

Washington University in St. Louis  
**Washington University Open Scholarship**

---

All Computer Science and Engineering Research

Computer Science and Engineering

---

Report Number: WUCS-91-13

1991-10-01

## Persistent Connections in High Speed Internets

Authors: James M. Anderson, Gurudatta M. Parulkar, and Zubin Dittia

We have proposed a very high speed internet (VHSI) abstraction for the next generation of internetwork. An important component of the VHSI is a novel Multipoint Congram-Oriented High Performance Internet Protocol (MCHIP). The congram is the basic service primitive of MCHIP which incorporates strengths of both connection and datagram approaches. MCHIP supports two types of congrams: the user congram (UCon) and the persistent internet congram (PICon). A UCon is similar to a soft connection and can be used to provide strict performance guarantees to applications that need it. PICon are long lived congrams between MCHIP entities, and their purpose is to provide a shared communication channel for a number of applications. This paper reviews the MCHIP protocol and then focuses on PICon management.

Follow this and additional works at: [http://openscholarship.wustl.edu/cse\\_research](http://openscholarship.wustl.edu/cse_research)

---

### Recommended Citation

Anderson, James M.; Parulkar, Gurudatta M.; and Dittia, Zubin, "Persistent Connections in High Speed Internets" Report Number: WUCS-91-13 (1991). *All Computer Science and Engineering Research*.  
[http://openscholarship.wustl.edu/cse\\_research/632](http://openscholarship.wustl.edu/cse_research/632)

**Persistent Connections in High Speed Internets**

**James M. Anderson, Gurudatta M. Parulkar  
and Zubin Dittia**

**WUCS-91-13**

**October 1991**

**Department of Computer Science  
Washington University  
Campus Box 1045  
One Brookings Drive  
Saint Louis, MO 63130-4899**

*This work is supported in part by the National Science Foundation, and an industrial consortium of Bellcore, BNR, DEC, Italtel SIT, NEC, NTT and SynOptics.*

**Copy to:**

Andreas Bovopoulos  
Neil Haller - Bellcore  
E.B. Juskevicius - BNR  
Deborah R. Stokes - BNR  
Alan Kirby - DEC  
Ken-ichi Yukimatsu - NTT  
Guru Parulkar  
Ron Schmidt - SynOptics  
Jonathan Turner  
Akira Arutaki - NEC America

**Abstract to:**

Milind Buddhikot  
Jerome R. Cox, Jr.  
Chuck Cranor  
Zubin Dittia  
Andy Fingerhut  
Larry Gong  
Rex Hill  
Nader Mirfakhraei  
Einir Valdimarsson  
Ellen Witte

## Persistent Connections in High Speed Internets\*

J.M. Anderson      Gurudatta M. Parulkar      Z. Dittia

### ABSTRACT

We have proposed a very high speed internet (VHSI) abstraction for the next generation of internet-work. An important component of the VHSI is a novel Multipoint Congram-Oriented High Performance Internet Protocol (MCHIP). The congram is the basic service primitive of MCHIP which incorporates strengths of both connection and datagram approaches. MCHIP supports two types of congrams: the user congram (UCon) and the persistent internet congram (PICon). A UCon is similar to a soft connection and can be used to provide strict performance guarantees to applications that need it. PICons are long lived congrams between MCHIP entities, and their purpose is to provide a shared communication channel for a number of applications. This paper reviews the MCHIP protocol and then focuses on PICon management.

To appear in the *Proceedings of IEEE Global Telecommunications Conference*, December 1991.

---

\*This work was supported in part by the National Science Foundation, and an industrial consortium of Bellcore, BNR, DEC, Italtel SIT, NEC, NTT, and SynOptics.



