Report Number: WUCS-89-20

1989-09-01

Specification of a Multipoint Congram-Oriented High Performance Internet Protocol

Tony Y. Mazraani and Gurudatta M. Parulkar

We have proposed a very high speed internet (VHSI) abstraction that can provide a variable grade service with performance guarantees on top of diverse networks. An important component of the VHSI abstraction is a novel Multipoint Congram-oriented High Performance Internet Protocol (MCHIP). Features of this protocol include support for multipoint communication, the congram as the service primitive which incorporates strengths of both connection and datagram approaches, ability to provide variable grade of service with performance guarantees, and suitability for high speed implementation. This document introduces the VHSI abstraction and then focuses on the description of MCHIP. The protocol description... **Read complete abstract on page 2.**

Recommended Citation

Mazraani, Tony Y. and Parulkar, Gurudatta M., "Specification of a Multipoint Congram-Oriented High Performance Internet Protocol" Report Number: WUCS-89-20 (1989). *All Computer Science and Engineering Research.*
https://openscholarship.wustl.edu/cse_research/733

Specification of a Multipoint Congram-Oriented High Performance Internet Protocol

Tony Y. Mazraani and Gurudatta M. Parulkar

Complete Abstract:

We have proposed a very high speed internet (VHSI) abstraction that can provide a variable grade service with performance guarantees on top of diverse networks. An important component of the VHSI abstraction is a novel Multipoint Congram-oriented High Performance Internet Protocol (MCHIP). Features of this protocol include support for multipoint communication, the congram as the service primitive which incorporates strengths of both connection and datagram approaches, ability to provide variable grade of service with performance guarantees, and suitability for high speed implementation. This document introduces the VHSI abstraction and then focuses on the description of MCHIP. The protocol description includes packet types, sequence of packet exchange, and representative scenarios.

SPECIFICATION OF A MULTIPOINT CONGRAM-ORIENTED HIGH PERFORMANCE INTERNET PROTOCOL

Tony Y. Mazraani
Gurudatta M. Parulkar

WUCS-89-20

September 1989

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

ABSTRACT

We have proposed a very high speed internet (VHSI) abstraction that can provide a variable grade service with performance guarantees on top of diverse networks. An important component of the VHSI abstraction is a novel Multipoint Congram-oriented High Performance Internet Protocol (MCHIP). Features of this protocol include support for multipoint communication, the *congram* as the service primitive which incorporates strengths of both connection and datagram approaches, ability to provide variable grade of service with performance guarantees, and suitability for high speed implementation. This document introduces the VHSI abstraction and then focuses on the description of MCHIP. The protocol description includes packet types, sequence of packet exchange, and representative scenarios.

This work is supported by Bell Communications Research, Bell Northern Research, Italtel SIT, NEC, and National Science Foundation grant DCI-8600947.

To appear in the Proceedings of IEEE Infocom '90.

Contents

List of Figures

Specification of a Multipoint Congram-oriented High Performance Internet Protocol

Tony Y. Mazraani
tonym@flora.wustl.edu

Gurudatta M. Parulkar
guru@flora.wustl.edu

Computer and Communications Research Center
Department of Computer Science
Washington University
Saint Louis, Missouri 63130

1. INTRODUCTION

Ongoing research in computer communication and telecommunication suggests two emerging trends. First, in the next few years we will witness communications networks which can support increasingly high data rates [8,16,18]. For example, networks with data rates of a few hundred Mbps are being prototyped, and networks with data rates of a few Gbps are being planned. Second, researchers from all disciplines of science, engineering, and humanities plan to use the communication infrastructure to access widely distributed resources in order to solve bigger and more complex problems [12]. These trends pose a number of new challenges and opportunities to the researchers in the field of communications.

One such challenge is how to deal with the ever increasing diversity of underlying networks at high speed and at high performance levels, and how to support a wide variety of applications on one communication substrate. Note that different applications require considerably different quality of service in terms of bandwidth, end-to-end latency, errors, and packet loss. Moreover, the diversity of networks and applications will remain a fact of life for at least the foreseeable future. Thus, a framework which will allow diverse networks to work together and will allow diverse applications to work on top of interconnected networks is essential.

In the ARPA Internet and ISO models, the internet level is responsible for providing a homogeneous networking abstraction on top of diverse networks [3,13,15]. The success of the TCP/IP protocol suite and the ARPA Internet can be largely attributed to its internet abstraction which allows diverse networks to work together, allows a network to become part of the Internet without requiring any changes to its internal structure, and finally allows higher level protocols to behave as if they operate in a homogeneous network. However, the existing internet abstraction is based on the best effort datagram delivery which is becoming increasingly outdated for a number of reasons: it cannot work well with the connection-oriented high speed networks; it does not do any explicit

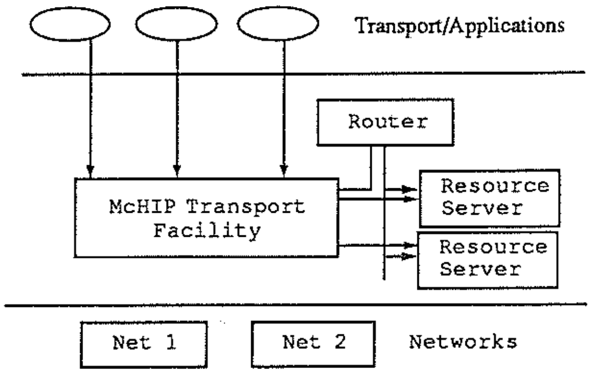

Figure 1: VHSI Abstraction

resource management, and thus cannot provide variable grade service with guarantees to different applications; and its gateway architectures are not designed to work at very high speeds.

We have proposed a very high speed internet (VHSI) abstraction that can meet these challenges [14]. An important component of the VHSI abstraction is a novel Multipoint Congram-oriented High Performance Internet Protocol (MCHIP). Features of this protocol include support for multipoint communication, the congram as the service primitive which incorporates strengths of both connection and datagram approaches, ability to provide variable grade of service with performance guarantees, and suitability for high speed implementation. This paper describes MCHIP, and is organized as follows. The next Section gives an overview of the VHSI abstraction. In section 2 we briefly present our arguments in favor of the congram approach. Section 3 describes the protocol using service primitives that the higher level protocols can request, descriptions of various packet types, and a couple of representative scenarios.

2. THE VHSI ABSTRACTION

In this section we present an overview of the very high speed internet (VHSI) abstraction. Reference [14] describes the VHSI abstraction, including its motivation, in considerable detail. The VHSI abstraction includes a number of significant improvements over the current internet abstraction and is shown in Figure 1. This includes a multipoint congram-oriented transport facility, resource management servers for diverse networks, an internet route server, and interface to various network access protocols.

Component Networks. The VHSI, like the existing Internet, will include a variety of local, regional, and national networks, and each class will consist of public networks, private corporate networks, and networks supported by the federal agencies. The network diversity includes speed, addressing, packet size and format, routing capabilities, access constraints, error control, resource allocation, and resource monitoring. We argue that the VHSI can successfully deal with the network diversity, only if it can impose certain requirements on the component networks and get them to cooperate at the internet level in a number of important ways. Two requirements that the VHSI abstraction imposes on each component network are that it provide its parametric description to the directly connected gateways, and either do internal resource management on a per *conversation* basis, or allow its directly connected gateways to do this.

Internet Resource Management. One of our major goals is to be able to make performance guarantees to applications across an internet of diverse networks. To achieve this goal, we have argued that it is necessary to monitor and control resource allocation and usage on a per congram/connection basis. However, most of the existing and emerging networks do not have this functionality. We argue that suitable mechanisms can be designed at the internet level to incorporate this functionality, with only minor modifications to the internal operation of the networks.

A simple but effective approach is to designate a gateway to serve as a resource manager or resource server — similar in spirit to a name or route server. A resource server is responsible on behalf of its network for keeping track of resource usage of active congrams and accepting new congrams only if there are resources to meet the performance needs of the congram. For details refer to [14].

Internet Routing. The important requirements for internet routing in the VHSI include efficient multicast and routing based on resource requirements, access constraints, and policy. Furthermore, there are standard requirements, such as stability, fast convergence, and load balancing, the routing protocols have to meet. Clearly, no existing internet routing protocol has all this functionality. However, there are a number of promising research efforts in progress which aim at developing routing protocols/models which would include some or all of this functionality [2,5,6,9,11].

