program (NCE) under project MS-4 of the Institute of Robotics and Intelligent Systems (IRIS). Their support is gratefully acknowledged.

Abstract

The purpose of this work is to develop an intelligent system capable of transferring data from many distributedly located sensors to a main processing unit. The data inputs received from the sensors are usually analog signals and the goal is to create smart integrable modules for storing the converted digital signals, interfacing with a shared bus, as well as performing parallel data transfers in an autonomous and intelligent manner.

On a small scale, the choice of such an integrated system is to avoid as much as possible the complexity and redundancy problems created by wiring single devices. The goal is to make a portable and practical system to interface to other subsystems, while reducing data communication wiring and optimizing data parallel processing. Micro-networking modeling issues are then studied for the implementation of the system and simulation results will be shown to illustrate the performance of the system.

Résumé

L'objectif de ce travail consiste à développer un système intelligent capable de transférer les données de plusieurs capteurs répartis, vers une unité centrale de traitement de signaux. Les données d'entrée reçues à partir des capteurs sont en général des signaux analogiques et le but est de créer des modules intégrables "intelligents" (smart) pour stocker les signaux transformés de type numerique et de les transmettre sur un bus réparti de façon autonome et intelligente.

Sur une petite échelle, le choix d'un tel système intégré permet d'éviter autant que possible les problèmes de complexité et de redondance des dispositifs isolés cablés. Nous visons à réaliser un système qui soit portable et commode à interfacer avec d'autres systèmes tout en réduisant le cablage pour la transmission des données et en optimisant le traitement parallèle des données. La modèlisation du "micro-réseau" sera ensuite étudiée pour la réalisation du système et les résultats de simulation seront présentés pour illustrer les performances du système.

Contents

List of Figures

Chapter 1

Introduction

In today's era, the reach and speed of networks has increased by remarkable strides. Information data exchange can reach speeds of up to a billion bits (gigabits) per second, in services and information networks, either for customer or for research applications.

At the scale of an engineering system, for example a robot, networking is needed to provide information for control and processing. Take for instance the example of a master-slave robot manipulator, the Sarco Dextrous teleoperational system such as the Sarcos Dextrous Arm, which includes more than twenty axes of freedom with two or three sensors per joints. As a result, at least forty sensors enter the count, with three wires per sensor to communicate the data. Consequently, a set of more than one hundred and twenty wires will be a large, bulky, and difficult to handle, data transmission system for every sensor to be interfaced to a central processor. Hence, the need of a distributed shared bus system is clearly necessary to service data communication. Many such examples can be found.

This thesis proposes an integration method for the application of data acquisition using the Carrier Sense Multiple Access with Collision Detection concept [14]. The purpose of the project is to design a low-cost shared bus micro-network, for transmitting data at a fair transmission rate, from distributedly placed sensors over a short distance (about 5 meters) to a central processing unit.

In this research, we propose that sensors acquire data periodically and send it over a micro-network to a receiving node within light time constraints. Each individual sensor is assumed to be intelligent or autonomous, and is assigned with a priority number for sending data. Each sensor is self-driven and has the capabilities of reading direct measurements regularly at discrete instants of time. Those data items, which are then converted and kept in some digital form, wait for their transmission period to be sent to a receiving node over the common shared bus system. Just like in most networking applications, data items will be stored in the form of packets and be associated to their characteristics, such as an identity number or priority number of the associated sensor, in order to regulate the bus use while providing identified information for processing at the receiving node. The scope of this work, however, does not deal with data format nor with data processing, but rather with data communication and transmission.

With the concept of shared bus, the behavior of the system can be affected in terms of performance, delays and other parameters of interest. Because it is a shared resource, collisions of simultaneous transmissions from different sensors can occur on the shared bus system. By properly structuring sensors, as low and fast transmission types, and defining their priorities in term of waiting delays, the network system is shown to converge to a stable transmission point while its throughput can be increased according to the various factors of concern, such as number of nodes, transmission and sleep rates of device types, etc. Test cases are carried out in simulation to evaluate the desired pattern of devices and observe its effect on the network behavior using the designed protocol. With an assumed transmission rate up to 10 Mbits per second (Mbps), it is observed that the network can exhibit better utilization of the shared bus for a certain distribution of sensor node types.

Although better performance could be achieved, most of today's systems are too complex and expensive because they usually provide for general-purpose data acquisition applications. This specific distributed data acquisition can be restricted to certain applications at a moderately fast speed. However, it can also be easily or flexibly adapted to various applications such as robotics and other fields. Hence, the design of a 10Mbps micro-network for distributed data acquisition is studied, developed and evaluated using VHDL, the Very

High Description Language. There is no optimal choice in the design of this distributed bus network, since it will depend on the purpose of the application of the network. Therefore, there is a need to study the effects of the distribution of sensors and of their parameters (transmission rates, waiting rates, and so forth) which can affect system response quality and traffic flow (bus occupancy). In the general case, the total load produced by all the sensors on the common shared bus system should be at the proper consumption rate of the bus capacity; the load should be adequately distributed over the bus bandwidth. A common performance degradation arises when traffic becomes overloaded or bursty, leading then to bus congestion or conflicts of transmissions on the shared bus.

The project is presented in the following order. In chapter two, the different approaches to networking for distributed data communications are briefly presented. The concept of Carrier Sens Multiple Access with Collision Detection (CSMA-CD) is studied with respect to the current existing the literature. The design of a micro-network protocol, a set of communications rules for the purpose distributed data acquisition, is then developed with respect to the CSMA-CD method and implemented in chapter three. The hardware model of the network is discussed in chapter four. The current design is based on Transistor-Transistor Logic (TTL) gates and integrated circuit chips for cost reasons, while maintaining an acceptable transmission rate (10Mbps). The operational behavior of such a network is then examined in terms of performance and efficiency through a series of simulation test cases. In the concluding chapter, final points of observations will be drawn while a brief discussion of network issues will also be given.

Chapter 2

Discussion of the literature and Design Rational

2.1 Basic Concept and Procedure

This section gives a brief overview of CSMA/CD networks. Different approaches are proposed to model the input/output relationships of the parameters of interest. The performances of CSMA/CD networks which exist in the literature are summarized at the end of this chapter.

2.1.1 Network Topology

Different transmission topologies are possible and compared [4] [5]. The topology of a network, as well as its control strategy, can considerably influence the performance and reliability of a network. Here, the objective is to develop a distributed data acquisition system for sensors, which can be applied to various general- purpose applications in robotics and other fields. Consequently, multiple access networks are preferred since self-driven sensor nodes can be made intelligent and can be added to the network for communicating physical data to a central processing unit, with a great deal of flexibility. In general, multiple access network protocols are divided into two classes:

One class of multiaccess protocols for packet communication systems use the random access (or contention) technique, where the entire bandwidth is provided to the users as a single channel to be accessed randomly. In random access protocols, transmission rights are simultaneously offered to a group of stations under the assumption that exactly one of the stations has a packet to send. However, if two or more stations send packets on the channel simultaneously, these messages interfere with each other. In such a case, a collision occurs and the transmitting stations abort their transmissions and retransmit packets until they are successfully sent within a transmission period of time. Such policies are sometimes referred to as Carrier Sense Multiple Access with Collision Detection (CSMA/CD) mechanisms.

On the other hand, a controlled access mechanism is one in which a token is first secured by a node in order to transmit its message through the medium. Controlled access protocols, such as in token rings or buses, avoid collisions by coordinating access of the stations to the channel by imposing either a predetermined or dynamically determined order of access. Access coordination is done by use of the channel itself. Each station indicates with a short message on the channel whether or not it requests access. This polling mechanism consumes some channel capacity regardless of whether stations require access or not. Although such protocols are efficient when traffic is heavy, under light traffic conditions they result in unnecessary packet delays as stations that want to transmit wait their turn.

Bux [4] has studied and showed the variations between those network topologies. Random access protocols exhibit small packet delays under light traffic conditions; stations transmit as soon as they request access to the channel, and the probability of a collision is low when traffic is light. Another attractive aspect of random access protocols is their extreme simplicity and flexibility, making them easy to implement, since by design, they are completely distributed. Also, failure of any node does not affect the overall operation.

Under low traffic load, CSMA/CD protocol has a distinct advantage because it uses no media overhead to determine which device has to access the network. Testing on networks using CSMA/CD protocol schemes, such as Ethernet, show superior performance to token bus at highway loads less than 40% of network capacity.

For high loads above 40%, the response time (in terms of transfer delay-throughput behavior) grows exponentially, and quickly becomes significantly worse than token bus or ring protocol. Of the three network protocols, the token bus is apparently more suited to control communication application. The CSMA/CD protocol is well suited for standard computer network applications, in which the network load rarely exceeds 40%.

Generally, the question to be addressed is how to deal with the data traffic with a low cost and reliable distributed system in order to acquire data from self-driven sensors. For most process-control applications, the essential requirement is to ensure that informative data is transmitted in a continuous and timely manner. A certain data loss ratio may nevertheless be acceptable. Token bus best satisfies these requirements, yet, a Token Ring or Bus topology would not be as reliable as a Carrier Sense network in terms of failure recovery since a failure node has to be removed from a Ring network and the system has to be restarted. It would also be more complex.

The local "micro-network" will adopt the CSMA/CD architecture, contrary to conventional wisdom, for locally distributed control, where parallelism and concurrency are among the necessary factors for high-speed and real-time signal processing. To recover lost data, we will consider data cyclicity as inherent to sensor state refreshing [12] [28], the spatial consistency will be lost by a single transmission error during a cycle but can be rectified automatically by the system in the next cycle. During this uncertain period, probabilistic data or data extrapolation could be employed to estimate the missing data values. The correct decision depends on the nature of the data and the applications which have to be taken into account carefully by the user.

Our approach will be characterized by a distributed network of concurrent data communication. By exploiting concurrency and distributedness, the Carrier Sense system only provides the communication medium while the nodes possess the ability to sense the state of the bus and transmit their data. A time division technique will be used and networks using such a technique are called baseband systems. That is, like a single-lane highway, only one message is passing at any given time.

2.1.2 CSMA/CD Approach

The advantages of the Carrier Sense protocol include a simple algorithm for operation of the network and almost no delays in a station access time. The disadvantages of this protocol include a limited cable length. However, this does not have a considerable impact in this project since only medium lengths of a few meters are considered.

The principal operation of Carrier Sense Multiple Access (CSMA), presented in [4] [14] [15] [32], consists of reducing the level of interference caused by overlapping packets by allowing stations to sense the carrier due to the other stations' transmissions, and aborting transmission when the channel is sensed busy. The additional feature of collision detection produces a variation of CSMA that is known as CSMA/CD. Because of its simplicity, CSMA/CD is perhaps the most popular contention- based protocol.

Although each station transmits only when it senses the transmission medium as being free, a collision with other transmissions may occur. This is because a transmission decision is made only on the basis of local information(i.e. the information that an interface sees at its interface), not on the basis of the overall state of the transmission medium. Figure 2.1 illustrates this operation.

In all the CSMA/CD protocols, given that a transmission is initiated on an empty channel, it takes at most τ seconds for the packet transmission to reach all nodes. Beyond this time, the channel will surely be sensed busy as long as data transmission is in process. The longer a segment, the more time it takes for a signal to propagate over it. To ensure the propagation timing limits are met, each media variety has a maximum segment length defined in the standard. A collision can only occur if another transmission is initiated before the current one is sensed, and it will take at most additional τ seconds before interference reaches all devices. Nevertheless, CSMA/CD has a collision consensus reinforcement mech-

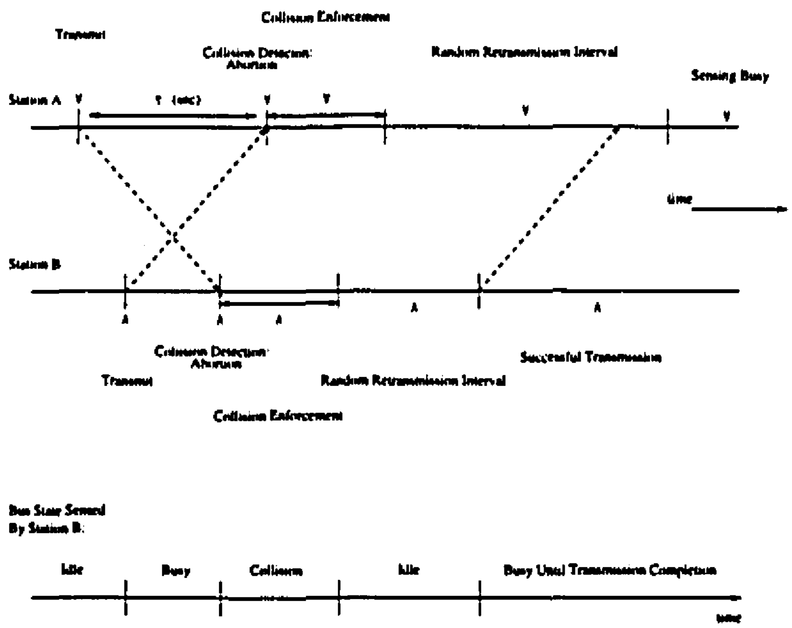

Figure 2.1: Example of CSMA/CD Bus Operation

anism by which a device, experiencing interference, jams the channel to ensure that all other interfering nodes detect the collision.

As a matter of simplification, it is usual to consider asynchronism of transmissions at the bit level; that is, if two stations happen to transmit at the same instant, their signals collide. The stations are notified of the collision, and reschedule their transmission at the next cycle. The collision will be detected within a one-bit clock cycle. To be appropriately corrected, it must be assumed that the bit time is long enough that each station can decide for its action (to transmit or stay idle) in the next clock cycle. Figure 2.2 illustrates the bit level asynchronization between two stations.

In general, a station is said to be backlogged if its packet either had a channel collision or was blocked because of a busy channel. On the other hand, as each local station is able to sense continuously whether the medium is idle or busy, the decision for transmitting or retransmitting data packets depends on the various schemes of the CSMA/CD protocol. To avoid consecutive collisions, each station involved should choose random time intervals to schedule their retransmission of the collided data packets. There exists numerous "retry

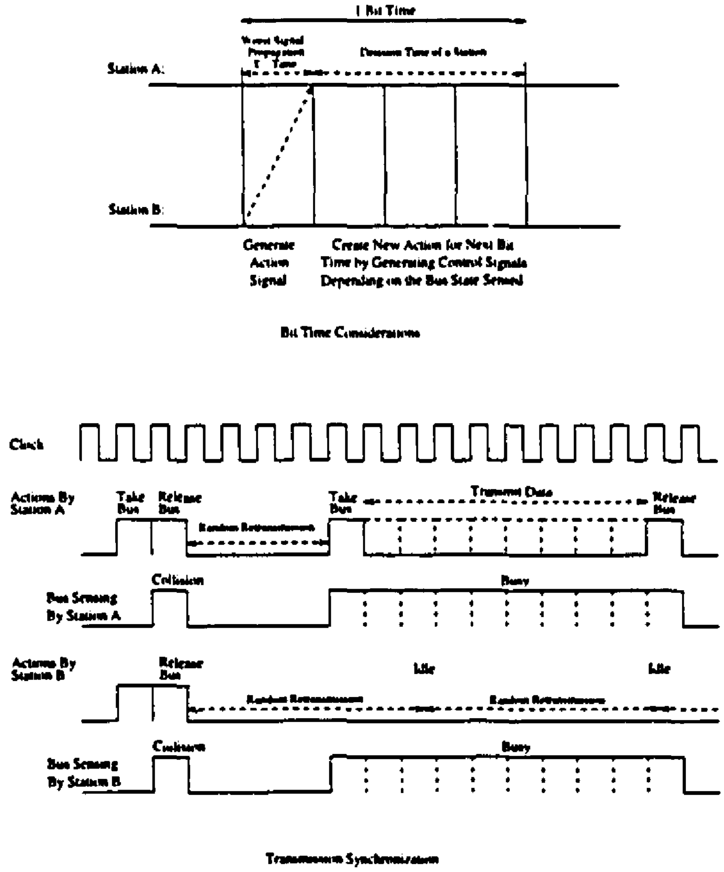

Figure 2.2: Asynchronism of Transmissions at Bit Level

strategies" which are useful once a collision is detected [6] [17] [19] [30] [35]. The most popular nodal schemes are: Non-persistent, One-persistent and P-persistent.

