Design of a Token Ring to ISDN Gateway

by

Jangkyung Kim

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

Department: Electrical Engineering and Computer Engineering
Major: Computer Engineering

Iowa State University
Ames, Iowa
1989

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The rapid and continuing decrease in computer hardware costs, accompanied by an increase in computer hardware capability makes it possible for us to have an increased number of systems at a single site: office building, home, school, factory and so on. In order to exchange data between these systems, and to share expensive resources, we need to interconnect these data processing systems in local environments. One of the most important consequences of this need is the advent of Local Area Networks (LANs). On the other hand, during recent years we have experienced a rapid evolution towards the use of digital techniques for telephony and data telecommunication services. The culmination of these digital techniques is the Integrated Services Digital Network (ISDN), which is a projected worldwide public telecommunications network that will service a wide variety of user needs. It would therefore be a great asset to an ISDN environment to offer to its customers some kinds of LAN services.

The most important reasons for connecting a LAN to the ISDN are to interconnect LANs that are located on two remote sites, to connect a LAN to the ISDN in order to connect any ISDN terminal to this LAN, and to make any ISDN service available to all LAN customers. But when we connect a LAN to the ISDN, some problems can be caused. The major problems are the inconsistency of addresses, the incompatibility of frame structures and the subaddressing within the LAN. There are three types of solutions for these problems. They are the Gateway solution, the Mapping Layers solution, and the Internet Protocol Layer solution [1]. The research done for this thesis was focussed on the design issues in implementing a *Token Ring to ISDN Gateway* which is one of the LAN-ISDN connection solutions.

1.1 Method

For the implementation of the Gateway, the token ring (IEEE 802.5) was selected as a specific LAN model. The LAN environment of this research is the token ring system which is installed on the IBM PC/XTs at the Computer Systems Research Groups's laboratory of Iowa State University. The token ring adapter of this system is TI (Texas Instruments)'s TMS380 Token Ring Adapter. As the ISDN environment simulation, Mitel's ISDN Express Card^{TM} Design Kit with ISDN Evaluation System(IES) software was used [2].

For the implementation of the *Token Ring to ISDN Gateway*, the S interface configuration method using Basic Channel Service (2B+D) was selected. The hardware of the gateway consists of a TI's TMS380 Token Ring Adapter, the *Gateway Controller* (IBM PC/XT), and the *ISDN S Interface Circuit* which was designed and assembled during this research. The *Gateway Controller* controls the TMS380 which is the token ring side interface and the *ISDN S Interface Circuit* which is the ISDN side interface.

The *Interconnection Protocol* designed for this gateway covers up to the OSI Network Layer and is implemented using C and 8088 Assembly Language. The *Interconnection Protocol* was designed based on the CCITT X.25 Virtual Call protocol and works between same token rings which are connected through the ISDN. The

Interconnection Protocol was realized by the *File Transfer Program* and the *Gateway Driver*. The *File Transfer Program* resides at the File Transfer TX station and File Transfer RX station and performs the file transfer between these two stations. The *Gateway Driver* runs at the gateway and makes the gateway work. The block diagram of the system environment in which the *Token Ring to ISDN Gateway* works is shown in Figure 1.1. Lastly, the performance of the gateway was measured.

1.2 Organization of the Thesis

In the next chapter, the related backgrounds and problems of this research are discussed. At first, OSI Model concept is discussed briefly. Then the token ring (IEEE802.5), ISDN, internetworking method, and X.25 Virtual Circuit protocol are discussed. Lastly, the LAN-ISDN interconnection problems are introduced.

Chapter 3 contains the design issues of the *Token Ring to ISDN Gateway* hardware. The design of the gateway along with the *Gateway Controller*, TMS380 Token Ring Adapter and the *ISDN S Interface Circuit* are discussed in detail.

Chapter 4 discusses the *Interconnection Protocol* design. In this chapter, the protocol design concept and the details of the *Interconnection Protocol* design are discussed. Next, the details of the *File Transfer Program* design and the *Gateway Driver* design are discussed.

Chapter 5 contains the performance analysis of the *Token Ring to ISDN Gateway*. The performance was measured using monitoring.

Chapter 6 concludes the thesis by recapitulating the work done and suggesting problems for future work.

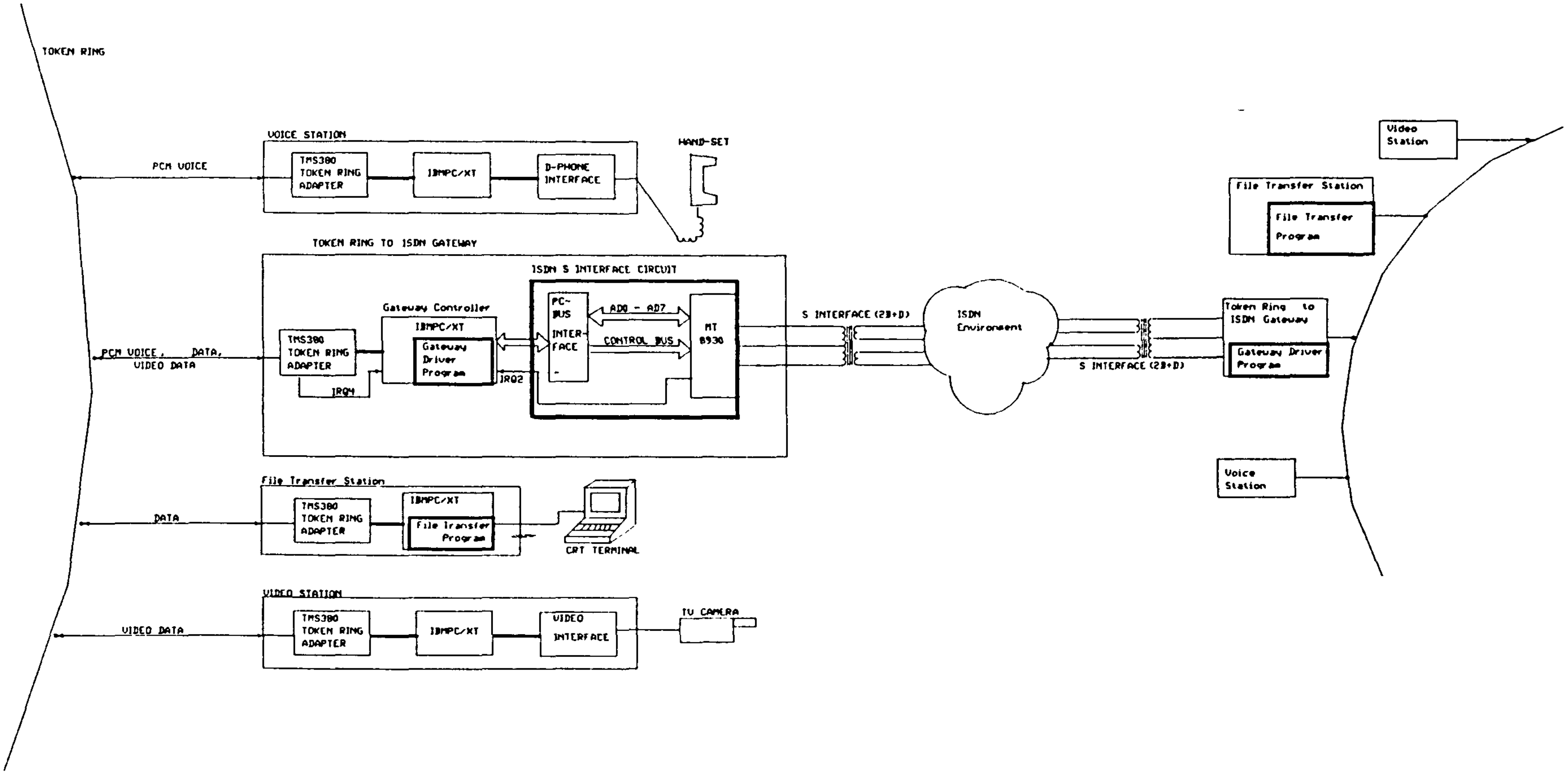

Figure 1.1: The block diagram of the system environment

2 INTERCONNECTION OF TOKEN RING AND ISDN NETWORKS WITH A GATEWAY

This chapter provides the background for this research and describes the problems involved in interconnecting the token ring and ISDN networks with the gateway. The OSI model, IEEE 802.5 token ring, ISDN, internetworking method, and the X.25 Virtual Circuit protocol are also discussed. Lastly, the LAN-ISDN interconnection problems are introduced.

2.1 OSI Model

2.1.1 Motivation

As the use of the computer communications and the computer networking proliferates, the related software development becomes too costly to be acceptable. Therefore, a set of international standards are needed to provide a common basis for the coordination of standards development for the purpose of interconnection. In 1977, the International Standard Organization (ISO) established a subcommittee to develop such an architecture. The result was the Open Systems Interconnection (OSI) model, which is a frame work for defining standards for linking heterogeneous computers.

Table 2.1: OSI layers

Layer	Functions
Physical	Deals with the mechanical,electrical, functional, and procedural characteristics to access the physical medium
Data link	Sends blocks of data (frames) with the necessary synchronization, error control, and flow control
Network	Responsible for establishing, maintaining, and terminating connections
Transport	Provides end-to-end error recovery and flow control
Session	Establishes, manages, and terminates connections (sessions) between cooperating applications
Presentation	Provides independence to the application processes from differences in data representation (syntax)
Application	Provides access to the OSI environment for users and also provides distributed information services

2.1.2 Layers

A structuring technique chosen by ISO is layering. The communications functions are partitioned into a hierarchical set of layers. Each layer performs a related subset of the functions required to communicate with another system. The layer provides services to the next higher layer and communicates with the same layer of another system using the services of the next lower layer. The OSI reference model has seven layers, which are listed with a brief function in Table 2.1.

Figure 2.1 shows the OSI operation where application X has a message to send to application Y. When the data is processed at the source system, each layer appends

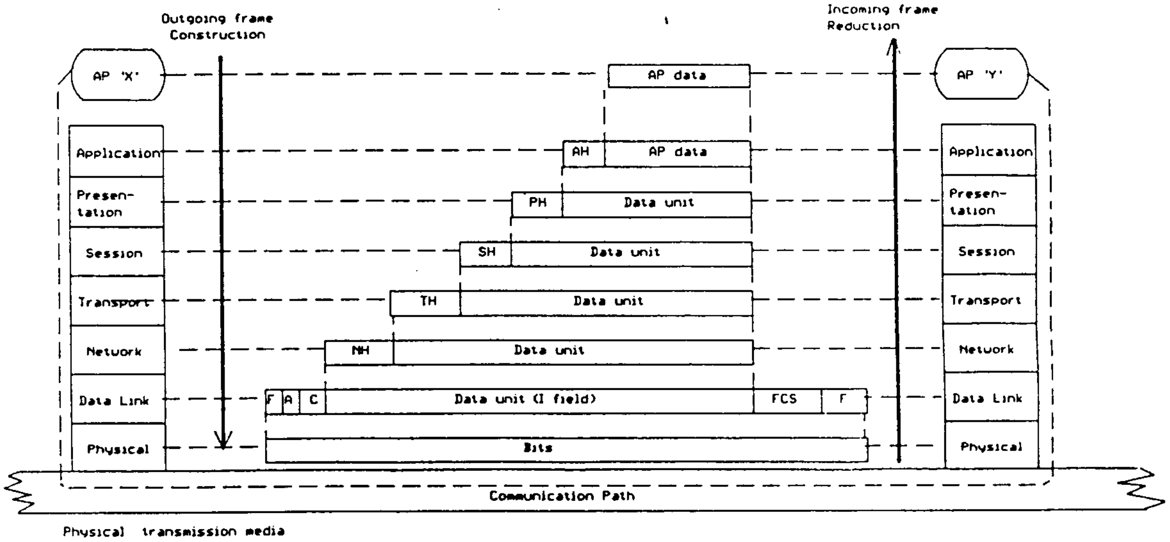

Figure 2.1: OSI operation

the header to the data. When the frame is received by the target system, each layer strips off the outmost header of ascending frame.

2.2 IEEE 802.5 Token Ring

The token ring is one of the most popular LAN used recently, originally proposed in 1969 and referred to as the Newhall Ring [3]. This technique is the one ring access method selected for standardization by the IEEE 802 Local Network Standard Committee [4] [5].

2.2.1 General Description

The token ring technique uses the single token that circulates around the ring when all stations are idle. A station wishing to transmit must wait until it catches a token passing by. It then changes the token from "free token" to "busy token." The station then transmits a frame immediately following the busy token. The frame on the ring will make a round trip and be purged by the transmitting station and the transmitting station will insert a new free token on the ring again. When a transmitting station releases a new free token, the next station downstream with data to send will be able to seize the token and transmit. Multiple levels of priority are available for independent and dynamic assignment depending upon the relative class of service required for any given message, for example, synchronous (real-time voice), asynchronous (interactive), immediate (network recovery). Figure 2.2 shows how the token ring works. The principal advantage of token ring is the fair access of the medium. The principal disadvantage of token ring is the requirement for token maintenance.

2.2.2 IEEE 802.5 Standard

This standard specifies the format and protocols used by token-passing ring Medium Access Control (MAC) sublayer, the physical (PHY) layer, and the means of attachment to the token-passing ring physical medium. The Local Area Network (LAN) model and its relationship to the Open Systems Interconnection (OSI) Reference Model of the International Organization for Standardization (ISO) is illustrated in Figure 2.3.

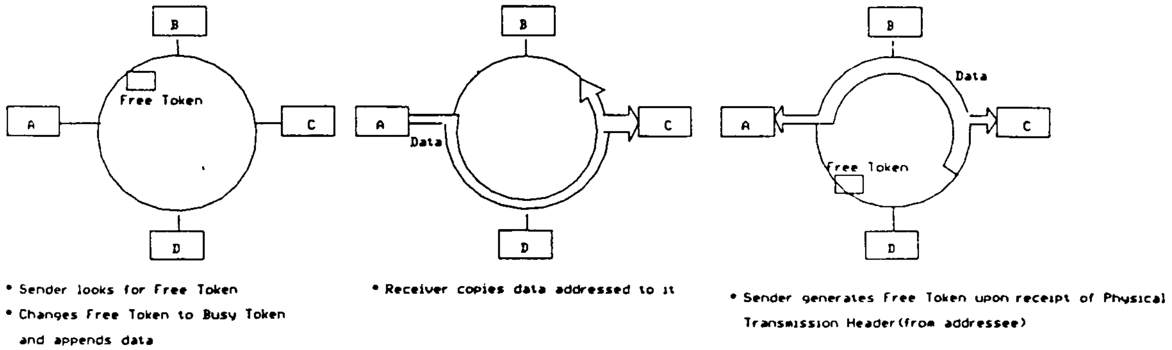

Figure 2.2: Token Ring

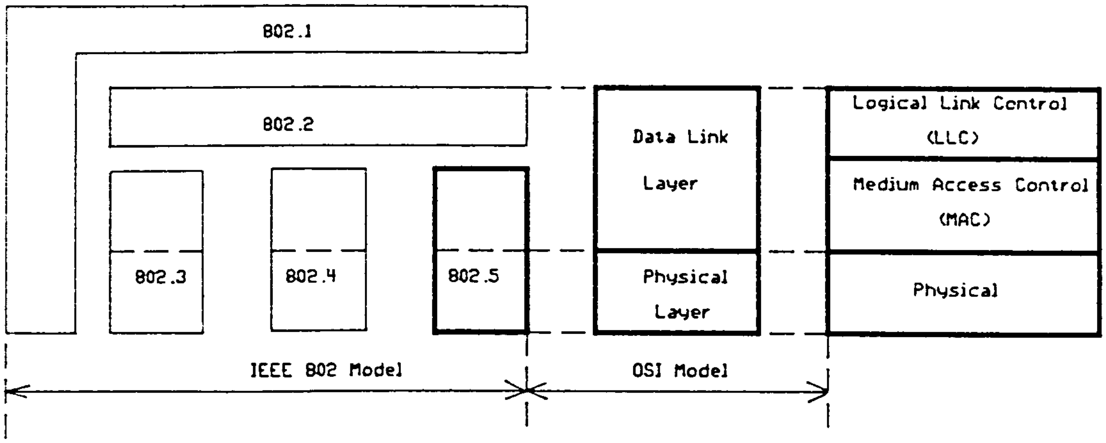

Figure 2.3: Relation of OSI reference model to IEEE 802 model

2.3 ISDN

The Integrated Services Digital Network (ISDN) is a projected worldwide public telecommunications network that will service a wide variety of user needs.

2.3.1 General Description

The ISDN is the network which has evolved from the basic digital telephone networks. The ISDN will provide end-to-end digital connectivity to support a wide range of services including voice, data, sound, and video applications. The user has access to the ISDN by means of a local interface to a "digital pipe" of a certain bit rate.