MCHIP. There is little doubt that the next generation internet protocol must provide high performance with high predictability. Also, an integrated multipoint communication facility is important for a number of applications such as video distribution, multimedia conferencing, LAN interconnect, network management, and other distributed systems applications [4,17]. However, one issue that researchers still argue about is that of connection vs. connectionless service. In the following paragraphs we briefly present our arguments in favor of a service which aims at combining the strengths of both the connection and the datagram.

The issue of connection vs. connectionless service is at least as old as computer communications and has been a continual source of ideological debates. The reason for the persistence of this debate is that the semantics of a connection have been evolving with the rapid changes in network technology and with new applications. Initially, a connection meant a physical circuit, which is inflexible and inefficient for the bursty applications found in computer communications. Subsequently, a connection meant a virtual circuit (as in X.25 networks) on top of packet switching, providing additional flexibility and efficiency over the physical circuit. However, this connection implied a relatively static path for packet routing and reliable delivery of packets. Reliability, in turn, implied complex and slow mechanisms for hop-to-hop flow and error control. The need for exploring a connection-oriented architecture is recently expressed by the National Research Network Review Committee in its report "Toward a National Research Network" [12]. The IAB (Internet Activities Board) has also recognized the need for a connection-oriented service and has recently started a new working group called "ST and the Connection-oriented Internet Protocol" [7].

MCHIP and emerging high speed networks are taking the concept of a connection a few steps further, in order to make it more suitable for a wide variety of applications and networks. We introduce an abstraction called a *congram*, because it incorporates important aspects of connection-oriented and datagram services, and because it avoids any prejudice resulting from using old terms, such as connection or datagram. A congram in our context means a plesio-reliable service with no hop-to-hop flow and error control.[1] A congram only implies a predetermined path for packets and some resources statistically bound to the congram (application). Also, appropriate low overhead mechanisms are provided to allow establishment and reconfiguration of the congram path. Note that reconfigurability is important to ensure survivability in the event of network failures. Thus, the important point to note about the plesio-reliable congram abstraction is that the connection

[1] *Plesio* comes from the Greek word *plesios* which means close to or almost.

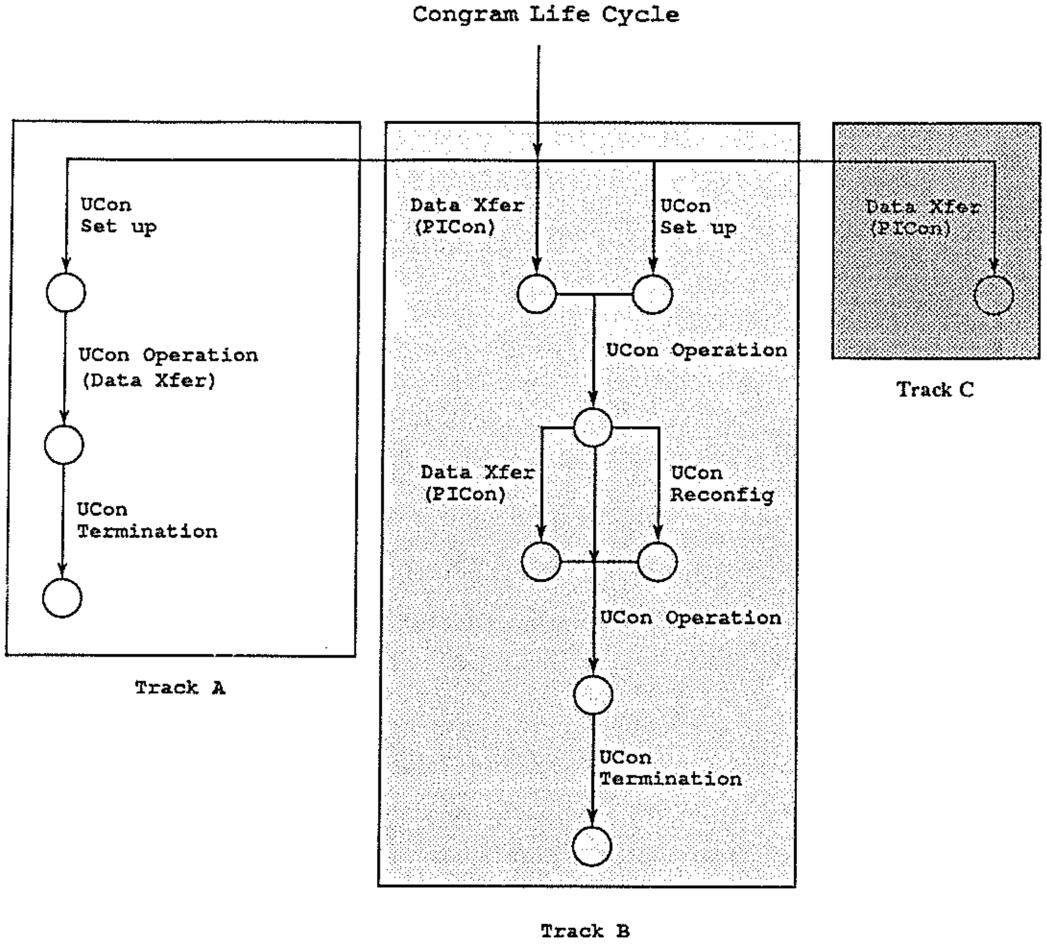

Figure 2: Congram Options

part of this abstraction provides efficient per packet processing and variable grade service with performance guarantees, and the plesio-reliability part provides survivability and flexibility as in a datagram model. In short, this abstraction has the potential to incorporate the valuable aspects of both connectionless and connection-oriented approaches. We argue that there is a need to explore the potential of such a congram-oriented service at the internet level.

MCHIP supports two types of congrams: perpetual internet congram (PICon) and user congram (UCon). PICons are long lived congrams between MCHIP entities, and their purpose is to carry data for UCons that are in the transient state. There are three possible ways for an application to send its data using congrams, as shown in Figure 2. First, an application establishes its own UCon, sends the data, and terminates the UCon (track A). Second, an application still establishes its own UCon, but uses PICons for sending its data while its UCon is being set up or reconfigured (track B). Finally, an application may not establish its own UCon but use PICons to send small amounts of data (track C). This idea of multiplexing data from user congrams in the transient state onto perpetual congrams has a lot of promise. With this functionality, we can have a congram abstraction which provides variable grade service and performance guarantees, which does not suffer from the typical connection setup delay, and which allows efficient reconfigurability. Thus, this kind of congram abstraction can have advantages of both congram-oriented and connectionless approaches. This paper presents the

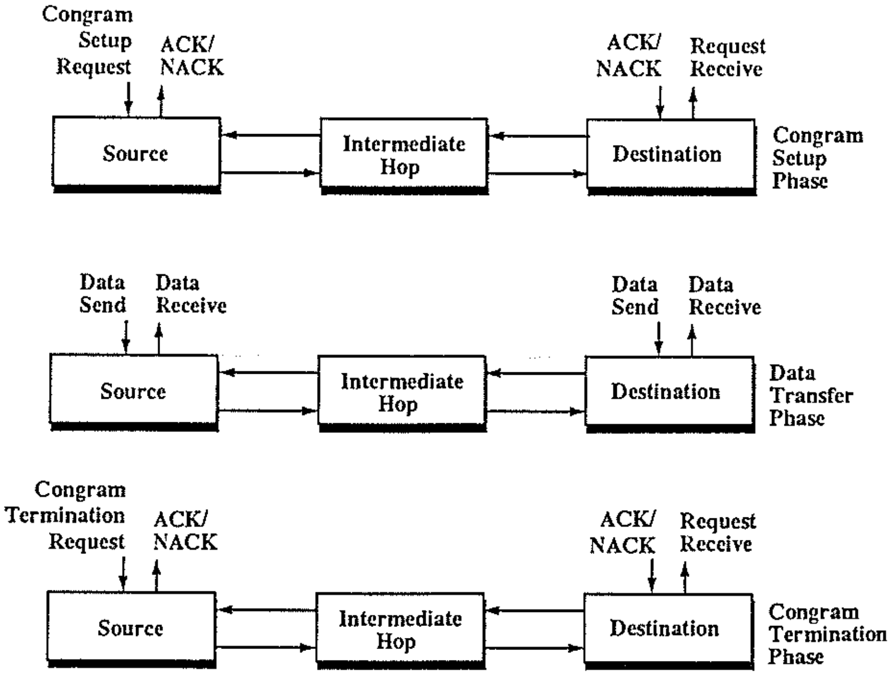

Figure 3: Message flow during the three phases of a congram

details of track A. The details of tracks B and C are left for other reports.