# Persistent Connections in High Speed Internets

J.M. Anderson, G.M. Parulkar, and Z. Dittia  
Department of Computer Science  
Washington University  
St. Louis, MO, 63031

## Abstract

*We have proposed a very high speed internet (VHSI) abstraction for the next generation of internetwork. An important component of the VHSI is a novel Multipoint Congram-Oriented High Performance Internet Protocol (MCHIP). The congram is the basic service primitive of MCHIP which incorporates strengths of both connection and datagram approaches. MCHIP supports two types of congrams: the user congram (UCon) and the persistent internet congram (PICon). A UCon is similar to a soft connection and can be used to provide strict performance guarantees to applications that need it. PICons are long lived congrams between MCHIP entities, and their purpose is to provide a shared communication channel for a number of applications. This paper reviews the MCHIP protocol and then focuses on PICon management.*

## 1 Introduction

In this paper, we examine persistent connections as a part of the very high speed internet (VHSI) abstraction that was proposed in Reference [1]. The goal of the VHSI abstraction is to provide variable grades of service with performance guarantees to a variety of applications on top of diverse networks. The target applications include voice and video distribution, multimedia conferencing, interactive distributed imaging and visualization, network file systems and multimedia mail. An important component of the VHSI abstraction is a novel Multipoint Congram-Oriented High Performance Internet Protocol (MCHIP). The congram is the basic service primitive of MCHIP which incorporates strengths of both connection and datagram approaches. MCHIP supports two types of congrams: the user congram (UCon) and the persistent internet congram (PICon). A UCon is similar to a soft connection and can be used to provide strict performance guarantees to applications. PICons are long

lived congrams between MCHIP entities, and their purpose is to provide a shared communication channel for a number of applications. PICons can also be used to carry data for UCons during their set up and reconfiguration. This paper provides motivation for PICons and describes how they are used and managed.

The paper is organized as follows. Section 2 presents an overview of the Multipoint Congram Oriented High Performance Internet Protocol (MCHIP). Section 3 concentrates on the need for PICons, and their place in the VHSI model. Section 4 focuses on the PICon control packet organization and presents examples of PICon setup and use. Finally, Section 5 identifies future research directions.

## 2 MCHIP

The internet level transport facility in the VHSI abstraction is the *Multi-Point Congram Oriented High Performance Internet Protocol* (MCHIP). The functionality of the VHSI abstraction is derived from the facilities provided by MCHIP, which in turn depends on the services provided by the congram router, the resource manager, and the various network access protocols. In subsequent paragraphs, we define the congram and show how MCHIP provides users with high performance internetworking over a range of diverse networks.

There is little doubt that the next generation internet protocol must provide high performance with high predictability. One issue that researchers still argue is that of connection versus connectionless service. We consider an application to be connection-oriented if it requires strict performance guarantees, a large fraction of link bandwidth, and can tolerate set up and termination delays. We consider an application to be datagram-oriented (or connectionless) if it does not require performance guarantees, can use resources on the basis of availability, and requires a relatively small fraction of link bandwidth. We believe that both

paradigms are useful for different applications, and we need to effectively support them both at the internet level.

We introduce a service abstraction called a *congram*, because it incorporates important aspects of connection-oriented and datagram services. A congram in our context is a plesio-reliable<sup>1</sup> service with no hop-to-hop flow and error control.

A congram only implies a predetermined path for packets with statistically bound resources. Also, appropriate low overhead mechanisms are provided to allow establishment and reconfiguration of the congram path.

MCHIP supports two types of congrams: the user congram (UCon) and the persistent internet congram (PICon)<sup>2</sup>. The UCon corresponds to a soft connection – it requires setup by the user (at some cost), and once the required data transfer is complete, it needs to be terminated. PICons are long lived congrams between MCHIP entities, and their purpose is to allow multiplexing of traffic from a number of users and applications when appropriate, and to carry data for UCons that are being set up or reconfigured. In this respect, PICons are like dynamic leased packet switched internet channels. It is important to note that UCons are created upon request by a user (and are thereby dynamic), while PICons are set up after negotiations between the users and the providers (and are relatively static). There are three possible ways for an application to send its data using congrams. These methods are described in detail in the subsequent paragraphs.

In the first method, an application which requires strict performance guarantees from the network establishes its own UCon, sends the data, and terminates the UCon. During the setup phase, a path is identified and statistical resource reservations, based on the application's resource requirements, are made along the path. During the data transfer, it is ensured that the application does not send traffic at a higher rate than it requested resources for. This helps make performance guarantees to the application. It is important to note that there is no complex hop-to-hop flow and error control during the data transfer phase, thus making the per packet processing relatively simple. Once data transfer has been completed, the path and its associated resources are released. Applications that can best be supported using this track are connection-oriented and typically demand high bandwidth, examples being video distribution, multimedia

conferencing, and interactive remote visualization.