In the Non-persistent CSMA/CD scheme, a node with a packet ready for transmission senses the channel and acts as follows:

1. If the channel is sensed idle, the node initiates transmission of the packet.
2. If the channel is sensed busy, the node schedules the retransmission of its packet at some later time. It waits for a random amount of time and resenses the channel.
3. If a collision is detected during transmission, the node aborts its transmission and schedules the retransmission of the packet later.

In the 1-persistent CSMA/CD protocol (which is a special case of p-persistent) [30], a node that finds the channel busy persists in transmission as soon as the channel becomes free. If it finds the channel idle, it transmits immediately with probability one. In the P-persistent protocol, a ready node senses the channel and proceeds as in the Non-persistent protocol except that when the channel is sensed busy, the node persists until the channel is idle:

1. with probability p to initiate the transmission of its packet,

2. with probability 1-p to delay by τ seconds (end-to-end propagation delay).

The CSMA/CD mechanism operates on a bus-type network and is sometimes referred to as Ethernet protocol. Boggs [3] has discussed the different discrepancies that exist between the theoretical ideals and the reality of Ethernet operation.

The issues involve access, collision detection, and retry strategies. A simple technique would let the devices transmit randomly. Tobagi [32] has shown that the Non-persistent CSMA/CD provides better performance of throughputs versus traffic loads than the p-persistent and 1-persistent modes, as it is shown in Figure 2.9). However, a random delay strategy is considered non-adaptive since it operates on the premise that after a collision, the device retransmits at a time unrelated to and independent from the number or frequency of collisions. In fact, these problems (of collision rates) can further be generalized as minimizing the delay time for a node to transmit its packet within an allocated period of time (after which, old data is lost to the new data) while maximizing the data transmission rate on the bus bandwidth (capacity).

The analysis suggests that a network constructed with a bus topology would be more reliable when transmissions are adjusted to make the traffic fluid, and optimize the utilization of the shared resource.

2.1.3 Hardware Characteristics

For the design of the micro-network, a geographic scope of the bus could be defined as short as 10 centimeters or as large as 100 meters in length. The actual size conceived in this design is limited to be 5 meters in length. The objective behind this is to develop a flexible, portable and adaptable micro-network which can be applied to various applications in robotics or others as a tool for collecting data from a group of sensors.

Due to the restrictive working environment, the hardware components must be selected for implementing a prototype. The components mainly consists of twisted pair wires, diodes, transceivers and simple digital technologies.

Twisted pair have many advantages for the network implementation. Twisted pairs are inexpensive, have less weight and greater flexibility of installation and node relocation than other media. However, the disadvantages of using twisted pair can involve crosstalk and interference of signals due to the ambient EMF perturbations. Yet, when two conductors are twisted together (twisted pair), external noise is likely to to be coupled equally in each conductor. This noise injection can effectively be cancelled out at the receiving end, leaving only the transmitted signal. Those problems do not seem to be major due to the limited scope of the network.

Figure 2.3 illustrates the configuration of the network by using simple diode concepts. The logic is primarily based on the logical addition of current signals, which are quantized in levels for preventing eventual electrical breakdowns.

With this principle, a simple dual communication bus system may be realized. First, a single line status bus is used to code the state of the network: Idle, Busy or Collision. The states would be represented as follows: Idle on a logical Low Level, Busy on a logical High (the first level of signal quantization) and Collision on a Saturation levels (greater than the logical High level) which can be sensed by transceivers of analog types. The roles of the transceivers will be described below. In essence, the status of the network will be determined upon the transmission decisions of the stations; initially, the network is cleared to the Idle state. When no station transmits or writes to the bus, the state will stay at zero current

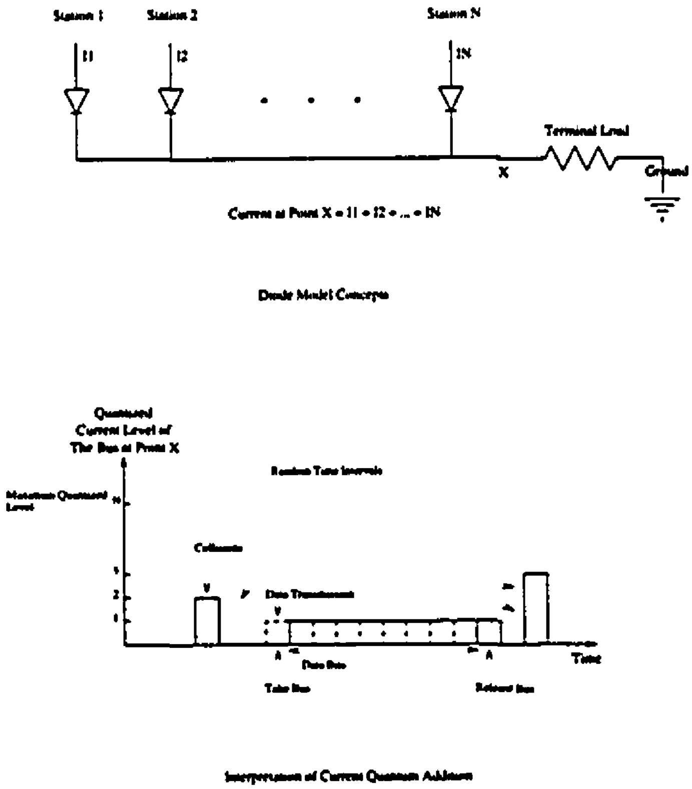

Figure 2.3: Illustration of Diode Model Concept and Principle of Current Quantum Addition

summation or Idle state; when exactly one station transmits, the current will be at High level which is the Busy state; when more than one station writes simultaneously, the network falls in a Collision state where current addition could become excessive. This condition is detected by the saturation circuitry from the transceivers. Secondly, the data bus is simultaneously synchronized with the status bus. When a station signals a transmission on the status bus, it also sends data information on the data bus.

Sensor data items are converted, stored and processed in some digital form. The data items are actually kept for a short fraction of a period at a local station and wait for their turn to be sent at most once during that period. The buffering capability of each station is implemented using digital technology. Two types of stations are modeled; one is for data transmission and the other for data reception.

The bus system status is continuously detected by the transceiver hardware from each

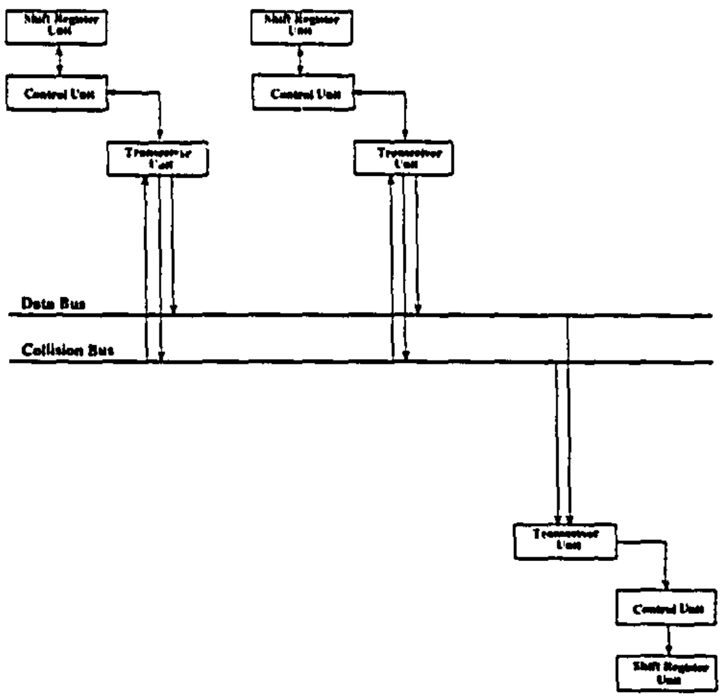

Figure 2.4: Overview of the Bus System

station. Two operation modes are found in the transceivers. Transmission-based detection monitors the DC current level of the network. Receive-based detection is dependent upon the DC persistence of the twisted pair network wire. The current level results from the summation of the numbers of stations transmitting. In the idle case, no station is transmitting. Exactly one station is transmitting when the bus system is busy, and more than one transmission results in collision. In all cases, transmitting and receiving nodes should be able to detect the current state to make further decision. Figure 2.4 is to show the overall bus system of the micro-network.

2.2 Distributed Architecture Models

When designing a distributed network, modeling the performance characteristics of a network before putting it to use is essential; it gives the designer the freedom and flexibility to adjust the various parameters of the network in the planning stage rather than in the operational phase. (It is, however, difficult to predict the performance of a network until a detailed analysis of a similar network becomes available.)

2.2.1 Mathematical Model

Analytical models are mathematical representations of the system which relate input variables to output variables. To make the mathematics tractable, several assumptions about the systems are usually made. These assumptions tend to reduce the accuracy of the models and may lead to misleading results. Analytical models may be difficult to develop, but they are still useful when gross answers are acceptable for rapid initial assessment because they show the qualitative relationships between input and output parameters.

Since data collisions are the major limitation to the channel capacity and network utilization, it is important to model them. For the sake of simplicity, the following assumptions are made concerning the characteristics of the individual stations and the operation of a CSMA/CD network system.

- Arrivals at all transmitting stations follow a Poisson Process, that is, interarrivals of packets are randomly distributed with exponentially distributed interarrival times.
- All stations generate traffic at different rates λ_i, and the mean arrival rate of the overall system is λ.
- Packet lengths are fixed and identical.
- The spacing between stations may be random.
- The propagation delay is estimated to be 25 nanoseconds for a network length of 5m.
- The transmission medium is assumed to be error-free, and errors are only due to collision.
- The receiving station has a service rate of μ.

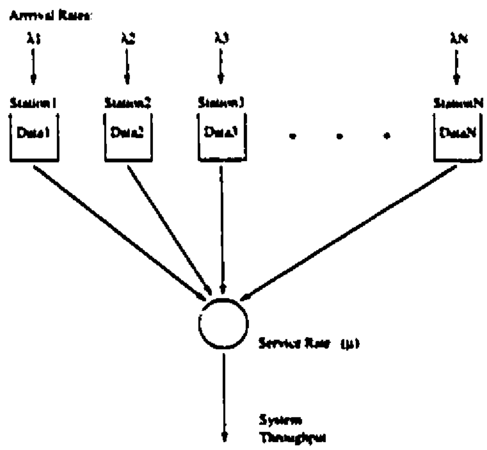

Figure 2.5: Illustration of System Model

This system can be viewed as illustrated in Figure 2.5.

Assuming that all arrival rates are identical and there are N mutually independent, nonsynchronized stations which communicate over a common broadcast channel, combinatorics can be used to define the mechanisms of collision probabilities. The probability of transmission is evenly distributed among the stations. Therefore,

$$P_i\{\text{a station transmits}\} = \frac{1}{N}$$

and the probability for a station not to transmit is

$$P_i\{\text{a station does not transmit}\} = (1 - \frac{1}{N}).$$

The probability that the bus channel is busy is seen as the probability that one out of N stations is transmitting while the rest remains idle. Hence,

$$P\{\text{channel is busy}\}^1 = \begin{pmatrix} N \\ 1 \end{pmatrix} P_i\{\text{a station transmits}\} \times P_i\{\text{a station does not transmit}\}^{N-1}.$$

$$P\{\text{channel is busy}\} = N(\frac{1}{N})(1 - \frac{1}{N})^{N-1} = (1 - \frac{1}{N})^{N-1}.$$

---

[1]By $\begin{pmatrix} N \\ k \end{pmatrix}$ we mean the combinatorial expression $\frac{N!}{k!(N-k)!}$.

The probability that the bus remains idle is:

$$P\{\text{channel is idle}\} = P\{\text{none station transmits}\}.$$

$$P\{\text{channel is idle}\} = \binom{N}{N}(1 - \tfrac{1}{N})^N(\tfrac{1}{N})^0 = (1 - \tfrac{1}{N})^N.$$

Thus, the probability that a collision occurs is then:

$$P\{\text{collision on the channel}\} = 1 - P\{\text{channel idle}\} - P\{\text{channel busy}\}.$$

For mathematical simplification purposes, the simplest model, the M/M/1 queue model, is considered to determine the statistical properties of the above system [2] [26] [11] [23]. The structure of the M/M/1 queue is presented in Figure 2.6.

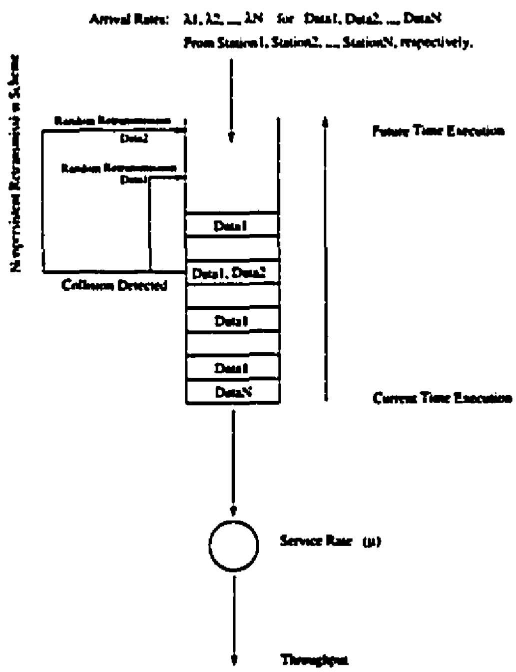

Figure 2.6: Illustration of the M/M/1 Queue Model

This M/M/1 notation refers to Poisson Arrivals, exponential service statistics and single server service, respectively. The analysis of the M/M/1 is commonly described because of its simplicity; the concept of a single server is well known from daily life such as waits at a supermarket, bank or movie house.

Let the following notations be defined as below:

P_n:	Steady-state probability of having n customers in the system
λ:	Mean arrival rate (inverse of average interarrival time)
μ:	Mean service rate (inverse of average service time)
N:	Average number of packets in the system
N_q:	Average number of packets waiting in queue
T:	Average packet time in the system
W:	Average packet waiting time in queue (not including service time)

P_n is the primary characteristics of the M/M/1 queue: P_n is the probability that there are n packets in the queue (including the one in the service). It is useful for determining the remaining statistical properties of the system such as the average queue occupancy, the average throughput, and so-forth. Figure 2.7 shows the transitions of the data populations existing in the system.

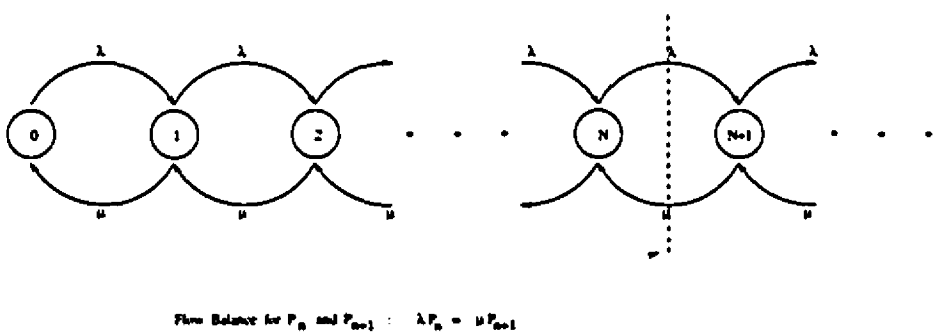

Figure 2.7: Transition State Diagram

The explanation is simple. The rate of packets entering state n must be equal to the the rate leaving state n. Hence,

$$\lambda * P_n = \mu * P_{(n+1)}.$$

By defining, $\rho = \frac{\lambda}{\mu}$, we obtain:

$$P_n = (\rho^n) \times P_0.$$

The only remaining unknown is P_0 and it can be solved by the normalization condition of probabilities:

$$\sum_{i=1}^{\infty} P_i = 1.$$

It thus results that $P_0 = 1 - \rho$.

The main results of M/M/1 systems are given in [2] [23] and can be summarized as:

1. Utilization factor (proportion of time the server is busy)

   $\rho = \frac{\lambda}{\mu}$.

2. Probability of n packets in the system

   $P_n = \rho^n(1-\rho), n = 0, 1, ...$

3. Average number of packets in the system

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

4. Average packet time in the system

   $T = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\lambda-\rho}$.

5. Average number of customers in queue

   $Nq = \frac{\rho^2}{1-\rho}$.

6. Average waiting time in queue of a packet

   $W = \frac{\rho}{\mu-\lambda}$.