3. DESCRIPTION OF MCHIP ALONG TRACK A

As mentioned above, this section describes the MCHIP operation along the track A, which has three phases: setup, data transmission, and termination. Even though track A looks similar to a generic connection, the operation details are different as will be shown in this section. Figure 3 shows the message flow among source, intermediate, and destination MCHIP entities during three phases. Of course, there could be more than one intermediate MCHIP along the congram path.

The point-to-point congram setup requires a two-way handshake: request and ACK/NACK. The congram setup request is issued by a higher-level protocol (transport for example), and is propagated from source to destination via zero or more intermediate MCHIPs. All intermediate MCHIPs do routing and tentative resource allocation based on the congram's attributes specified in the request. The destination MCHIP forwards the request to the appropriate higher-level protocol, then propagates the ACK/NACK back to the source. Intermediate MCHIPs commit, or release, their tentatively-allocated resources during the propagation of the ACK/NACK. Assuming that the congram is set up, data can be exchanged between source and destination. Before transmitting a data message, MCHIP stamps the message with the appropriate congram identifier. Since the congram identifier changes from hop to hop, intermediate MCHIPs map the incoming identifier to an outgoing one during data transfer. The congram termination also follows a two-way handshake. The congram termination request is issued by a higher-level protocol, and is propagated to the destination MCHIP, which forwards it to the appropriate higher-level protocol. An ACK or a NACK is then propagated from the destination back to the source. Intermediate MCHIPs use the ACK/NACK response to decide whether or not to release the resources already allocated for the congram.

The multipoint congram phases are basically similar to the point-to-point case. However, an intermediate MCHIP may exchange messages with one upstream MCHIP and one or more downstream MCHIPs. Furthermore, the congram setup requires a three-way handshake: request, ACK/NACK, and commit/abort. The source may either send a commit or an abort message to all destination MCHIPs. The commit message instructs intermediate MCHIPs to commit to their tentatively-allocated resources. The abort message, which may also be issued by an intermediate MCHIP, instructs intermediate MCHIPs to release all tentatively-allocated resources and abort the congram setup. The reasons for aborting the congram setup will be covered in section 3.3. Now let us consider each phase of the congram in detail.

3.1. POINT-TO-POINT CONTROL MESSAGES

Point-to-point (PTP) control messages correspond to congrams involving only two endpoints for the duration of the congram. We deal with PTP and multipoint congram management separately in order to minimize the length of message headers, to simplify the per-packet processing, and to minimize the overhead of managing PTP congrams.

The format of both PTP and multipoint control messages consists of two parts: *message name* specified in [ ] and *message fields* specified in < >. Hence, a MCHIP generic control message is as follows:

[NAME] <field 1> <field 2> ...

Message names with *.PTP* extension correspond to point-to-point control messages. Multipoint control messages have names with *.MTP* extension. We now consider PTP control messages in detail.

[OPEN_CON.PTP] <ICN> <PROTOCOL> <SRC_ADDR> <DEST_ADDR> <ATTRIBUTES>

This message requests the establishment of a PTP congram between a given source and destination whose addresses are *SRC_ADDR* and *DEST_ADDR*, respectively. The congram is identified by a 2-octet hop-by-hop number referred to as Internet Channel Number (*ICN*). The selection of the *ICN* is still a matter for further study. During congram setup, the higher-level protocol type is specified in *PROTOCOL*, which is then used by MCHIP to demultiplex incoming data.

The congram attributes (*ATTRIBUTES*) allow an application to specify its transmission requirements and resource needs. An application may also specify its desired as well as its minimum-acceptable resource needs. The proposed congram attributes are as follows:

- *Bandwidth* requirements of a congram are specified by three measures: average bandwidth in *bits/sec*, peak bandwidth in *bits/sec*, and average burst length in *bits*.
- *End-to-end delay* specifies an upper bound in *milliseconds* on the average delay acceptable to the application.
- *Rate of packet loss* represents the maximum number of packets that the application can afford to lose for every 10^6 packets transmitted.

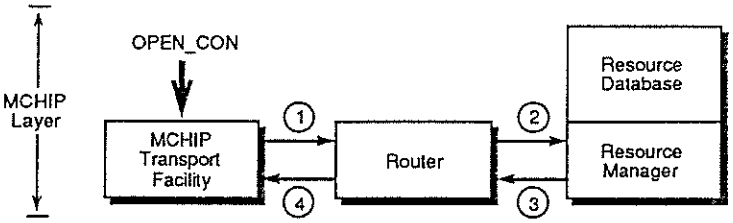

Figure 4: Routing and resource allocation during congram setup

- *Rate of out-of-sequence packets* represents the maximum number of out-of-sequence packets acceptable to the application for every 10^6 packets transmitted.
- *Access constraints* represent the routing constraints that the application may have on its packets traversing the internet. We expect the access constraints to be a set of predicates to be partly decided by the Policy Based Routing effort [2]. For example, it could very well be a *policy route.*
- *Congram setup time* is the maximum time in *milliseconds* that the application can afford to spend on setting up a particular congram.
- *Maximum Transfer Unit* (MTU) is a fixed upper bound (in *octets*) on the size of a packet belonging to a particular application. This parameter, when specified, means that the congram should traverse only those networks whose MTUs are greater than or equal to the specified MTU (in order to prevent packet fragmentation).

OPEN_CON.PTP requires four important steps that occur inside the MCHIP layer (see Figure 4). These steps are related to the routing of, and the resource allocation for, a PTP congram. For most cases the router is local, hence routing is done via a local procedure call to the router. Resource allocation, however, may be done either through a local procedure call (local resource manager), or through an explicit control message (remote resource manager).

This unique routing and resource management model helps MCHIP make performance guarantees, and offer variable grade service. The resource manager (RM) is particularly tailored for each network to efficiently use the network's resources. The RM computes the *effective bandwidth*[1] using the congram attributes, and then reserves the appropriate network resources.

When the MCHIP transport facility (MTF) receives an *OPEN_CON.PTP* message, it issues a procedure call to the router (step 1) for routing the PTP congram whose resource needs are specified in the *ATTRIBUTES* field. The congram attributes are used by the router in its negotiation with the resource manager (RM). If the underlying network supports resource allocation, the router simply decides the next hop and lets the network itself take care of the resource allocation. In the case of networks with no resource allocation capabilities, the router may provide the RM (step 2) with several possible routes to the next hop. In this case the RM chooses one of the routes depending upon the resource availability, and allocates appropriate resources along the chosen route. If the RM is on a remote machine, the router sends an explicit control message (*ALLOCATE.PTP*) to the remote RM requesting resource allocation.

[ALLOCATE.PTP] <ICN> <ATTRIBUTES> <PATH>

ALLOCATE.PTP includes the *ICN* of the PTP congram, the congram attributes specified in the *ATTRIBUTES* field, and the possible routes to the next hop specified in the *PATH* field.

Once the resource allocation is done, the RM indicates to the router (step 3) the type (desired or minimum-acceptable) of allocation and the chosen route. If the RM is on a remote machine, this information is sent in a *R_ALLOCATE.PTP* message.

[R_ALLOCATE.PTP] <ICN> <RESPONSE> <REASON/RALLOC> <EXPLANATION> <PATH>

If all the requirements of *ALLOCATE.PTP* are met, *RESPONSE* contains an ACK. Moreover, the *RALLOC* field is zero if the desired attributes are met, or 1 if only the minimum-acceptable attributes are met. The *PATH* field contains the necessary routing information. If *RESPONSE* is a NACK, the reason(s) for the failure are specified in the *REASON* field (see Table 1). The *EXPLANATION* field contains more information about the failure, if necessary.

Upon receiving a response from the RM, the router informs the MTF (step 4) of the selected route, as well as of the type of resource allocation along that route. At this point, MCHIP forwards the congram request (*OPEN_CON.PTP*) to the next hop. When the end MCHIP receives the congram request, it forwards the request to the appropriate higher-level protocol. It then sends a response message (*R_OPEN_CON.PTP*) back to the source MCHIP describing the status of congram request.