In the second method, an application still establishes its own UCon, but uses PICons for sending its data while its UCon is being setup or reconfigured. While using PICons, the application is sharing a PICon with other applications and therefore cannot expect the same level of performance as it could with its own UCon. Clearly, applications should use this track only if they cannot tolerate setup and reconfiguration delays but can tolerate reduced performance resulting from sharing a PICon with other applications.

Finally, in the third method, an application may use PICons to send its data rather than setting up a UCon. Since a PICon provides a shared channel, the user cannot expect performance guarantees, however, the major advantage lies in the fact that the application does not need to wait for connection setup. Also, there is no requirement for termination of a connection. The third method provides datagram-like access to appropriate applications. Because datagram traffic is confined to PICons, it does not adversely affect the UCon traffic, nor does it make a compromise on the ability of the UCon to make performance guarantees to its application. Applications that can best be supported using this track are datagram oriented and typically demand a small bandwidth with little or no performance guarantees, examples being file transfer for small files, electronic mail and distributed processing.

The UCon and connection-oriented MCHIP communication are described in detail in [5]. In this paper, we shall be primarily concerned with the motivation and management of PICons.

### 3 MOTIVATION FOR PICONS

Connection oriented sources can best be supported using some form of soft connections (UCons in our case). However, there is no consensus on how to support both connection oriented sources and datagram oriented sources on a single communication substrate, thus requiring justification for our PICon approach [3]. We justify the need for PICons in two steps: first, we argue that for datagram oriented sources, having PICons is better than having UCons alone. Our second argument is based on the fact that PICons essentially help in creating a virtual datagram network, which is at least as good as the existing datagram networks.

<sup>1</sup> *plesio* comes from Greek, meaning *almost*.

<sup>2</sup> In previously published work, we referred to PICons as "perpetual internet congrams".

### 3.1 PIcon Justification

The following reasons justify supporting PIcons in addition to UCons:

- The cost and time required to establish a UCon (which depends on many factors such as internet loading and physical distance between hosts) is unacceptable to a number of datagram applications. PIcons provide the application with an alternative: instead of having to wait for a UCon to be established, the data may be forwarded to the destination immediately using the PIcon service.
- Given the fact that datagram applications have good multiplexing characteristics, using PIcons to multiplex their traffic leads to higher utilization of communication resources. In other words, having a separate UCon for each datagram applications would lead to poorer utilization of resources.
- The time that would be wasted while an application waits for a UCon to be set up can be utilized if the application initially uses a PIcon and then switches to the UCon once it has been established. Likewise, PIcons can act as backup (redundant) connections to provide some service even during connection reconfiguration resulting from network failures.

### 3.2 PIcons in a Virtual Datagram Network

Today's datagram networks are built using a set of gateways, leased circuit switched lines (channels), and private LANs. The backbone network itself is a virtual packet switched network on top of a circuit switched network. Circuit switched channels are generally part of a complex TDM hierarchy, and tend to be static, inflexible, and expensive. The packet switched network provider pays for the peak rate of the channel, irrespective of its usage. Reconfiguration of lines involves significant overheads. Further, since the line lessor provides a very basic service with statically allocated bandwidth, revenues are low.<sup>3</sup>

We now discuss how the PIcon approach overcomes or bypasses the problems discussed in the previous paragraph. In a congram oriented network, various private local area networks will be connected to gateways which are in turn connected to a backbone network that provides packet switched service

<sup>3</sup>This should not be taken to mean that leasing lines is inexpensive!

between the gateways. In order to be able to communicate between various hosts using datagrams, PIcons are established between the gateways. These PIcons are long lived, and are in general accessible to anyone wishing to use them.<sup>4</sup> For those applications demanding connection-oriented service with performance guarantees, UCons can be established on demand. For all other applications, PIcons can be used. This architecture results in a significant savings in bandwidth by ensuring that most of the bandwidth is allocated based on demand. Of course, it is possible that a significant portion of the bandwidth allocated for a PIcon is unused under low load conditions. However, this bandwidth will usually be a small percentage of the total, since a major portion of the available bandwidth will be utilized by UCons. Thus, a combination of UCons and PIcons would prove far more cost-effective than using leased lines strictly for datagram traffic. This scheme is also extremely lucrative from the point of view of the provider — packet switched services are considered value-added, and can thus lead to increased revenues.