2.3.2 Transmission Structure

The digital pipe between the central office and the ISDN user will be used to carry a number of communication channels. The capacity of the pipe, and therefore the number of channels carried may vary from user to user. The transmission structure of any access link will be constructed from the following types of channels:

- B channel : 64 kbps
- D channel : 16 kbps
- C channel : 8 or 16 kbps
- A channel : 4 khz analog

The B channel is the basic user channel and can carry the PCM-encoded digital voice, digital data, and a mixture of lower-rate traffic. The D channel is the control channel and can carry the control information between user and network. It can also support lower-speed digital data requirements. The A channel is used in the transitional period to carry conventional analog voice signals. The C channel is used with the A channel to form a combined access channel.

These channel types are grouped into transmission structures that are offered as a package to the user. They are the basic access (basic channel structure), the combined access (hybrid access channel structure), and the primary rate access (primary channel structure). The Basic Access consists of 2B+D channels and the total bit rate is 192 kbps including synchronization and framing bits. The combined access consists of A+C channels where A is 4 khz analog channel and C is 8 or 16 kbps digital channel. The primary rate access which the US, Canada, and Japan support consists of 23B+D channels where B is 64 kbps channel and D is also 64 kbps channel. This corresponds to the T1 transmission facility, using the DS-1 transmission. The Europeans prefer their own 2.048 Mbps link containing 30B channels and one D channel, each at 64 kbps. In addition to the above, intermediate service, broadband service, and higher rate structures are being considered as standard offering.

2.3.3 User Access

This subsection briefly describes the current international trends in user access area. The ISDN reference points and functional grouping were defined by the Consultative Committee of International Telegraph and Telephone (CCITT). To define the requirements for ISDN user access, an understanding of the anticipated configuration of user premises equipment and of the necessary standard interfaces is critical. The functions of user's premises may be grouped using the reference points and functional groupings.

The CCITT suggested some kinds of functional groupings. They are Network Termination 1 (NT1), Network Termination 2 (NT2), Network Termination 1,2 (NT12), Terminal Equipment type 1 (TE1), Terminal Equipment type 2 (TE2), and Terminal Adapter (TA). The NT1 has functions that belong to the OSI layer 1, that is, functions associated with the physical and electrical termination of the ISDN on the user's premises. The NT2 is an intelligent device that may include up through OSI layer 3 functionality. The NT2 perform switching and concentration functions. The NT12 is a single device that contains the combined functions of the NT1 and NT2. The TE1 is the device that support the standard ISDN interface. Examples are a digital telephone, integrated voice/data terminal, and digital facsimile equipment. The TE2 encompasses existing non-ISDN equipment that requires a TA to plug into an ISDN interface.

The CCITT also defined the reference points. The ISDN reference points are S, T, R, and U. The S interface is between the TE1 and NT2. The T interface is between the NT2 and NT1. The U interface is between the NT1 and the outside of the user premises. The R interface is between the TE2 and TA. Figure 2.4 shows the CCITT approach to the ISDN reference points and functional groupings.

2.4 Internetworking Method

There exist many different kinds of computer networks around the world. Therefore, the interconnecting the various computer networks are needed so that any two stations on any of the constituent networks can communicate. There are four kinds of ways we can interconnect the different networks. They are Gateway (protocol

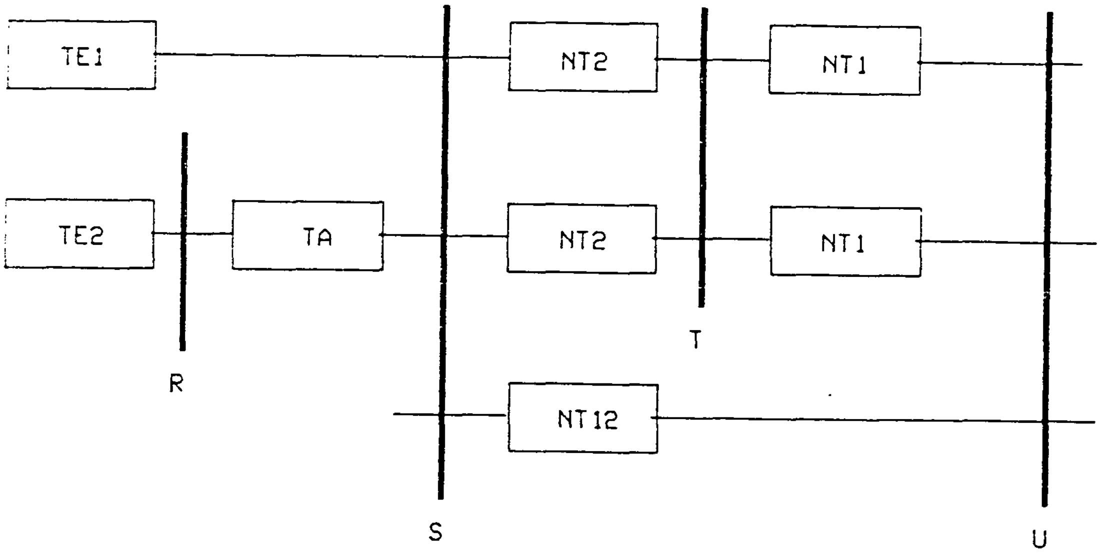

Figure 2.4: ISDN reference points and functional groupings

translator), Internet Protocol, Bridge, and X.75 [6]. The overall requirements on the internetworking facility are the followings: First, It should provide a link between networks. Secondly, It should provide for the routing and delivery of data between processes on different networks. Third, It should provide an accounting service that keeps track of the use of the various networks and gateways and maintains status information. Lastly, it should provide the services listed above in such a way as not to require modifications to the networking architecture of any of the constituent networks.

2.4.1 Bridge

The bridge is used to connect homogeneous networks. The bridge implements physical and data link layer protocols. The main functions of the bridge are to read all frames transmitted on source network and accept those addressed to destination network. Then it retransmits the frames onto the destination network using the medium access control protocol of the destination network.

2.4.2 Gateway

The gateway serves as a link between two dissimilar networks. For example, a gateway is needed to connect two independent local networks, or to connect a local network to a long-haul network. The gateway may include all seven layer functions. The main functions of the gateway is to solve the frame incompatibility problem and subaddressing problem between different networks.

2.4.3 Internet Protocol

The Internet Protocol (IP) is the name given to a protocol standard developed by DOD as part of the DARPA Internet Project. The IP provides a datagram service between stations. The IP can implement the end-to-end station level architecture using the IP layer protocol which resides above the network layer. The functions of IP layer are addressing, routing, fragmentation, reassembly, error control, and flow control.

2.4.4 X.75

The X.75 is a standard developed by CCITT as a supplement to X.25. It is used between public X.25 networks or between the private X.25 networks. The X.75 specifies Signal Terminating Equipment (STE) that acts as DCE-level gateways to connect two X.25 networks and defines the internetwork STE-STE interface. The functions of X.75 are to set up the Virtual Circuits across each network and route them by a relay mechanism.

2.5 CCITT Recommendation X.25 Virtual Circuit Protocol

The *Interconnection Protocol* design of the *Token Ring to ISDN Gateway* is based on the X.25 Virtual Circuit protocol. Therefore, the CCITT Recommendation X.25 is described briefly in this section.

2.5.1 Overview

In 1976, The CCITT approved the X.25 packet-switching network access standard which defines the method and sequence of communications over a packet-switched network or public data network. Now the X.25 is the best known and the most widely-used packet-switching protocol.

2.5.2 Structure

The X.25 conforms to the first three layers of the OSI model. They are Physical level, Link level, and Packet level. For the Physical level, the physical-level portion of X.21 is specified. Optionally (and at present more commonly), X.21 bis may be used; This is similar to RS-232-C. The Link level is LAP-B, which is a subset of the asynchronous balanced mode (ABM) of HDLC. The Packet level specifies a virtual-circuit service. The virtual-circuit service of X.25 provides for two types of virtual circuit: Virtual Call and Permanent Virtual Circuit. The structure of X.25 is shown in Figure 2.5.

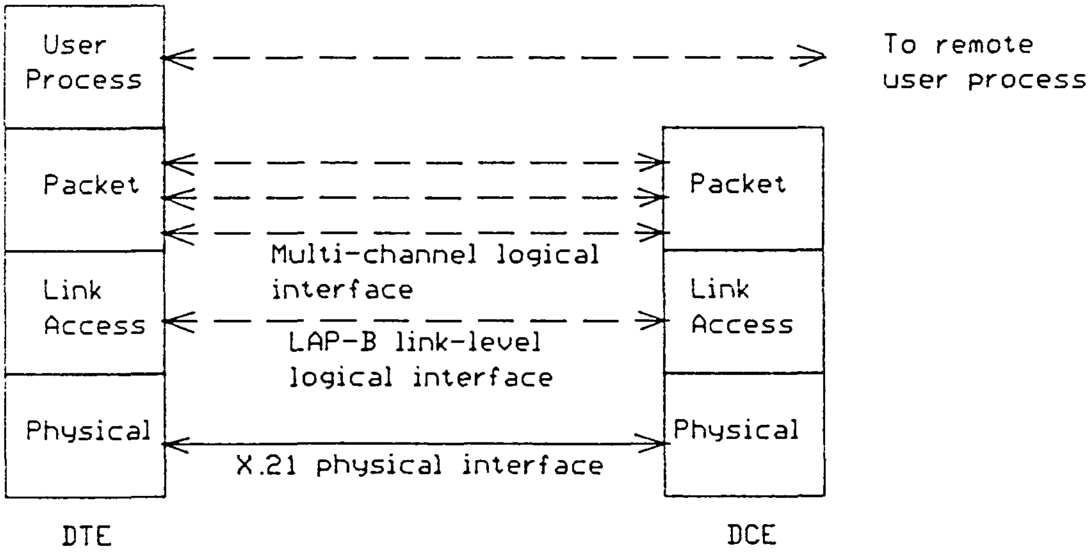

Figure 2.5: X.25 structure

2.5.3 X.25 Virtual Call Protocol

The basic services offered at the X.25 packet-level interface are Virtual Call (VC) and Permanent Virtual Circuit (PVC). A Virtual Call is a dynamically-established virtual circuit using a call-set up and call-clearing procedure. The Virtual Call is the typical traffic handled by the packet-switched network.

In the X.25 Virtual-Call service, the DTE constructs control packets and encapsulates data in data packets. These are then transmitted to the DCE via LAP-B. Thus the packet is encapsulated in a layer 2 frame (one packet per frame). The DCE strips off the layer 2 header and trailer and may encapsulate the packet according to some internal network protocol. The routing of packets between the DCEs is the responsibility of the internal logic of the network.

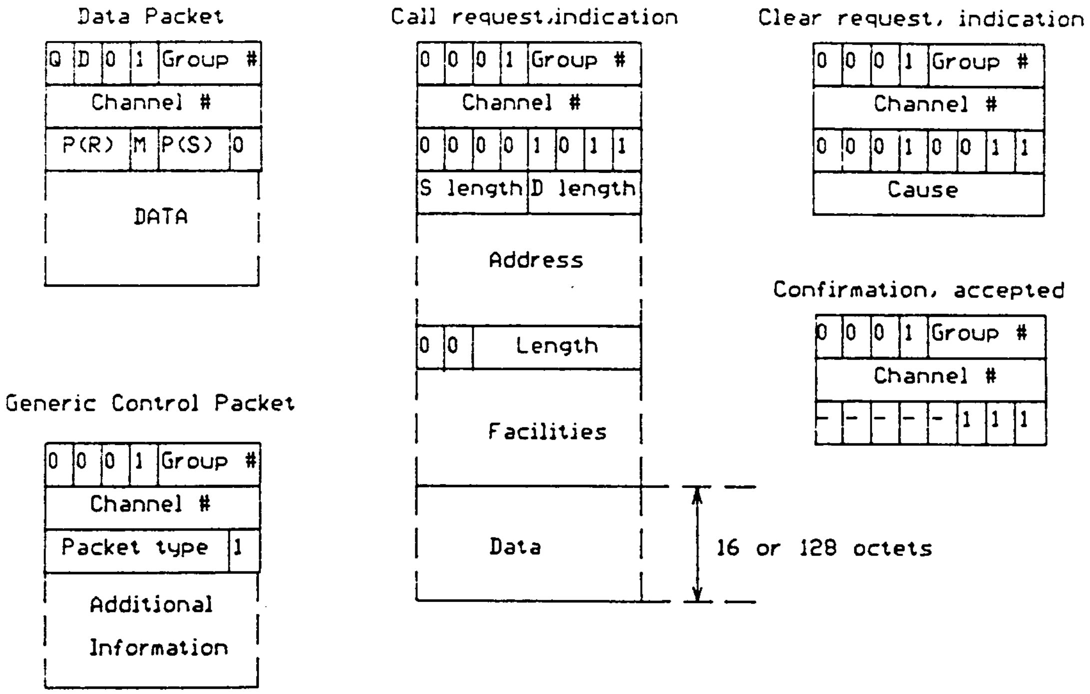

Figure 2.6: X.25 packet formats

The user data transmitted through the network has to be in a well-defined format called a packet. The packet is a sequence of binary digits, consisting of data and control elements arranged in a special format, which is transmitted as a composite whole. The packet may be of any length up to some fixed maximum, and must include, within its control information, identifying end-point station address or Logical Channel Number. In order to implement the Packet level, information about the Virtual-Call service has to be provided to the correct destination. Figure 2.6 shows X.25 packet formats.

2.6 LAN-ISDN Interconnection

Before describing the design of a *Token Ring to ISDN Gateway*, the general LAN-ISDN interconnection problems are discussed in this section.

2.6.1 Motivation

The most obvious reasons for connecting a LAN to the ISDN are: to interconnect LANs that are located on two remote sites, to connect a LAN to the ISDN in order to connect any ISDN terminal to this LAN, and to make any ISDN service available to all LAN customers. In the first two cases, existing LANs can be connected to the ISDN, requiring only moderate changes to the LANs. In these cases, they are connected via a gateway to the S interface of the ISDN.

2.6.2 LAN-ISDN Connection Problems

In the S Interface configuration, the major problems which can be caused when connecting a LAN to the ISDN are like followings. The first problem is the inconsistency of addresses which is caused because the LAN uses only layer 2 (data link layer) addresses, whereas the ISDN also uses layer 3 (network layer) addresses. The second problem is the incompatibility of frame structures. The last problem is how to handle the subaddressing within the LAN. Above problems should be overcome for the LAN-ISDN connection. There are three types of solutions for these problems [7]. They are the Mapping Layers solution, the Gateway solution, and the Internet Protocol Layer solution.

2.6.3 Gateway Solution

Of the above three solutions, the gateway solution is adopted for the implementation of the *Token Ring to ISDN Gateway.* The gateway can solve the frame incompatibility problem by removing all frame structure elements. The gateway can also solve the subaddressing problem by requiring that the first in-slot received data conforms to a set of command words that the LAN gateway can interpret to direct the following data to an appropriate address within the LAN. The gateway solution can fit within the OSI reference model.

2.6.4 LAN Adaptation

If an existing LAN must be connected to the ISDN, some adaptations to the LAN must be done. First, a gateway to the S interface must be provided that uses the access method of specific LAN. But more important is the introduction of an internet layer in all terminals of the LAN, to analize the destination address and forward the data to the gateway or to the local destination. Another important function of the internet layer is the fragmentation or reassembly of the data in order to give acceptable sized packets to the ISDN.

However, if the LANs that are to be connected have a uniform frame structure and a unique addressing scheme (e.g., when connecting two Token Rings over the ISDN like this research), no internet layer is needed. Since the frame structure are the same, no fragmentation/reassembly at the internet layer is necessary, and the gateway can easily determine the ISDN network number of the destination LAN, since each has a unique addressing number.

3 HARDWARE DESIGN OF THE *TOKEN RING TO ISDN GATEWAY*

3.1 Overview

This hardware design chapter describes the details of the *Token Ring to ISDN Gateway* design issues. The *Token Ring to ISDN Gateway* connects the Token Ring to the ISDN environment. By using this gateway, we can send (receive) the digital data, PCM voice, and video data to (from) the ISDN environment. The functions of the *Token Ring to ISDN Gateway* are:

- to exchange data and control signal between the token ring and the ISDN.
- to solve the frame incompatibility problem between the Token Ring and the ISDN.
- to solve the subaddressing problem by using the Interconnection Protocol.
- to provide S interface at the ISDN side and token ring interface at the token ring side.

3.2 Gateway Design

3.2.1 ISDN Access Point Decision

Connecting a LAN to the ISDN requires an internetworking facility ("gateway") in the LAN to perform the necessary signaling adaptation functions and the rate conversion. The ISDN therefore perceives the LAN as a whole a data terminal equipment with an ISDN S or T or U interface. Therefore,there are three different user-network interface configurations we can use for the LAN to ISDN interface. They are the S interface configuration, the T interface configuration, and the U interface configuration. Each configuration has different problems to be solved [7].