[R_OPEN_CON.PTP] <ICN> <RESPONSE> <REASON/RALLOC> <EXPLANATION>

This control message is issued by the destination MCHIP in response to an *OPEN_CON.PTP* message. The *ICN* identifies the original congram. If all the requirements of the *OPEN_CON.PTP* are met, the *RESPONSE* field contains an ACK and the *RALLOC* field is 0 or 1. In this case the *REASON* and *EXPLANATION* fields are omitted from the message. If the *OPEN_CON.PTP* message fails, the *RESPONSE* field contains a NACK. The cause of the congram's failure is indicated in the *REASON* field (see Table 1). The *EXPLANATION* field contains more information about the failure, if necessary.

R_OPEN_CON.PTP is used by intermediate hops to adjust the tentatively-allocated resources during the processing of *OPEN_CON.PTP*. During congram setup, resource allocation is done based on the desired congram attributes specified in *OPEN_CON.PTP*. However, if at some intermediate hop the desired resources are not available, then allocation is done instead on the basis of the minimum-acceptable congram attributes. If this is the case as indicated by *R_OPEN_CON.PTP*, the intermediate hops that reserved the desired level of resources change their reservation to the minimum-acceptable level. If the RM is local, Changes in resource reservation are done via a procedure call to the RM. If the RM is remote, however, the intermediate hop sends an explicit control message (*DEALLOCATE.PTP*) to the RM. The format of *DEALLOCATE.PTP* is as follows:

CODE	REASON
00 000000	No explanation
01 000001	Destination unreachable
01 000010	Destination host not responding
01 000100	Congram reset
01 001000	Congram broken by network fault
10 000000	MTU requested cannot be met
10 000001	Bandwidth requested not available
10 000010	Delay requested not available
10 000100	Packet loss rate requested not available
10 001000	Out-of-sequence packet rate requested not available
10 010000	Access constraints requested cannot be met
10 100000	Congram setup time cannot be met

Table 1: Possible values for the *REASON* field

[DEALLOCATE.PTP] <ICN> <RALLOC>

This control message instructs the RM to change the level of tentatively-allocated resources to the level specified in the *RALLOC* field.

[SEND.PTP] <ICN> <DATA>

During congram setup, gateways do various state table initializations to facilitate subsequent packet forwarding. The congram state table may include the address of the next hop (or next hops in the case of a multipoint congram), fragmentation/reassembly information, congram resource requirements, and resource usage/availability.

When the congram setup phase is over, data can be exchanged between the source and destination using *SEND.PTP*. At this point the congram state information is established. Hence, each data message need only be identified by the appropriate *ICN*. The data portion of the message is carried in the *DATA* field. When an intermediate hop receives a *SEND.PTP* message, it maps the incoming *ICN* to an outgoing *ICN*, and then forwards the message to the appropriate next hop.

[CLOSE_CON.PTP] <ICN>

CLOSE_CON.PTP is issued by the source to take down an already established PTP congram whose internet channel number at the source site is *ICN*. *CLOSE_CON.PTP* expects a response.

[R_CLOSE_CON.PTP] <ICN> <RESPONSE>

R_CLOSE_CON.PTP is returned by the destination to indicate the status of the *CLOSE_CON.PTP* message. If all the requirements of *CLOSE_CON.PTP* are met, then the *RESPONSE* field contains

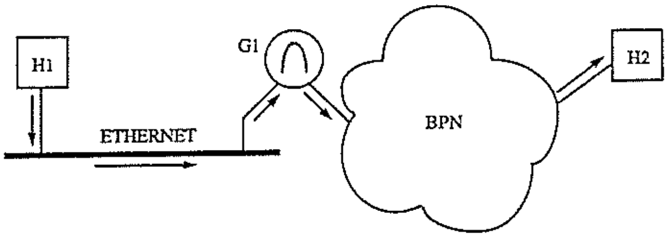

Figure 5: Internetwork configuration for an example PTP congram setup

an ACK instructing all intermediate hops to release the resources reserved for the congram. If the RM is part of the intermediate hop, resource deallocation is done via a local procedure call to the RM. However, in the case of remote RMs, the intermediate hop sends an explicit control message to the RM. The format of the control message is as follows:

[DEALLOCATE.PTP] <ICN>

Note that *RESPONSE* is always expected to be an ACK unless an intermediate hop or the destination goes down. Hence, a NACK is considered an error. This is due to the assumption that when the source issues a *CLOSE_CON.PTP*, higher-level protocols have already negotiated the termination of the congram.

3.2. AN EXAMPLE OF POINT-TO-POINT CONGRAM SETUP

This section describes in detail the setup, data transfer, and termination of an example PTP congram. Figure 5 shows the internetwork configuration used in this example. It consists of two networks: a typical broadcast LAN (Ethernet) and a high speed multipoint connection oriented network (BPN). (BPN[16] is a Broadcast Packet Network being designed and prototyped at Washington University. It is an ATM-like network that supports multipoint communication.) Resource management in the Ethernet is handled by an RM which is part of gateway G1. In BPN, however, resource management is handled by the network itself. For the sake of simplicity, it is assumed that the desired congram attributes can be met and that the destination accepts to join the congram.

In this example, host H1 wishes to establish a PTP congram to host H2. The setup starts with the MCHIP layer in H1 receiving an *OPEN_CON.PTP* request from its higher-level protocol (say transport) which is specified in the *PROTOCOL* field. This request specifies the source and destination as H1 and H2, respectively.

```
H1: [OPEN_CON.PTP] <ICN = 123> <PROTOCOL> <SRC_ADDR = H1> <DEST_ADDR = H2>
          <ATTRIBUTES>
```

Host H1 first consults its router to determine the next hop. The router then communicates the routing information to the RM for appropriate resource allocation based on the congram attributes. Since the router is part of H1, routing can be done via a procedure call from the MTF (MCHIP Transport Facility) to the router. The routing information returned specifies that the next hop is G1. Since the RM is part of G1, an explicit message should be sent from H1 to G1 for resource allocation.

```
H1->G1: [ALLOCATE.PTP] <ICN = 123> <ATTRIBUTES>
```

ALLOCATE.PTP is sent in an Ethernet packet. Note that the *PATH* field has been omitted because resource allocation on the Ethernet, which is a broadcast network, does not require any routing information.

When the RM in G1 receives the *ALLOCATE.PTP* message, it allocates the appropriate resources, updates its resource usage tables, and returns a response message to H1 indicating that all the requirements of *ALLOCATE.PTP* have been met. The *RALLOC* field is zero in this case, indicating that the desired resources have been allocated for the congram.

```
G1->H1: [R_ALLOCATE.PTP] <ICN = 123> <RESPONSE = ACK> <RALLOC = 0>
```

Realizing that the appropriate resources are available for the congram, H1 forwards the *OPEN_CON.PTP* message in an Ethernet packet to the next hop, G1.

```
H1->G1: [OPEN_CON.PTP] <ICN = 123> <PROTOCOL> <SRC_ADDR = H1> <DEST_ADDR = H2>
                       <ATTRIBUTES>
```

Upon receiving the *OPEN_CON.PTP* message, G1 selects a new *ICN* for the congram then contacts its router to find that the next hop is H2. Since resource management in BPN is handled by the network itself, G1 sends the *ALLOCATE.PTP* to the BPN interface.

```
G1->BPN: [ALLOCATE.PTP] <ICN = 456> <ATTRIBUTES> <PATH = H2>
```

Since BPN is a connection oriented network, the only routing information needed for resource allocation is the address of the next hop (H2). At this point, the BPN interface in G1 translates the *ALLOCATE.PTP* message to a series of BPN-specific messages. These messages result in establishing a BPN connection from G1 to H2, and in allocating the appropriate resources along the connection. Assuming that all the requirements of *ALLOCATE.PTP* are met, a response message is returned by BPN. The BPN interface in G1 translates the response to an *R_ALLOCATE.PTP*.

```
BPN->G1: [R_ALLOCATE.PTP] <ICN = 456> <RESPONSE = ACK> <RALLOC = 0>
```

Again, the *RALLOC* field is zero to indicate that the desired congram attributes have been met. At this point, G1 marks the congram request with the new *ICN*, and then forwards it to H2:

```
G1->H2: [OPEN_CON.PTP] <ICN = 456> <PROTOCOL> <SRC_ADDR = H1> <DEST_ADDR = H2>
                       <ATTRIBUTES>
```

Assuming H2 agrees to join the congram, it returns a response message that propagates back to the source via G1:

```
H2->G1: [R_OPEN_CON.PTP] <ICN = 456> <RESPONSE = ACK> <RALLOC = 0>

G1->H1: [R_OPEN_CON.PTP] <ICN = 123> <RESPONSE = ACK> <RALLOC = 0>
```

Note how the *ICN* field changes from hop to hop. Before propagating *R_OPEN_CON.PTP* toward the source, G1 examines the content of the *RALLOC* field. Since *RALLOC* is zero on both BPN and Ethernet, G1 realizes that the desired attributes have been met on both networks. Therefore, there is no need for any changes in resource allocation. Hence, G1 simply forwards the response to H1.