This analysis can also be extended to the case of a finite population of data when the system can only contain a maximum number of packets, which is equal to the number of active stations in this design. Blocking probabilities will be useful in the sense that during a certain amount of time, the system can only serve (or transmit) a finite amount of packets. The remaining packets will be blocked, backlogged and will eventually be lost. Given that a duration of time for which an average number of packets, N, can be served by the system, the blocking probability for which the system can not satisfy any further transmission requests is given by:

$$P_N = \frac{(1-\rho)\rho^N}{1-\rho^{N+1}}.$$

Figure 2.8 illustrates the effective throughput of a system.

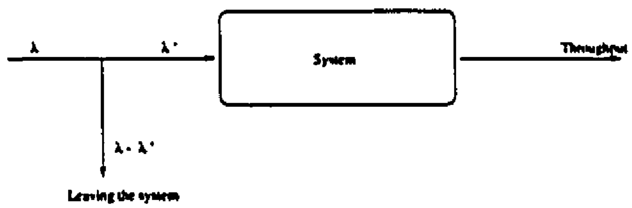

Figure 2.8: Actual System Throughput

By definition,

$$\begin{aligned}
\text{Utilization of the system} &= \text{fraction of time that the system is busy} \\
&= \text{Prob(system is not empty)} \\
&= \text{1 - Prob(system is empty)} \\
&= 1 - P_0 = 1 - e^{\frac{\lambda}{\mu}}.
\end{aligned}$$

In association,

$$\begin{aligned}
\text{Throughput of the system} &= \text{utilization of the system} \times \text{mean service rate} \\
&= (1 - e^{\frac{\lambda}{\mu}})\mu.
\end{aligned}$$

From Figure 2.8, it can be seen that, for a system with a mean data arrival rate λ, data packets are lost at a rate of $(\lambda - \lambda')$. By conservation of packets in the system, $\lambda' = \mu(1 - e^{\frac{\lambda}{\mu}})$. Therefore, the average rate that packets leave the system without being served is:

$$(\lambda - \lambda') = \lambda - \mu(1 - e^{\frac{\lambda}{\mu}})$$

More complex systems can furthermore be developed from the M/M/1 model. The M/M/m system designates the multiserver exponential queueing system with m servers.

There, the Erlang distribution and the blocking probabilities are used to approximate the collision and service statistics. The concepts of data service classes will also be introduced with respect to the different service rates from the m servers.

Lam and various authors [2] [14] [26] have also extended this analysis to a generalized case of CSMA/CD systems. The CSMA/CD mechanism is referred to as a M/G/1 system model which treats the single server of general type.

Although theoretical studies have examined several operating regimes, these are not characteristic of actual networks, and their results are also of limited utility in comparing network technologies. On the other hand, software simulations can be used in experiments to see the intrinsic performance of a network, where the theory can not. Collisions are statistical and interdependent with many factors like packet size, node count or average transmission rate. Non-linear methods in software programs are sometimes required to discretize the system in order to understand (in more depth) the transmission mechanics.

2.2.2 Systematic/Protocol Model

Using simulations, a network may be modeled to any desired level of detail if the necessary system relationships are known. Many authors had to resort to simulation of their system models [36] [33] because those simulation models can give more accurate results than analytical models and most of the assumptions (in the analytical model) can be relaxed. However, the level of detail dictates the simulation runtime. This can make detailed simulation slow and expensive.

Better information on performance and functionality of a data communication network, will then be applied and analyzed with a suitable model and simulated in software. In chapter three, a simulation model will be implemented to study a Non-persistent CSMA/CD protocol. In such a model, data must respect some synchronisation constraints at the bit level. For a distributed context, all correlated data are categorized by classes of sensors and must be acquired within the same time interval for the same class of sensors.

In general, an optimization problem must be solved to increase the network performance (utilization) in global terms. From a local viewpoint, the data must be transmitted successfully at most once without being lost during its transmission period, and with a minimum delay time. In general, a network used for most applications must appear as an "ideal" information channel without loss of information. In classical networks, the "delay" notion is very subjective and must be referred to as "reasonable delays".

2.2.3 Hardware Model

The hardware model implements the functionality of the design while respecting the physical limitations imposed on the network. It demonstrates the operational feasibility of local nodes as well as of the global traffic control.

Each station in this network is modeled as a finite state machine and associated to local variables. Associated to each transition in the machine is an action, which may alter the variable values. Enabling predicates determine whether a transition may be made. Actions are taken when the transition is executed. The action assigns new values to some subset of the variables.

Using this model, we studied the system stability behavior, the throughput and the average packet delay performance. This model is implemented from a hardware design viewpoint in chapter four.

2.3 Performance Analysis

To demonstrate suitable functionality and operationality, a network must detect and remedy collisions quickly and compensate for them before system performance degrades to an unacceptable level. Hence, various performance analyses are studied to resolve such degradation of the networks.

First, the input/output parameters of the system will be defined. The performance values that describe the character of CSMA/CD networks will then be presented.

2.3.1 Definition of Parameters

Often traffic on real networks is seldom so well-behaved because there may be brief-burst of high loads separated by long periods of silence. For mathematical conveniences in modeling, most systems assume a simple distribution of Poisson arrivals. The performance of the network depends on a variety of parameters, in addition to those fixed by the specification. These include the following:

• Bit rate	(10Mbps)
• Bit Asynchronisation	(25ns/5m propagation delay, 75ns jam time)
• Fixed packet size	(up to 256 bit packet length)
• Number of nodes	(of the order of 100/5m)
• Non-persitence	(random retransmission time)
• Arrival rates of packets	(1Hz - 16MHz)
• Distribution of packet arrivals	Poisson arrivals

The performance of a network cannot be quantified with a single dimension of parameters; theoretical studies have tended to examine a small set of measures, both for analytic tractability and because it is hard to decide what to measure without knowing what applications are involved. These measures are the following:

- average delay: the average time it takes to send a packet, measured from the time the host first wishes to acquire the channel.
- throughput: the fraction of the nominal network bandwidth that is actually used for carrying data. packet headers are considered useful data in calculating this value.
- channel capacity: the maximum achievable throughput for a given set of parameters. Capacity is function of such parameters as packet length and network length.
- fairness: in a fair network, each host with pending traffic should have an equal probability of acquiring the channel.

- stability: if the throughput actually drops at high loads then the network is said to be unstable in that region.

The performance measures are usually described as a function of some parameter of the model; offered load is the most common. Several different definitions of offered load are used in the literature. Here is a definition of an offered load:

The offered load at each host is the fraction of the network bandwidth that the host would use if it had complete access to the network, then G, the offered load on the network as a whole, is simply the sum of the offered load at each host. Each host's offered load on the network is less than 1, G can therefore be greater than 1, although the throughput cannot. The common strategy is to overload a network for more than its capacity and analyze how well it can respond. Criteria of performance can vary from applications and importance of measures can differ. However, the offered load is important in this project, in terms of the number of stations in the system as well as their frequency of transmissions, so as to test the various configurable working conditions of the network [22].

2.3.2 Measurements

A number of criteria, useful for evaluation are given below. Their respective importance depends on the constraints put on the systems. Those usually include the following:

- Response time and throughput.
- Convergence, stability and fairness.
- Constraints as regards utilization.

In ideal cases, the CSMA/CD models must yield intuitively efficient performances as shown in Figure 2.9. This shows in summary that, although CSMA/CD is quite workable, as network traffic becomes heavier, the chance of a collision increases and throughput degrades as the collision detection mechanism (and consequent retries) come into frequent operation.

Bux et al., [4] [7] [17] [24] [30] [32], have also shown the limitations of those CSMA/CD models.

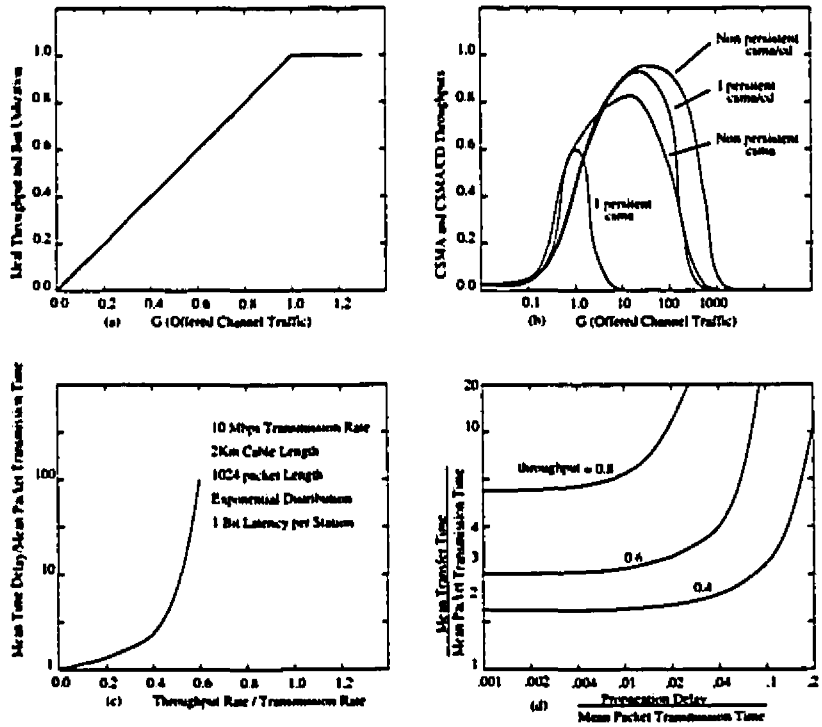

Figure 2.9: Measurements of CSMA/CD networks

CSMA/CD is a contention-based protocol. Systems under the CSMA/CD protocol can exhibit unstable behavior due to retransmissions of packets. Consequently, to evaluate the system performance precisely, we will consider the stability problem by limiting the number of nodes. Each can only send at most one packet during its transmission period.

2.3.3 Example: Ethernet

This subsection presents measurements of the behavior of an Ethernet as an application to CSMA/CD [3] [21] [27] [29], under varying combinations of packet lengths, number of hosts, period of transmissions, and sleep times.

When many hosts are waiting for a long packet to go by on a busy Ethernet, the instantaneous load on the network routinely exceeds its capacity for short periods. Stressing

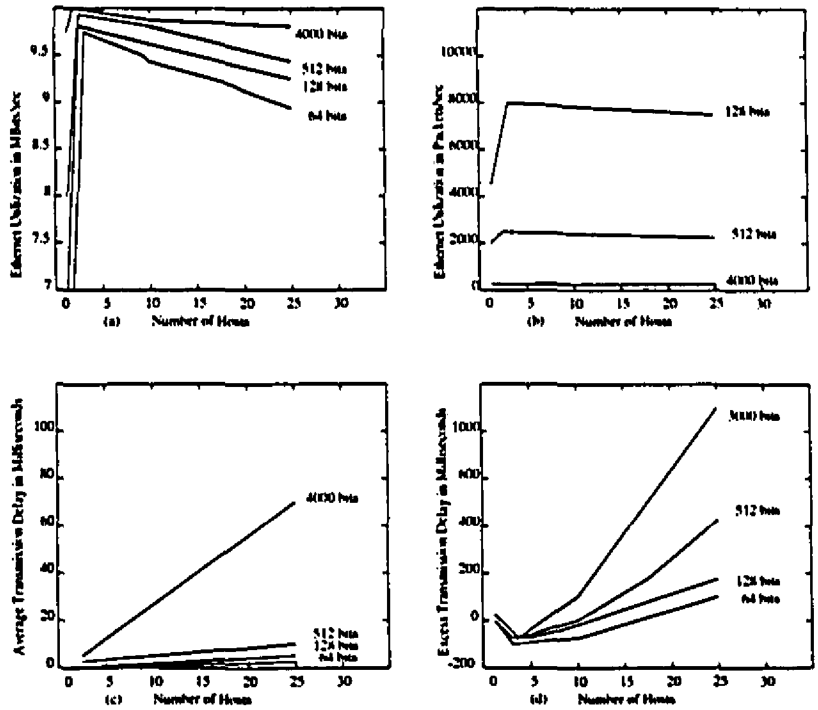

Figure 2.10: Measurements of Ethernet

an Ethernet with an unusual steady overload yields insights into how the networks handles the usual momentary overloads encountered in normal operation. Therefore, in these tests, attempts were made to generate total offered load that continuously exceeds the capacity of the network.

The statistics and modeling confirm the practical limits that network designers have learned by trial and error. Figure 2.10 shows the measures made on Ethernet networks. As an actual fact [3], it is usually suggested to plan for a maximum of 30 nodes per Ethenet channel when most of these nodes are diskless and require a client/server support. Most single networks tend to support a maximum limit of 60 nodes, and try to avoid subnets with more than 30 nodes, particularly when the nodes utilize frequent remote data transfers. A fully populated Ethernet (more than 60) is fine if the network is sparsely used, or provides for only minimal functionality. When saturation is imminent, the network must be able, however, to provide adequate alternative channels or clear all nonessential traffic from the channel for a manually employed priority enforcement.

It would be interesting to explore different approaches for achieving this asynchronous CSMA/CD operation and the effect of these approaches on the time-constrained performance of the protocol. We will test the sensitivity of our former assumptions as well as numerical parameters by comparing subsequent simulation and operational results against results based on existing network analysis as well as the mean values analysis. On the basis of the analytical results that will be obtained, we will also examine the effect of retransmission strategies.

Chapter 3

Micro Network Protocol

3.1 Protocol Definition

A bus system is a trucking system that carries packages of data between stations with no regard to what is inside the packages. Just like any traffic control however, chaotic processes are generally not desired and certainly not permitted for data communications. Some convention or protocol needs to be set to control the data flows on the bus system.

The component structures are first described to conform with the system architecture. The variables associated need to satisfy the hardware characteristics described in chapter two and also, to obey the rules of the protocol which will be implemented. The algorithm of the protocol is then explained afterwards.

3.1.1 Carrier Sense Multiple Access with Collision Detection Modeling

There exist different kinds of protocol that can be defined for a given network. Under this project, a CSMA/CD based protocol is used to establish the control mechanism or the set of conventions via which data elements are transmitted. Although CSMA/CD is a well-proven and effective network access technique, it suffers from inherent deficiencies which limit its effectiveness as traffic densities approach the limit of the transmission capacity and which make it difficult to implement in a low-cost reliable, high-capacity baseband system. Hence, the intent of the simulations is to enhance the effectiveness of the data communications applications in reality and maintain their performances at respectable values.

A simulation model was developed in the C programming language to specify a Non-persistent CSMA/CD protocol as described in chapter two. This model is used to evaluate the performance of the distributed protocol for a data acquisition micro-network and analyze the liveliness for a large number of users sharing a common serial bus system. The modeling first consists of describing the structures of the entities involved. These include multiple transmitters, a single receiver and the shared bus system. Once those structures have been defined, the operation of the protocol can be interpreted as a computer program in which those entities interact with each other by altering/modifying their variables. The results of the simulation model is shown ulteriorly.

3.1.2 Bus Structure

The bus system is composed of a data bus line and a status bus line. The data bus line is to carry the the data being sent by a station, and a status bus line to inform the present state of the bus occupancy.

The structure of the data bus is defined as follows:

```
struct DBusType {
  Mesg  Packet;
};
```

A message packet here contains the identity and the data of the transmitting station. It has the following structure:

```
struct MesgType {
```

```
  int            addr;
  int            data;
};
```

Here, the data bus only carries the data packets and a packet designates the contents of the message that are sent from a transmitting station.

The structure of the status bus line is:

```
struct SBusType {
  int            Level;
};
```

The status of the bus is defined to give the actual number of nodes transmitting on the bus system; the level will state that the bus system is idle by a 0, only one node is transmitting on the bus by a 1, and a collision has occurred for any higher integer, indicating the number of simultaneous transmitting stations. Collisions are hence modeled by simultaneous writes to this variable.