Among the above configurations, the S interface configuration can be used as a local distributed interface. The local distributed interface provides multidrop connections where several devices are directly connected to the same termination at S interface. This configuration does not impose an undue functional or economic burden on single point-to-point connections, either on the network termination device or on devices designed to only connect to a single termination [8]. In the S interface, the master-slave method is used for the network synchronization. In this method, a primary reference clock (the master clock) controls all of the exchanges directly or via intermediate stages and thus determines the synchronization clock frequencies. Therefore, the slaves need not have the own clock and get the clock from the master. In our design, the S interface configuration is used.

3.2.2 ISDN Channel Decision

As discussed in Subsection 2.3.2 , ISDN has three kinds of channel structure. They are the basic channel structure, the hybrid access channel structure, and the primary channel structure. We have decided to use the basic channel structure which has 2B+D channels and is the simplest structure among them. The reason why we selected the basic channel structure is the simple feature of this structure and the availability of the basic channel structure devices.

3.2.3 Gateway Configuration

The hardware of the gateway consists of TI (Texas Instruments)'s TMS380 Token Ring Adapter, *Gateway Controller*, and the *ISDN S Interface Circuit*. The *Gateway Controller* (IBM PC/XT) controls the TMS380 which is the token ring side interface and the *ISDN S Interface Circuit* which is the ISDN side interface.

The IBM PC/XT works as a main controller of the gateway. This runs the *Gateway Driver* which performs many gateway functions. This also handles the interrupts from TMS380 and *ISDN S Interface Circuit*. The interface between the *Gateway Controller* and the TMS380 and ISDN S Interface Circuit is through PC Bus of the IBMPC/XT. The TMS380 and the ISDN S Interface Circuits are inserted in the expansion slots of the IBM PC/XT mother board. Because we don't have real ISDN environment, MITEL's MB89000 ISDN Express Card^{TM} Design Kit with ISDN Evaluation System (IES) software was used as the simulation of real ISDN environments [9]. Figure 3.1 shows the block diagram of the *Token Ring to ISDN Gateway* with the ISDN Express Card^{TM}. In this configuration, the *Token Ring to ISDN Gateway* operates as the slave of the S interface and the MB89000 operates as the master of

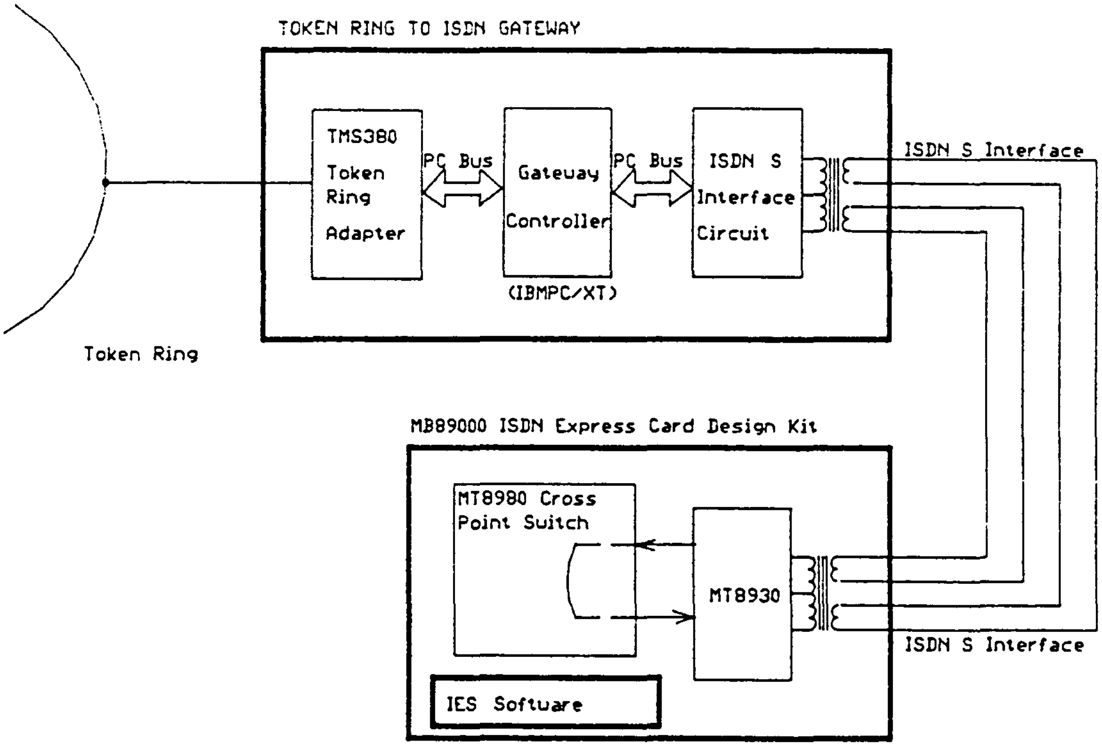

Figure 3.1: The block diagram of the *Token Ring to ISDN Gateway* with the ISDN Express Card^{TM}.

the S interface.

3.3 Gateway Controller

The *Gateway Controller* is the IBMPC/XT which controls the TMS380 Token Ring Adapter and *ISDN S Interface Circuit*. The main function of the *Gateway Controller* is to control the data exchange between the token ring and the ISDN, to solve the frame incompatibility problem, to solve the subaddressing problem, and to provide the flow control and error control.

The *Gateway Controller* handles two interrupts. One interrupt is from the TMS380 Token Ring Adapter which is asserted when TMS380 has received data from the token ring or is ready to receive data from the *Gateway Controller* or wants to report the status of the TMS380. During the data interchange, this interrupt occurs in every 64 msec on the average. This interrupt uses IRQ4 interrupt. Another interrupt is from the MT8930 of the *ISDN S Interface Circuit* which is asserted in every 125 usec. This interrupt rate is due to the data rate of the ISDN basic channel structure. This interrupt uses IRQ2 interrupt. The details of how these interrupts are handled and what these interrupt service routines do are described in detail at Section 4.4. The details of frame conversion, subaddressing, data transfer, data buffering, flow control, and error controls are described in Section 4.2.

3.4 TMS 380 Token Ring Adapter

This section describes the TMS380 Token Ring Adapter cards which is the token ring side interface of the *Token Ring to ISDN Gateway.* The TMS380 cards were supplied by Texas Instruments. The block diagram of the TMS380 Token Ring Adapter card is shown in Figure 3.2.

3.4.1 TMS380 Hardware Configuration

The TMS380 architecture integrates the functions of a token ring adapter into a five-chipset. The five chips are:

- TMS38030 System Interface (SIF) chip.
- TMS38010 Communications Processor chip.

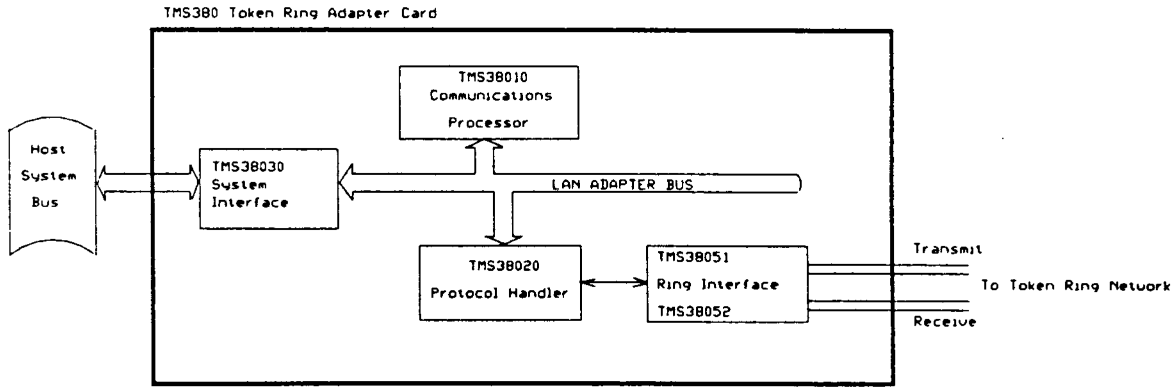

Figure 3.2: The block diagram of the TMS380 token ring adapter cards

- TMS38020 Protocol Handler chip.
- TMS38051 Ring Interface Transceiver.
- TMS38052 Ring Interface Controller.

The details of the TMS380 are in [10].

3.4.2 Frames

A message is passed through the token ring in a format referred to as a "frame". The frame format used in TMS380 is the standard format of IEEE 802.5 standard. The information carried by frames may be one of the two types:

- Medium Access Control (MAC) information.
- Non-Medium Access Control information (e.g., Logical Link Control (LLC) information).

Exchange of MAC frames is transparent to the attached system. These frames are handled by the TMS380 card itself. Therefore, the user doesn't need to consider the MAC frames. The frame which can be handled by the user is the LLC frame. In this gateway design, the LLC frame is used for the *Interconnection Protocol.* The details of the LLC frame design will be discussed in Subsection 4.2.3.

3.4.3 TMS380 to Token Ring Interface

The TMS380 Token Ring Adapter connects to the token ring through two twisted pairs of wires: one set to transmit and one set to receive. These twisted pairs connect through a wiring concentrator to form lobes. Each lobe is serially connected to the network to form a ring topology. The wiring concentrator provides the electrical and mechanical function necessary to physically insert and de-insert the ring station onto the ring. This part of interface is totally done by TMS380 and thus the task of interfacing to the token ring is reduced to write the driver program for the adapter card. This means that the TMS380 makes the user free from the details of PHY and MAC layer of the token ring.

3.4.4 TMS380 to IBMPC/XT Interface

The TMS380 Token Ring Adapter interface to the *Gateway Controller* is through the TMS38030 System Interface chip (SIF). This interface provides two user selectable modes: 808x mode and 680xx mode. In this gateway design, 808x mode is used.

The System Interface of the TMS380 provides both direct I/O and direct memory access (DMA) interfaces to the attached system. Direct I/O is used to handshake with the adapter and to pass initialization parameters to the TMS380 Adapter. All message transfers between the attached system and the ring are accomplished via the DMA interface. This provides for high-performance data transfer to and from the attached system. The TMS380 can provide four kinds of data transfer interface techniques to the attached system. They are I/O Mapped Design, Memory Mapped Design, IBMPC/AT DMA Design, and 68000 Bus Interface Design. In this gateway design, we used the Memory Mapped Design technique which provides an interface to the 8-bit PC bus that uses string moves through a memory mapped register. The efficiency gained in using the string move instruction within the interrupt service routine allows the average data rate through the PC I/O channel to be 2.8 megabits-per-second for the IBMPC/XT. In this design, the TMS38030's SHRQ (hold request) DMA signal is used as a IRQ4 (interrupt request4) signal to the IBMPC/XT. This causes the PC to be interrupted, and execution of an interrupt service routine is started. This interrupt service routine is discussed in Section 4.2.8. The block diagram of this Memory Mapped Design is shown in Figure 3.3.

3.5 *ISDN S Interface Circuit*

This section describes the design of the *ISDN S Interface Circuit* which is the ISDN side interface of the *Token Ring to ISDN Gateway.* The *ISDN S Interface Circuit* was designed and assembled during this research. The block diagram of the *ISDN S Interface Circuit* is shown in Figure 3.4.

3.5.1 *ISDN S Interface Circuit* Configuration

This circuit consists of the PC-Bus Interface part and the ISDN S Interface part. This circuit was assembled on the JDR PR-1 prototype card and inserted into one

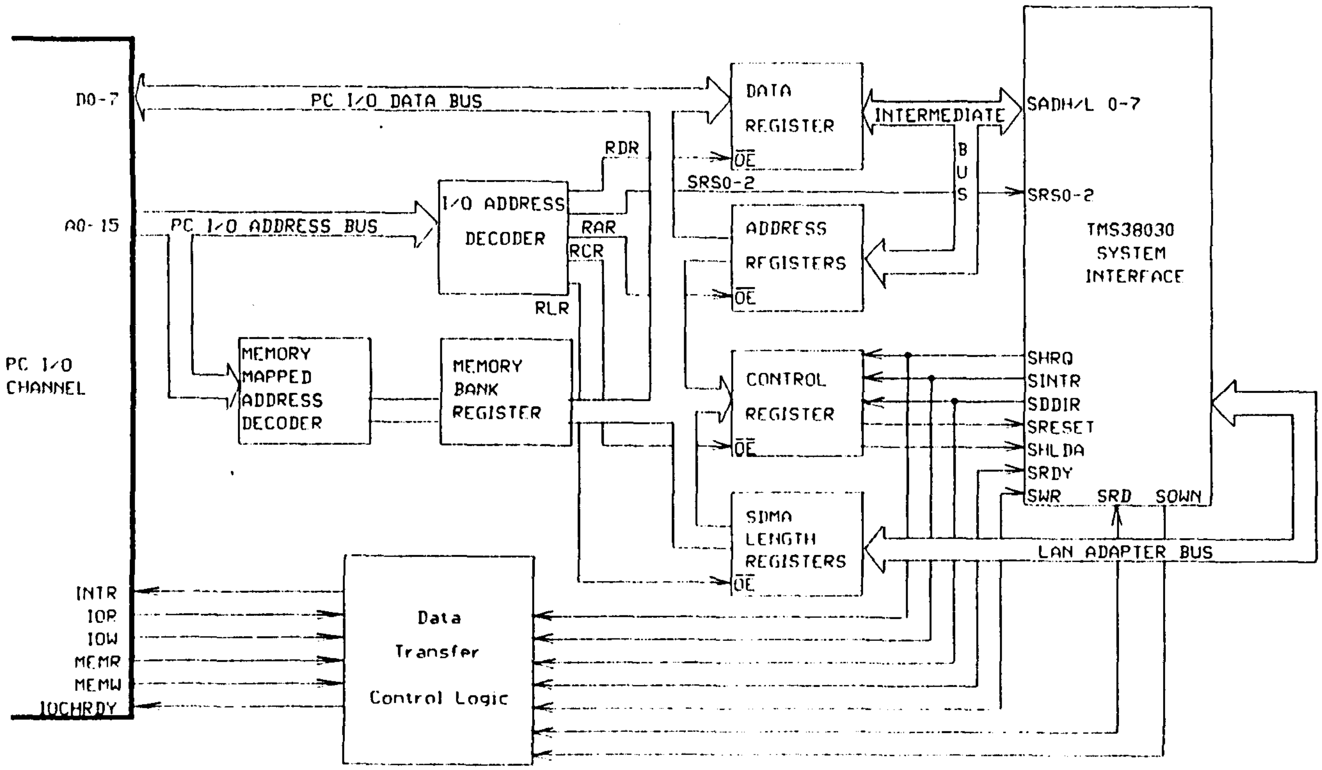

Figure 3.3: TMS380 to *Gateway Controller* interface block diagram

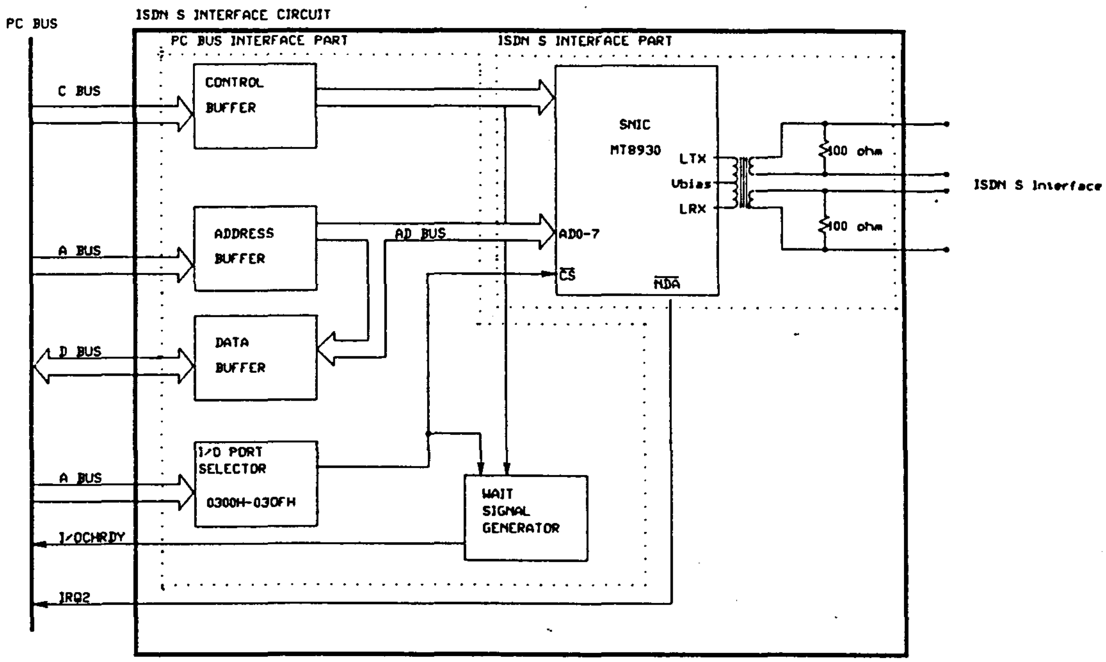

Figure 3.4: The block diagram of the *ISDN S Interface Circuit*

of the expansion slots of the IBMPC/XT [11]. This circuit generates one interrupt signal from the MT8930 ISDN S interface chip. This interrupt signal is activated in every 125 usec and connected to the IRQ2 signal of the PC-Bus. Some techniques are used to modify the interrupt priority. These are discussed in detail in Subsection 4.2.8.