At this point, the PTP congram between H1 and H2 is established. All data packets exchanged between the source and destination are marked only with the appropriate *ICN*. In this example, H1 and H2 mark their data packets with *ICN*s 123 and 456, respectively. When G1 receives packets from H1, it maps the incoming *ICN* of 123 to an outgoing *ICN* of 456. The opposite *ICN* mapping (456 to 123) occurs when packets are received from H2. The transfer of one data packet from H1 to H2 is shown below.

```
H1->G1: [SEND.PTP] <ICN = 123> <DATA>

G1->H2: [SEND.PTP] <ICN = 456> <DATA>
```

At some later time H1 decides to terminate the congram. It sends a *CLOSE_CON.PTP* message to H2:

```
H1->G1: [CLOSE_CON.PTP] <ICN = 123>

G1->H2: [CLOSE_CON.PTP] <ICN = 456>
```

Based on the assumption that the higher-level protocols have already negotiated the termination of the congram, H2 returns an *R_CLOSE_CON.PTP* message in which *RESPONSE* is an ACK.

```
H2->G1: [R_CLOSE_CON.PTP] <ICN = 456> <RESPONSE = ACK>
```

This message causes the termination of the BPN connection between G1 and H2. In the BPN context, the connection termination means deallocation of resources also. On the Ethernet side, however, resource deallocation is done via a procedure call to the RM in G1. At this point, G1 forwards the *R_CLOSE_CON.PTP* to H1 then reclaims the *ICN* for future use.

```
G1->H1: [R_CLOSE_CON.PTP] <ICN = 123> <RESPONSE = ACK>
```

When MCHIP in H1 receives the the response message, it deletes the congram's state information, reclaims the *ICN* for future use, and indicates to the higher-level protocol that the congram has been terminated.

3.3. MULTIPOINT CONTROL MESSAGES

Multipoint (MTP) control messages correspond to congrams involving one source and multiple endpoints. An MTP congram is strictly a broadcast channel. That is, each packet sent by any endpoint of the congram is received by all other endpoints. During the congram setup, every MCHIP remembers its upstream and downstream neighbors. A data packet received from a neighbor is sent to all other neighbors. The number of endpoints that can be active simultaneously is left to the higher-level protocol. MCHIP always attempts to add all endpoints at the beginning of the congram. If any endpoint does not join the congram, the congram setup is either carried through or aborted, depending upon a congram attribute described later. Once the congram is set up, the current version of MCHIP does not support either the addition/deletion of endpoints or changes in the congram attributes. We now consider the multipoint control messages in detail.

[OPEN_CON.MTP] <CID> <ICN> <PROTOCOL> <NUM_ENDPTS> <ENDPT_LIST> <ATTRIBUTES>

OPEN_CON.MTP requests the setup of an MTP congram from one source to multiple endpoints. MTP congrams are identified by two parameters: *CID* (Congram Identifier) and *ICN* (Internet Channel Number). *CID* is a unique identifier of the congram. It comprises the address of the source concatenated with a 2-octet number selected to make the *CID* unique. *ICN* is a 2-octet hop-by-hop identifier which is selected during congram setup. It is then used during the data transfer phase as a short identifier for data packets. As in the PTP (point-to-point) case, *PROTOCOL* specifies the higher level protocol, and is used by MCHIP to demultiplex incoming data. *NUM_ENDPTS* specifies the number of endpoints; and *ENDPT_LIST* specifies the endpoint addresses. The *ATTRIBUTES* field specifies the transmission requirements and resource needs of the MTP congram. MTP congram attributes include the PTP congram attributes, in addition to one attribute which we refer to as congram *FLEXIBILITY*. An MTP congram is considered *flexible* (*FLEXIBILITY* = 1) if the application is willing to set up the congram to as many endpoints in *ENDPT_LIST* as possible. However, if the application requires that all endpoints be added, the congram is considered *inflexible* (*FLEXIBILITY* = 0). In this case, the congram setup is aborted if any endpoint does not join the congram.

Once the source sends an *OPEN_CON.MTP*, it waits for a response (*R_OPEN_CON.MTP*) from each endpoint specified in the congram request. If the congram is flexible, and assuming that at least one of the endpoints joins the congram, the source indicates to the higher-level protocol that the MTP congram is setup. Furthermore, it returns a list of endpoints, if any, that did not join the congram. However, if the congram is inflexible, it is considered setup only if all endpoints specified in the congram request join the congram successfully. Hence, if the source receives any NACK, it aborts the congram setup and sends an *ABORT.MTP* message which instructs all hops involved to delete the state information about, and release the resources reserved for, the MTP congram.

Routing and Resource Management. As in the PTP communication model, *OPEN_CON.MTP* requires four important steps involving routing and resource allocation (see Figure 4). Routing is usually done via a local procedure call to the router (step 1). The routing information returned by the router (step 4) in the MTP case is a list of addresses ordered as follows:

{next hop 1: endpoint x, endpoint y, endpoint z; next hop 2: endpoint u, endpoint v; ...}.

This list is interpreted by MCHIP as follows: to get to endpoints x, y, and z, forward packets to next hop 1; to get to endpoints u and v, forward packets to next hop 2; and so on.

The interaction between the router and the RM (resource manager) is basically similar to the PTP case. If the underlying network supports resource allocation, the router decides next hops and leaves the resource allocation task to the network itself. However, in the case of networks with no resource allocation capabilities, the router provides the RM (step 2) with the routing list. The list may contain multiple routes, if available. The RM in this case chooses the appropriate routes depending upon the resource availability, and allocates requested resources along the chosen routes. Once resource allocation is done, the RM informs the router (step 3) of the chosen routes as well as the type (desired or minimum-acceptable) of resource allocation.

For remote RMs, steps 2 and 3 are carried via two explicit control messages: *ALLOCATE.MTP* and *R_ALLOCATE.MTP* respectively. In this case *ALLOCATE.MTP* contains the *CID* of the congram, the routing list (*NEXTHOP_LIST*), and the congram attributes (*ATTRIBUTES*) as shown below.

[ALLOCATE.MTP] <CID> <ICN> <NEXTHOP_LIST> <ATTRIBUTES>

The exact amount of resources allocated along an MTP channel depends on the expected number of active sources which is usually determined by the higher-level protocol, and is beyond the scope of this report. The format of the response message is as follows:

[R_ALLOCATE.MTP] <CID> <ICN> <RESPONSE> <NEXTHOP_LIST>
<REASON/RALLOC> <EXPLANATION>

If all the requirements of *ALLOCATE.MTP* are met, *RESPONSE* contains an ACK, and *NEXTHOP_LIST* contains the routes chosen by the RM. Furthermore, *RALLOC* contains zero if the desired attributes are met or 1 if the minimum-acceptable attributes are met. In the case of a failure, the *RESPONSE* field contains a NACK. The reason(s) for the failure are specified in the *REASON* field (see Table 1), and the *EXPLANATION* field contains more information about the failure, if necessary.

Assuming that the four steps are completed successfully, MCHIP forwards the congram request to the next hops. Note that a PTP congram is a special case of an MTP congram in which the number of endpoints is one. However, applications are not advised to use the MTP facility to set up a PTP congram, because the overhead of setting up the congram in this case is relatively higher.