3.1.3 Transmitter Structure

Each station in the network is modeled by a finite state machine and associated to its respective local variables. The structure associated to a station is presented below.

```
struct DeviceType {
  int            DeviceId;            /* device  specifications */
  int            Start;                 /* starting time      */
  float          SleepRate;             /* sleep time         */
  float          TransRate;             /* transmission time */
```

```
  float        sleep_time;            /* communications times   */
  float        traffic_time;
  int          status;                /* bus status flag   */

  int          jobs;                  /* device statistics */
  int          unsent_jobs;
  float        efficiency;
  float        avg_latency;
  float        total_latency;
  float        worst_latency;
  float        time_ping;
  float        time_pong;

  pMesg        Packet;                /* device data memory */
  pDevice      Next;
};
```

Three main parameters define the autonomy of a station. The Start variable indicates that a station begins its data communications process until the end of simulation time. The TransRate variable is expressed in terms of number of clock cycles (or bit time) and is used to define the data refreshing cyclicity. That is, at every beginning of a transmission period, a station updates its data with a new one and requests to transmit it. As a simple illustration, we take the example of 'incrementing data':

```
  ...
     if ( ((d_time - s_device->Start)%period) == 0 ) {
              (s_device->Packet)->data++;

              ...
     }
```

For simplification, we designate the transmission periodicity by T, during which a station has to perform at most one successful transmission with its actual data. When the bus is found busy or enters a collision state, the `SleepRate` variable, simply denoted by S, is used to define the time duration that a station must wait until a new transmission request is issued or a new transmission period begins.

The current state of a station is described by the variables `sleep_time`, `traffic_time`, and `status`. The `status` variable indicates the status of the bus system that a station senses. This would allow a station to make a decision or take an action depending on the bus state and the three other parameters presented above. The `sleep_time` variable is used to specify the amount of time a station is held in its awaiting state. When a station is inactive, idle or transmitting, this variable equals zero. The `traffic_time` variable denotes the duration required by a station to send its data packet. The cost of accessing the bus and releasing it is one bit time each and is not included in the `traffic_time` parameter.

As the simulation evolves, the course of the variables in a station varies. Correspondingly, each transition of those variables results into an action. Each station may further alter its own states and variables as well as the state of the system bus. Therefore, it would be interesting to keep track of some useful characteristics of each individual station such that when put together, the global system performance can be estimated.

For each individual station, the set of performance attributes includes:

- the number of jobs sent (successful transmissions);
- the number of jobs unsent (no success in a transmission period);
- the average latency (average delay time before a transmission);
- the worst latency (worst delay time);
- the efficiency (the ratio of jobs sent over the maximum number of jobs that the bus can allocate from its capacity).

3.1.4 Receiver Structure

On the other hand, the receiver is seen as a buffer, remaining idle when the status bus is sensed idle, receiving the transmitting data when it has sensed that only one device is transmitting, and capturing erroneous data when there is a collision detected on the status bus. The main point, however, is not to deal with the data contents at the receiving end, but rather, the objective is to gather characteristic statistics of individual stations to relate the system performance and response to the variety of applications using the micro network protocol.

3.1.5 Protocol Implementation

The concurrency arising from sharing the common bus among N distributedly connected stations need to be resolved by mutual exclusion. That is, at most one station at a time can have access to the bus. In a busy-waiting solution, stations attempt to make progress or re-attempts at a later time.

For simulation purposes, the protocol response to parallel processes is treated in a sequential manner for each station over a discretized time interval. The procedure is then repeated for every station before time is incremented by one time quantum. A flow chart is given in Figure 3.1. It shows the logical steps needed to process a single station request at a discrete time t.

The control for data communications is dependent on two points of view; global view and local views. In the global view, the status of the bus must be known to all stations. If the state denotes Idle, then the bus may process a bus grant to a device. If it is Busy, then the bus is processing data communications to one of the granted stations. On a colliding bus, the bus is accessed by many stations at the same time and need to be freed as soon as possible on the next clock and all transmissions involved must be aborted. The process of transmitting a data packet involves sensing the global state of the bus, while it also depends on the local views of each stations. On the local sites, a station must know if it has

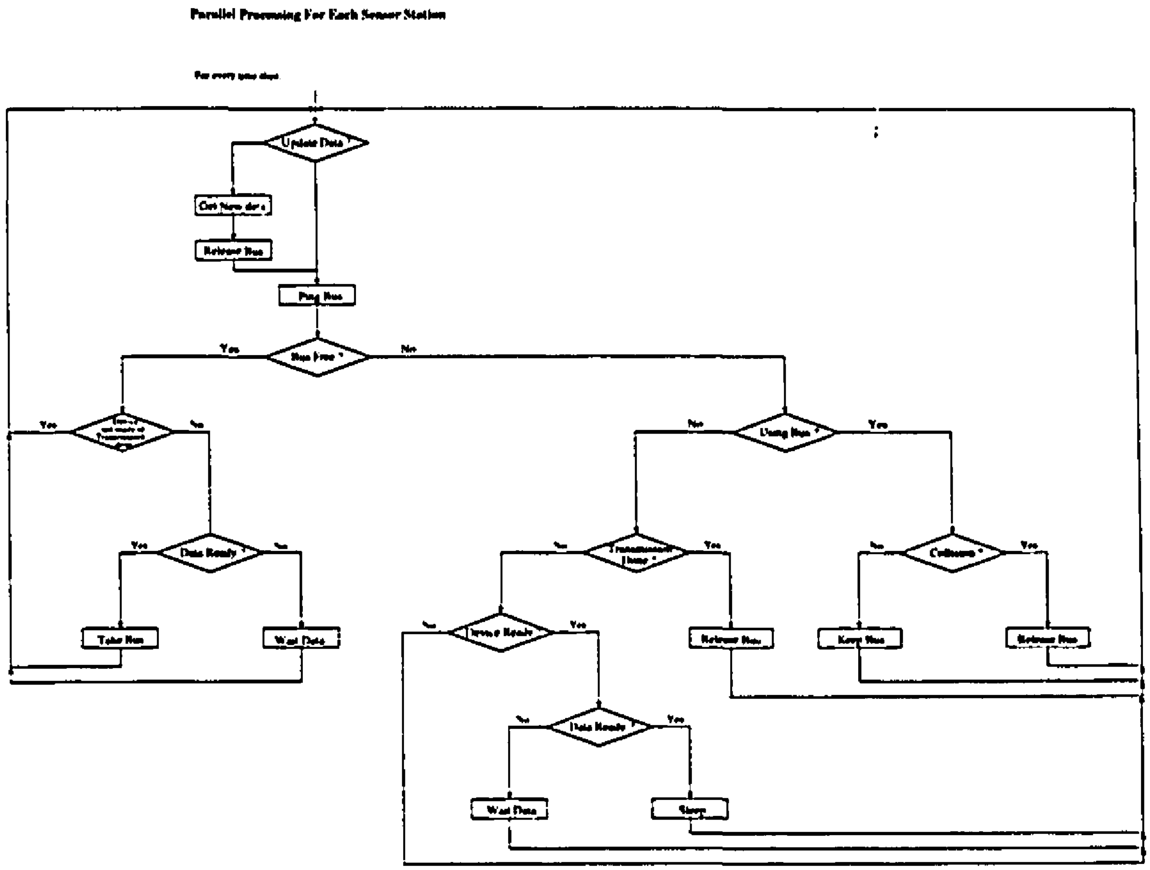

Figure 3.1: Flow chart of the protocol algorithm

to proceed a transmission request, if no data has been transmitted for a given transmission period, or whether its data is available for transmission before a new period begins. The program flow thus starts by updating data if a device reaches a new transmission period, and then proceeds by sensing if the bus state is free.

If the bus is free, a station will determine whether it must initiate transmission and take the bus. If not, that station either has its job sent or must wait for its data to be available for the new transmission period.

If the bus is not free, and depending on the local situation, five possibilities can occur:

1. If the station had completed a transmission, it stays idle.

2. If it requests transmitting, it must wait for its new data if there is no time to transmit the old one.

3. If it requests transmitting, it must wait for the bus to be free while its data is available.

4. If a collision is not sensed and the bus is not freed, the station is using the bus and must continue with its transmission.

5. If a collision is sensed, then it must release the bus immediately.

Figure 3.2 shows the different action states of a station before the bus updates its state.

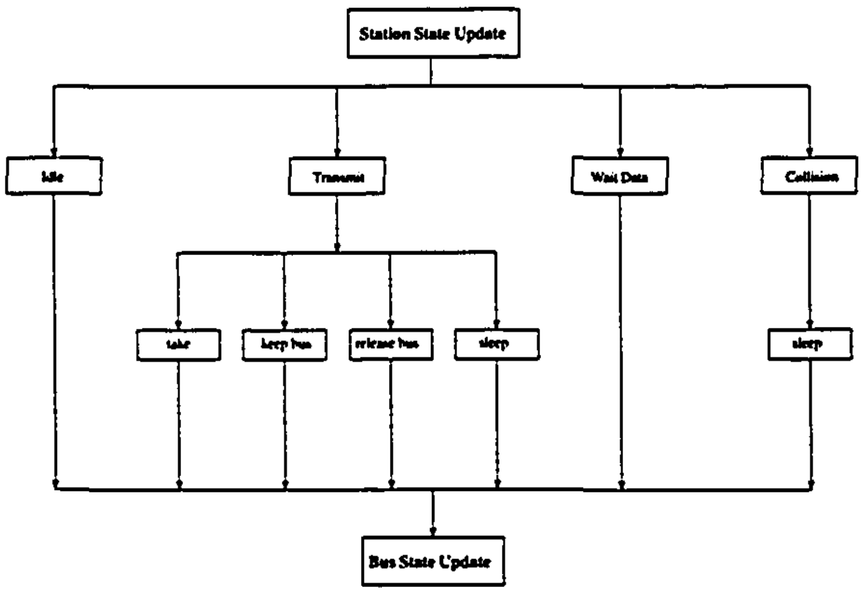

Figure 3.2: Action states of a station

In the next section, we describe the assumptions used to define the traffic population and the control characteristics. To avoid accumulation of retransmissions, in other words, to achieve stability, the population size must be restricted to avoid traffic overload. Traffic control schemes also allow dynamic adjustments of the traffic.

3.2 Traffic Control Handling

Traffic control is a key element for obtaining good performance in terms of delays, throughput and bus utilization of a system. A priori considerations involve the traffic population as the input of a system. The size of the population in a system can affect it considerably to make the traffic flow fluid, dense or jammed. During the run-time operation, some procedures must be introduced to improve the situations where data is congested over a shared communicating route. Congestions in system intuitively suggest to define some prioritization schemes in order to favor communications of relevant data over less important ones.

3.2.1 Traffic Processing

The traffic is determined by the number of stations that are connected to the bus and their activities. The number of stations in a system can be large and varies according to their contributions to the system. For instance, sensors that generate critical data may be multiply connected to the network, which allows a multiple diffusion of those variables to increase their availability.

The network behavior is totally controlled by local activities and there is no global controlling or monitoring task over the bus. The behavior model of each station is oriented to the basic activities which reside at each node of the system. These basic activities have attributes such as period, waiting priority rates, packet length, and so on. The transmission activities of these stations altogether establish the traffic flux of data on the network. In real-time operation, transmission activities are only visible at the local nodes. With fast transmission rates, the traffic population on the bus can grow rapidly. With low rates, traffic becomes less dense.

The fundamental problem, when the traffic becomes overloaded, is to improve the performance of the system and the utilization of the bus.

Sharing the common bus evenly is important for efficiency. Communication must be balanced so that all devices can communicate in very short time frames, preferably in a small number of bit time.

The smallest time that elapses between detection of a collision and detection of a clear channel is one bit cycle unit time (the idle detection time). Once a conflict is detected, transmissions are not processed entirely for the whole packet length; all transmissions cease at the next clock cycle. This greatly reduces the waste of the bus bandwidth and hence improves the throughput of the network. Also, the CSMA/CD Nonperistent scheme allows one to defer concurrent transmissions to a later time and regulate the traffic more efficiently.

3.2.2 Collision Resolution Procedures

The system must provide for a means to detect and remedy traffic failures, or else it no longer has a fluid flow and becomes jammed, leading to a degradation of the network performance. The problem we now want to address is how to select the alternatives for operation that will minimize the amount of messages with delays below a given time constraint.

One of the main advantages of the proposed protocol lies in the fact that collisions can be detected and aborted by some specified policy, before they waste too much bandwidth on the channel. The usual approach for finding the optimal policy is to choose some initial policy and then to tune it for better policies.

Whenever a collision does occur, a collision must be resolved by rescheduling the colliding stations, with the hope that a better distribution of packet transmissions can be created. This can be accomplished in several ways. A simple way is to reschedule attempts of transmission at random instants. In general, a retransmission is attempted according to the type of station, set by its sleeping time. The sleeping time may be fixed, however, and a dynamic adjustment can be created by adding a small random variation to the fixed reschedule time, to avoid the repeated collisions from the same stations. A fast response time is important for successful implementation of error confinement.

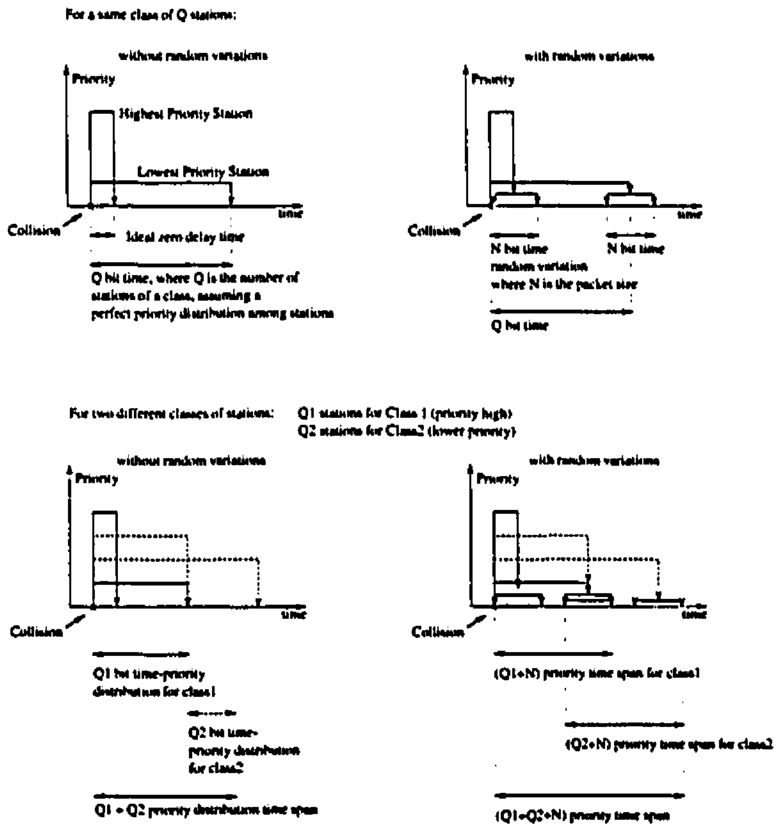

Figure 3.3: Priority classification

The protocol developed can be extended to data communications with different classes of devices at same transmission speed. In the case of data collisions for identical devices, the best retransmission scheme is to grant randomly re-access time with minimum delay, that is, zero delay time. When there are several classes of devices, an intuitive approach is to reschedule the lowest priority class with zero delay. The next class will reschedule a transmission after a Q-bit bit time (where Q is the number of stations in the class) so as to let one of the station from the lower class win access to the bus. After a Q bit time, stations from this class still have to compete with the stations from the lower class (with higher priority) for access to the bus. Of course, random variations will have to be added to resolve the fairness problems among the stations. In simulations, the smallest delay time considered is the N packet bit time, plus the number of devices from the lower classes in an effort to "spread evenly but randomly the priorities" among the stations. Figure

3.3 illustrates the operations. This scheme will be used in simulations wherever random retransmissions should be involved.

The problems of the network, once understood, are quickly recognized from their symptoms. A poorly functioning network results in high collision rates. Stations need to be tuned or removed to match the capacity of the bus.

The sampling frequency must be selected in a global way to make full use of the bandwidth of the network, and also locally at a node site so that a node does not have to sample uselessly. The data sampling rate must be just necessary for the consumption rate at the receiver end. The receiving node(s) can perform a better productive task with the specific transmitted data. A certain amount of message loss is usually tolerable. The population of retransmitted messages can be reduced to improve better performance of the network; messages which are not successfully transmitted within a certain amount of time are considered lost, regardless of whether or not it is eventually received at the destination station. These factors greatly reduce the waste of the bus bandwidth and hence improve the throughput of the network. All of the above issues must therefore be taken in considerations in order to understand the simulation results and the behavior of the network.

3.3 Protocol Simulation

The simulation has three major sections: initialization, control processing and output.