## 4 PICON MANAGEMENT

PIcon management refers to specification of PIcon setup, usage, and termination. In this section, we detail the format of PIcon packet formats and present the steps required for PIcon setup, usage, and termination.

### 4.1 PIcon Data Packet Format

The layout of a PIcon packet is based on the layout of other MCHIP packets [2]. The format of a PIcon data packet is fixed, which minimizes the amount of PIcon packet processing that is required at each node. It is important that a PIcon packet contain enough information so that each node can ensure that the packet travels towards its destination. The PIcon data packet format is fully defined in Reference [3] and will not be described here due to space limitations.

### 4.2 PIcon Control Packet Format

In order for an application to make use of the services that PIcons provide, the MCHIP interface must provide certain facilities to applications. These facilities are accessed using the PIcon control messages.

<sup>4</sup>It is possible to have PIcons reserved explicitly for people who are willing to pay the extra cost.



The format of the PIcon control messages closely resembles the UCon control message format described in Reference [5]. These messages are described using the notational format:

[MESSAGE\_NAME] <FIELD\_1> <FIELD\_2> ...

where MESSAGE\_NAME is the name of the appropriate control message and the optional FIELD\_*n* describes a parameter associated with the particular message. We now consider the point-to-point PIcon specific control messages in detail. Note that the extension .PTP will be used to differentiate point-to-point messages from multipoint messages. Also, most of the messages will contain the fields PID, which is the PIcon ID, and SRC\_ADDR and DEST\_ADDR, which are the source and destination addresses respectively. In addition, some of the control messages will include an ATRB field, which contains information specific to the application's resource requirements. These attributes are fully defined in Reference [5]. The control messages are listed below with brief explanations:

[OPEN\_PICON.PTP]<DEST\_ADDR><ATRIB>

This control message originates as a request from an application to MCHIP requesting establishment of a PIcon. The message will be passed from the MCHIP layer on one host to the next hop's MCHIP layer as the request is evaluated.

[R.OPEN\_PICON.PTP]<PID>

This message is an acknowledgement sent by one MCHIP layer to another MCHIP layer in response to the message OPEN\_PICON.PTP, confirming the setup of the requested PIcon, and the assigned PIcon ID is returned as PID. When the originating host's MCHIP layer receives this control message, the response is passed to the requesting application.

[SEND\_PICON.PTP]<DEST\_ADDR><DATA>

This control message is sent by an application to the local MCHIP layer requesting the use of PIcons for sending the specified data. The destination address is provided as a parameter. The local MCHIP layer may send a resource request to a remote Resource Manager, if appropriate.

[RECV\_PICON.PTP]<SRC\_ADDR><DATA>

This control message is sent by MCHIP to an application to indicate the arrival of a PIcon data packet from the host with address SRC\_ADDR.

[CLOSE\_PICON.PTP]<PID>

This control message signals the termination of a PIcon. It is sent to the local MCHIP interface, along with the PIcon ID of the PIcon to be terminated. The local MCHIP layer then sends the control message to the next hop on the PIcon path in order to free all resources associated with the PIcon.

[R.CLOSE\_PICON.PTP]<PID>

This message is an acknowledgement sent by one MCHIP layer to another MCHIP layer in response to the message CLOSE\_PICON.PTP, confirming the termination of the requested PIcon, and the terminated PIcon ID is returned as PID. When the originating host's MCHIP layer receives this control message, the response is passed to the application requesting the termination.

### 4.3 PIcon Setup

In this subsection, we present how MCHIP is used to setup and terminate PIcons. Note that while the PIcon setup and termination is similar to that of a UCon, the basic difference appears in the form of the entity responsible for setting up the congram. In the case of a UCon, an application is responsible for setting up the connection, while a PIcon is setup by the PIcon Manager portion of the operating system on a host or gateway, normally after negotiations between the various parties involved. Other more subtle differences also exist and will be described below.