[R_OPEN_CON.MTP] <CID> <RESPONSE> <ENDPT> <REASON/RALLOC> <EXPLANATION>

R_OPEN_CON.MTP is returned by an endpoint or an intermediate hop to indicate the status of the congram setup. If *OPEN_CON.MTP* reaches an endpoint (*ENDPT*) which accepts to join the congram, the endpoint is expected to respond with an ACK. During the propagation of the response message, all intermediate hops check the *RALLOC* field to find out the type of resources allocated on incoming and outgoing links. If the resources allocated on both links are not the same, the intermediate hop is expected to release resources on the appropriate link so that only the minimum-acceptable attributes are guaranteed. Changes in the level of resource allocation is done via either a procedure call (local RM) or an explicit control message (remote RM). The format of the control message is as follows:

[DEALLOCATE.MTP] <CID> <ICN> <RALLOC>

If the requirements of *OPEN_CON.MTP* are not met at an intermediate hop or at an endpoint, a response message is returned in which the *RESPONSE* field contains a NACK. If the congram is flexible, an intermediate hop that receives a NACK is expected to release the resources tentatively allocated along this path. Then, it propagates the NACK toward the source. In this case the congram setup to other endpoints is not affected. If the congram is inflexible, however, an intermediate hop that receives the NACK aborts the congram setup to all its next hops, and then propagates the NACK toward the source. In this case the source aborts the congram setup to all endpoints from which it has not yet received a response.

[COMMIT.MTP] <CID> <RALLOC>

A *COMMIT.MTP* is issued by the source after all endpoints respond to the congram request. *COMMIT.MTP* instructs all intermediate hops to commit to the type of resources specified in the *RALLOC* field. That is, an intermediate hop that had tentatively reserved the desired resources is expected to release surplus resources if it receives a *COMMIT.MTP* in which *RALLOC* is 1. Therefore, after the *COMMIT.MTP* reaches all endpoints, either the desired or the minimum-acceptable attributes are guaranteed for the whole congram.

[SEND.MTP] <ICN> <DATA>

As mentioned above, various congram state tables at all hops are initialized during the congram setup phase. This facilitates the data transfer phase carried out by *SEND.MTP*. Data messages are identified by *ICN*, and carry data in the *DATA* field. An multipoint congram is a spanning tree connecting the endpoints. Each packet sent by any endpoint in the tree is broadcast to all other endpoints in the tree. Hence, when an intermediate hop receives a *SEND.MTP*, it uses its state table information to map the incoming *ICN* to an outgoing *ICN*, and to decide the output links. It then forwards the *SEND.MTP* message to all (downstream and upstream) its neighbors in the tree, except the neighbor from which the message was received.

[ABORT.MTP] <CID>

In the case of inflexible MTP congrams (i.e. *FLEXIBILITY* = 0), if the source receives a NACK from an endpoint, it sends an *ABORT.MTP* message to all endpoints that have not responded yet. An intermediate hop that receives an *ABORT.MTP* releases the resources tentatively allocated for the congram, and then forwards the message to all its downstream neighbors. If the received *ABORT.MTP* carries an unrecognizable *CID*, the message is discarded. While *ABORT.MTP* is being propagated to all endpoints, the source discards all response messages that are in transit.

[CLOSE_CON.MTP] <CID>

CLOSE_CON.MTP is sent from the source to the endpoints to terminate an already established congram. The identifier of the congram to be terminated is specified in the *CID* field. The intermediate hops propagate the *CLOSE_CON.MTP* message toward the endpoints. When an endpoint receives a *CLOSE_CON.MTP*, it is expected to return a response message to the source and disconnect itself from the congram.

[R_CLOSE_CON.MTP] <CID> <RESPONSE>

R_CLOSE_CON.MTP is returned by each endpoint of the MTP congram to indicate the status of *CLOSE_CON.MTP* at that endpoint. The source in this case keeps track of the number of responses received, to decide whether or not the congram is terminated. Upon receiving a *CLOSE_CON.MTP*, each endpoint is expected to terminate its participation in the congram. Hence, the *RESPONSE* field is expected to contain an ACK. This is due to the assumption that when the endpoint receives a *CLOSE_CON.MTP*, higher-level protocols have already negotiated the termination of the congram. During the propagation of the ACK, intermediate hops deallocate the resources reserved for the congram. Resource deallocation is done via either a procedure call (local RM) or an explicit control message (remote RM). The format of the control message is as follows:

[DEALLOCATE.MTP] <CID> <ICN>

Assuming all the requirements of *CLOSE_CON.MTP* are met, an ACK is propagated back to the source. In this case all intermediate hops delete their congram state information, release the resources already reserved for the congram, and reclaim the *CID* for future use.

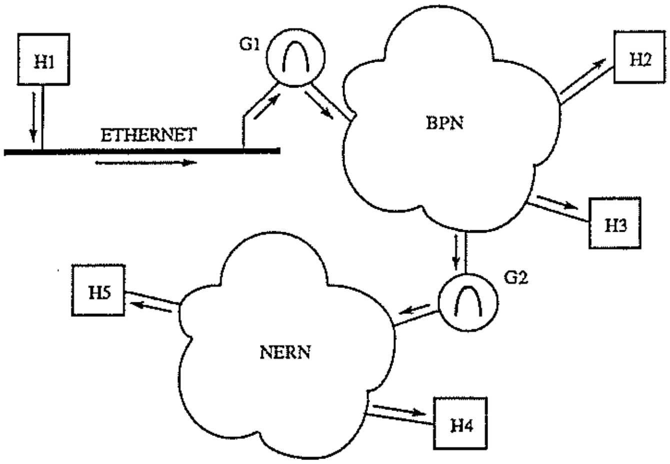

Figure 6: Internetwork configuration for an example MTP congram setup

3.4. AN EXAMPLE OF MULTIPOINT CONGRAM SETUP

This section describes in detail an example of MTP congram setup. The internetwork configuration used in this example is shown in Figure 6. It consists of an Ethernet, a high speed multipoint connection oriented network (BPN), and a point to point datagram network (NERN: National Education and Research Network). Resource management in Ethernet and NERN is handled by gateways G1 and G2, respectively. BPN, however, is capable of managing its own resources. The example assumes that the congram is *flexible* (i.e. *FLEXIBILITY* = 1), and that the desired congram attributes can be met.

Let H1 be the source and H2 through H5 be the other endpoints of the MTP congram. The setup starts with the MCHIP layer in H1 receiving an *OPEN_CON.MTP* from its higher-level protocol specified in the *PROTOCOL* field.

```
H1: [OPEN_CON.MTP] <CID = H1.24> <ICN = 123> <PROTOCOL> <NUM_ENDPTS = 4>
                   <ENDPT_LIST = {H2,H3,H4,H5}> <ATTRIBUTES>
```

This message is a request to MCHIP in H1 to setup an MTP congram to four endpoints whose addresses are H2, H3, H4, and H5. The resource requirements of the congram are specified in the *ATTRIBUTES* field. The congram is identified by a unique identifier, H1.24, and a hop-by-hop identifier, 123.

MCHIP in H1 first determines, using its router, that the next hop is G1 which is also the RM for the Ethernet. It thus sends an *ALLOCATE.MTP* message in an Ethernet packet to the RM in G1 for resource allocation.

```
H1->G1: [ALLOCATE.MTP] <CID = H1.24> <NEXTHOP_LIST = NULL> <ATTRIBUTES>
```

The next hop list is null in this case because Ethernet is a broadcast medium, hence resource allocation can be done irrespective of the number of endpoints.

The RM in G1 allocates appropriate resources for the congram, updates its resource usage tables, and returns a response message to the router in H1.

```
G1->H1: [R_ALLOCATE.MTP] <CID = H1.24> <RESPONSE = ACK> <NEXTHOP_LIST = NULL>
                         <RALLOC = 0>
```

The *RESPONSE* field contains an ACK to indicate that all the requirements of *ALLOCATE.MTP* have been met. The *RALLOC* field of zero indicates that the desired resources have been allocated for the congram.