3.5.2 PC-Bus Interface Part

This interface converts the demultiplexed data-address bus signal of the PC-Bus into the multiplexed data-address bus signal for the MT8930 and vice versa. This interface also has the buffers for the control signal of the PC-Bus and the I/O port selection function. In this design, the I/O address range of this *ISDN S Interface Circuit* are from 0300H to 030FH.

3.5.3 ISDN S Interface Part

The main chip of this part is MITEL's MT8930 Subscriber Network Interface Circuit (SNIC) [12]. SNIC is a device which implements the CCITT I.430 Recommendation for the ISDN S and T reference points. The selected configuration and the modes of SNIC for this circuit are point-to-point configuration, TE mode, microprocessor control mode, and NDA mode. In TE mode which is selected by lowering the pin 8 (NT/$\overline{TE}$) of SNIC, SNIC is configured as a Terminal Equipment (TE). At this TE mode, the SNIC operates as a slave to the MB89000 (master) which is configured as a Network Termination (NT) and the clock of the MT8930 is phased-locked to the line data signal from NT. The microprocessor control mode is selected by raising the pin 7 (Cmode). In this mode, the pin 15 through pin 22 are assigned as a multiplexed data-address bus and the *Gateway Controller* can access and control this chip by I/O operation. The NDA mode is selected by giving the suitable command to the Master Control Register of SNIC. In this mode, pin 13 is assigned as New Data Available ($\overline{NDA}$) signal which indicates the access time when the synchronous registers of SNIC are available. The New Data Available signal is lowered in every 125 usec and has 50 % duty cycle. This signal is inverted and used to strobe the interrupt request 2 signal (IRQ2) of the *Gateway Controller.*

The SNIC supports 192 kbit/s (2B+D+overhead) full duplex data transmission on a 4-wire balanced transmission line. Transmission capability for both B and D channels, as well as related timing and synchronization functions are provided on the chip. An HDLC transceiver is included on the SNIC for link access protocol handling via the D-channel. Depacketized data is passed to and from the transceiver via the microprocessor port. Two 19 byte deep FIFOs, one for transmit and one for receive, are provided to buffer the data. In this design, B channel is used for data transmission and D channel is used for control transmission. B channels are accessed through the synchronous registers and D channel is accessed through the HDLC transceiver. The C (overhead) channel provides a means for the system to control and monitor the functionality of the SINC. The data rates of C channel is 48 Kbits/s.

The line code used on the S interface is a Pseudo Ternary code with 100 % pulse width . Binary zeros are represented as marks on the line and successive marks will alternate in polarity [13]. The output driver characteristics of the SNIC requires a 2 : 1 transformer with the electrical specifications as provided by the PE 64350 (Pulse Engineering #PE64350, San Diego, California, USA) [12]. The transformers used in this circuit are PE 64995 transformers which have the similar characteristics

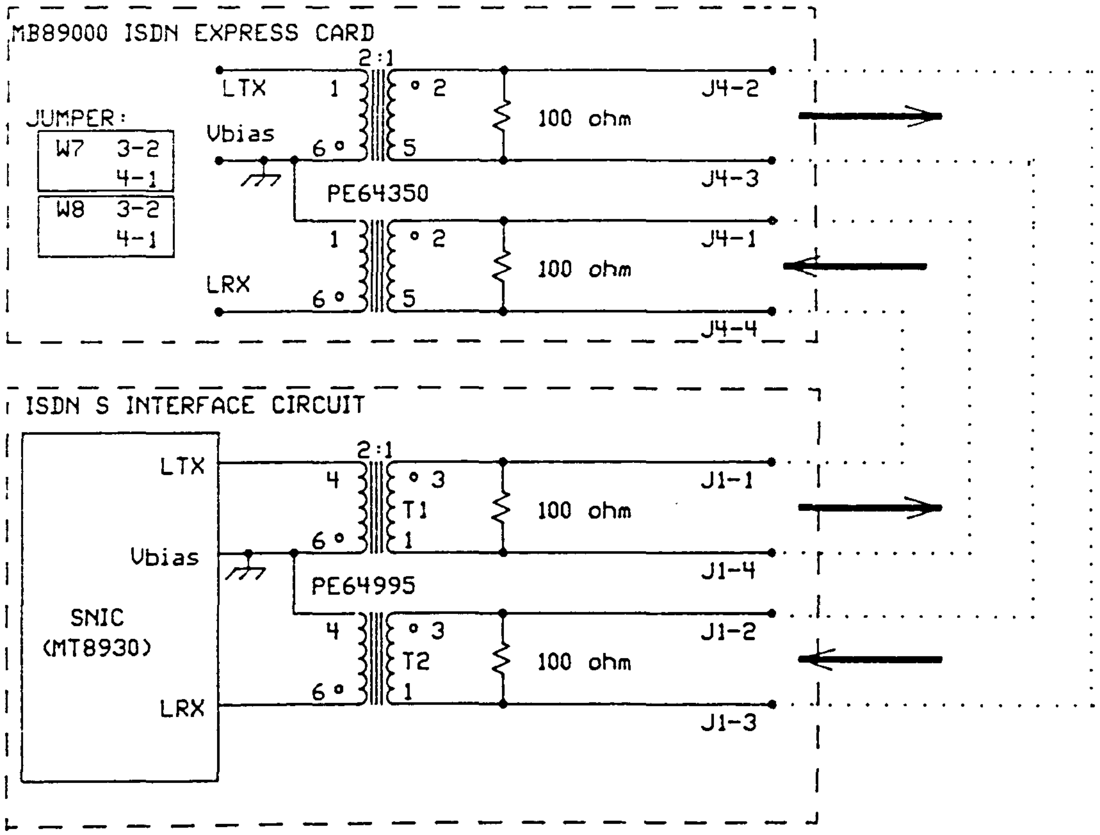

Figure 3.5: Interface between *ISDN S Interface Circuit* and MB89000 Card^{TM}

to PE 64350 transformers. The connection between the transformers of the *ISDN S Interface Circuit* and the transformers of the MB89000 ISDN Express Card^{TM}is shown in Figure 3.5. The Figure 3.6 shows the circuit diagram of the *ISDN S Interface Circuit*.

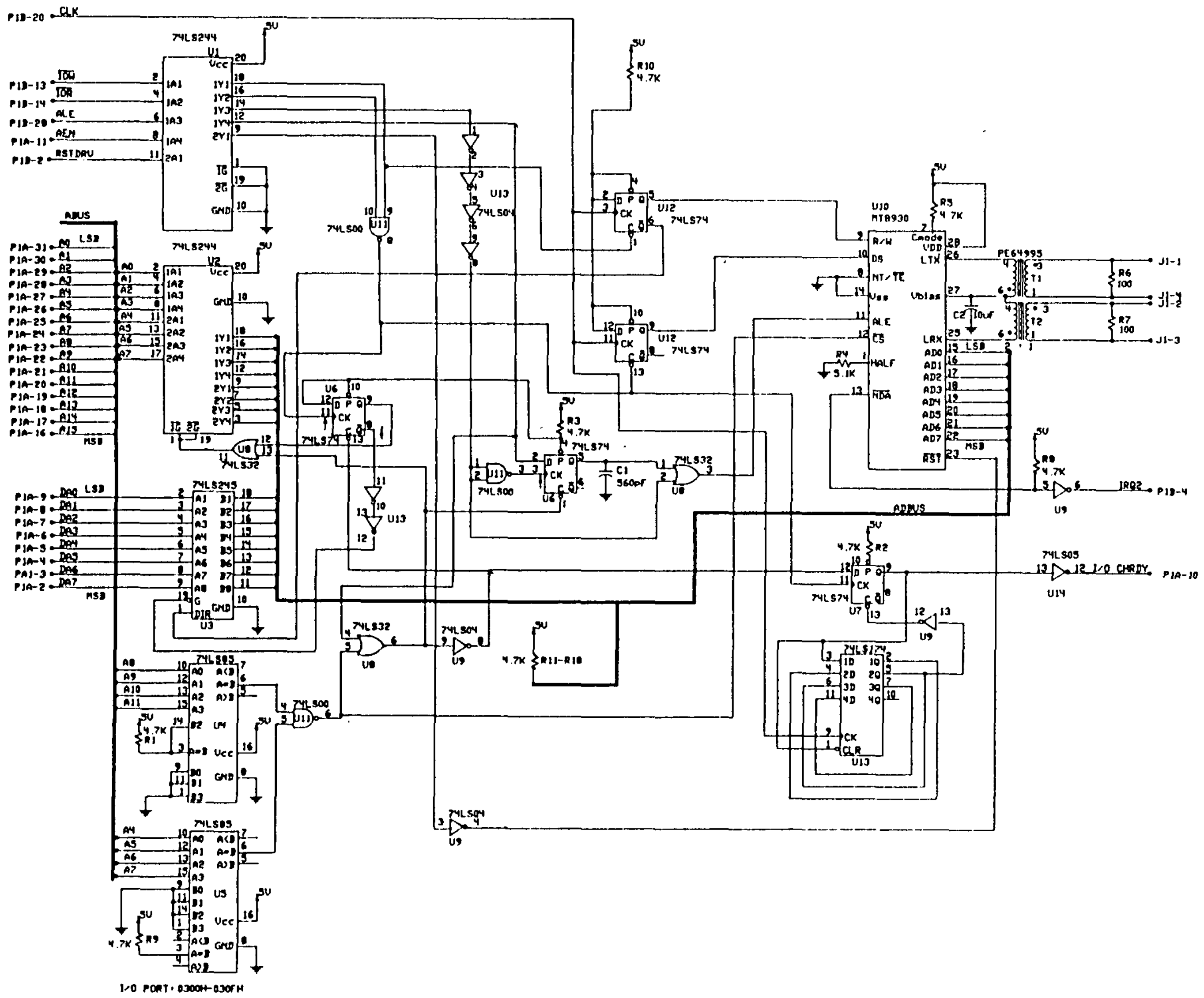

Figure 3.6: Circuit diagram of the *ISDN S Interface Circuit*

4 *INTERCONNECTION PROTOCOL* DESIGN

4.1 Overview

This chapter describes the details of the *Interconnection Protocol* design. The purpose of the *Interconnection Protocol* is to transfer a file from one token ring station to another token ring station across the ISDN through the *Token Ring to ISDN Gateway*. The *Interconnection Protocol* was realized by the *File Transfer program* which runs on the Token Ring File Transfer stations and the *Gateway Driver* which runs on the *Token Ring to ISDN Gateway*. This *Token Ring to ISDN Gateway* can handle any types of digital data including PCM voice, video data, and file data. But in this research, only the file transfer function is implemented by writing a *File Transfer Program*.

The design of the *Interconnection Protocol* is based on the CCITT X.25 Virtual Call protocol which is described in Section 2.5. This protocol works between same token rings which are connected through the ISDN.

4.1.1 Network Architecture

The network architecture of the *Token Ring to ISDN Gateways*, Token Ring File Transfer stations, and the ISDN Switch station is shown in Figure 4.1. As shown in Figure 4.1, the *Interconnection Protocol* covers the OSI Network layer and the part

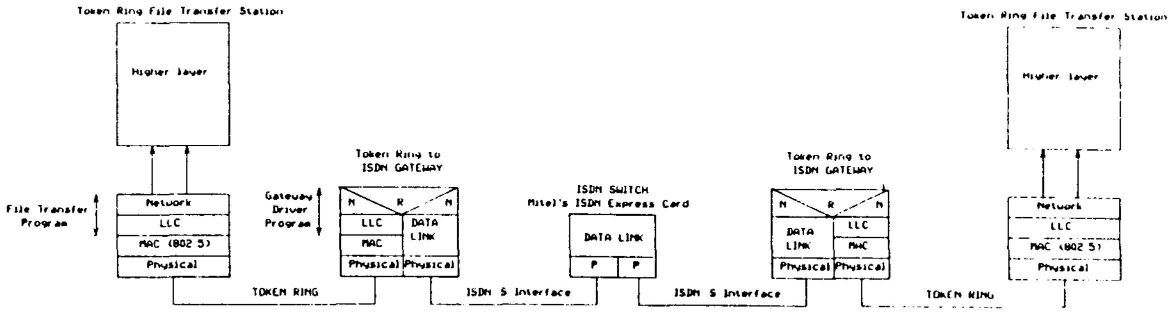

Figure 4.1: Network architecture of the *Token Ring to ISDN Gateway* network

of Data Link layer (LLC layer). This protocol is implemented by the *File Transfer Program* and *Gateway Driver*. The *Token Ring to ISDN Gateway* runs *Gateway Driver* and the File Transfer station runs the *File Transfer Program*. The MAC layer and the Physical layer of the token ring side interface at the Token Ring File Transfer station and the *Token Ring to ISDN Gateway* are implemented by the TMS380 Token Ring Adapter. The Data Link layer and the Physical layer of the ISDN side interface at the *Token Ring to ISDN Gateways* are implemented by the SNIC (MT 8930) chip.

4.2 *Interconnection Protocol* Design Detail

4.2.1 Virtual Call Protocol

The *Interconnection Protocol* implements the Virtual Call packet level which is described in Subsection 2.5.3. A Virtual Call is a dynamically established virtual circuit using a call setup and call clearing procedure. Figure 4.2 shows a sequence of events in this protocol. The File Transfer station and the *Token Ring to ISDN Gateway* are connected through the token ring network. The interface between the *Token Ring to ISDN Gateway* and the ISDN Switch is the ISDN S interface.

The sequence of events is as follows:

1. File Transfer Tx station (A) requests a virtual circuit to File Transfer Rx station (D) by sending a Call Request packet (0BH) to the *Token Ring to ISDN Gateway* Tx station (B). The packet includes source and destination addresses. Future incoming and outgoing data transfers will be identified by the virtual circuit number which chooses the B1 channel or B2 channel using routing table of the *Token Ring to ISDN Gateway* Tx station (B).

2. B assigns the virtual circuit number, saves it in the routing table, and routes this Call Request packet to the *Token Ring to ISDN Gateway* Rx station (C) across the ISDN D channel. However, if station C finds both B1 and B2 channels are used already, it sends the Call Rejected packet (09H) back to the station A. Then, station A gives up to transfer the file.

3. The station C receives the Call Request packet, sets up the routing table, and sends a Call Indication packet (0BH) to station D. This packet has the same

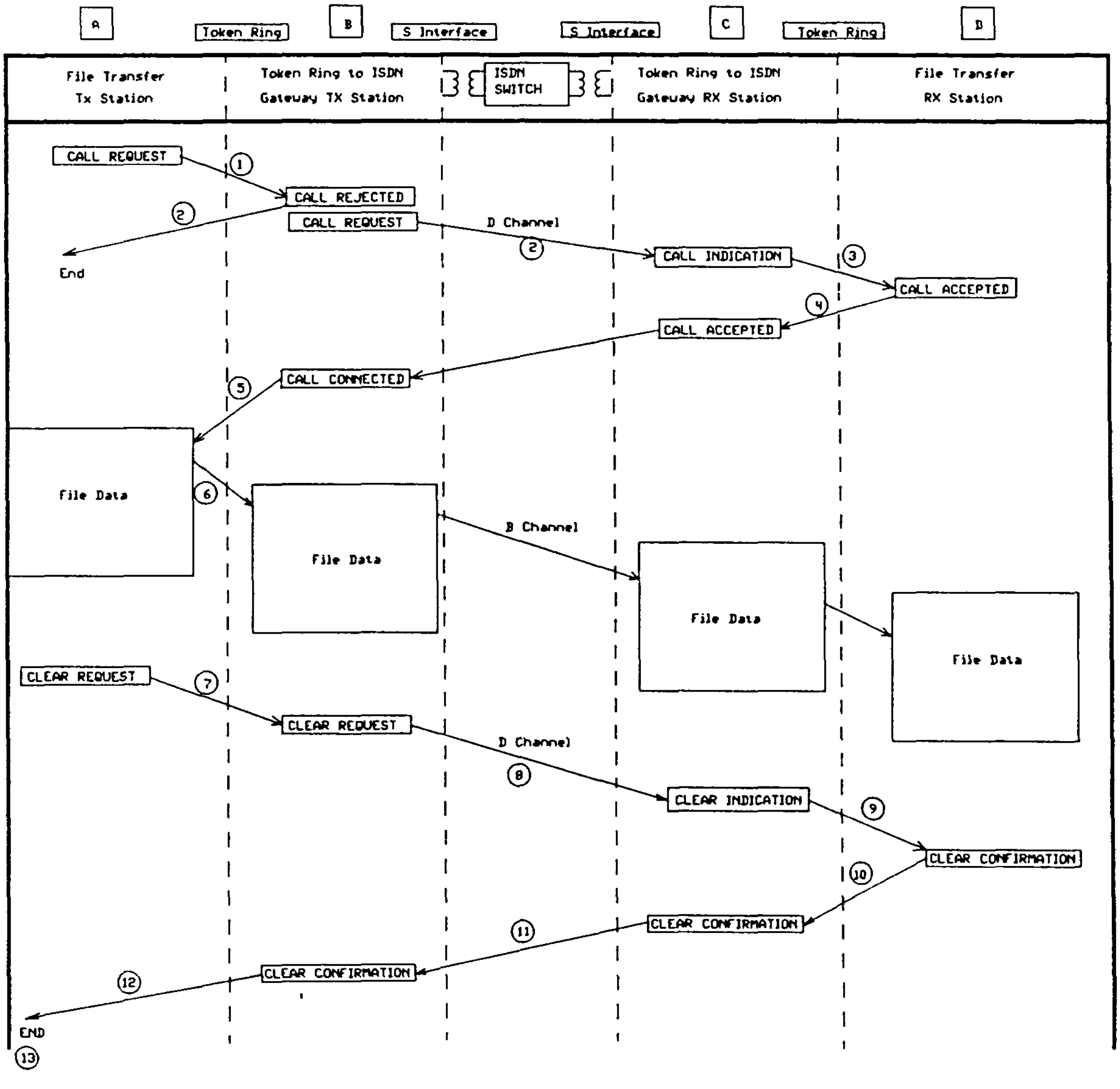

Figure 4.2: Sequence of events - *Interconnection Protocol*

format as the Call Request packet.