In the initialization section, the structures for transmitting stations are created. Input parameters are read and variables initialized to their appropriate values. All the states of the stations, as well as the state of the bus need to be reset. The next step is to run the simulation over a time span. An transition event may occur as illustrated in Figure 3.1 for each station during one clock cycle time. The program executes the event and updates the values of all the variables affected by the event. This process continues until a time limit is reached. All relevant data is recorded.

3.3.1 System Parameters

In real-time applications, different kinds of information producers are found, each one with different communication needs: Data refreshing rates, synchronization requirements, etc. Many actual systems are so complex that it is desirable to resort to simulations for investigation of its properties. It is possible to use simulations to solve problems in a systematic way, by varying the grain size of the simulation. Here, the grain size of interest in terms of system performance for networks often include channel utilization, packet delays and transmission efficiency.

As system parameters, it is reasonable, for this project, to set the bus capacity C, to 10Mbps for reasons of signal propagation delay time as was presented in chapter two. Each station is given a transmission rate (T) and sleeping rate (S). The length of the packets is fixed for every station and it is kept short, since we would like to achieve rapid information transmission from distributed sites.

The population of transmitting stations is defined by the number of stations, which are divided in various classes according to their transmission rate. Typically, slow stations have their transmission rates in the range of 20Hz to 100Hz, medium fast stations in 100Hz-1Khz and fast stations above 1Khz range. In simulation, stations of a particular class are given identical transmission rates and identical wait times. To simulate the traffic operation efficiently, transmission loads are randomized to spread over time. For a homogeneous class of stations (that is, with an identical transmission speed), small random variations in wait times is used to eliminate any deadlock conditions. In the next subsections, simulation test results will show the effect of varying the mentioned parameters and their importance will be discussed.

3.3.2 Performance Measures

The primary performance criteria are system performance (or throughput) and delay as a function of the traffic load. On the other hand, the bus utilization (U) is defined as the ratio of the number of successfully transmitted packets over the maximum number of packets that the bus capacity can handle.

$$U = \frac{number\ of\ successful\ packets \times packetsize}{capacity \times simulation\ duration}$$

Another important measure is the accumulated transfer delay (ATD). The accumulated transfer delay is the time interval between the instant the packet is available at the sending station and the completion of its successful reception at the receiving station. When a station is unsuccessful at sending a packet during a transmission period, the mean transfer time is not equal to the time duration of the transmission period. The delay time is accumulated at every new transmission period. This quantity is expressed in terms of bit time, a quantum of time of 100ns for 10Mbps bus capacity rate. The variable measures below are recorded in the simulation.

For each individual station, the average latency time (L) can be derived from the measure of their respective ATD:

L = ATD / (number of transmission periods).

We also define the worst latency (WL) for a single station by:

WL = Max (L) as L varies throughout the simulation.

In the worse case, the average latency could be equal to the transmission period of the station in concern. The efficiency (E) of each station is also measured:

E = number of successful transmissions / number of transmission periods.

Overall, the system can be quantified by the following measures:

Overall Average Latency (OAL) = Sum of latencies of each station / number of stations.

Overall Worst Latency (OWL) = Sum of worst latencies of each station / number of stations.

Overall Accumulated Efficiency (OAcE) = Sum of Efficiencies of all the stations.

Overall Average Efficiency (OAE) = OAcE / number of stations.

The characteristic changes of those measures will be produced with respect to the type of traffic population presented to the system. The outcome of the changes will be presented in the next section.

3.3.3 Simulation Results

The micro network protocol was shown to operate appropriately to solve mutual exclusions among simultaneous transmission, and to avoid deadlock states by using random rescheduling. The results of the simulations are varied; depending on stations population profile in the network system.

In order to achieve the highest performance and maintain at the same time the desired flexibility in a system configuration, the trade-offs involving the system must be well-understood. The first series of simulations shows the variations of stations with and without random rescheduling. Here, T = 1KHz, S = (64 bits length to distribute stations' priorities over bit time + 8 packet bits margin = 72 bits) = 140KHz (approximately), N = 8 bits.

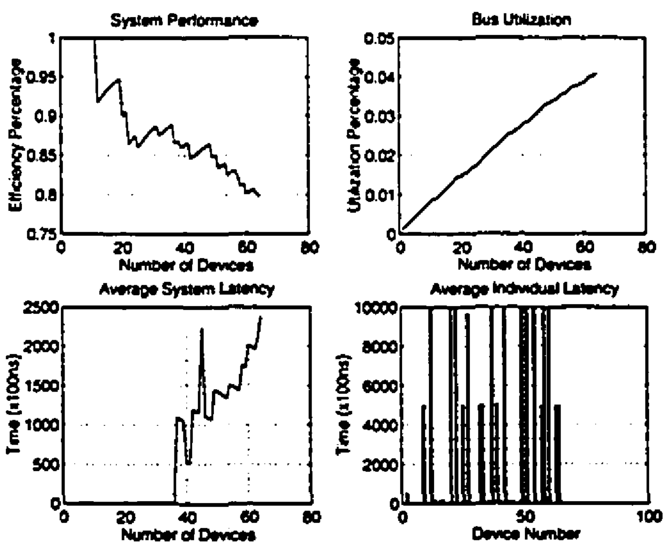

Figure 3.4: Protocol Simulation without random rescheduling, T = 1KHz, S = 140KHz, N = 8 bits.

Figure 3.4 shows the simulation without random rescheduling. On the x-axis, the system is populated first with slow transmission hosts and then, is filled with faster ones. As presented in the previous section, the system performance is defined by the ratio of successful transmissions over the total number of transmissions requested in the system. Here, when the number of stations in the system is less than about 10, the system performance stays at 100% without any data loss per transmission request. From there, repeated collisions resulted in data loss in the system causing the system performance to drop about linearly. The bus utilization U is linearly increased by two factors; first, by the number of hosts in the system and secondly, the frequency of transmission. The average system latency is to say the average time delay for a station to transmit one packet of data among the other stations. Ideally, this variable should be close to zero. When the number of stations is small, this value is practically zero. However, as the number of stations increases, this value also increases rapidly. On the individual latency graphs, some stations experienced starvations and hence the maximum latency allowed which is the duration of a transmission period (1,000,000ns or 1KHz). The discrepancy is also enormous between the stations, leading to inequitable transmission rates between stations.

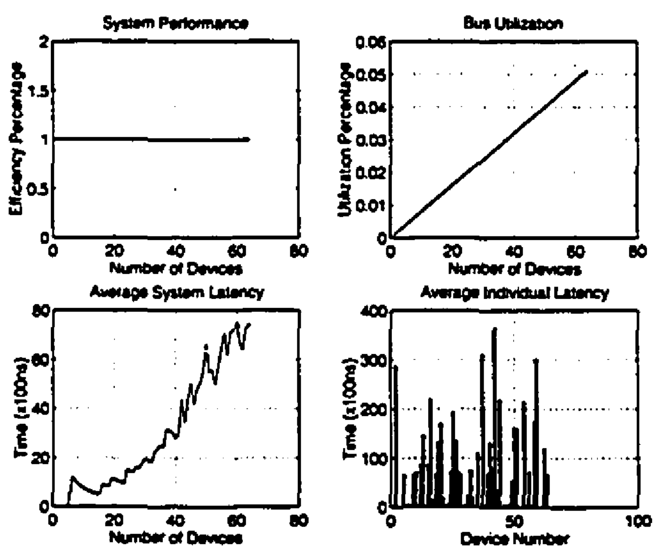

Figure 3.5: Protocol Simulation with random scheduling, T = 1KHz, S = 140KHz, N = 8 bits.

Figure 3.5 shows the above simulation varied with random rescheduling. In general,

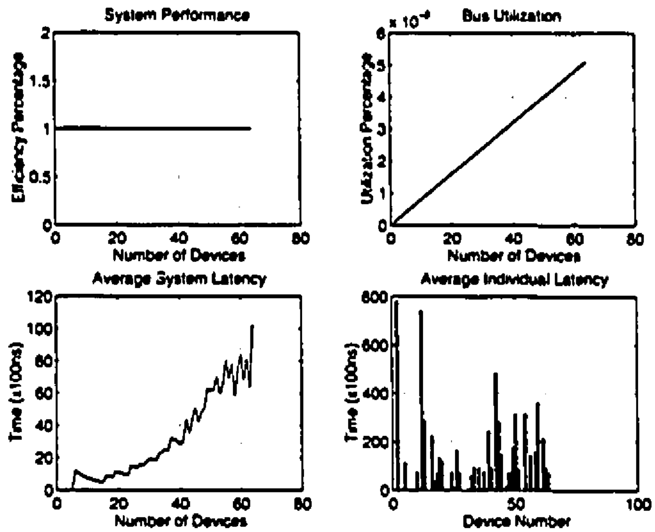

Figure 3.6: Protocol Simulation with T = 100Hz, S = 140KHz, N = 8 bits.

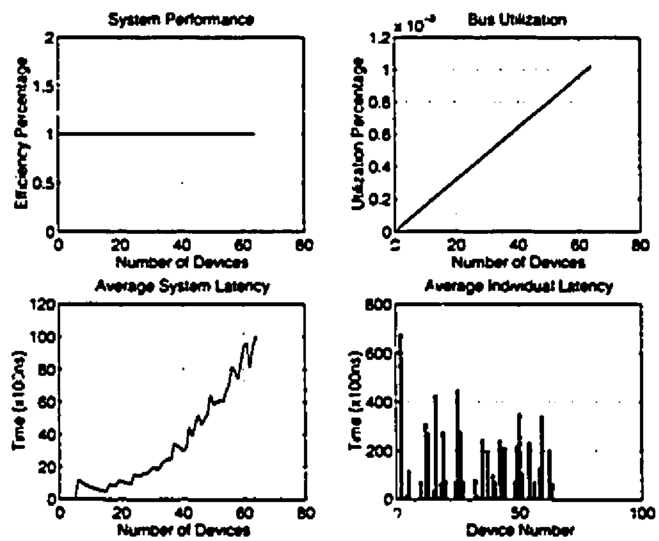

Figure 3.7: Protocol Simulation with T = 20Hz, S = 140KHz, N = 8 bits.

random variation is to add higher probability for all stations to be successful in their transmissions. By doing so, better transmission distribution can be obtained over the total bandwidth. It also hence reduces the problems of fairness and starvations among the stations. As compared to Figure 3.4, the results indicate that the stations are successful in transmitting their packets with higher rates and in faster response time. The system performance stays at 100% efficiency as the system has no data loss. The average system latency and individual latency is also much reduced (to the order of 100ns instead of 2,500ns or more). The average latency has however an exponential growth versus the linear increase of the number of host stations. The outcome of this simulation is comparable to Figures 2.9c and 2.10d. Individual average latency shows the average delay for each specific station to transmit a packet. This variable gives an idea of fairness of a station among the population in the system.

Figures 3.6 and 3.7 varied the previous simulation with lower transmission rates, with T = 100Hz and then T = 20Hz, respectively. S = 140KHz, N = 8 Bits. With lower transmission rates, the average and individual latencies are slightly increased, while the bus is less utilized as the number of transmission requests also drops. Also, with the advantage of longer transmission duration, the performance of the system is still effective at 100%.

Next, the above results are compared with mixed classes of stations. 20 stations will be assigned to the first transmission and wait rates, 20 to the second and 24 to the third. In the first example, T_1 = 20Hz, 100Hz and 1KHz. Correspondingly, S_1 = 360KHz, 208KHz and 140KHz. N = 8 bits. The population of the system is first filled with the slow transmitting host stations and then with the faster ones. The idea is to leave the fast stations to operate their transmissions during the time discrepancies between two transmissions of the slow stations.

Figure 3.8 shows the discontinuities in the piecewise-linear increase of the bus utilization, which is stressed by the incremental insertion of each class of stations. The bus utilization is increased faster when faster transmitting stations are introduced in the system. By alternating fast transmissions with slow ones, the average and individual latencies are also decreased.

In the second example, T_2 = 100Hz, 1KHz, and 10KHz. S_2 = 360KHz, 208KHz and 140KHz. N = 8 bits. As the transmitting frequencies are increased in the hybrid class, the latency times slightly increased, whereas the latencies' discrepancies indicate that less fairness is achieved and the stations are competing among themselves. A very small system performance drop is observed although it is apparently negligible.

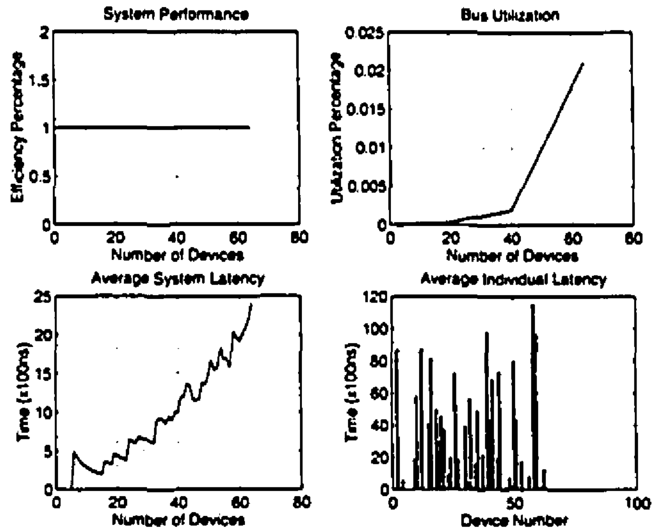

Figure 3.8: Protocol Simulation of 3 classes of transmission, T_1 = 20Hz, 100Hz and 1KHz, S_1 = 360KHz, 208KHz and 140KHz, N = 8 bits.

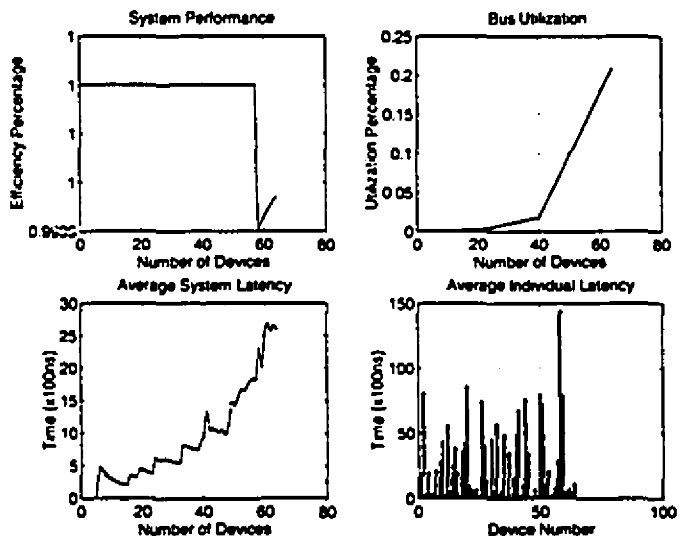

Figure 3.9: Protocol Simulation of 3 classes of transmission, T_2 = 100Hz, 1KHz and 10KHz, S_2 = 360KHz, 208KHz and 140KHz, N = 8 bits.

By using the previous example of hybrid class, that is, T_2 = 100Hz, 1KHz, and 10KHz; S_2= 360KHz, 208KHz and 140KHz; The system is simulated in variation of the packet length, with N = 16, 32 and 64 bits, increasingly varied.

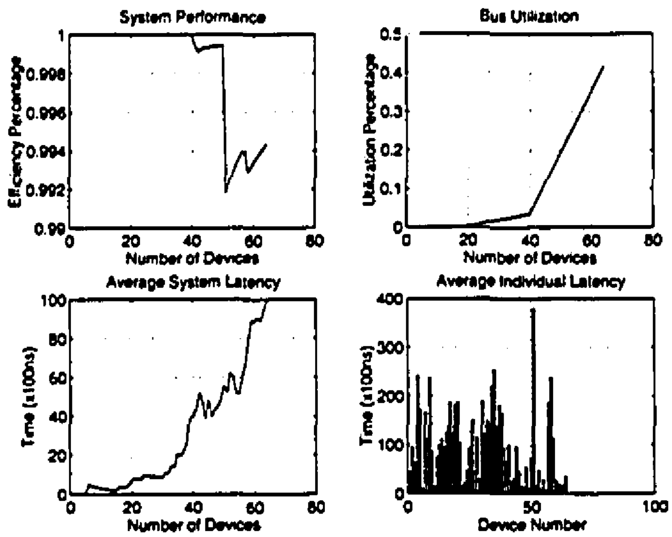

Figure 3.10: Protocol Simulation of 3 classes of transmission at 16 bit packet length, T_2 = 100Hz, 1KHz and 10KHz, S_2 = 360KHz, 208KHz and 140KHz, N = 16 bits.

