Report Number: WUCS-89-19

1989-01-01

The Next Generation of Internetworking

Gurudatta M. Parulkar

This paper describes a research effort concerned with the design of the next generation of internet architecture, which has been necessitated by two emerging trends. First, there will be at least a few orders of magnitude increase in data rates of communication networks in the next few years. For example, researchers are already prototyping networks with data rates of up to a few hundred Mbps, and are planning networks with data rates up to a few Gbps. Second, researchers from all disciplines of science, engineering, and humanities plan to use the communication infrastructure to access widely distributed resources in... Read complete abstract on page 2.

The Next Generation of Internetworking

Gurudatta M. Parulkar

Complete Abstract:

This paper describes a research effort concerned with the design of the next generation of internet architecture, which has been necessitated by two emerging trends. First, there will be at least a few orders of magnitude increase in data rates of communication networks in the next few years. For example, researchers are already prototyping networks with data rates of up to a few hundred Mbps, and are planning networks with data rates up to a few Gbps. 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. These trends provide new challenges and opportunities to researchers in the communication field. One such challenge is the design of what we call the very high speed internet (VHSI) abstraction which can help efficiently support guaranteed levels of performance for a variety of applications, and can cope with the ever increasing diversity of underlying networks with rapidly growing user population and needs. Our strategy towards achieving this ambitious goal comprises the following: • Design, specification, and prototype implementation of novel multipoint congram-oriented service that can work well with connection-oriented and datagram high speed networks, can provide variable grade service on a need basis to its applications, and can provide adequate reconfigurability to deal with survivability requirements due to network failures. • Design and implementation of gateway architecture than can support data rates of a few hundred Mbps, can interface with diverse networks, and can implement the congram-oriented service without becoming a performance bottleneck. • Development of analytical and simulation models to evaluate important tradeoffs associated with the design of a congram-oriented protocol, the resource management on diverse networks, and the design of new gateway architectures.

THE NEXT GENERATION OF INTERNETWORKING

Gurudatta M. Parulkar

WUCS-89-19

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

An abridged version of this report appeared in the ACM SIGCOMM Computer Communication Review, Volume 20, Number 1, January 1990.

Contents

1 INTRODUCTION 1

2 BACKGROUND 3

2.1 Internet Objectives 3

2.2 High Speed Networking Approach 5

2.3 Limitations of the Existing Internet Abstraction 6

3 THE NEXT GENERATION OF INTERNETWORKING 7

3.1 Component Networks 8

3.2 Multipoint Congram-oriented High Performance Internet Protocol 11

3.3 Internet Resource Management 17

3.4 Gateway Architecture 18

3.5 Internet Routing 23

4 WORK IN PROGRESS 23

4.1 Multipoint Congram-oriented Internet Service 24

4.2 Resource Management 25

4.3 Gateway Architecture 26

5 CONCLUSION 28

The Next Generation of Internetworking

Gurudatta M. Parulkar

Department of Computer Science
Washington University in St. Louis MO 63130
guru@flora.wustl.edu

ABSTRACT

This paper describes a research effort concerned with the design of the next generation of internet architecture, which has been necessitated by two emerging trends. First, there will be at least a few orders of magnitude increase in data rates of communication networks in the next few years. For example, researchers are already prototyping networks with data rates of up to a few hundred Mbps, and are planning networks with data rates up to a few Gbps. 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. These trends provide new challenges and opportunities to researchers in the communication field. One such challenge is the design of what we call the very high speed internet (VHSI) abstraction which can help efficiently support guaranteed levels of performance for a variety of applications, and can cope with the ever increasing diversity of underlying networks with rapidly growing user population and needs. Our strategy towards achieving this ambitious goal comprises the following:

- Design, specification, and prototype implementation of a novel multipoint *congram-oriented* service that can work well with connection-oriented and datagram high speed networks, can provide variable grade service on a need basis to its applications, and can provide adequate reconfigurability to deal with survivability requirements due to network failures.
- Design and implementation of gateway architectures that can support data rates of a few hundred Mbps, can interface with diverse networks, and can implement the congram-oriented service without becoming a performance bottleneck.
- Development of analytical and simulation models to evaluate important tradeoffs associated with the design of a congram-oriented protocol, the resource management on diverse networks, and the design of new gateway architectures.