4. The station D indicates acceptance of the call by sending a Call Accepted packet (08H) specifying the same virtual circuit number as that of the Call Indication packet.
5. The station A receives a Call Connected packet (08H) with the same virtual circuit number as that of the routing table of station B.
6. The station A sends data to the station D through the ISDN B channel. The station B and C use routing tables to route the data to the correct destination station.
7. The station A sends a Clear Request packet (13H) to terminate the virtual circuit and waits for the Clear Confirmation (16H).
8. The station B sends a Clear Request packet to the station C through the ISDN D channel.
9. After receiving a Clear Request packet, the station C sends Clear Indication (13H) packet to the station D.
10. After receiving a Clear Indication packet, the station D finishes the file transfer job and sends Clear Confirmation (16H) packet to the station C.
11. The station C sends Clear Confirmation packet to the station B through the ISDN D channel.
12. The station B renews the routing table and sends a Clear Confirmation packet to the station A.

13. After receiving Clear Confirmation packet, the station A finishes the file transfer job completely.

The *Interconnection Protocol* packet level described above is very similar to that of X.25 Virtual Call [14]. But this protocol does not have the flow control and error control functions which X.25 Virtual Call has.

4.2.2 Frame Incompatibility Problem

Two types of physical frames are used for the *Interconnection Protocol.* One is the Token Ring LLC Frame which meets the IEEE 802 standard format and exists on the token ring network. The other is the ISDN S-Bus Frame which meets the Recommendation I.430 format and exists on the ISDN S interface. Figure 4.3 shows where the Token Ring LLC Frame and the ISDN S-Bus Frame exist. The format of these two frames are different. Therefore, the *Token Ring to ISDN Gateway* should solve this frame incompatibility problem. The details of each frame format is described in Subsection 4.2.3 and 4.2.4 respectively. Figure 4.4 shows how this frame incompatibility problem can be solved at the *Token Ring to ISDN Gateway.* The File Transfer TX Station sends two types of Token Ring LLC Frames. One is the control frame which includes the control data. The other is the data frame which includes the actual file data. When the *Token Ring to ISDN Gateway* receives the Token Ring LLC Control Frames from the token ring side, it saves only the control data part out of the control frame into the RECDATA buffer temporarily. Then it sends the control data to the HDLC Transceiver which constructs the HDLC Frame with these data, disassembles this HDLC Frame, and puts them on the ISDN S-Bus Frame. On the other hand, if the *Token Ring to ISDN Gateway* receives the Token Ring LLC Data

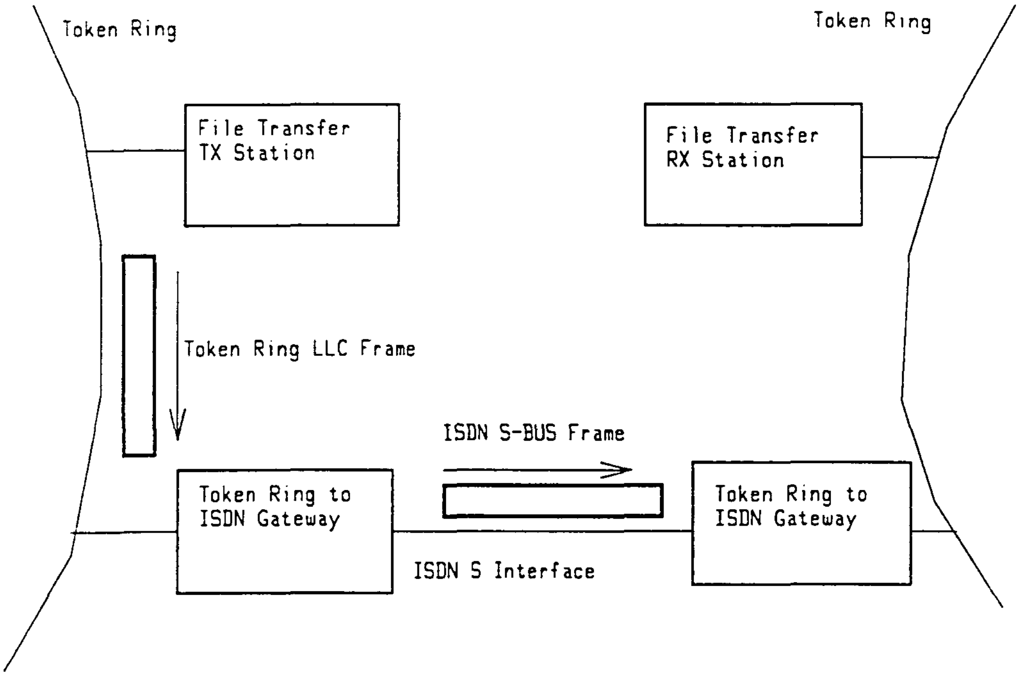

Figure 4.3: Token Ring LLC frame and ISDN S-Bus frame

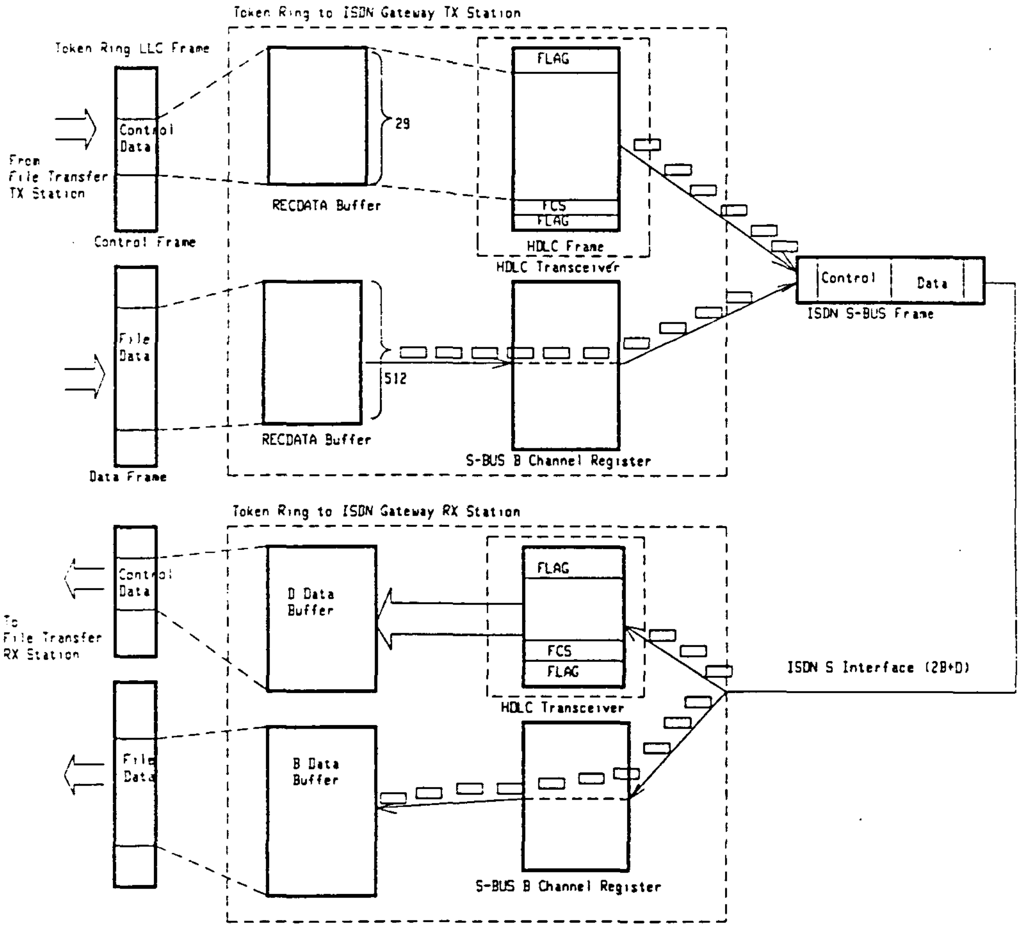

Figure 4.4: Frame incompatibility solution at the *Token Ring to ISDN Gateway*

Frames from the token ring side, it saves only the 512 bytes long file data part out of the data frame into the RECDATA buffer momentarily. Then it sends the file data to the S Bus B Channel Register byte by byte. Thess file data are then put on the ISDN S-Bus Frame. Therefore, the ISDN S-Bus Frame contains the control data and file data on one frame. Actually, the ISDN S-Bus Frame carry the control data, file data, and control/status signal through the 2B+D+C channels.

When the *Token Ring to ISDN Gateway* RX Station receives the ISDN S-Bus Frame from the ISDN S interface, it splits the frame into control data and file data. It then accumulates them in the D Databuff and B Databuff respectively. When all HDLC Frame is accumulated in the D Databuff, the gateway makes Token Ring LLC Control Frame with the HDLC Frame and sends it to the token ring network. On the other hand, when 512 bytes of file data are accumulated in the B Databuff, the gateway makes Token Ring LLC Data Frame with those file data and sends it to the token ring network.

4.2.3 Token Ring Frame

The *Token Ring to ISDN Gateway* exchanges data with the other token ring stations which are on the same token ring network by means of Token Ring Frame [15].

4.2.3.1 Frame Types A message is passed through the token ring network in a format referred to as a "frame". The physical frame format is strictly defined by the Token Ring Architecture. The information carried by frames may be one of two types.

- Medium Access Control (MAC) information
- Non-medium Access Control information (e.g., Protocol Data Units destined for the Logical Link Control sublayer)

Between those two types of frames, the *Interconnection Protocol* handles only the LLC frames.

The services provided by the Token Ring Adapter provide a comprehensive set of problem determination, resolution, and reporting functions, so that ring communication problems are rapidly diagnosed and automatically corrected. These functions are carried out through exchange of MAC frames. Exchange of MAC frames is transparent to the *Interconnection Protocol.* The LLC implemented by the *Interconnection Protocol* and the MAC layer implemented by the Token Ring Adapter provide the Data Link Control layer of the ISO Open Systems Interconnection reference model for a local area network communication system.

4.2.3.2 Frame Format The physical frame format is strictly defined by the Token Ring Architecture and consists of the following fields within the frame's bit stream:

- Starting Delimiter Field (SDEL)
- Physical Control Fields (PCF) consisting of the Access Control (AC) and Frame Control (FC) fields
- Source and Destination Address Fields (SA/DA)
- Information Fields (data)

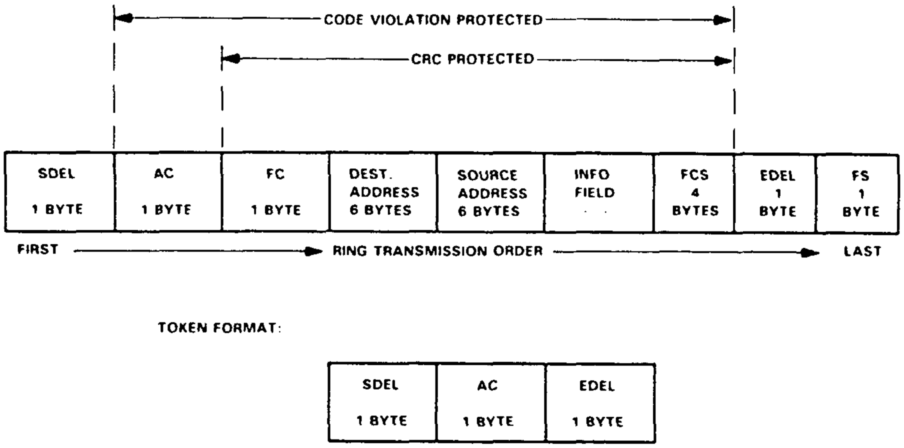

Figure 4.5: Token Ring frame format

- Frame Check Sequence (FCS)
- Ending Delimiter (EDEL)
- Frame Status Field (FS)

The frame format is shown in Figure 4.5. The *File Transfer Program* should fill the AC, FC, SA, DA, and information field of the Token Ring Frame. The AC field contains the priority, token indicator, monitor count and priority reservation. SA field is six bytes long and contains the address of the File Transfer TX Station. DA field is also six bytes long and contains the address of the *Token Ring to ISDN Gateway* (777777777777H) if the destination of the frame is on the other token ring network across the ISDN. If the destination is on the same token ring network, the DA field contains the address of the File Transfer RX Station on the same token ring network. The information field contains the data which needs to be sent to the File

Transfer RX Station across the ISDN. There are two types of Token Ring Frames; control frame and data frame. The control frame brings the control data which are needed for Virtual Call Protocol. The data frame brings the file data. As shown in Figure 4.6, the information field of the control frame includes group number, control type, cause, source address, and destination address fields. The details of each field are described in Subsection 4.2.5. The data frame has 512 bytes data field in addition to the control frame.

4.2.4 ISDN Frame

The *Token Ring to ISDN Gateway* sends and receives the data through the ISDN S interface by means of the ISDN Frames.

4.2.4.1 Frame Types As described in Subsection 4.2.2, the *Token Ring to ISDN Gateway* uses the ISDN S-Bus Frame as the physical frame on the ISDN S interface. The *Token Ring to ISDN Gateway* also uses the HDLC Frame which is the logical frame between two HDLC Transceivers when the gateway sends control data through the D channel. That is to say, the HDLC Frame is the Data Link layer frame and the ISDN S-Bus frame is the Physical layer frame.