Figure 1 shows that several different modules together constitute what we refer to as the MCHIP PIcon service. These specialized modules work together to interface different types of sources to the underlying physical network. The modules and their associated functions are as follows:

**ARBITRATOR:** This module provides the interface between an applications's packet streams and the other modules that make up MCHIP. The Arbitrator examines each packet received from an application and determines if it is a control packet or a data packet. The packet is then passed on to the appropriate MCHIP module.

**CONGRAM SETUP MODULE:** Requests to setup a PIcon are managed by this module. Request information is exchanged between the Congram Setup Module, the Congram Router, the Resource Manager, the End-To-End PIcon Directory, and the PIcon ID Translation Table during PIcon setup.

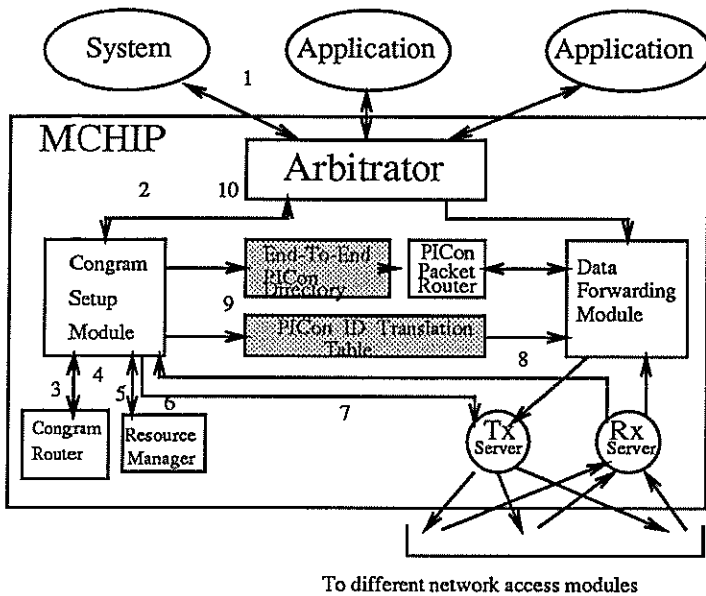


Figure 1: MCHIP Block Diagram of PICon Setup

**CONGRAM ROUTER:** Next hop addresses for PICon setup requests are generated by this module. The Congram Setup Module provides this module with PICon setup request information and a number of possible next hops are returned.

**RESOURCE MANAGER:** The list of possible next hop addresses are filtered by the Resource Manager. This filtering is based on the information contained in the resource manager's database about resource availability on different paths.

**END-TO-END PICON DIRECTORY:** This is a table which contains the PICon IDs and destination addresses of all PICons having the current host as one of its end points. The Congram Setup Module uses this module to record new PICons. The PICon Packet Router module uses this module to read the existing PICon data when calculating next-hop destinations.

**PICON ID TRANSLATION TABLE:** This table is used to map PICon IDs<sup>5</sup> to network interface identifiers or application port numbers. The Congram Setup Module provides updated data to this module and the Data Forwarding Module uses the data to process PICon packets.

**PICON PACKET ROUTER:** Contains a table of PICon IDs and a description of the path

<sup>5</sup>These PICon IDs need not have an end-point at the current host.

through the VHSI that each PICon constitutes. This module is used by the Data Forwarding Module to determine the PICon ID of a PICon along which to send a packet.

**DATA FORWARDING MODULE:** This module is used when an application requests the use of PICons. It uses the PICon Packet Router and PICon ID Translation Modules to process the PICon data packet's header information. Once a PICon has been selected for this packet, the packet is passed to the Transmit Server.

**TRANSMIT SERVER:** The Transmit Server Module forwards a PICon packet to the appropriate network access module in a lower layer.

**RECEIVE SERVER:** The Receive Server Module accepts a PICon packet from a lower layer and passes it to either the Congram Setup Module or the Data Forwarding Module.

In Figure 1 we show a block diagram which will help demonstrate the interactions between MCHIP and various applications during PICon setup. The numbers in Figure 1 correspond to steps in the following sequence. The description of PICon setup is divided into the actions at the host which originates the request and at an intermediate gateway.