1 INTRODUCTION

The ongoing research in the computer communication and telecommunication fields suggests two emerging trends which are complementary to one another. First, in the next few years we will witness communications networks which can support increasingly high data rates [7,8,14,34,36]. 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 [24]. 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. In the case of existing and emerging networks, the diversity includes speed, addressing, packet size and format, routing capabilities, access constraints, error control, resource allocation, and resource monitoring. The application set includes video distribution, computer imaging, distributed scientific computation and visualization, distributed file and procedure access, and multimedia conferencing. The challenge here is to support different applications that require considerably different quality of service in terms of bandwidth, end-to-end latency, errors, packet loss, etc. Moreover, the diversity of networks and applications will remain a fact of life for at least foreseeable future. Thus, a framework which will allow diverse networks to interwork together and 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 [25,33,3]. 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 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.

Federal agencies that support the Internet have recognized its problems and have proposed a three phase plan to create the next generation of communication infrastructure for scientific communications [11,24]. Clearly, phase I and II can be achieved with modest research and with the existing technology. However, phase III is quite revolutionary and poses a number of interesting and challenging research and technological problems which require new solutions and pushing the state of the art in a number of areas. One such challenge is that of developing what we call the very high speed internet (VHSI) abstraction. This abstraction must efficiently support guaranteed levels of performance for a variety of applications, and cope with the ever increasing diversity of underlying networks with rapidly growing user population and needs. An internetworking abstraction is very useful because it decouples the issues specific to higher level applications from the underlying network technology. That is, if an internet abstraction is designed carefully and is rich in its functionality, it can protect transport and application level protocols from the technological changes and evolution of the underlying networks. We claim that the VHSI abstraction must include the following functionality:

- The internet can allow networks to be diverse and autonomous, but must require that they provide their parametric description to the internet, and either do their own resource management or allow directly connected gateways to do it on their behalf.
- The internet abstraction should provide mechanisms for applications to request the quality of service they need and to specify any routing constraints they may have.
- The internet must also include a basic building block or an abstraction which can overcome limitations of the classical connection and datagram abstractions. We introduce a new abstraction called *congram*, which incorporates the strengths of both connection and datagram abstractions, and (hopefully) leaves out their weaknesses[1]. Such a service is key to providing variable grade service to different applications with acceptable reconfigurability to deal with network failures.
- The gateway architectures should be such as to allow variable number of input ports with variable data rates. Also, they have to provide all the internet level per packet processing in hardware in order to carry traffic at full data rate with low latency.

Why these are the essential elements of the future internet, how to provide the functionality stipulated by them, and how effective they are in a real internet environment are some of the questions that we try to address in this paper.

[1] A congram has a path and some statistical resources associated with it, and it is set up by an application usually for the duration of a dialogue/conversation. It is not the same as a virtual circuit, because it does not guarantee delivery of packets in sequence without being dropped or duplicated. Also, it allows low overhead set up and reconfigurations, and allows resource management for each application dialogue/conversation. Finally, the congram and *flow*, introduced by Dave Clark of MIT [3], are similar in semantics. However, congram is different from a flow in terms of its set up and reconfiguration capabilities.