4.2.4.2 ISDN S-Bus Frame Format The S-Bus is a four wire, full duplex, time division multiplexed transmission facility which exchanges information at 192 kbit/s rate including two 64 kbit/s PCM voice or data channels, a 16 kbit/s signalling channel and 48 kbit/s for synchronization and overhead. The SNIC makes use of three types of channels to transmit (receive) control/status and data to (from) the S interface port. These are the B, D, and C channels. The ISDN S-Bus frame structure

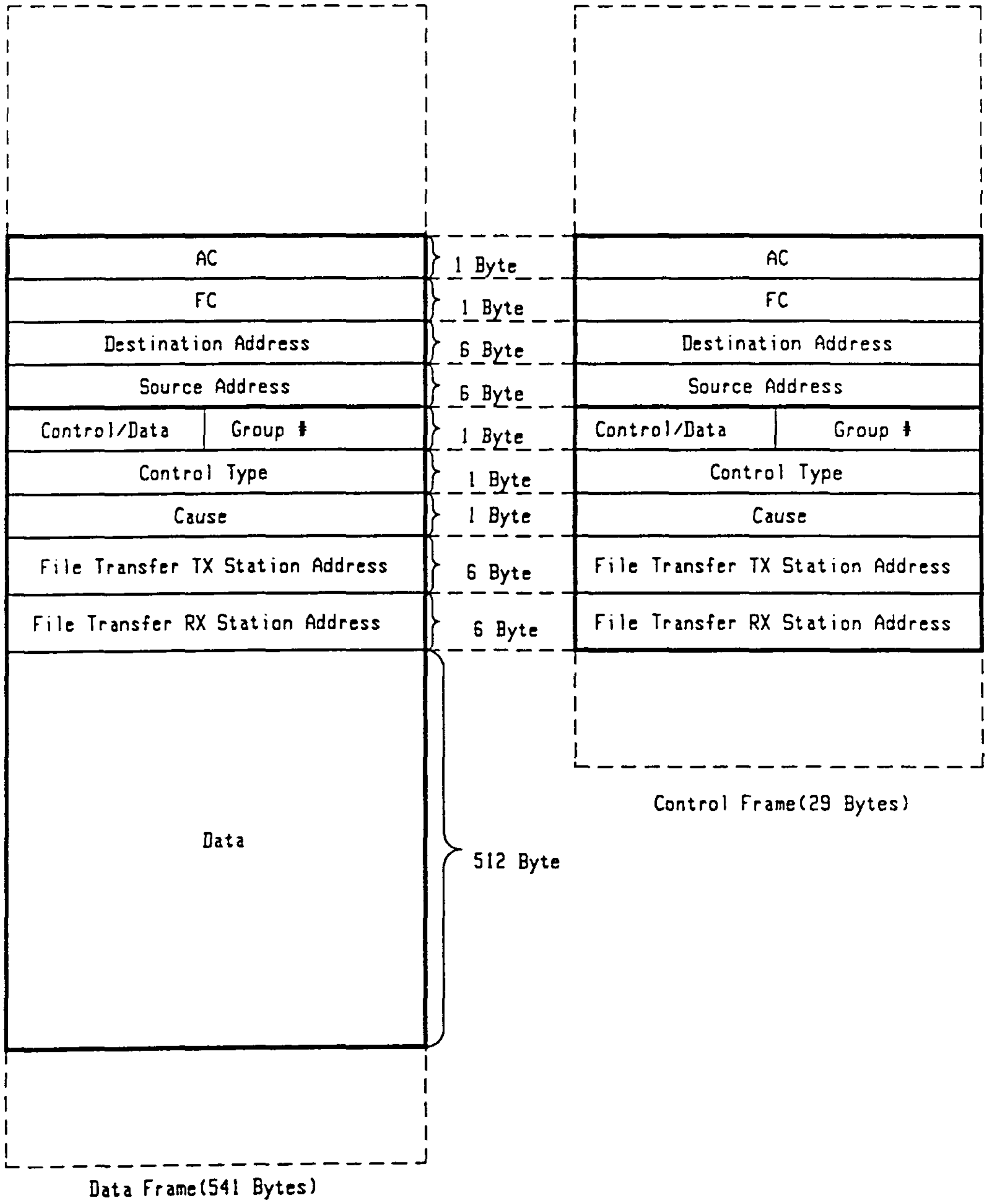

Figure 4.6: Token Ring control frame and data frame

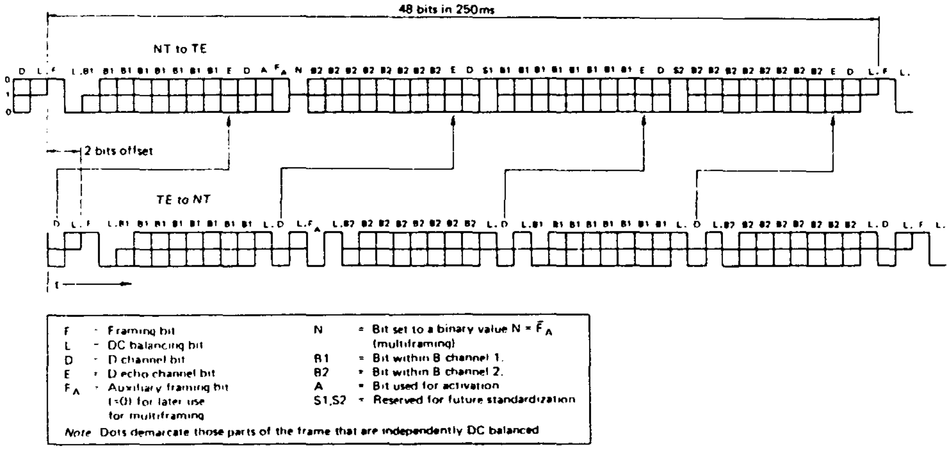

Figure 4.7: S-Bus frame structure

transmitted by the NT and TE contains the followings and shown in Figure 4.7.

NT to TE ISDN S-Bus Frame:

- Framing bit (F)
- B1 and B2 channels (B1,B2)
- DC balancing bits (L)
- D channel bits (D0,D1)
- Auxiliary framing and N bit (Fa, N)
- Activation bits (E)
- Multiframing bit (M)
- Spare bit (S)

TE to NT ISDN S-Bus Frame:

- Framing bit (F)
- B1 and B2 channels (B1, B2)
- DC balancing bits (L)
- D channel bits (D0, D1)
- Auxiliary framing bit (Fa) or Q-channel

The ISDN S-Bus frame is 48 bits long which includes 16 bits B1 channel, 16 bits B2 channel, 4 bits D channel, and 12 bits C channel. The transmit rate of this frame is 192 kbit/s. Therefore, it takes 250 usec to transmit one ISDN S-Bus Frame. The details of the ISDN S-Bus Frame is in [16].

4.2.4.3 HDLC Frame The HDLC Transceiver handles the bit oriented protocol structure and formats the D channel as per level 2 of the X.25 packet switching protocol defined by CCITT. It transmits and receives the packetized data serially in a format shown in Figure 4.8. The details of the HDLC Frame are handled by the HDLC Transceiver of the SNIC. Therefore, the *Interconnection Protocol* only needs to give some command to the SNIC by writing a command word to the HDLC Control Register. The details are in Subsection 4.4.3.

4.2.5 *Interconnection Protocol* Packet Format

The *Interconnection Protocol* implements the Virtual Call protocol which was described in Subsection 4.2.1. A variety of packet types are used for this Virtual Call

FLAG	DATA FIELD	FCS	FLAG
1 Byte	N Bytes(N >= 2)	2 Bytes	1 Byte

Figure 4.8: HDLC frame format

protocol. The function of each packet is described already in Subsection 4.2.1 and the formats of each packets are shown in Figure 4.9. These packets are the information fields of the Token Ring Frame. These information fields are used for Virtual Call setup and clearing between the File Transfer Stations if they are control frames. This information fields contains the file data if they are data frames.

4.2.6 Gateway Addressing

The network used in this research interconnects the same token ring networks through the ISDN S interface. In this network, the addressing mechanism of both token ring networks are same and the *Token Ring to ISDN Gateways* are treated as one of the token ring station whose address is 777777777777H. Therefore, when the File Transfer TX Station wants to send the data across the ISDN, it first sends the data to the *Token Ring to ISDN Gateway* which is one of the token ring station. Then the *Token Ring to ISDN Gateway* sends it through the ISDN S interface. To do

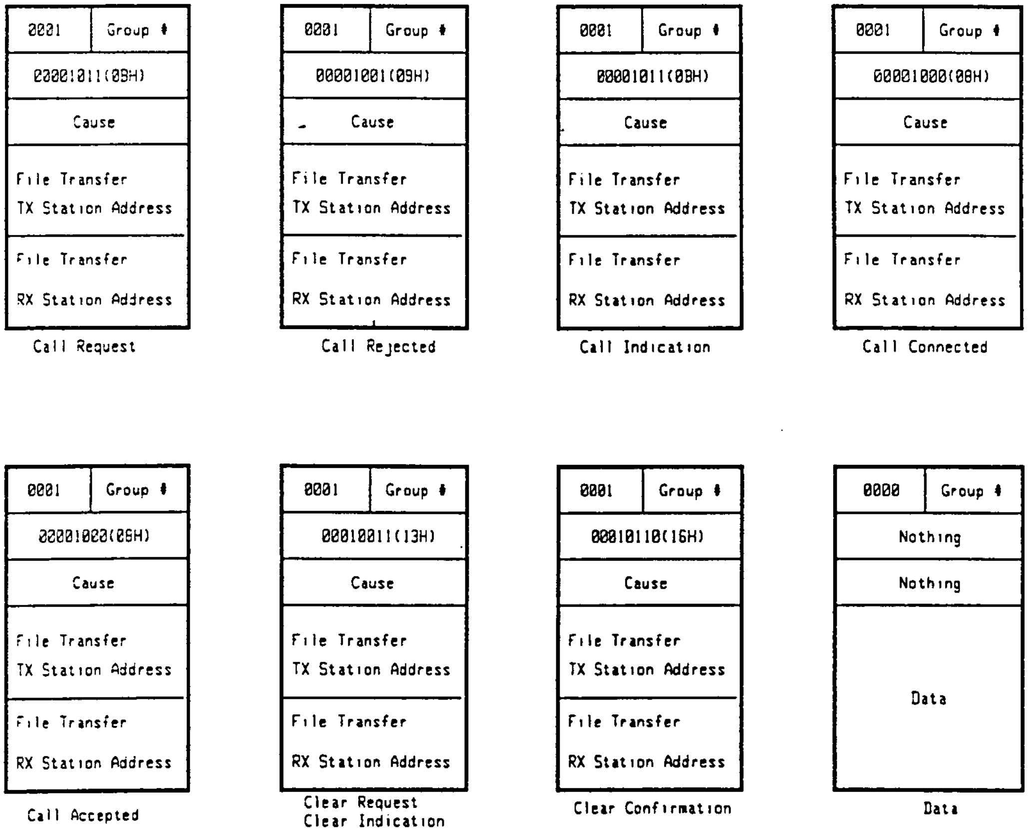

Figure 4.9: *Interconnection Protocol* packet format

Status
B1 Group Number
B1 Control
B1 Cause
B1 TX Address High
B1 TX Address Mid
B1 TX Address Low
B1 RX Address[6]
B2 Group Number
B2 Control
B2 Cause
B2 TX Address High
B2 TX Address Mid
B2 TX Address Low
B2 RX Address[6]

Figure 4.10: Routing table

this, the DA field of the token ring frame contains 777777777777H and the RX field contains the File Transfer RX Station address. Figure 4.11 shows how the frames are converted when they go through the *Token Ring to ISDN Gateway.*

4.2.7 Routing Table

The *Token Ring to ISDN Gateway* has the routing table which is shown in Figure 4.10. The routing table contains the information which is needed when the *Token Ring to ISDN Gateway* routes the data frame to the correct ISDN channel. When the gateway receives the Call Request frame or Call Indication frame, it checks if there is available B channel. If there exists available B channel, it selects one of the B channels for this file transfer operation. If not, it sends Call Rejected frame back to the File Transfer TX Station. Then it stores this routing information in the routing table, establishes the Virtual Call connections, and constructs the ISDN Frame or Token Ring Frame for the next coming file data. The gateway can route the data frame to the correct channel by using the information in the routing table. The gateway uses virtual circuit number (VC#) in order to select the correct routing table. The details how this is done is in Figure 4.11.

4.2.8 Token Ring Device Driver Modification

The File Transfer stations and the *Token Ring to ISDN Gateway* load the Token Ring Device Driver (Family4.dev) program at its system bootup time. The Token Ring Device Driver was originally written by the Texas Instrument and provided with the TMS380 Token Ring Adapter. The name of the Token Ring Device Driver is "Skysup.asm" which is written in 8088 assembly language [2].

In this research, this program is used as the Token Ring Adapter driver with a few modification. The modifications were done because the original Token Ring Adapter was designed to use the interrupt 2 signal but the Token Ring Adapter in the *Token Ring to ISDN Gateway* uses interrupt 4 signal. Therefore, the interrupt vector table and the interrupt masking of the interrupt controller should be changed. The *Token Ring to ISDN Gateway* uses interrupt 2 signal for the ISDN S interface communication which needs higher priority than the Token Ring Adapter communication. The File Transfer Stations use the original Token Ring Device Driver without modification.

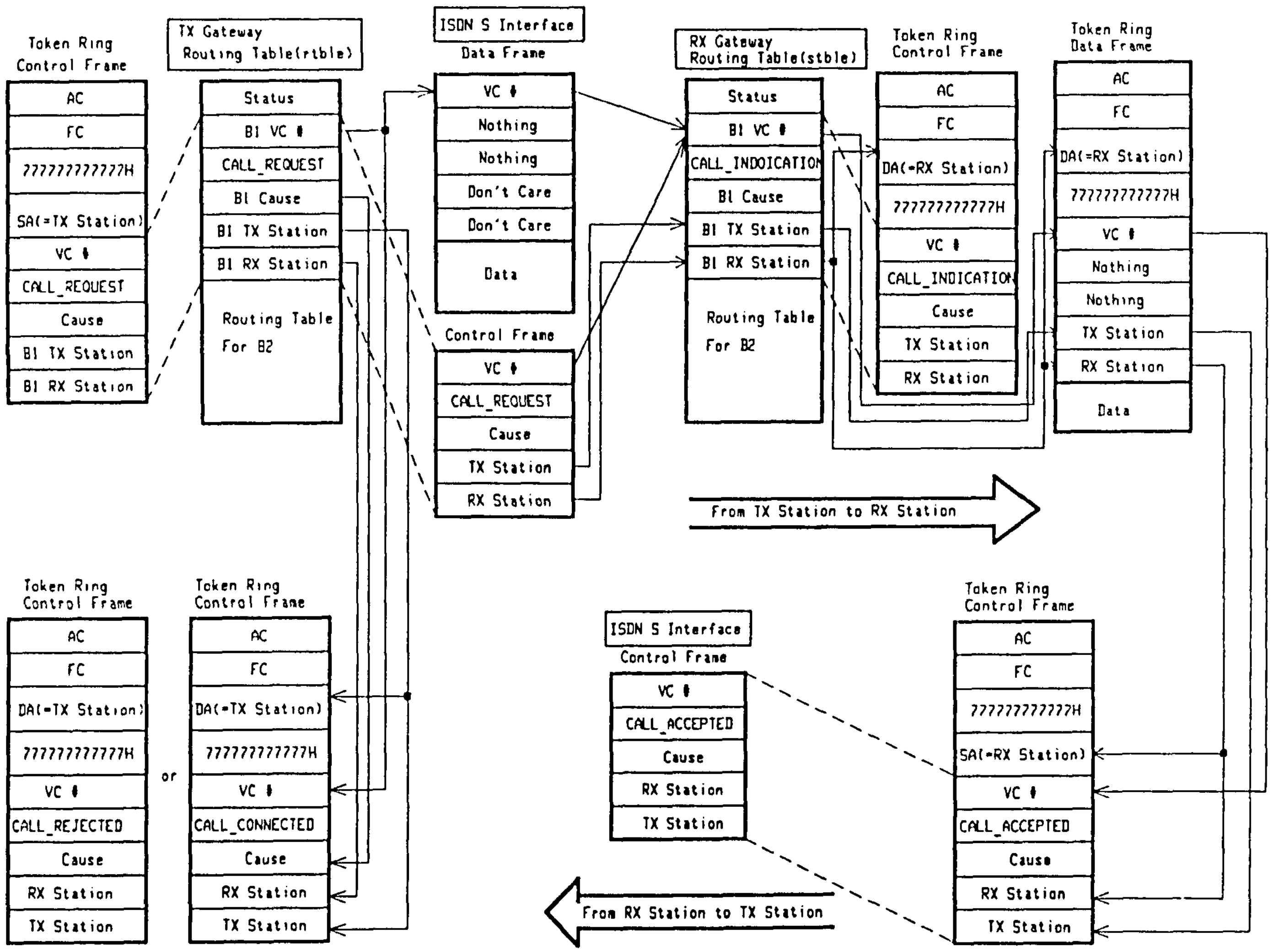

Figure 4.11: Frame conversion using routing table

4.3 *File Transfer Program* Design

4.3.1 Overview

The *File Transfer Program* performs the file transfer function between two token ring stations and resides at the File Transfer TX Stations and the File Transfer RX Stations. This program can be used in one token ring network or two token ring networks across the ISDN. The *File Transfer Program* establishes the Virtual Call connection between the two File Transfer Stations and reads one file from the disk driver of the File Transfer TX Station and sends it to the File Transfer RX Station. The File Transfer RX Station then reads the file and stores it to the disk driver of the File Transfer RX Station. This program doesn't have any error control and flow control facilities.

4.3.2 Structure

The *File Transfer Program* consists of seven C program files, one 8088 assembly language file, and one header file. The functions of each file is in Table 4.1.

4.3.3 Files

In this subsection, the details of each file is described.

4.3.3.1 Main.c This file has two routines which are main () and tics (). Main () is a main routine of the *File Transfer Program.* This routine first gives the message on the screen and waits for the command from the keyboard. The command options are;

Table 4.1: *File Transfer Program* files

File Name	Functions
Main.c	Selects the program running option and control the program flow
Init.c	Initializes the Token Ring Adapter
Open.c	Opens the Token Ring Adapter
Trans.c	Reads the file and transmits it to the token ring
Recv.c	Receives the file from the token ring and stores it to disk
Callreq.c	Does Virtual Call control function for the File Transfer TX Station
Callacpt.c	Does Virtual Call control function for the File transfer RX Station
Ttt.asm	Has some Token Ring Adapter interface routines and time measure routines
Ring.h	Includes the definitions and the data structures

- Initialize and open
- Transmit station
- Receive station
- Exit

At Initialize and open option, the Token Ring Adapter is initialized and opened. At Transmit station option, the File Transfer station works as a transmit station. The transmit station first establishes the call connection, reads the file from the disk driver, sends it to the File Transfer RX Station, and clears the connection. It also measures the file transfer delay. At receive station option, the File Transfer station works as a receive station. The receive station first establishes the call connection, receives the file from the ISDN interface, and stores it into the disk. At Exit option, the control returns to DOS.