At this time, MCHIP in H1 knows that G1 is the next hop, and that Ethernet has resources allocated for the congram. Next, it forwards the congram request to G1.

```
H1->G1: [OPEN_CON.MTP] <CID = H1.24> <ICN = 123> <PROTOCOL> <NUM_ENDPTS = 4>
                       <ENDPT_LIST = {H2,H3,H4,H5}> <ATTRIBUTES>
```

Upon receiving the *OPEN_CON.MTP* message, G1 selects a new *ICN* for the congram, determines using its router that H2 and H3 are directly connected to BPN, and that H4 and H5 can be reached via G2. G1 initiates a BPN multipoint connection among H2, H3, and G2 by sending an *ALLOCATE.MTP* message to the BPN interface.

```
G1->BPN: [ALLOCATE.MTP] <CID = H1.24> <NEXTHOP_LIST = {H2,H3,G2}> <ATTRIBUTES>
```

The interface translates *ALLOCATE.MTP* to a series of BPN-specific messages which result in establishing the BPN multipoint connection. A connection setup in BPN includes allocation of appropriate resources. Hence, a response message is returned by BPN which is translated by the BPN interface to an *R_ALLOCATE.MTP* message. The *RESPONSE* field contains an ACK, and *RALLOC* is zero indicating that the desired attributes have been met.

```
BPN->G1: [R_ALLOCATE.MTP] <CID = H1.24> <RESPONSE = ACK> <NEXTHOP_LIST = {H2,H3,G2}>
                          <RALLOC = 0>
```

At this point, MCHIP in G1 forwards the congram request to all three next hops. Note that G1 inserts the address of G2 in *ENDPT_LIST* which instructs G2 to set up the congram to only H4 and H5.

```
G1->H2,H3,G2: [OPEN_CON.MTP] <CID = H1.24> <ICN = 456> <PROTOCOL> <NUM_ENDPTS = 4>
                             <ENDPT_LIST = {H2,H3,G2:H4,H5}> <ATTRIBUTES>
```

Assuming that H2 and H3 accept to join the congram, each returns a response message which is propagated back to the source. The response message contains the address of the corresponding endpoint (*ENDPT*) as well as the type (desired in this case) of resources allocated for the congram.

```
H2->H1: [R_OPEN_CON.MTP] <CID = H1.24> <RESPONSE = ACK> <ENDPT = H2> <RALLOC = 0>

H3->H1: [R_OPEN_CON.MTP] <CID = H1.24> <RESPONSE = ACK> <ENDPT = H3> <RALLOC = 0>
```

G1 examines the *RALLOC* field and finds that the desired resources have been allocated on both BPN and Ethernet. Since no resource adjustment is needed, it simply forwards the response message to H1. For simplicity, we show the processing of *R_OPEN_CON.MTP* by the source after all the endpoints send their responses.

When the congram request is received by G2, it checks the *ENDPT_LIST* to find that it is expected to setup the congram to H4 and H5. It then determines, using its router, that H4 and H5 are directly connected to NERN. The router communicates the routing information to the RM (which is part of G2) for resource allocation. Note that the routing information may consist of multiple routes to both hosts. The RM chooses one route based on the resource availability, and allocates appropriate resources along the chosen route. MCHIP in G2 subsequently sends two separate congram requests to H4 and H5. Two separate requests are sent because NERN is a point to point network. Thus G2 maintains two PTP congrams which are remembered as part of the congram state information.

```
G2->H4: [OPEN_CON.MTP] <CID = H1.24> <ICN = 789> <PROTOCOL> <NUM_ENDPTS = 4>
                       <ENDPT_LIST = {H2,H3,H4,H5}> <ATTRIBUTES>

G2->H5: [OPEN_CON.MTP] <CID = H1.24> <ICN = 789> <PROTOCOL> <NUM_ENDPTS = 4>
                       <ENDPT_LIST = {H2,H3,H4,H5}> <ATTRIBUTES>
```

In this example we assume that H4 accepts and H5 does not accept to join the congram. Thus, H4 returns an ACK, and H5 returns a NACK. Both response messages are propagated back to the source via G1 and G2.

```
H4->H1: [R_OPEN_CON.MTP] <CID = H1.24> <RESPONSE = ACK> <ENDPT = H4> <RALLOC = 0>

H5->H1: [R_OPEN_CON.MTP] <CID = H1.24> <RESPONSE = NACK> <ENDPT = H5>
                         <REASON = NO EXPLANATION>
```

When the NACK from H5 is received at G2, MCHIP issues a procedure call to the RM for resource deallocation. As result, the RM in G2 releases all the resources tentatively allocated along the route to H5. It then forwards the NACK to the source via G1. The ACK from H4 is forwarded to G1 via G2. G2 does not take any action because there is no need to change the resources tentatively allocated along the route to H4.

During the congram setup, the source MCHIP listens to all responses from the endpoints. The responses in this example include all but one ACK. Since the congram is flexible, the source indicates to the higher-level protocol that the desired attributes have been met, and that all endpoints except H5 have joined the congram. H5's refusal to join the congram is indicated in the *ENDPT* field.

```
H1: [R_OPEN_CON.MTP] <CID = H1.24> <RESPONSE = ACK> <ENDPT = H5> <RALLOC = 0>
```

At this point, the source MCHIP determines, using the *RALLOC* field, that the desired congram attributes have been met at all hops. It thus sends a *COMMIT.MTP* to all hops instructing them to commit to their tentatively allocated resources.

```
H1->H2..H4: [COMMIT.MTP] <CID = H1.24> <RALLOC = 0>
```

During the data transfer phase, data is exchanged using *SEND.MTP*. Data messages are marked by the appropriate *ICN* (123, 456, and 789 in this example), and are broadcast to all enpoints of the congram. The congram termination phase is similar to the PTP case described in section 3.2. The source issues a *CLOSE_CON.MTP* in which the *CID* is H1.24. Intermediate hops release all resources allocated for the congram, delete the state information, and reclaim the *CID* and the *ICN* for future use. When the source MCHIP receives the ACK response to *CLOSE_CON.MTP*, it indicates to appropriate higher-level protocol that the congram has been terminated.

4. WORK IN PROGRESS

Considerable work is in progress at Washington University on the next generation of internetworking. The purpose of this paper is to give an overview of what we call the very high speed internet (VHSI) abstraction, and present the MCHIP protocol specification. As mentioned before, features of MCHIP include support for multipoint communication, the *congram* as the service primitive which incorporates strengths of both connection and datagram approaches, ability to provide variable grade of service with performance guarantees, and suitability for high speed implementation. We have used the packet format (see the Appendix) and packet exchange sequence that is familiar to, and recommended by, the IETF (Internet Engineering Task Force) working group on the connection-oriented internet protocol. This working group is evaluating MCHIP as the long term connection-oriented internet protocol.

Work in progress includes implementation of a prototype gateway interconnecting Ethernet, FDDI, and BPN networks with MCHIP as the internet protocol, evaluation of the proposed resource management strategy for connectionless networks, and a more systematic exploration of the necessity and advantages of PICons. It is important to note that the per-packet processing in the case of MCHIP is simple enough to be implemented in hardware and to support FDDI and BPN data rates. The congram management functions, such as congram setup, maintenance, reconfigurations, and termination are complex and require software processing at hosts and gateways. We are also working on the performance analysis of MCHIP's congram management functions.

— APPENDIX —

PACKET FORMATS

A. POINT-POINT CONTROL PACKETS

- OPEN_CON.PTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    OPCODE     |   PROTOCOL    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LENGTH             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-        SOURCE ADDRESS       -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-     DESTINATION ADDRESS     -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     ATTRIBUTES BIT MAP        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AVERAGE    |     PEAK      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| BURST FACTOR  |    AVERAGE    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      PEAK     | BURST FACTOR  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     DELAY     |     DELAY     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  PACKET LOSS  |  PACKET LOSS  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  OOS PACKETS  |  OOS PACKETS  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CON-TIME    |   CON-TIME    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      MTU      |      MTU      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
/            ACCESS             /
/          CONSTRAINTS          /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           CHECKSUM            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- OPCODE specifies the type of control packet.

- PROTOCOL specified the higher-level protocol that issues the request.
- LENGTH is the packet length in octets.
- CHECKSUM covers the whole packet.
- ICN is the internet channel number. It is an arbitrary number that identifies the congram to which the packet belongs.
- SOURCE ADDRESS is a 4-octet internet address of the source host.
- DESTINATION ADDRESS is a 4-octet internet address of the destination host.
- ATTRIBUTES BIT MAP indicates if a particular congram attribute is specified in the packet or not by setting the appropriate bit in the map high or low. The organization of this bit map is as follows:

```
15                             0
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

bit 0: desired bandwidth
bit 1: minimum acceptable bandwidth
bit 2: desired delay
bit 3: minimum acceptable delay
bit 4: desired packet loss rate
bit 5: minimum acceptable packet loss rate
bit 6: desired out-of-sequence (OOS) packet rate
bit 7: minimum acceptable OOS packet rate
bit 8: desired congram time
bit 9: minimum acceptable congram time
bit 10: desired MTU
bit 11: minimum acceptable MTU
bit 12: access constraints
bit 13: unused
bit 14: unused
bit 15: unused
```

Note that the congram attributes should be specified in the order shown in the bit map. For example, a packet in which only the desired bandwidth and MTU attributes are specified should look as follows:

```
/                               /
/                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AVERAGE    |      PEAK     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  BURST FACTOR |      MTU      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                               /
/                               /
```

- The ACCESS CONSTRAINTS field is expected to be a set of predicates to be partly decided by the Policy Based Routing effort.