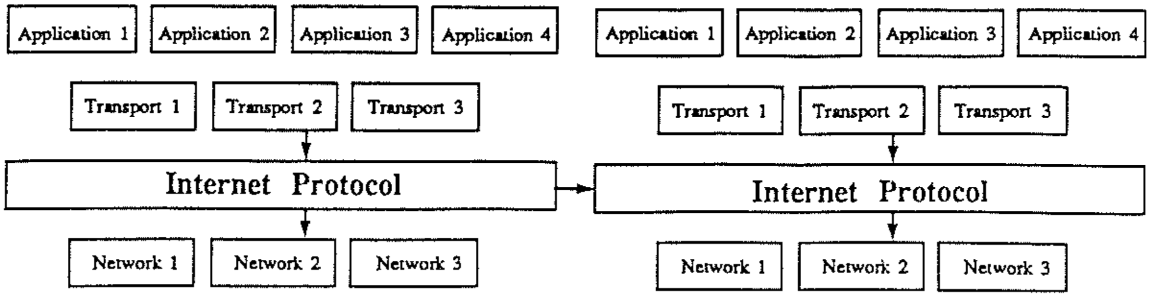


Figure 1: Internet Protocol Hierarchy

In the next section, we briefly describe the important objectives of internetworking, present some fundamental principles of high speed networking, and outline weaknesses of the existing Internet. In Section 3, we propose a design of the next generation internet abstraction which includes comments on component networks' functionality, design of a multipoint congram-oriented internet protocol, resource management across diverse networks, and design of gateway architectures. Clearly, there are a number of questions unanswered about the next generation internet abstraction, and in Section 4 we summarize the work in progress which aims at getting answers to some of these questions. Finally, Section 5 is the summary.

2 BACKGROUND

The purpose of this section is to present an internet framework, necessary to keep the research on the next generation of internetworking in perspective. First, we outline the high level and high priority objectives of the next generation of internetworking. Then, we summarize in abstract terms the principles of high speed networking, derived from the extensive research on this topic in past few years. Finally, we conclude this section by outlining the limitations of the existing internet abstraction to motivate the need for the next generation internet.

2.1 Internet Objectives

In terms of the protocol hierarchy, an internet level protocol interfaces with transport protocols and various network access protocols as shown in Figure 1. In terms of its overall objectives, the internet level creates a virtual homogeneous network on top of diverse networks. That is, it allows transport protocols and applications to operate as if in a homogeneous network and not be concerned with the underlying networks as shown in Figure 2. In terms of packet forwarding, the internet protocol forwards packets from one gateway to another, using the transport facilities of different networks, and using gateways to switch packets between networks. There are also control and routing protocols at the internet level which help gateways to gather routing information and to manage the operation of the internet.

Considering that the internet is essentially a virtual network, its design objectives in part are the same as those of a component network. Thus, the internet has to efficiently support guaranteed levels of performance for a wide variety of applications with growing user population and demands. Note that this statement of goals does not distinguish between network users and network providers, and does not explicitly include a number of other goals, such as a fair billing policy, network security, and network privacy. In the following paragraphs we elaborate on various aspects of our statement of goals:

Efficiency. There are four major components of a computer (inter)network: applications, hosts,

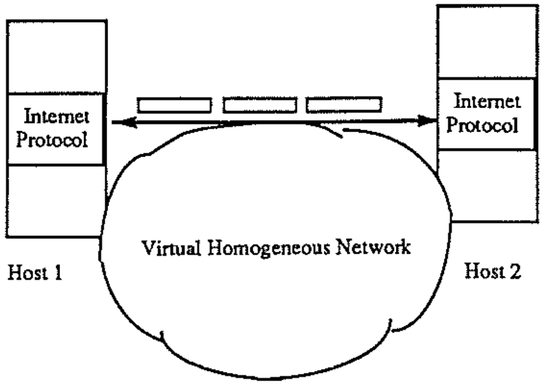


Figure 2: Virtual Homogeneous Network

switching systems, and communication links. Here, we are primarily concerned with the communication links and switching systems. In a high speed environment, communication links can support data rates from 100 Mbps to a few Gbps, and various applications can send and receive data at similar rates. The efficient use of communication links means they have high utilization, that is, most of the time, they are busy carrying useful user information. Efficient use of switching systems means that data streams are switched from input ports to output ports with little queuing delay, allowing high link utilization, and without requiring excessive hardware and software.

Diverse Applications. The communication substrate has to support a variety of applications which require transporting voice, video, and data in an integrated fashion. For example, video distribution, computer imaging, distributed scientific computation and visualization, distributed file access, and multimedia conferencing are all target applications. These applications generate traffic with very different characteristics and have different requirements in terms of expected throughput, delay, errors, and their ability to dynamically adjust resource requirements. Networks should provide mechanisms for these applications to request the quality of service they need and to specify any routing constraints they may have. Of course, networks have to provide mechanisms to provide such a variable grade service and to meet routing constraints of the applications.