4.3.3.2 Init.c This routine initializes the TMS380 Token Ring Adapter. It also allocates and initializes the System Command Block (SCB) and System Status Block (SSB).

4.3.3.3 Open.c This routine opens the TMS380 Token Ring Adapter by writing OPEN command to the adapter.

4.3.3.4 Trans.c This file includes transmit () and send () routines. Transmit () routine reads 512 bytes of data from the disk file at a time and sends them to the *Token Ring to ISDN Gateway* repeatedly until the end of the file is read. Send () routine gives TRANSMIT command to the TMS380. The flow chart of the transmit () routine is in Appendix A.

4.3.3.5 Recv.c This file includes receive (), initrecvlst (), and freerecvlst () routines. Receive () routine receives the token ring frames and stores the 512 bytes of data to the disk driver repeatedly until the end of the files are received. Initrecvlst () routine initializes the receive buffer of the Token Ring Adapter which has the linked list type data structure. Freerecvlst () routine makes the receive buffer available in order to make it possible for the Token Ring Adapter to receive one more data frame.

4.3.3.6 Callreq.c This file includes callreq () and clearreq () routines. Callreq () routine makes Call Request control frame, sends it to the File Transfer RX Stations, and waits until Call Confirm control frame is received. Clearreq () routine makes Clear Request control frame, sends it to the File Transfer RX Station and waits until Clear Confirmation control frame is received.

4.3.3.7 Callacpt.c This file includes callaccept (), clearconfirm (), and framemodify () routines. The callaccept () routine waits until Call Indication control frame is received, modifies the frame by calling the framemodify () routine, and sends Call Accepted control frame to the File Transfer TX Station. Clearconfirm () routine waits until Clear Indication control frame is received, modifies the frame by calling the framemodify () routine, and sends Clear Confirmation control frame to the File Transfer TX Station. Framemodify () routine modifies the token ring frame by exchanging TX and RX fields and DA and SA fields of the frame.

4.3.3.8 Ttt.asm This file includes mmiow (), mmior (), and gett () routines. Mmiow () routine writes the data to the TMS380 Token Ring Adapter. Mmior () routine reads the data from the TMS380 Token Ring Adapter. Gett () routine gets the time from DOS by calling the DOS function (INT 21H).

4.4 *Gateway Driver* Design

4.4.1 Overview

The *Gateway Driver* is a control program for the *Token Ring to ISDN Gateway*. This program exchanges the data and control signal between the token ring and the ISDN. This program manipulates two B channel data and one D channel control data. Figure 4.12 shows the operation block diagram of this program.

For the ISDN B1 (B2) channel data receive manipulation, this program reads

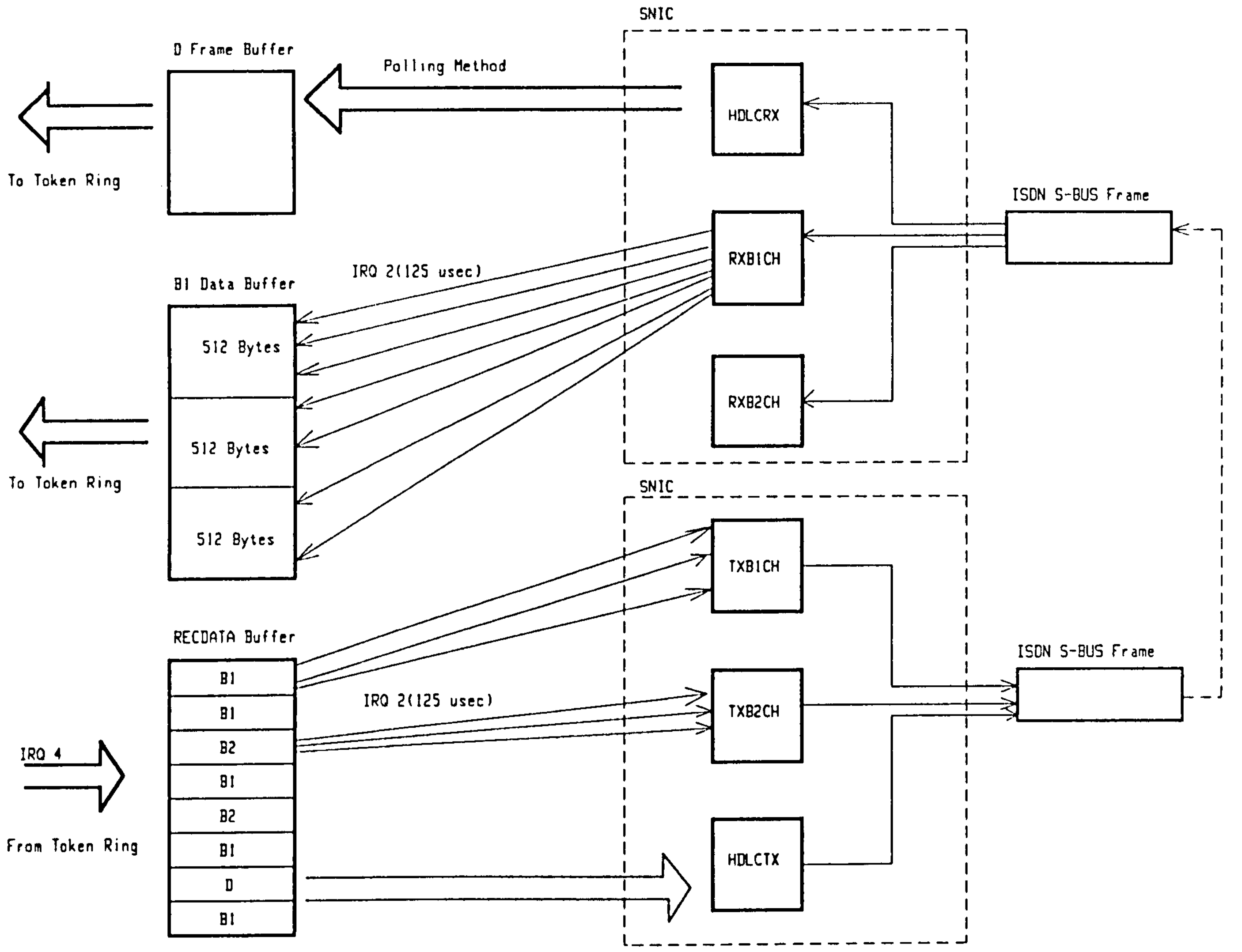

Figure 4.12: *Gateway Driver* operation

the one byte B1 (B2) data from the SNIC RXB1CH (RXB2CH) register when the interrupt 2 signal (IRQ2) is asserted in every 125 usec and accumulates it to the B1DATA (B2DATA) circular buffer. If the received data accumulates to the 512 bytes, it makes one token ring frame and sends this frame to the Token Ring Adapter. For the ISDN D channel control data manipulation, polling method is used. In the main () routine, the status of SNIC FIFO is continuously checked until the Receive FIFO is filled with the whole control data frame. Then it makes one token ring control frame and sends this frame to the Token Ring Adapter.

For the ISDN B1,B2, and D channel data transmit manipulation, this program gets the one data frame from the TMS380 buffer and puts it to the RECDATA buffer whenever interrupt 4 signal (IRQ4) is asserted. At the same time this program checks if RECDATA buffer is filled and what type of data is in RECDATA. If the data are B type data, they are sent to the TXB1CH (TXB2CH) buffer in every 125 usec when interrupt 2 signal (IRQ2) is asserted byte by byte. If the data are D type data, they are sent to the HDLCTX FIFO at once. Then SNIC makes ISDN S-Bus Frame with these data and sends them to the ISDN S interface.

4.4.2 Structure

The *Gateway Driver* consists of seven C program files, two 8088 assembly language files, and one header file. The function of each file is in Table 4.2.

4.4.3 Data Structure

The *Token Ring to ISDN Gateway* has separate data structures for the ISDN data transmit and the ISDN data receive. The operations on these data structures

Table 4.2: *Gateway Driver* files

File Name	Functions
Main.c	Controls the program flow
Init.c	Initializes the Token Ring Adapter
Open.c	Opens the Token Ring Adapter
Beg.c	Initializes the gateway data structure
Recv.c	Receives the data from the Token Ring Adapter and puts them in RECDATA buffer and fills the B1RXBUF (B2RXBUF) buffer
Tx.c	Sends the data which are received from ISDN to the token ring
Isdnd.c	Handles the ISDN D channel control data
Tt.asm	Has Token Ring Adapter interface routines
Intsvc.asm	Has a interrupt 2 service routine and some interrupt handling routines

are already described in Subsection 4.2.2 and Subsection 4.4.1. The details of the data structure for the B1 data which are received from ISDN and transmitted to the token ring are in Figure 4.13. The *Token Ring to ISDN Gateway* has the same data structure for B2 data and similar data structure for D control data to the data structure for B1 data.

The detail of the data structure for the data which are received from the token ring and transmitted to the ISDN is in Figure 4.14. This data structure handles all B and D type data.

4.4.4 SNIC Parallel Port

The *Gateway Driver* controls the SNIC through the SNIC parallel port. By writing or reading the SNIC parallel port, the *Gateway Driver* controls the HDLC transceiver and accesses all data, control, and status registers. The internal registers

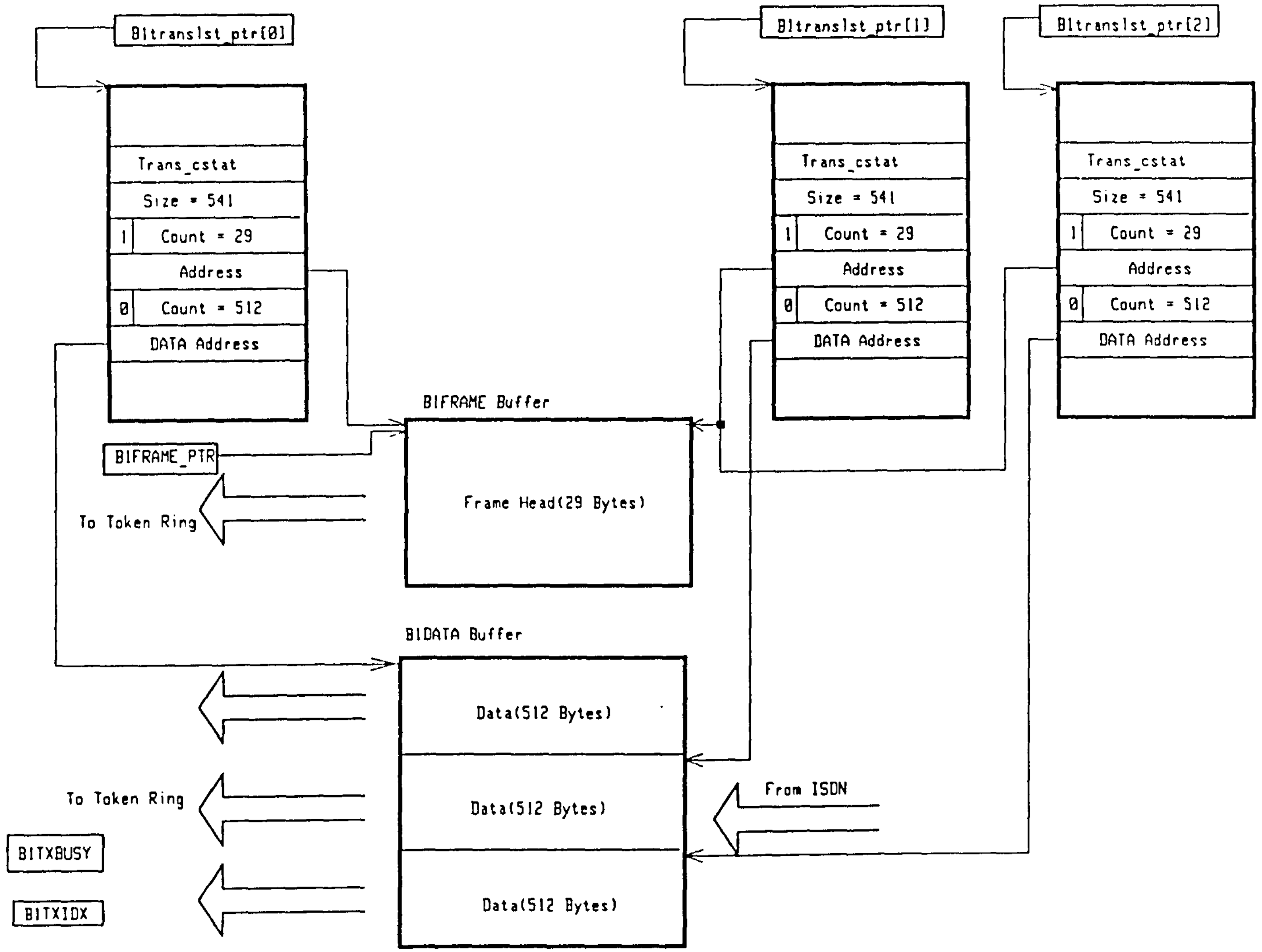

Figure 4.13: Token Ring TX B1 data structure

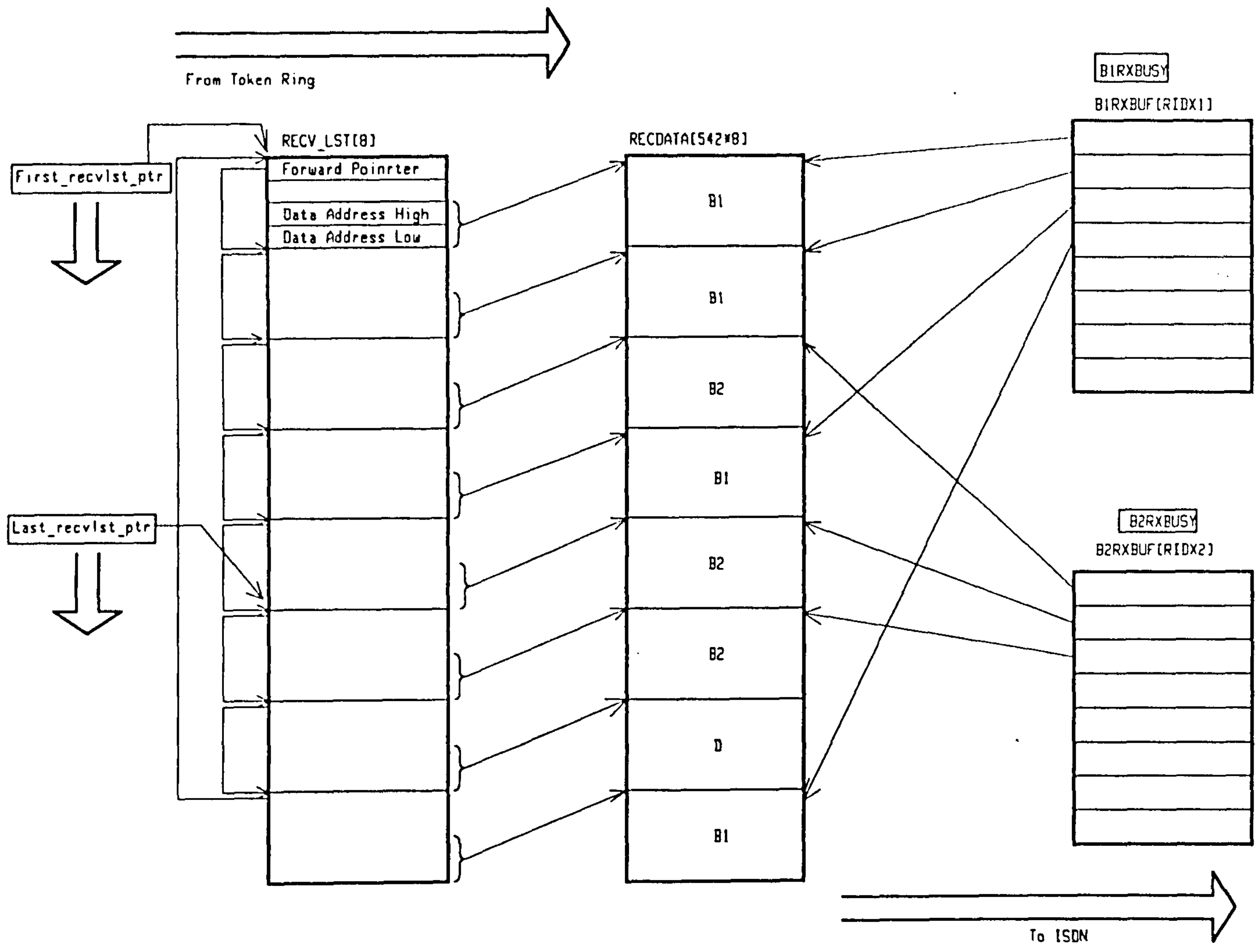

Figure 4.14: Token Ring RX data structure

Table 4.3: SNIC parallel port

	SNIC Parallel Port Name	Description
ASYNC	MCR	Master Control Register
	STCR	ST-BUS Control Register
	HDLCR1	HDLC Control Register 1
	HDLCR2	HDLC Control Register 2
	HDLCSR	HDLC Status Register
	HDLCIMR	HDLC Interrupt Mask Register
	HDLCISR	HDLC Interrupt Status Register
	HDLCRX	HDLC Tx FIFO (D Channel)
	HDLCTX	HDLC Rx FIFO (D Channel)
	HDLCARR1	HDLC Address Byte #1 Register
	HDLCARR2	HDLC Address Byte #2 Register
SYNC	NTCCR	NT Mode C-channel Register
	NTCDR	NT Mode DSTi C-channel Register
	NTSR	NT Mode C-channel Status Register
	TECCR	TE Mode C-channel Register
	TEMDR	TE Mode DSTi C-channel Register
	TEMSR	TE Mode C-channel Status Register
	TXB1CH	S-Bus Tx B1-channel
	RXB1CH	S-Bus Rx B1-channel
	TXB2CH	S-Bus Tx B2-channel
	RXB2CH	S-Bus Rx B2-channel

can be accessed through the microprocessor port. Asynchronous registers operate independently from the S or ST-BUS timing and can be accessed at any time. Synchronous registers, on the other hand, are single buffered registers and require synchronous access. All interactions with the synchronous registers must be performed while the New Data Available signal is low, that is when interrupt 2 is requested with the IRQ2 signal is high. The SNIC parallel port names corresponding to the SNIC internal registers and their descriptions are shown in Table 4.3.