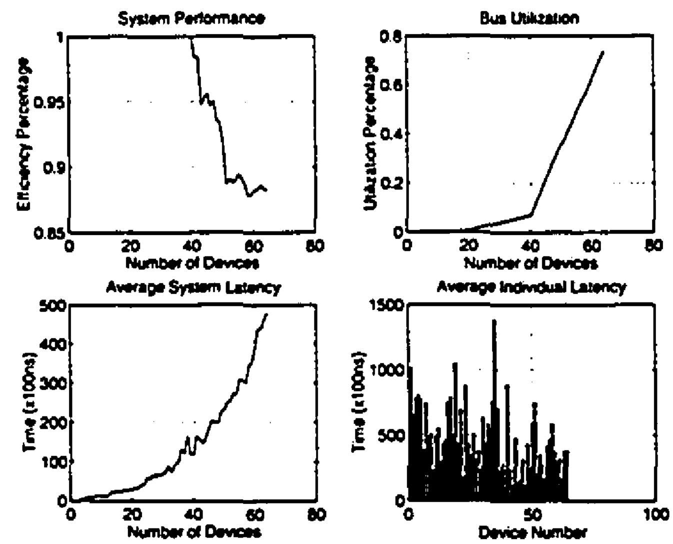

Figure 3.11: Protocol Simulation of 3 classes of transmission at 32 bit packet length, T_2 = 100Hz, 1KHz and 10KHz, S_2 = 360KHz, 208KHz and 140KHz, N = 32 bits.

Logically, the bus channel bandwidth is also reduced with successful transmissions as the packet length increases, and hence, the system performance decreases (exponentially) as the traffic grows in the system. As a result, the system can only allocate a smaller number

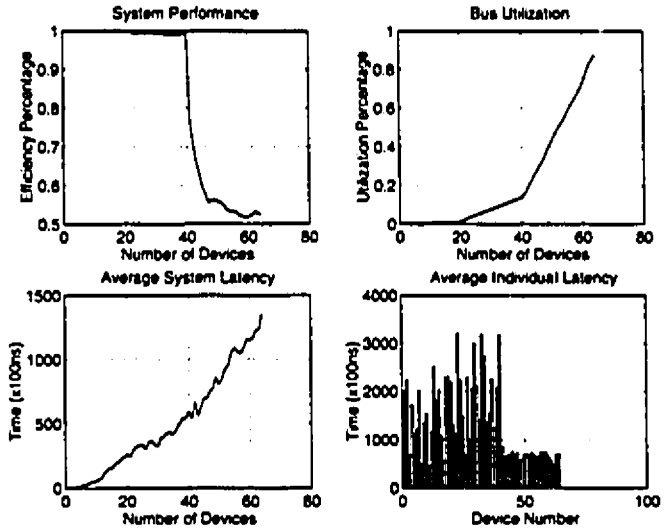

Figure 3.12: Protocol Simulation of 3 classes of transmission at 64 bit packet length, T_2 = 100Hz, 1KHz and 10KHz, S_2 = 360KHz, 208KHz and 140KHz, N = 64 bits.

of successful transmissions, inducing a important ratio of data losses in the system and a faster exponential increase in the latency times, as the packet length is increased. With N = 64 bits, the individual latency times indicate that the fastest stations suffer the most from the data losses, whereas the remaining stations at lower transmission rates suffer practically no data loss.

Finally, we would like to test a system under heavy traffic load and observe how the system performance would respond. In the Figures 3.13 and 3.14, we show the operation of a homogeneous system from light to dense population conditions, with up to 120 stations. There, we chose T = 10KHz, S = 78KHz, with N = 8 bits and N = 64 bits, respectively.

For the first simulation, the system performance stabilizes around 73% with a dense population. The individual latency time shows that the stations fairly shared the bus channel between themselves. The bus utilization is constantly increased although the average latency time is exponentially growing with the growth of the population. In the second simulation, the system performance is unproductive with more than 80% data loss. Although nearly full capacity of the bus is utilized, the average and individual latency times tend to converge to the limit of the transmission period.

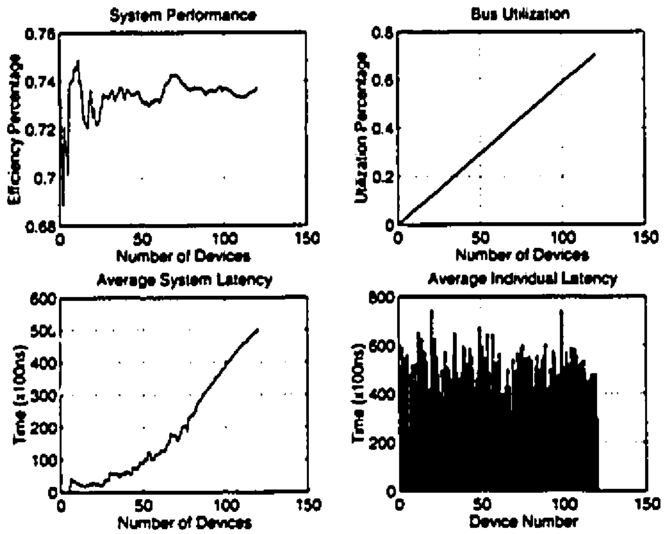

Figure 3.13: Protocol Simulation of Heavy Traffic with a Homogeneous Class, T = 10KHz, S = 78KHz, N = 8 bits.

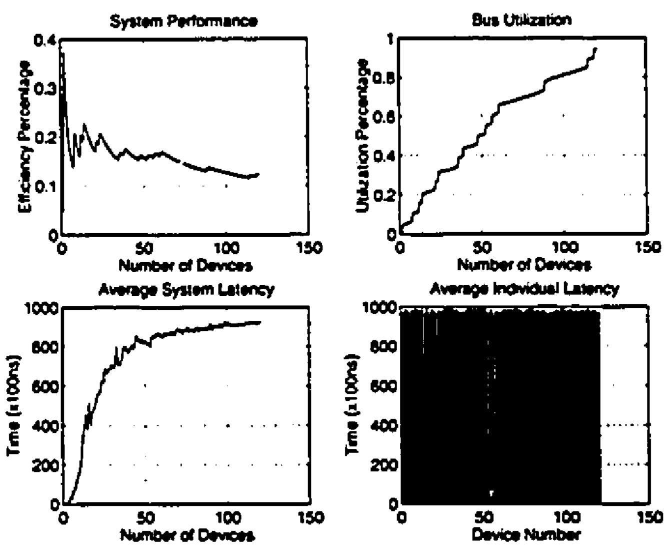

Figure 3.14: Protocol Simulation of Heavy Traffic Load with a Homogeneous Class, T = 10KHz, S = 78KHz, N = 64 bits.

Similarly as the above examples, the system is also simulated with hybrid classes under heavy traffic loads. First, we set T = 5KHz, 10KHz and 20KHz, S = 208KHz, 114KHz and 78KHz, and N = 8 bits. Secondly, the packet length is then increased to N = 64 bits. In

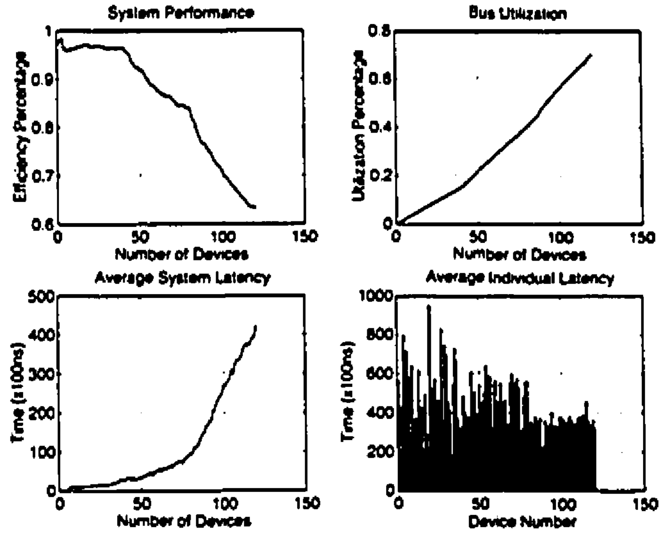

Figure 3.15: Protocol Simulation of Heavy Traffic Load with Hybrid Classes, T = 5KHz, 10KHz and 20KHz, S = 208KHz, 114KHz and 78KHz, N = 8 bits.

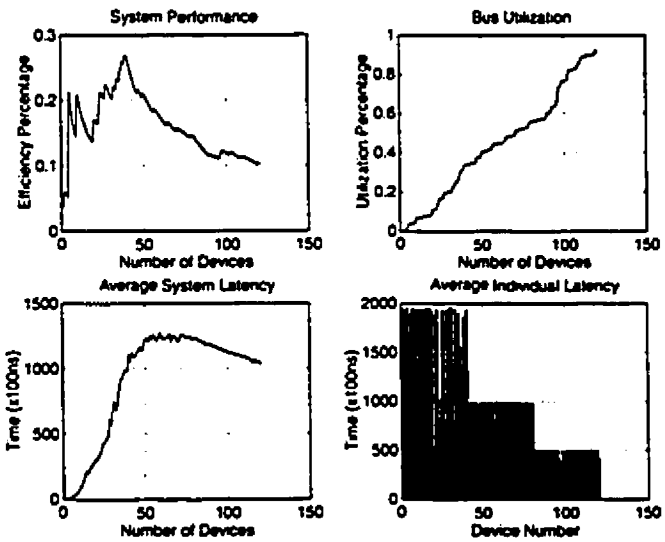

Figure 3.16: Protocol Simulation of Heavy Traffic Load with Hybrid Classes, T = 5KHz, 10KHz and 20KHz, S = 208KHz, 114KHz and 78KHz, N = 64 bits.

Figure 3.15, the system performance expresses the efficiency drop of data transmission as the traffic load is increased first with slow transmission stations and then with faster stations. Despite of the drop, the individual latency times indicates an equally stable fairness among the stations. In contrast, the results in 3.16 shows the system performance degradation with a poor throughput, the individual latency times discriminating the non-transmission times of the different classes. The average latency time shows an optimal value with about 60 stations in the system, indicating the threshold of the traffic saturation of the system. The bus utilization is however constantly increased.

The results from the simulation indicates that the network performance degrades as expected when the bus system is overloaded. This problem can be overcome using long periods of transmission and random deviation in retransmission priorities at much shorter periodic rates. In engineering applications, while the number of stations is being specified, the 10% range of information loss is considered as acceptable and recoverable, while the bus capacity must not be saturated. Simulations suggests that the traffic load should be adjusted to the right consumption of the bus capacity whereas using small packet lengths can help avoid fairness inequalities among the stations.

In general, these numerical results indicate some resemblance with the theoretical predictions. The fundamental trade-off lies in the bus utilization and system throughput and delays and fairness among the stations. Better yet, high performance could be achieved by tuning simulations according to the specifications of the application. The best approach is to separate useful stations into different classes according to their data communications rates.

Chapter 4

Hardware Design

4.1 Network Architecture

In this section, a specific architecture for a physical realization of the network is presented. Values of interest are described as the simulated data variables satisfying the protocol implemented in chapter three. We simulate the functional behavior of the micro network in VHDL.

4.1.1 Communications System

The hardware of each individual sensor station provides all the transmission control and the data storage by using Transistor-Transistor Logic gates (and/or PLA chips) and interfacing from external inputs.

In Figure 4.1, the basic block diagram of a data-transmitter node is shown. Recall that sensors acquire periodic data and are required to send it over the network towards a central receiving unit, at most once within a constrained period of time, after which new data overwrites old data and the sensors reattempt to process the transmission of the newly received data.

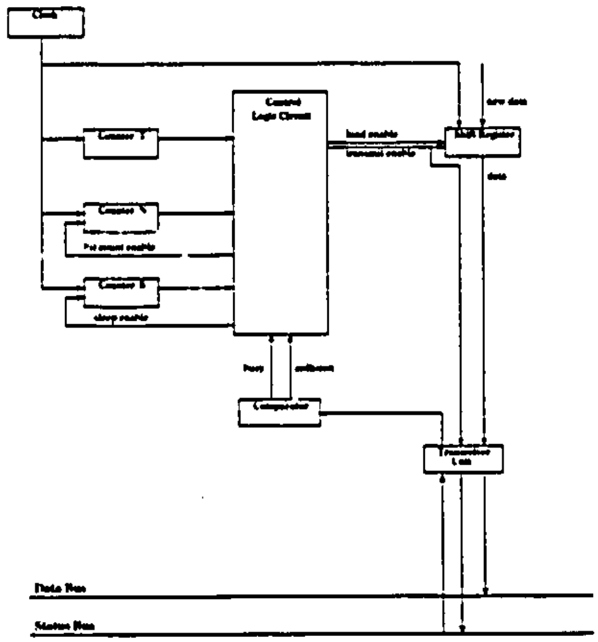

Figure 4.1: Diagram of a transmitter station

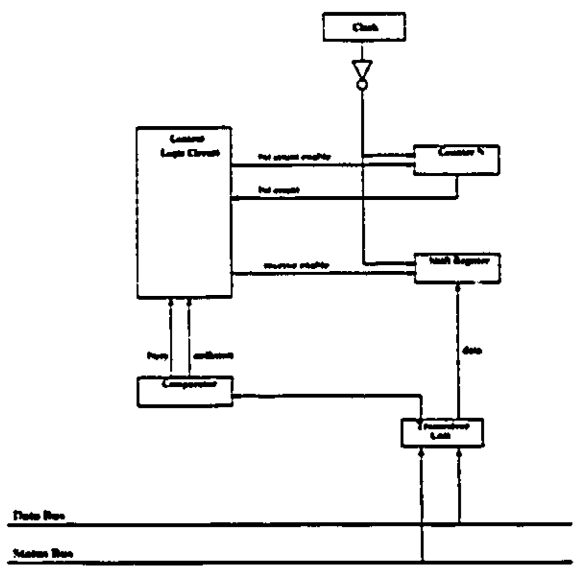

Figure 4.2: Diagram of a receiver station

The network general receiver unit is illustrated in Figure 4.2. The receiving node has the ability to sense the transmission medium and detect for idle, transmission, or collision states appearing on the bus. When a transmission is detected on the single-line data bus, the receiver unit has to shift the transmitted bits of a packet in serial from the bus into the register. On a collision or idle status of the bus system, the receiver will remain idle. The overall network was presented as in the Figure 2.5.

4.1.2 Hardware Related Limitations and Constraints

Several constraints must be considered in the design of the network. First, the propagation delay has little effect on the micro network since the propagation times are always short but not negligible. A maximum end-to-end travel time of signals must be considered on the shared channel. In Figure 2.2, a sensor station propagates a single bit signal at the rising edge of the clock cycles and holds that signal long enough so that every other station on the network can identify it. Considering that signals roughly travel at two thirds of the speed of light, i.e., 2×10^8 meters/second, and the geographic scope of the micro network reaches up to 5 meters, then the propagation delay to transmit a signal from one end to another is of the order of:

$$\begin{aligned} t_p &= \frac{length}{signal\ speed} \\ &= \frac{5}{2 \times 10^8} \\ &= 25\ nanoseconds\ per\ 5\ meter\ lengths \end{aligned}$$

At the bit level asynchronism, it is appropriate to assume that propagation delay is short relative to the clock cycle unit (approximately 25%). The remaining time of the cycle allows the stations to generate control signals for the next cycle.

A station which processes a packet transmission resulting in a collision on the shared bus has the ability to abort the bus and release it, so the bandwidth of the bus channel is not wasted. An attempt to randomize the colliding packets is made for each station involved in collisions. Packets involved in collisions must incur a back-off delay, even when the network is idle in the mean time. In other words, for the response time of the system not to become too high, the bus access must be granted distributedly over time.

4.1.3 VHDL Network Realization

With VHDL programming, a digital system is described as a module with inputs and outputs. The electrical signals on the outputs are a function of the values of the inputs. One way to specifying the function of a module is to describe its hierarchical structure or sub-modules. Each of the sub-modules can include existing elementary structures such as flip-flops, registers, logical gates and so on, to the lowest level. It is also possible to describe the function of a module by its behavior. The description of the function performed by the module is specified by the relationship between its output and inputs without reference to its actual internal structure. In Figure 4.1, each single transmitter node is described as a module consisting of:

- a transceiver unit reporting the status of the bus system
- three counting units
- a control unit
- a shift-register

The structure of a transmitter module is shown in Figure 4.3. The functional operation of the transmitter is sensitive to the input clock (10Mbps clock) and the values changes of the bus status. The transmitter has an interfacing circuit to report continuously the states of the bus system. Depending on those states, the control unit generates the control signals to process the operational states during the transmission period from the protocol presented in chapter three.