Guaranteed Levels of Performance. The ability to have guaranteed levels of performance means that the network should allow applications to meet their performance needs with high probability. Clearly, performance guarantees are statistical and do not imply inflexible and high levels of resource needs. For example, a file transfer application may request the transfer of a several megabit long file to take place within an hour. In this case, the network has to guarantee that the file is transferred in the given time, but the actual transfer may require a small fraction of the link bandwidths, allowing considerable flexibility.

Scalability. Experience with existing networks suggests that the network user community and their needs are rapidly increasing, and that this trend will continue for the foreseeable future. Thus, scalability means that the switching systems, protocols, and other mechanisms should be such that they can grow with the increasing demands and still be efficient and be able to make the required performance guarantees.

It is important to note that these objectives are more difficult to achieve at the internet level than within a homogeneous network, because of a number of administrative and technological reasons: the underlying networks are under different administrative control; they are technologically diverse; the internet cannot be optimized for a few application classes because it is the interface to users of a large number of interconnected networks; and finally an internet abstraction should be designed such that it can survive technological changes to the underlying networks. Table 1 in Section 3.1 lists attributes of various example networks to provide an overview of the network diversity.

2.2 High Speed Networking Approach

In this section we describe, in relatively abstract terms, how high speed networks (HSN) have tried to achieve these goals in a homogeneous environment, which is relatively simpler and more tractable problem. However, this discussion will help us formulate a few principles of high speed networking which are useful at the internet level as well.

Guaranteed Level of Performance. Two approaches are used to make performance guarantees in a network. The first approach involves monitoring and control of resource allocation and usage of each application. This approach requires that an application specify its resource needs a priori, and unless its needs can be met, it is blocked. Once started, mechanisms are provided to ensure that it does not use more resources than requested. Because every application uses only its share of resources, it can meet its performance needs, and help avoid network congestion.

The second approach is to over-engineer the network to the extent that an application is certain to get the resources it needs. Thus, the applications can be unconstrained and still be sure to meet their performance needs and get the guaranteed level of performance.

Clearly, networks use both approaches with one of them being more dominant. Most HSNs use the first approach as the dominant one, because it allows a network to deal with the statistical behavior of applications better, and it can avoid frequent short term congestion in parts of the network [1]. Also, in a large HSN environment, the second approach would require prohibitively large amount of resources to provide adequate over-engineering.

Thus, the first principle of HSN is to do tight monitoring and control of resource allocation and usage of each application to make performance guarantees.

High Efficiency. With the commercial viability of the fiber optic medium already established, the performance bottleneck has moved from communication media to switching systems. Thus, in order to keep up with the data rates of communication links, it is essential that most of the packet processing and other control operations be performed at high speeds, which implies hardware implementations. However, if all the switching logic is to be implemented in hardware or in custom VLSI, the resulting switching system will be very expensive, complex, and inflexible.

The solution used in HSNs involves separating control and data paths, simplifying the data path as much as possible, and implementing the data path in hardware and control path in software. The simple data path can be implemented economically in hardware and can help achieve high link utilization. This strategy works well assuming that the bulk of communication information exchanged is data, and since control operations are done in slower software, they are less frequent and the subsequent data can tolerate this delay.

Thus, the second principle of HSNs is to separate time critical functions from non time critical functions, and try to make the time critical functions as simple as possible in order to implement them economically in hardware.

Scalability. The bottleneck with reference to scalability is within the switching systems' interconnection network. We need to use interconnection networks which scale well with the number of communication links, where the number of links could be as high as few thousands. HSN switching systems use binary multistage routing networks, which are relatively complex and involve multiple store-and-forward operations within the switch, but have excellent scalability. Thus, with these switching systems, the HSNs can scale for a large number of communications links and can potentially be used to create a network as big as the existing telephone network.

Thus, the third principle of HSN is to choose switching and protocol architectures that scale well with the growing user population, with a maximum target population being the users of existing telephone network.

Diverse Applications. There have not been any particularly novel solutions to the problems of dealing with the diverse applications in a high speed environment, except for the following basic consensus: it is too early to decide on an architecture that is suitable for all applications, and so the underlying substrate should allow sufficient flexibility; packet switching is flexible and can potentially support different applications; an integrated multipoint communication facility is useful for a set of applications, and also includes point-to-point communication as a special case. Unfortunately, newer networks have also been (un)intentionally tailored to a subset of applications. The most vivid example of this trend is the ATM effort which has adopted a standard for packet length (64 bytes) which is appropriate for voice traffic, but clearly too short for many other applications and for higher speed communications.