4.4.5 Files

In this subsection, the details of each file is described.

4.4.5.1 Main.c This file controls the program flow of the *Gateway Driver.* This file also includes some routines which is needed to initialize the interrupt controller (8259) and SNIC (MT8930).

4.4.5.2 Init.c This file is same as the init.c of the *File Transfer Program.*

4.4.5.3 Open.c This file is same as the open.c of the *File Transfer Program.*

4.4.5.4 Beg.c This routine initializes the token ring tx B1 data structures, token ring tx B2 data structures, token ring tx D data structures, and token ring rx data structures which are described in Subsection 4.4.3.

4.4.5.5 Recv.c This file includes recvring () and freerecvlst () routines. Recvring () routine waits until the data is received from the Token Ring Adapter and decides which types of data they are. If the data is B channel data, this routine initializes B1RXBUF buffer, increases B1RXBUSY flag, and frees the receive list. If the data is D channel control data, this routine calls the txisdnd () routine and frees the receive list. Freerecvlst () routine frees one more receive list in order to receive more data from the Token Ring Adapter. The flow chart of the recvring () routine is in Appendix A.

4.4.5.6 Tx.c This file includes the txb1 () and txb2 () routines. Txb1 () routine transmits the data which is stored in B1DATA buffer to the token ring and decreases the B1TXBUSY flag by one. Txb2 () routine does the same job for the B2 data.

4.4.5.7 Isdnd.c This file includes the txisdnd (), dchsend (), xdchsend (), rxisdnd (), and txtoring () routines. Txisdnd () routine reads the control data from the token ring and transmits them to the ISDN D channel. Dchsend () routine sends the frame to the ISDN D channel by using the routing table of the transmit gateway. Xdchsend () routine does the same job as dchsend () for the receive gateway. Rxisdnd () routine reads the control data from the ISDN D channel and transmits them to the Token Ring Adapter. Txtoring () routine transmits control data to the Token Ring Adapter.

4.4.5.8 Tt.asm This file includes mmiow () and mmior () routines which have the same function as those in *File Transfer Program.*

4.4.5.9 Intsvc.asm The most important part of this file is the int2svc () routine which is the interrupt 2 service routine. This file also includes some routines which are used to setup the interrupt vector table.

Int2svc () routine is executed in every 125 usec when the interrupt 2 signal is asserted at the SNIC. The main function of this routine is to read the B channel data from the SNIC B channel receive port (RXB1CH,RXB2CH) and accumulate them into the B1DATA (B2DATA) buffer. At the same time, it reads data from the RECDATA buffer and sends them to the SNIC B channel transmit port (TXB1CH,TXB2CH).

The most important thing considered when this routine was designed was to reduce the execution time of this routine. This routine should be executed in every

125 usec. That means the execution time of this routine should be shorter than 125 usec in order not to miss any interrupt request. Actually the shorter execution time guarantees the better performance of this gateway, because the shorter interrupt service routine can give more time for the other parts of the program. Therefore, this routine is written in 8088 assembly language and the data structure is designed in order to reduce the manipulation time. The actual execution time of this interrupt service routine is reduced to 85 usec in normal operation.

5 GATEWAY PERFORMANCE

5.1 Overview

This chapter describes the performance of the *Token Ring to ISDN Gateway.* The performance of the gateway was evaluated by finding how quickly the files can be transmitted (Transfer Delay) and how many bits are lost during file transmission (*Missing Data Rate*). The Transfer Delay is the total time needed to finish the transmitting one file from the File Transfer TX Station to the File Transfer RX Station. The *Missing Data Rate* is the number of bytes which are missed during the transmission divided by the number of bytes of the original file.

5.2 Transfer Delay Measurement

Two types of Transfer Delay were measured. One is the *File Transfer Delay* and the other is the *File Data Transmitting Delay.* The *File Transfer Delay* is defined as the time difference between the time when the File Transfer TX Station begins to set up the call connection by sending Call Request and the time when that station receives Clear Confirmation. The main () routine of the *File Transfer Program* gets the time by calling the MS-DOS function (INT 21H). The *File Transfer Delay* is measured for different size of files and the result is shown in Table 5.1 and Figure 5.1.

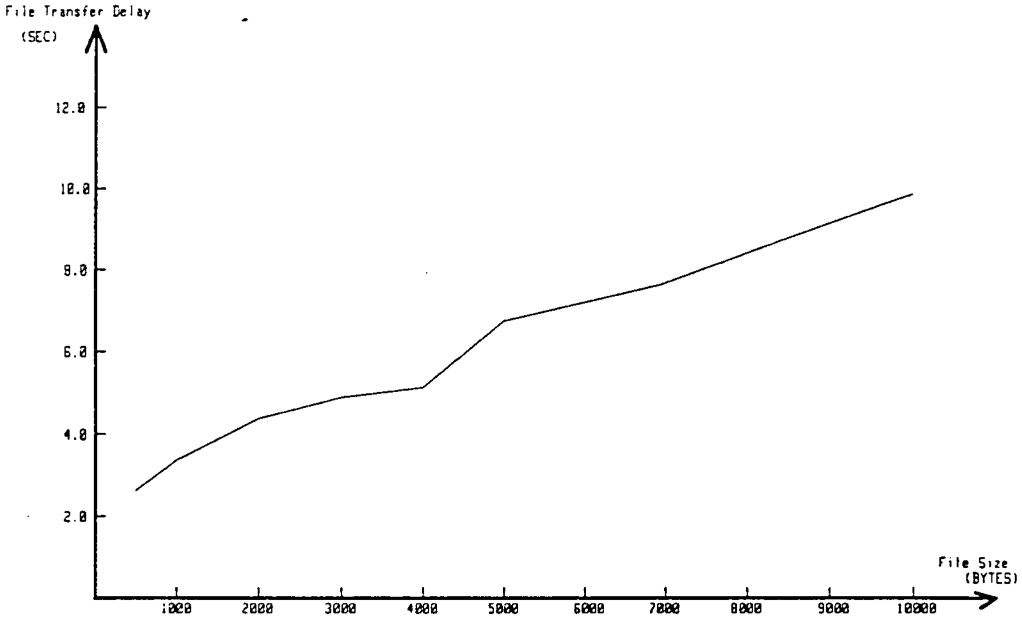

Figure 5.1: *File Transfer Delay*

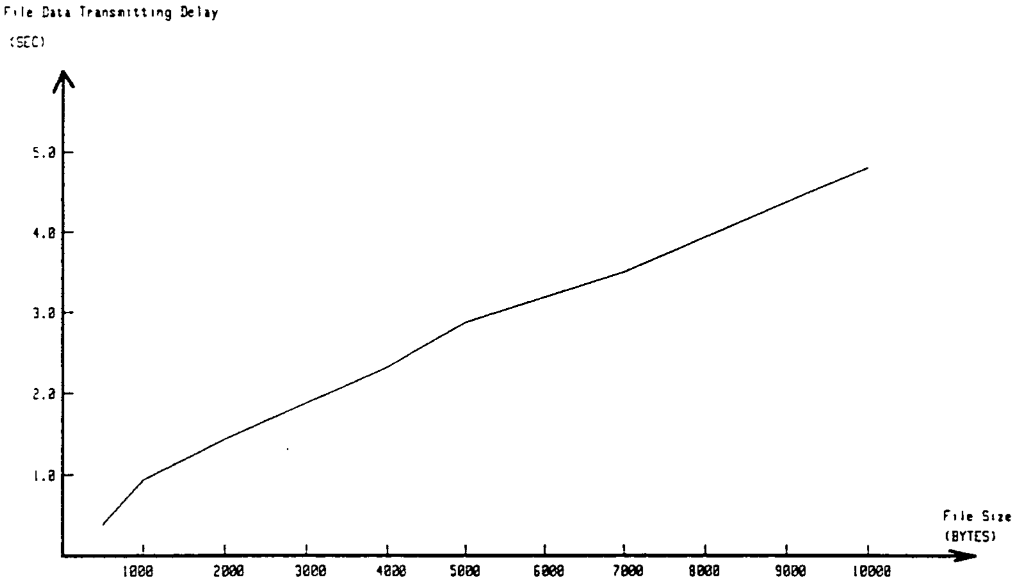

Figure 5.2: *File Data Transmitting Delay*

The *File Data Transmitting Delay* is defined as the time needed to transmit the file data only. That is the *File Transfer Delay* subtracted by the call setup time. The result is shown in Table 5.1 and Figure 5.2.

5.3 *Missing Data Rate* Measurement

The *Missing Data Rate* is measured by comparing the size of the original file at the File Transfer TX Station with the size of the received file at the File Transfer RX Station. The *Missing Data Rate* can be calculated by the following equation.

$$Missing\,Data\,Rate = \frac{S_{TX} - S_{RX}}{S_{TX}} \times 100(\%)$$

Figure 5.3: *Missing Data Rate*

where,

S_{TX} = size of the original file

S_{RX} = size of the received file

Many measurement trial shows that the file data is only missed and not changed. The result is shown in Table 5.1 and Figure 5.3.

5.4 Performance Analysis

The performance measurement was conducted with one File Transfer TX Station, one File Transfer RX Station, and one *Token Ring to ISDN Gateway*. Therefore, only one B channel and D channel are used for this file transfer performance measurement

Table 5.1: Gateway performance measurement result

File Size	File Transfer Delay	File Data Transmitting Delay	Missing Data Rate (MDR)
(bytes)	(sec)	(sec)	(%)
500	2.64	0.39	3.00
1000	3.46	0.94	4.50
2000	4.40	1.38	4.30
3000	4.77	1.87	4.00
4000	5.16	2.31	3.87
5000	6.81	2.91	2.04
7000	7.58	3.57	2.00
10000	9.83	4.78	1.91

experiment.

The result of the performance measurement shows that the *Token Ring to ISDN Gateway* can not transfer the file without any missing data. In this experiment, the *Missing Data Rate* ranges from 0.32% to 4.50% depending on the file size as shown in Table 5.1. The missing data are due to the slow speed of the gateway. The clock speed of the *Token Ring to ISDN Gateway* (IBMPC/XT) is 8 MHz and it takes 85 usec to execute one interrupt 2 service routine in normal operation. Therefore, in normal operation the gateway works well and no data is missed. But the gateway should reinitialize the data structures everytime after it receives or transmits one data frame (512 bytes of data). This reinitialization procedure takes more than 125 usec and the gateway can not service the interrupt 2 during this period and some data are missed. The experiment results show that the gateway performs better for the larger files.

The measurement of the Transfer Delay shows that the longer file takes more time to be delivered as we can expect. Comparing the *File Transfer Delay* with the

File Data Transmitting Delay, the experiment shows that the significant time is spent for the call setup.

6 CONCLUSIONS

The *Token Ring to ISDN Gateway* was designed and implemented. This gateway connects the token ring network to the ISDN environment.

In Chapter 3, the hardware design of the *Token Ring to ISDN Gateway* was discussed. For the implementation of the gateway, the S interface configuration method using Basic Channel Service (2B+D) was selected. The hardware of the gateway consists of a TI's TMS380 Token Ring Adapter, the *Gateway Controller* (IBM PC/XT), and the *ISDN S Interface Circuit.* In Chapter 4, the *Interconnection Protocol* design was discussed. This protocol covers the OSI Network layer and the part of the Data Link layer (LLC layer) and is implemented by the *File Transfer Program* and the *Gateway Driver.* This protocol was designed based on the CCITT X.25 Virtual Call protocol and works between same token rings which are connected through the ISDN. In Chapter 5, the performance measurement shows that the *Token Ring to ISDN Gateway* works with certain Transfer Delay and *Missing Data Rate.* The measurable *Missing Data Rate* is due to the slow speed of the gateway.

We will suggest some future works to improve the performance of the *Token Ring to ISDN Gateway.* First, the faster hardware should be used as a gateway. The IBM PC/AT may be used instead of the IBM PC/XT. The hardware with the second processor which does a dedicated communication job for the ISDN side interface may be used as a gateway. Second, the more improved *Interconnection Protocol* should be developed. This protocol can cover up to OSI layer 7 and be used between any kinds of LANs through ISDN. Finally, the *Interconnection Protocol* can be modeled and evaluated. The performance of the *Interconnection Protocol* as well as the performance of the end-to-end protocols can be evaluated by simulation. And the LAN to ISDN Gateway with the improved hardware and protocol can be implemented and the total performance with the LAN is to be evaluated by monitoring.

7 ACKNOWLEDGEMENTS

I wish to express my sincere thanks to my major professor, Dr. Douglas W. Jacobson for his guidance and encouragement during this research.

Thanks are also due to my graduate committee, Dr. Arthur V. Pohm and Dr. Way Kuo for taking time to help me with my research.

Special thanks are extended to Electrical and Computer Engineering Department for providing a good environment to do this research and financial support.

I thank my wife, Youngsun, for her selfless sacrifice, encouragement, and love, and my son, Jiwon for his smile.

Finally but most importantly, I would like to thank our parents for their continuous support and encouragement during the preparation of this thesis.

8 BIBLIOGRAPHY

[1] Prycker, De Martin. "LANs in an ISDN: consistency with the OSI Reference Model." *Computer Communications*, vol. 8, no. 2, (April 1985), 74-78.

[2] *TMS 380 Token Ring PC Adapter Kit Guide.* Houston, Texas:Texas Instruments, 1987.

[3] Farmer, W.D., and Newhall, E.E. "An Experimental Distributed Switching System to Handle Bursty Computer Traffic." *Proceedings, ACM Symposium on Problems in the Optimization of Data Communications*, 1969.

[4] Andrews, D. W., and Schultz, G.D. "A Token-Ring Architecture for Local Area Networks: An Update." *Proceedings, COMPCON Fall 82*, IEEE, 1982.

[5] Markov,J.D., and Strole, N.C. " Token-Ring Local Area Networks: A Perspective." *Proceedings,COMPCON Fall 82*, IEEE, 1982.

[6] Stallings,W. *Data and Computer Communications.* New York: Macmillan Publishing Company, 1985.

[7] Jacobson,D. W. *Local Area Network Protocols for Integrated Voice, Data, and Video Services.* Ames, Iowa: Iowa State University, 1987.

[8] Stallings, William. *Tutorial: Integrated Services Digital Networks (ISDN).* Washington, D.C.: IEEE Computer Society Press, 1985.

[9] *The ISDN Express Card: Designer's Manual(issue 2).* San Diego, CA: Mitel Corporation, 1988.

[10] *TMS380 Adapter Chipset User's Guide.* Houston, Texas:Texas Instruments, 1986.

[11] *JDR PR-1 and PR-2 Users Manual.* 110 Knowles Drive, Los Gatos, CA: JDR Microdevices, 1986.

[12] *Microelectronics Data Book : Don't Design without it (issue 5).* San Diego, CA: Mitel Corporation, 1988.

[13] Bocker, Peter. *ISDN: The Integrated Services Digital Networks: Concepts, Methods, Systems.* Berlin Heidelberg, Germany: Springer-Verlag, 1988.

[14] Deasington,R.J. *X.25 Explained: Protocols for packet switching networks, second edition.* Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, England: Ellis Horwood Limited, 1988.

[15] Currie, W. Scott. *LANs Explained: A Guide to Local Area Networks.* Market Cross House, Cooper Street,Chichester, West Sussex, PO19 1EB, England:Ellis Horwood Limited, 1988.

[16] Ronayne,John. *The Integrated Services Digital Network: from concept to application.* 128 Long Acre, London, WC2E 9AN, England:Pitman Publishing, 1987.