#### Source PICon setup procedure

1. The application, in this case the PICon Manager portion of the host operating system, sends a request to the **arbitrator** asking for a PICon to be set up. This request contains information pertaining to the transmission requirements and access constraints of the desired PICon. The request contains the address of the remote host which serves as the destination for the desired PICon.
2. The **arbitrator** forwards this request to the **congram setup module**, which is responsible for setting up both UCons and PICons.
3. The **congram setup module** forwards the destination address to the **Congram router** (service agreement based router).
4. The **Congram router**, based on the destination address and the information in its routing tables (which include access constraints), selects a number of possible next hops and returns this information to the **congram setup module**.

5. The **congram setup module** forwards the list of next hops and PICon resource requirements to the resource manager.
6. The **resource manager**, based on its knowledge (from databases) of the bandwidth of various links to the next hop connected to the host directly, eliminates those next hops which cannot meet the resource requirements. From the remaining choices it selects one and makes a tentative reservation on the corresponding path.
7. The **congram setup module** assigns, to the PICon setup request, a PICon ID which is not already in use on the selected outgoing link to the next hop. It then sends the PICon setup request in a control packet to the next hop, using the **transmit server** for forwarding to the appropriate network access module. It is important to note that this packet is a control packet and as such does not use a PICon or UCon for its delivery.
8. An *ACK* (acknowledgment) or *NACK* (negative acknowledgment) is received by the **transmit server** which is then forwarded to the **congram setup module**.
9. If the response is an *ACK*, the **congram setup module** performs three actions. First, it creates an entry for the PICon in the **end-to-end PICon directory**, which is a table containing the PICon IDs and destinations addresses of all PICons having the current host as one of its end-points. Second, it adds an entry to the **PICon ID translation table** which maps PICon IDs to network interface identifiers or application port numbers. Thus, the table contains information that can be used to decide which network interface module or application a received packet is to be forwarded to, given its input link number and its PICon ID. Finally, the **congram setup module** acknowledges the setup of the desired PICon by sending an appropriate control message through the **arbitrator**.
10. If, on the other hand, the response to the control packet was a *NACK*, the **congram setup module** asks the **resource manager** to release the tentatively reserved resources on the failed path and attempts alternate paths suggested by the **Congram router**.

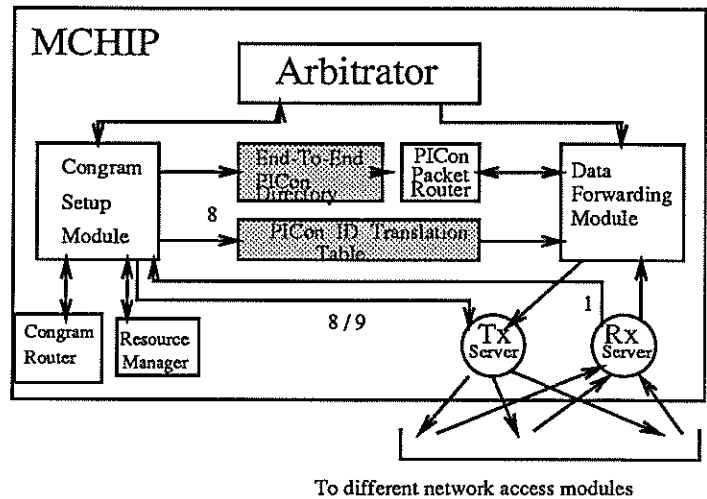


Figure 2: Intermediate Gateway PICon Setup

11. If the **congram setup module** finds that all paths have been tried without success, it sends a signal to the PICon Manager portion of the operating system through the **arbitrator** informing it that its request for a PICon setup has failed. The response to this signal is unspecified, but would typically involve another request for a PICon, but with lower bandwidth requirements.