- On an idle bus, the transmitting module takes the bus or stays inactive.
- On a busy bus, the module either keeps transmitting, decreases/resets its wait time or stay inactive if the transmission has already been successful.
- On a colliding bus, the module stays inactive if transmission has already been done or retries at a later time otherwise.

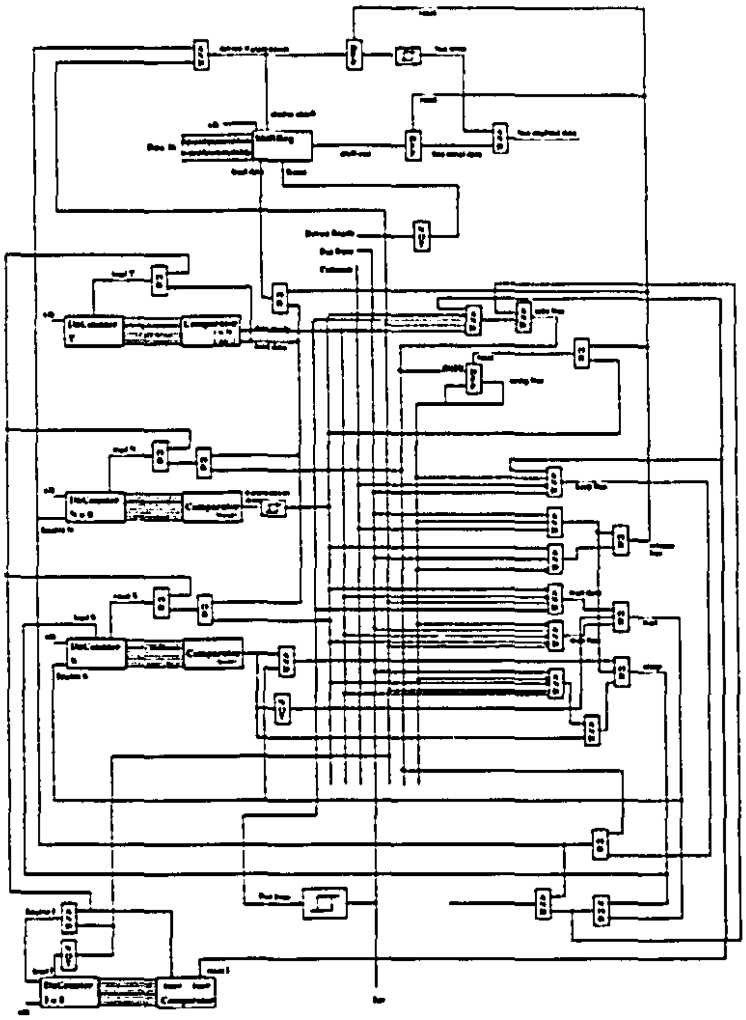

Figure 4.3: Architecture of a transmitter station

In this transmission process, the T-counter defines the periodicity of the programmed transmission time. When a module is granted a transmit, the N-counter counts the number of bits that have to be sent and enables the transmission signal during that amount of time. When a station is sensing busy transmission(s) on the bus other than its own, it must reschedule its transmission at a future time. In that case, the S-counter is enabled to count the amount of time associated to the module to perform ulteriorly the same transmission during the same transmission period.

When a module transmits, the transmission enable signal will be generated for the duration of the transmission and enable the shift-register to shift-out serially the data. Each transmitter connected to the network is autonomous and self-driven with the ability to be configured with fixed or random periods of transmission or retransmission, according to a desired probability function associated to the module.

This process is reversed for the receiving module. The general receiver has three distinct parts as shown in Figure 4.2:

- a receiving shift-register
- a collision detection transceiver unit
- a bit counter counting the length of receiving packet.

Figure 4.4 shows the implemented architecture of the receiver using TTL logic gates:

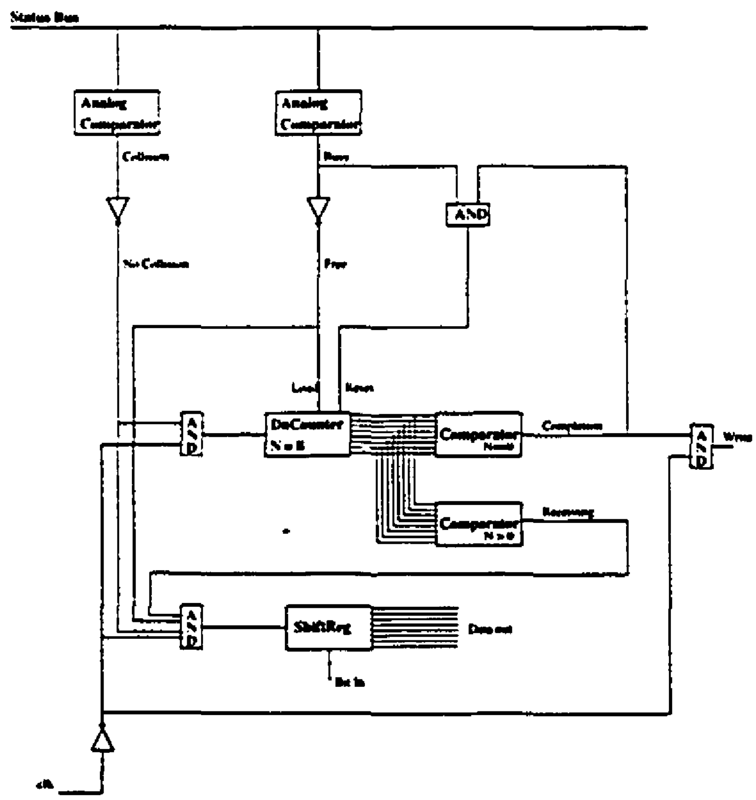

Figure 4.4: Architecture of a receiver station

When the bus is found busy, exactly one module is transmitting and the receiving module is enabled. The N-counter makes sure that N bits have been shifted serially into the shift-register. After that data have been received successfully, it can be loaded in parallel to any other host system. If the bus system is found idle or colliding, the receiving block remains inactive.

The subsequent sections will test the behavior of these modules. The integrated system will also be tested and its performance will be compared to the numerical analysis presented earlier in the previous section.

4.2 Network Simulation

Before performing simulations of the hardware model, some characteristics of the system are presented. Unit testing will show the functional operation of the transmitter and receiver blocks. The system is then tested in its integrity (with many transmitters and a receiver) to give the operational responses for various types of traffic population.

4.2.1 Real-time Simulation Characteristics

There are two principal features that are considered in this hardware design. A first consideration concerns the quantization of simultaneous signal writes. When several nodes request simultaneous transmission writes on the shared bus, the current signals are summed. A second consideration involves the configurability of each transmitting node to profile the behavior of the nodes as specified by the requirements. Different classes of transmission rates and wait priorities can be given respectively depending on (S) and (T).

In the architecture shown in Figure 4.3, the transmission rate is expressed by the T-counter in terms of the number of clock cycles. Typically, for a channel capacity C (bits/sec) and a desired transmission rate r (1/sec), the number of clock cycles required would be:

$$T = \frac{C}{r} \ (bits)$$

In this instance, with a channel capacity of 10Mbps, a slow transmission device of 20Hz would require its transmission period to be 500,000 clock cycles. Fast devices are typically of 1Khz or above. This VHDL implementation supports S-counter and T-counter with 24 bits. As a result, frequency rates for transmission and retransmissions (waits and sleeps), can be specified in the order of $\frac{C}{10^{24}-1}$ (1/sec) and above. Once the structure and behavior of the modules have been specified to model the actual sensors, it is possible to run the network system through simulations time.

The control signals for transmission are generated to include the processing states which are described in figure 3.2 in chapter three. To perform a transmission operation, a device must enter in the following states in this order: take bus, keep bus and release bus. The process of aborting a transmission consists of the following states: take bus and release bus. If the bus is found busy at the time of transmission request, the module enters directly to the sleep state. If its new data is not available, it enters in the wait state for later transmission. Each of those states above is of course dependent on many variables. These involve (with their respective hardware element):

• Device active/inactive	(device switch)
• Bus busy	(transceiver)
• Bus collision	(transceiver)
• Data not ready	(T and N counter: $T - N < 0$)
• Transmission done	(N counter)
• Sleep	(S counter)

All of those control signals will be illustrated in the transmitter simulation section.

Simulation starts with an initialization phase, and proceeds by repeating a two-phase simulation cycle. In the initialization phase, all signals are given initial values, the simulation time is set to zero and each module's behavior program is executed. In the first stage of a simulation cycle, the simulation time is advanced to the earliest time at which a transaction has been scheduled. All transactions scheduled for that time are executed, and this may cause events to occur on some signals. In the second stage, all modules which react to events occurring in the first stage have their behavioral programs executed. These program usually schedule further transactions on their output signals. When all the behavioral programs have finished execution, the simulation cycle repeats until the end of simulation time or there are no more scheduled transactions.

The simulation gathers information about the changes in system state over runtime, and possibly optimizes for good performance of the network with different combinations of configurations that satisfy system requirements.

4.2.2 Transmitter Simulation

The timing diagram shown in Figure 4.5 illustrates the time simulation of a transmitter node. On this diagram, the variables dev0, col0 and ext0 indicate the states of the transmitter node and the bus. The signal dready0 is to report to the transmitter if its data

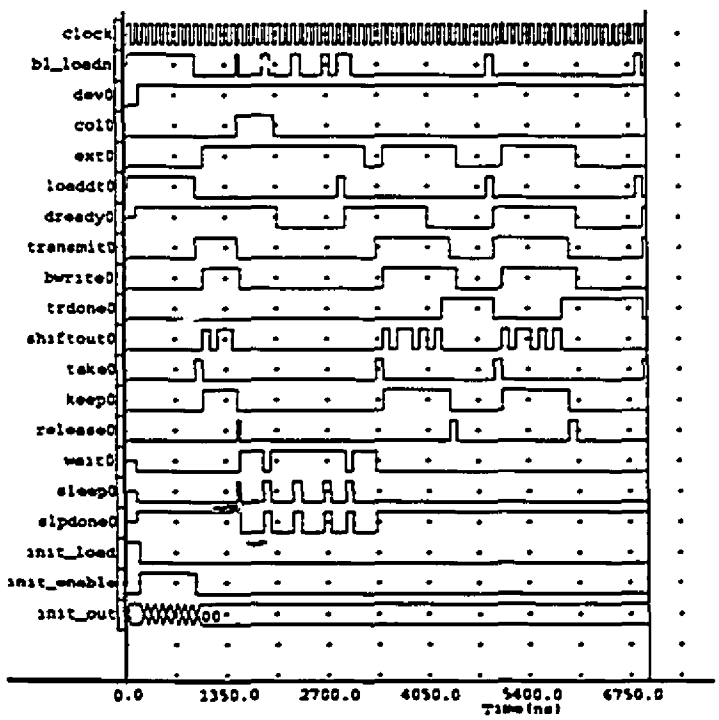

Figure 4.5: Timing diagram of a transmitter node

is available for a transmission request. The signals take0, keep0, release0, wait0, sleep0 specify the action states of the transmitter, as described in Figure 3.2. The variables trdone0, slpdone0 indicate the completion of a transmission and the completion of a waiting period of the transmitter. The serial bit signals on the data bus line are indicated by the shiftout0 signals, while transmit0 and bwrite0 indicate that a transmitter processes for transmission and writes to the bus. The signals loaddt0, b1_loadn are to illustrate the duration of a transmission period and the times when the N-bit count is reset for a new transmitting packet.

When a transmitter is first switched to active, it activates an initialization circuitry to initialize those control variables; init_load, init_enable and init_out show an arbitrary 8-bit time initialization. At the collision stage (at time=1500ns in the Figure 4.5), the transmitter first takes access of the bus for one bit time. Collision is detected at the next bit time and bus abortion is immediately decided. In Figure 4.5, a wait time of 3 clock cycles was chosen to retry for the next transmission. When transmission is finally successful at time 3450ns; the transmitter initially takes access onto the bus for one bit time, keeps it during (N=8) bit time for transmission and finally releases the bus after another bit time. The total overhead in transmitting N-bit packets is (N+2) bit times due to the extra bits required for accessing on the bus and releasing it. Note that the (N+2) bit time corresponds to the transmit enable time. The transmit enable signal will provoke a write to the actual state of the bus by causing a quantized current to add to the actual current state of the bus system. If there is more than one quantum, a data collision results and transmission abortions will proceed immediately at the next bit time for all the devices involved. If there is only a unique current quantum, the bus is granted to the requested transmitter and transmission will be communicated appropriately. For the remaining transmission duration, the transmitter becomes idle or inactive since it has successfully transmitted its data packet during the required transmission period.

4.2.3 Receiver Simulation

At the receiving end, the process is reversed and depends on the state of the status bus. Figure 4.6 illustrates the timing diagram of receiving module. The signals **collision** and **busy** which are translated by the transceiver unit, indicate the current state of the bus. The signal **bit_in** designates the current propagating bit signal on the data bus, and the signals **bits_out** the current data value stored in the shift register. The signal flags **a_s2**, **a_s3** and **cmp_s1** are to set a load enable when a N-bit packet reception has been completed. The signal bits **a_s1** and **dncnt_s1** are used to count down the number of bits upon a packet reception.

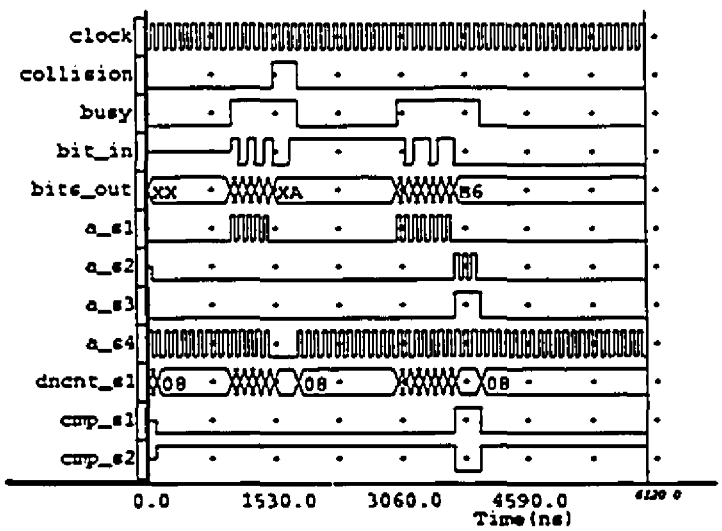

Figure 4.6: Timing diagram of a receiver node

In Figure 4.6, if the state indicates Idleness or Collision (which is shown at time=1500ns), there is no reception of data. If the state is otherwise busy (at time=3000ns), the receiving unit ensures to capture the N-bit data packet that is embedded in (N+2) communications time. After N bits have been received, a load enable signal can be generated to shift the new data to some other place.

4.2.4 Performance Tests

In this section, several tests were conducted to study the functionality and operationability of the system with regards to a certain distribution of device classification (by transmission rates and wait priorities). Ideally, the system response should spread evenly and distributedly data packets over its available bandwidth in order to obtain the ideal response described earlier in chapter two. In practice, the system response is far from ideal and simulations will prove it. The performance behavioral analysis of the system already studied in chapter three, is not emphasized here. Also, each station operates locally its activities as it was shown in Figure 4.3. For simulation purposes, we will only see the functionality and operationality of the system, in terms of channel utilization and the distribution of packet transmissions representing the use of the bus capacity by a group of stations.

Figure 4.7 shows the simple functionality and operationality of the network with 10 stations, each starting at a random time and carrying a light load (T = 100Hz, S = 100KHz, N = 8 bits). On this timing diagram, the signals `rec_bits` denote the data value that is currently read by the receiving shift register, and `17_bit1` the current bit value (or DC persistence) on the data bus line. The signals `17_busy1` and `17_collision1` which are perceived here by the transceiver unit of the receiver, indicate the current state of the bus.