- R_OPEN_CON.PTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OPCODE    |   RESPONSE    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          EXPLANATION          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RALLOC/REASON |   CHECKSUM    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- CLOSE_CON.PTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|     OPCODE    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CHECKSUM    |
+-+-+-+-+-+-+-+-+
```

- R_CLOSE_CON.PTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OPCODE    |    RESPONSE   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CHECKSUM    |
+-+-+-+-+-+-+-+-+
```

- ALLOCATE.PTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|     OPCODE    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      ATTRIBUTES BIT MAP       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
/                               /
/          ATTRIBUTES           /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
/                               /
/              PATH             /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           CHECKSUM            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- R_ALLOCATE.PTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OPCODE    |    RESPONSE   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
/                               /
/              PATH             /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          EXPLANATION          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RALLOC/REASON |   CHECKSUM    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- DEALLOCATE.PTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|     OPCODE    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     RALLOC    |   CHECKSUM    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- SEND.PTP

```
         0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
        +-+-+-+-+-+-+-+-+
        |     OPCODE    |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |         TOTAL LENGTH          |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |              ICN              |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |          FRAGMENT ID          |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
FLAG -->| |       FRAGMENT OFFSET       |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                               |
/                               /
/             DATA              /
/                               /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           CHECKSUM            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- TOTAL LENGTH specifies the length in octets of the data area.
- FRAGMENT ID, FRAGMENT OFFSET, and FLAG control fragmentation and reassembly of data packets. FRAGMENT ID is used to uniquely identify the data packet to which the fragment belongs. FRAGMENT OFFSET specifies the offset in octets of this fragment in the original packet. FLAG specifies whether this is the last fragment by setting the bit to 1.
- CHECKSUM covers the whole data packet.

B. MULTIPOINT CONTROL PACKETS

- OPEN_CON.MTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    OPCODE     |   PROTOCOL    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LENGTH             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          NUM_ENDPTS           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-                             -+
|              CID              |
+-                             -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
/                               /
/          ENDPT_LIST           /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      ATTRIBUTES BIT MAP       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AVERAGE    |     PEAK      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| BURST FACTOR  |    AVERAGE    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     PEAK      | BURST FACTOR  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     DELAY     |     DELAY     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  PACKET LOSS  |  PACKET LOSS  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  OOS PACKETS  |  OOS PACKETS  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CON-TIME    |   CON-TIME    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      MTU      |      MTU      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  COND |  NUM  |    NETADDR    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
/             ACCESS            /
/          CONSTRAINTS          /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           CHECKSUM            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- OPCODE specifies the type of control packet.
- PROTOCOL specifies the higher-level protocol that issues the request.
- The length of the packet in octets is specified in the LENGTH field.
- NUM_ENDPTS is the number of endpoints in the MTP congram.
- CID is the congram identifier.
- ICN is the internet channel number of the congram.
- ENDPT_LIST is the list of endpoints internet addresses.
- The congram attributes and attributes bit map are similar to those used in the PTP case. Bit 13 of the attributes bit map, however, which is unused in the PTP case, represents the congram's FLEXIBILITY in the MTP case.
- CHECKSUM covers the whole control packet.

- R_OPEN_CON.MTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    OPCODE     |   RESPONSE    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-                             -+
|              CID              |
+-                             -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-            ENDPT            -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          EXPLANATION          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RALLOC/REASON |   CHECKSUM    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- RESPONSE specifies the response (ACK or NACK) of a particular endpoint to an MTP congram request.
- CID is the congram identifier specified in OPEN_CON.MTP.
- ENDPT is the internet address of the endpoint sending the control packet.
- EXPLANATION contains additional information about the congram, if necessary.
- RALLOC specifies the resources (desired or minimum acceptable) allocated for this congram.
- If the congram setup to ENDPT fails, the reason(s) for the failure are specified in the REASON field.

- CLOSE_CON.MTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|    OPCODE     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-                             -+
|              CID              |
+-                             -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CHECKSUM    |
+-+-+-+-+-+-+-+-+
```

- R_CLOSE_CON.MTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    OPCODE     |   RESPONSE    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-                             -+
|              CID              |
+-                             -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CHECKSUM    |
+-+-+-+-+-+-+-+-+
```

- ALLOCATE.MTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    OPCODE     |  NUM_ENDPTS   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LENGTH             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-                             -+
|              CID              |
+-                             -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
/          NEXTHOP_LIST         /
/                               /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      ATTRIBUTES BIT MAP       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
/                               /
/          ATTRIBUTES           /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           CHECKSUM            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- R_ALLOCATE.MTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    OPCODE     |   RESPONSE    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LENGTH             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-                             -+
|              CID              |
+-                             -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
/         NEXTHOP_LIST          /
/                               /
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          EXPLANATION          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RALLOC/REASON |   CHECKSUM    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- COMMIT.MTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    OPCODE     |    RALLOC     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-                             -+
|              CID              |
+-                             -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CHECKSUM    |
+-+-+-+-+-+-+-+-+
```

- ABORT.MTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|    OPCODE     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-                             -+
|              CID              |
+-                             -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CHECKSUM    |
+-+-+-+-+-+-+-+-+
```

- DEALLOCATE.MTP

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|    OPCODE     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-                             -+
|              CID              |
+-                             -+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ICN              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    RALLOC     |   CHECKSUM    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- SEND.MTP (refer to SEND.PTP)

References

[1] Akhtar, S., "Congestion Control in a Fast Packet Switching Network," *MS Thesis, Department of Computer Science, Washington University in St. Louis,* December 1987.

[2] Clark, D., "Policy Routing in Internet Protocols," DARPA RFC 1102, SRI Network Information Center, May 1989.

[3] Clark, D., "The Design Philosophy of the DARPA Internet Protocols," *Proceedings of the ACM SIGCOMM'88.*

[4] Deering, S. and D. Cheriton, "Host Groups: A Multicast Extension to the Internet Protocol," DARPA RFC 966, SRI Network Information Center, December 1985.

[5] Estrin, D., "Inter-Organization Networks: Implications of Access Control Requirements for Interconnection Protocols," *Proceedings of the ACM SIGCOMM'86.*

[6] European Computer Manufacturers Association, "Inter-Domain Intermediate Systems Routing," ECMA/TC32-TG10/89/24, 7th Draft, January 1989.

[7] Forgie, J. W., "ST – A Proposed Internet Stream Protocol," DARPA IEN 119, SRI Network Information Center, September 1979.

[8] Huang, Alan and Scott Knauer, "Starlite: a Wideband Digital Switch," *Proceedings of Globecom 84*, 12/84, 121–125.

[9] IETF Open Routing Working Group (Ed. Callon, R.), "Requirements for Inter-Autonomous Systems Routing," DARPA IETF IDEA 007, SRI Network Information Center.

[10] Mazraani, Tony Y. and Gurudatta M. Parulkar, "Specification of a Multipoint Congram-oriented High Performance Internet Protocol," Technical Report WUCS-89-20, Department of Computer Science, Washington University in St. Louis, 1989.

[11] Nakassis, T., "A Model/Approach for Policy Based Routing," Proceedings of the Internet Architecture (INARC) Workshop, December 1987.

[12] National Research Network Review Committee, "Towards a National Research Network," Computer Science and Technology Board, National Academy Press, 1988.

[13] Network Information Center, "Internet Protocol Transition Workbook," SRI Network Information Center, March 1982.

[14] Parulkar, Gurudatta M., "The Next Generation of Internetworking," Technical Report WUCS-89-19, Department of Computer Science, Washington University in St. Louis, 1989.

[15] Tanenbaum, A. S., "Computer Networks," Second Edition, Prentice Hall, 1988.

[16] Turner, Jonathan S., "Design of a Broadcast Packet Switching Network," *IEEE Transactions on Communications, Vol. 36, No. 6,* June 1988.

[17] Turner, Jonathan S., "The Challenge of Multipoint Communication," Technical Report WUCS-87-6, Department of Computer Science, Washington University in St. Louis, 1987.

[18] Yeh, Y. S., M. G. Hluchyj, and A. S. Acampora, "The Knockout Switch: a Simple Modular Architecture for High Performance Packet Switching," *International Switching Symposium,* 3/87.