#### Intermediate gateway PICon setup

Figure 2 is a block diagram which shows the interaction between MCHIP and various applications during PICon setup at an intermediate gateway. When an intermediate gateway receives a PICon setup request (a control packet), it behaves exactly as if the request was local, i.e., the procedure is identical to that described above with three exceptions:

- **Step 1** The request comes as a control packet rather than a message from the system, and so the packet has to be forwarded by the **receive server** to the **congram setup module**.
- **Step 8** When an *ACK* is received, no entry is made in the **end-to-end PICon directory**. Note however, that an entry is made in the **PICon ID translation table**.
- **Step 8/9** If an *ACK* is received, it is sent to the previous hop (gateway or end-point host) which sent the control packet containing the request. If no route can be found, a

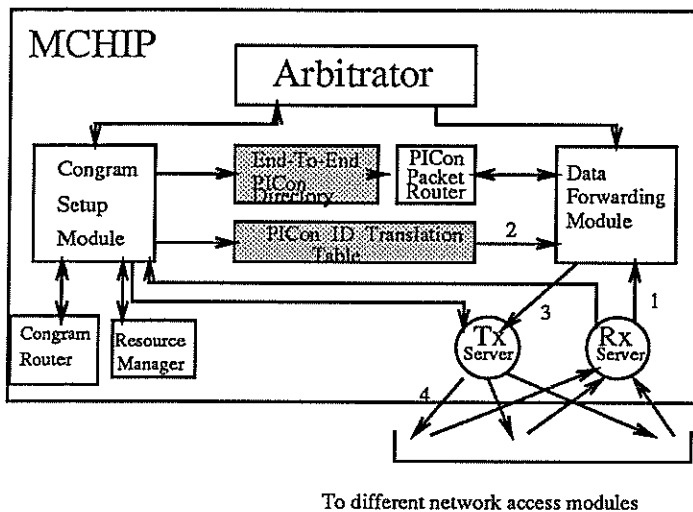


Figure 3: MCHIP Block Diagram Of PIcon Usage

NACK is sent instead.

#### 4.4 PIcon Usage

When an application requests the use of PIcons for data transfer, the arbitrator forwards the request to the data forwarding module. Routing of the data packet from the source to the destination can occur at two levels: routing within the PIcon network (handled by the PIcon packet router) and routing within a single PIcon (using the PIcon ID translation table).

Routing within a network of PIcons is much like routing in the current internet, except that PIcons replace actual links to the next hop. Thus, a PIcon network can be looked upon as a virtual network, and similar routing algorithms can be applied as would be applied to actual physical networks. These algorithms are implemented in MCHIP by the PIcon packet router, which refers to the end-to-end PIcon directory in order to find the PIcon ID of a PIcon along which to forward an inbound packet.

Figure 3 shows the steps associated with using an existing PIcon. The data forwarding module gets this information from the PIcon packet router and then refers to the PIcon ID translation table in order to determine the network interface identifier to which the data packet is to be forwarded, given the PIcon ID. It then encapsulates this port number, the PIcon ID, and the data packet into a single frame and forwards it to the transmit server. The transmit server strips the port number and forwards the frame to the appropriate network access module in a lower

layer. Thus, forwarding a packet on a PIcon is very simple as in the case of a Ucon or a connection.

## 5 Conclusion

At Washington University there is a great deal of work in progress on the next generation of internet-working. A UNIX kernel based implementation of a subset of MCHIP has been successfully achieved. Design and simulation of a two-port ATM-FDDI gateway using MCHIP as the internet protocol is nearing completion. In addition, multiplexing effectiveness of PIcons has been studied. This has led to a scheduling algorithm which can provide bounded delay and fair service to sources being multiplexed on the PIcon [4].

## References

- [1] G.M. Parulkar, "The Next Generation of Internet-working", *ACM SIGCOMM Computer Communications Review*, pp. 18-43, January 1990.
- [2] T.Y. Mazraani, *High Speed Internet Protocols and Resource Management in the Ethernet*, M.Sc. Thesis, Washington University, St. Louis, Missouri, August 1990.
- [3] J.M. Anderson, G.M. Parulkar, and Z. Dittia, *Persistent Connections in High Speed Internets*, Technical Report WUCS-91-13, Department of Computer Science, Washington University in St. Louis.
- [4] J.M. Anderson, *PIcon Management In High Speed Networks*, M.Sc. Thesis, Washington University, St. Louis, Missouri, August 1991.
- [5] T.Y. Mazraani and G.M. Parulkar, *Specification of a Multipoint Congram-Oriented High Performance Internet Protocol*, Proceedings of the IEEE INFO-COM'90