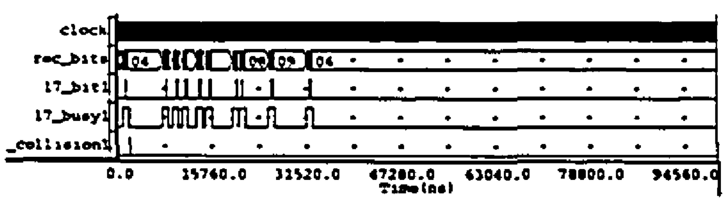

Figure 4.7: Simulation timing diagram of 10 sensor nodes

During the simulation time, the network utilization[2] is estimated to reach 30.75% after that the last node has successfully transmitted its packet. whereas the performance of the system is at 90.91% and with practically low transmission latency time.

[2] We define *network utilization* as $\left(\frac{\text{number of successful transmissions} \times \text{packet period}}{\text{total simulation time}}\right)$

Figure 4.8 shows a population of 64 devices with identical transmission (T = 1KHz) and wait times (S = 1.25MHz). The simulation involves starting transmissions at zero time and consequently results in 0% performance and 0% bus utilization (for achieving in any successful transmissions or retransmissions). The overall latency time is thus maximum and is the transmission periods of the stations.

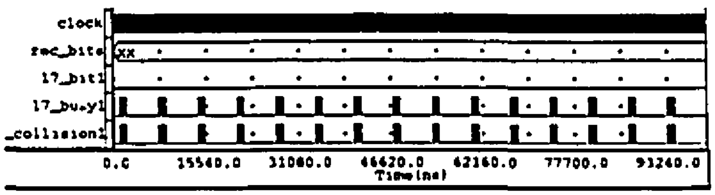

Figure 4.8: Simulation timing diagram of 64 identical sensor nodes

Figure 4.9 differs from the previous simulation by starting transmissions of each station at a random time, with the hope that data will be better distributed and successfully transmitted in time. The result obtained is shown for approximately one transmission period (or 64 expected successful transmission). Actual simulation shows 58 successful transmissions (versus 41 collisions). The system performance over the 100,000ns simulation period is hence $\frac{58}{64} = 90.63\%$. The effective bus utilization is also much improved, reaching nearly 58% after approximately one transmission period.

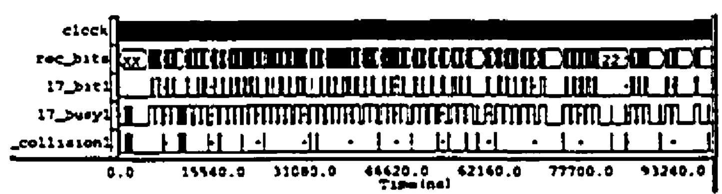

Figure 4.9: Simulation timing diagram of 64 sensor nodes with random variations

Next, the simulations are carried over with classified device types. The simulation in Figure 4.10 consists of 64 devices, with 20 at 20Hz transmission rate, 20 at 100Hz, 24 at 1KHz. The simulation also includes randomness in starting times as well as random variations in retransmission times (as was mentioned earlier). Note that, in this case, slow devices have less wait delay times than others. The figures obtained indicates 56 collisions

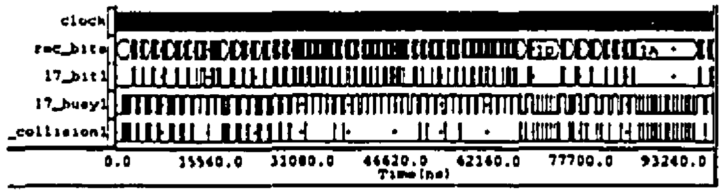

Figure 4.10: Simulation timing diagram of a hybrid class

and 37 collisions. The number of collisions is hence slightly reduced. Two observations were made. First, slow transmission stations attempts to resend collided packets as fast as possible. Secondly, faster stations tried to complete the time gaps betweens transmissions of the slow ones.

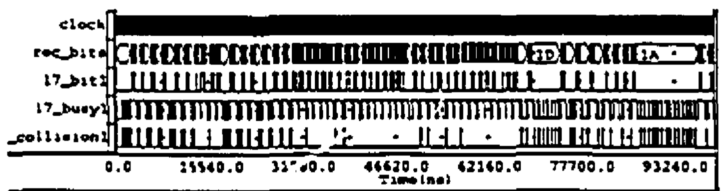

Figure 4.11: Simulation timing diagram of a hybrid class, with fast transmission frequencies

In Figure 4.11, 20 devices are at 100Hz transmission rate, 20 at 1KHz, and 24 at 10KHz. In this simulation, 56 successful transmissions have been completed and 40 collisions had resulted. With faster transmission rates, the frequency of collisions had increased. Since transmission periods are faster, stations refresh their data faster and request transmission. Under light loads, or light host loads, this improves the bus utilization of the system.

In Figure 4.12, the simulation is done with a packet length of 64 bits with the same

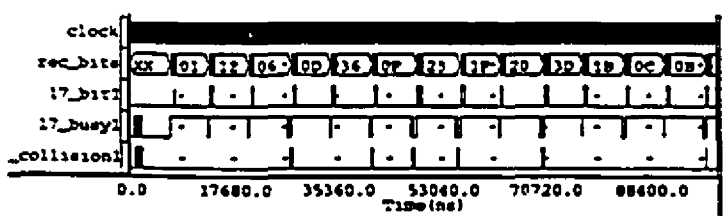

Figure 4.12: Simulation timing diagram of a hybrid class, with 64 bit packet length

devices as above. Overall, better utilization of the bus is obtained with longer packet lengths, yet the number of successful transmissions have much decreased over the simulation time span. The performance at the local sites have also become less effective as fairness and starvation problems have been increased inversely.

Through a number of simulation tests, it has been noted that the communications system is operational with the capability of resolving simultaneous processes and deadlock states with random variations. Nevertheless, a good, but not optimal, performance of the distributed concept was shown to be dependent on the fundamental trade-off between global bus utilization and local information transfer efficiencies. Obviously, this indirectly implies that the configurations of the local sites need to satisfy the type of data acquisition applications as well as its requirements.

4.2.5 Network Behavioral Performance

Based on our simulations, we showed that for a wide class of applications, the micro network is capable of carrying its nominal bandwidth of useful traffic and allocates the bandwidth fairly. Although the performance is not optimal in a global sens, this network system has:

- integration flexibility of installing new nodes.
- asynchronized ability of reception of two consecutive packets.
- possibility of configuring sensoring transmitter blocks as specified by the application requirement.

The results from the simulations indicate that the bus system increases its utilization as traffic is not overloaded. Improvements can be made by introducing randomness in starting transmission time, intentional small variations in retransmitting wait times, and increasing the packet bit lengths. It can then be observed over the simulation time that, the system is increasing the number of transmissions with faster transmissions rates as long as the host load is not saturating the bus capacity. However, if the traffic is too high, the distributed design is not appropriate since there would be too many collisions and loss of packets. Starvations and fairness problems also result.

The feasibility of hardware functionalities and operationalities have been demonstrated in performing distributed data communications up to 10Mbps speed using twisted pair wires. The speed, performance and flexibility of the micro-network allow to cost-effectively implementation of different types of configurations that satisfy a wide range of system requirements.

Chapter 5

Conclusion

With the simulations of the protocol and the hardware model, it has been shown that the implementation of a distributed CSMA/CD micro network is feasible for a large range of hardware specifications and communications constraints. The configuration of the transmitting stations easily defines a particular application of data communications. The considered parameters or the grain size of a system include data refreshing cyclicity, transmission and retransmission rates with respect to the clock cycles. The network population and data packet size can also be specified flexibly in both simulation models.

Essentially, this development of networking has brought a number of significant possibilities and demands for data acquisition. The sensors that generate critical data (variables) may be multiply connected to the network, which allows a multiple diffusion of those variables to increase their availability. However, more components and hence transmitting nodes also means that the system performance degrades with traffic growth. The common performance measures usually involve the fundamental trade-offs: Data delays, system throughput, bus utilization in function of the traffic loads; those quantities vary according to the specification and requirements of an application.

From the simulation analysis, the general observation suggests that data transmissions must be adjusted to the bus system bandwidth in order to make the traffic fluid and obtain better system performance. The common objective for each station is to maximize the number of transmissions on the shared allocated bandwidth. This CSMA/CD network design also relies on each station hearing the transmissions of another within a single bit time. Therefore, colliding transmissions can immediately be aborted after one bit time to reduce the waste of bus bandwidth.

There are various issues in applying CSMA/CD bus networks. First, by using a CSMA/CD protocol, there is no polling algorithm that guarantees fairness in transmissions to each station and in particular, stations with low priorities may experience low transmission efficiencies or even starvation situations. To resolve the fairness problem, small packets are to be preferred for dividing the bandwidth into smaller frames and allowing better interactive traffic communication and equitability among the stations. Secondly, when the traffic becomes dense and slow, data collisions become more frequent. It is important to prioritize stations with slow transmission rates, carrying relevant data in order to limit the loss of essential and unrecoverable information. This leads to classifying transmitting stations by their transmission rates and attempting to distribute their retransmissions in the case of collisions, at small random variations for the same group of stations.

The simulations have shown that the behavioral system performance can be improved by taking these issues into consideration. Mainly, the current network design is important in its flexibility and its aspects of handling, simulating and estimating diverse traffic types with dissimilar performance goals (high throughput, low delay, real-time communications, etc). Further improvements on this development can be addressed to involve data formats, time-stamps and reassembling at the receiving end. Routines can also be included with the transmission to detect failures in communications, so that data errors are not transmitted. The reliability may be as simple as a parity check, the simplest form of error checking.

To finalize, we will talk about future work that can be done. First, a dual bus system can be implemented using this concept of twisted pair so as to favor the high-speed data on a primary bus and lower speed data on a secondary one. With a dual service system, the data acquisition system performance can then be enhanced by separating data and reducing traffic population on a single bus line. Secondly, it is possible to extend the scope of this network to interconnections of other networks. The limited scope of this network is essentially to reduce the wiring complexity and redundancy on a data communications system. Once collected at a receiver site, data may possibly be forwarded by interconnected networks at a larger scale. Finally, recent developments in optical technology have made it possible to transmit data by pulses of light. Visible light has a frequency of about 10^8MHz, so the bandwidth of an optical transmission system is potentially enormous.

Bibliography

[1] E. Arthurs, G.L. Chesson, B.W.Stuck, Theorical Performance Analysis of Sliding Window Link, Level Flow Control for a Local Area Network, 8th Proceedings, Data Communication, 1982.

[2] D. Bersekas and R. Gallager, Data Networks, 2nd Ed., Prentice-Hall, 1992.

[3] David R. Boggs, Jeffrey C. Mogul, Christopher A. Kent, Measured Capacity of an Ethernet: Myths and Reality, WRL Reasearch Report 88/4, Digital Western Research Laboratory, Proceedings of the SIGCOMM, ACM SIGCOMM, August 1988.

[4] Werner Bux, Local-Area Subnetworks: A Performance Comparison, IEEE transactions on Communications, VOL. COM-29, NO. 10, October 1981.

[5] D.D. Clark, K.T. Pogran, D.P. Reed, An Introduction to Local Area Networks, Proceedings of the IEEE, Vol. 66, No. 11, Nov. 1978.

[6] Alan Colvin, CSMA With Collision Avoidance, A New Technique for Cost Reduction, Local Network, 1983.

[7] Edward J. Coyle and Bede Liu, Finite Population CSMA/CD Networks, IEEE Transactions on Communications, VOL. COM-31, NO. 11, NOVEMBER 1983.

[8] IEEE computer Society, Technical Committee on computer communications, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Standards for Local Area Networks, ANSI/IEEE Std. 802.3, 1985.

[9] Anura P. Jayasumana and P. David Fisher, A Low-Overhead Token-Passing Scheme for Bus Networks, Local Computer Networks, pp. 32-38, 1984.

[10] K. Kant, Performance Analysis of Hierarchical Ring Networks, 12th Conference of Local Computer Networks, 1987.

[11] Karl Kümmerle, John O. Limb, Fouad A. Tobagi, Advances in Local Area Networks, Frontiers in Communications, IEEE Press, 1987.

[12] James F. Kurose, Mischa Schwartz and Yechiam Yemini, Controlling Window Protocols For Time-Constrained Communication in a Multiple Access Environment, 8th Proceedings, Data Communication, 1982.

[13] Claude E. LaBarre and Paul J. Brusil, Architecture for Evaluating LAN Traffic and Performance, 12th Conference of Local Computer Networks, 1987.

[14] Simon S. Lam, A Carrier Sense Multiple Access Protocol for Local Networks, Computer Networks, v. 4, pp. 21-32, 1980.

[15] G.M. Lundy, R.E. Miller, Analyzing a CSMA/CD Protocol Through a Systems of Communicating Machines Specification, IEEE Transactions on Communications, Vol. 41, No. 3, March 1993.

[16] N. M'robet, G. Naguez, D. Trecourt, Réseaux Locaux à Très Haut Débit, l'équivalent du Code de Transmission asynchrone "Start-Stop", Performance of Data Communication Systems and Their Applications, North-Holland Publishing, G. Pujolle (ed.), 1981.

[17] Michael K. Molloy, Collision Resolution on the CSMA/CD Bus, Local Computer Networks, pp. 44-47, 1984.

[18] Henri Nussbaumer, Computer Communication Systems, VOL. 1, Data Circuits, Error Detection, Data Links, EPF Lausanne, 1990.

[19] S.S. Panwar and Y. Armoni, Collision Resolution Algorithms for a Time-Constrained Multiaccess Channel, IEEE transaction on Communications, Vol. 41, No. 31, July 1993.

[20] Radia Perlman, Incorporation of Multiaccess Links Into a Routing Protocol, 8th Proceedings, Data Communication, 1982.

[21] Kanti Prasad, Ashwani Singhal, Simulation of Ethernet Performance Based on single Server and Single Queue Model, 12th Conference of Local Computer Networks, 1987.

[22] C. Meraud and B. Maurel, Réseau D'Echange reconfigurable pour Contrôle du Processus réparti, Performance of Data Communication Systems and Their Applications, North-Holland Publishinh, G. Pujolle (ed.), 1981.

[23] T.G. Robertazzi, Computer Networks and Systems: Queueing Theory and Performance Evaluation, Telecommunication Networks and Computer Systems, Springer-Verlag, 1990.

[24] Matthew N.O. Sadiku, Mohammad Ilyas, Simulation of Local Area Networks, 1995.

[25] Adarshpal S. Sethi and Tuncay Saydam, Performance Analysis of Token Ring Local Area Networks, Local Computer Networks, pp. 26-31, 1984.

[26] Mischa Shwartz, Telecommunication Networks: Protocols, Modeling and Analysis, Addison Wesley, 1987.

[27] John. Shoch and Jon A. Hupp, Measured Performance of Ethernet Local Network, Communications of the ACM, VOL. 23, NO. 12, December 1980.

[28] Lansing J. Sloan, Limiting the Lifetime of Packets in Computer Networks, Computer Networks, v. 3, pp. 435–445, 1979.

[29] Otto Spaniol, Modeling of Local Computer Networks, Computer Networks, v. 3, pp. 315-326, 1979.

[30] Hideaki Takagi and Leonard Kleinrock, Throughput Analysis for persistent CSMA Systems, IEEE Transactions on Communications, VOL. COM-33, NO. 7, pp. 627-600, July 1985.

[31] Shuji Tasaka. Dynamic Behavior of a CSMA-CD System with a Finite Population of Buffered Users, IEEE Transactions on Communications, VOL. COM-34, NO. 6, pp. 576- 586, June 1986.

[32] Fouad A. Tobagi and Bruce Hunt, Performance Analysis of Carrier Sense Multiple Access with Collision Detection, Computer Networks, v. 4, pp. 245-259, 1980.

[33] W. Bruce Watson, Simulation Study of the Traffic Dependent, Performance of a Prioritized, CSMA Broadcast Network, Computer Networks, v. 3, pp.427-434, 1979.

[34] John D. "J.D." Wheelis, Process Control Communications: Token Bus, CSMA/CD or Token Ring?, Advances in instrumentation and Control, 1992.

[35] Y.T. Wu and J.F. Chang, Collision Resolution for Variable-Length Messages, IEEE Transactions of Communications, Vol. 41, No. 9, September 1993.

[36] Jeffry W. Yeh, Simulation of Local Computer Networks - a Case Study Computer Networks, v. 3, pp. 401-417, 1979.