2.3 Limitations of the Existing Internet Abstraction

The functionality typically associated with the internet level consists of packet forwarding between similar and/or dissimilar networks, internet congestion control (or internet resource allocation and management), network management[2], a uniform addressing method, and internet routing [26,2]. We claim that the current internet abstraction has weaknesses in all these areas and needs major improvements to make it appropriate for the next generation of internet. We summarize the weaknesses in the following paragraphs:

Predictable Performance. The internet level does not allow applications to explicitly request the amount of resources and the quality of service desired. Moreover, the internet model is unable to support a predictable level of performance for its applications primarily for two reasons: first, the internet uses a datagram approach for all applications and does not do any explicit resource allocation; second, it expects too little of its component networks. As mentioned earlier, the internet requires that a component network try its best to forward a datagram toward its destination, but allows the network to lose, resequence, and duplicate datagrams. Additionally, there is no attempt to even characterize the throughput that a network can deliver to an application, and no functionality to reserve or preallocate resources for an application. We argue that without proper characterization of component networks and without any resource allocation, it is extremely difficult, if not impossible, to provide predictable performance to applications.

Resource Management and Congestion Control. The congestion control strategy in the current internet has not been working well. It allows gateways and networks to become congested and then uses two mechanisms to clear the congestion: gateways drop (selectively or randomly) packets while congested; and gateways send ICMP *source quench messages* to a host which is sending too many packets too fast [17,27].

It is not difficult to see that these mechanisms are not appropriate for the VHSI environment. First, a number of applications simply cannot tolerate gateways dropping their packets. Second, the source quench message is reacted to with excessive delay, especially in the VHSI, because it takes too long (measured in number of bits) for these messages to propagate and to initiate an action in response. In the mean time, a large number of packets would have been dropped, and the congestion situation might have even changed. We claim that in a VHSI environment, it is more effective to avoid congestion rather than to let the congestion happen and then try to clear it.

Transport protocols also do some congestion control in the internet. They use estimates of round-trip-delays and packet acknowledgements to detect congestion in the internet [18,28]. When

[2] We include security, billing, capacity management, etc., within the network management

congestion is detected, they shrink their window sizes to reduce the offered load to the network. In other words, the end-to-end flow control is used to do the congestion control. This does not work well because it is too slow and conservative to be effective in the VHSI for the same reasons as explained above, it is based on an inaccurate estimate of the congestion, and it would lead to unpredictable performance to applications.

Packet Forwarding in Gateways. A gateway typically consists of two or more network interfaces connected to a general purpose processor and memory. Network interfaces do most of the network specific tasks, and the processor does most of the internet level processing, which is relatively complex, in software. Clearly, such a gateway architecture cannot keep up with the high data rates, and thus, cannot deliver acceptable throughput, measured in number of packets per second, to its users. Also, if a bus is used as the gateway's internal interconnection network, it can saturate even for a modest number of input ports.

The recent designs of gateways have moved the internet level processing to the network interfaces which contain their own processor and memory. The main processor deals with only the exceptional situations, such as setting up routing tables for the network interfaces and internet routing protocol functions. This is only marginally better, because the per packet processing is still done in software on a stored-program processor.

Internet Routing. The Internet is divided into a number of autonomous systems (AS), and an AS uses exterior gateways to spread and gather reachability information about the Internet connectivity for inter-AS routing [21,30,16]. Gateways, responsible for the inter-AS routing, do not use any elaborate routing protocol and do not exchange all the information about the Internet needed by other protocols to perform optimal routing. In short, the inter-AS routing model (essentially the EGP model) does not have sufficient functionality to ensure optimal routes in any sense.

Addressing. The current internet addressing scheme cannot support the expected growth of the Internet, mobile hosts, truly diverse networks (telephone network, for example), and does not provide assistance with routing of packets. Thus, there is a need to design an addressing scheme which can account for these problems.

In summary, although the Internet has been the center of most of the computer networking research and has led to a number of fundamental contributions, the time has come to explore a revised internet abstraction which can correct the above mentioned weaknesses and can work well with the emerging networks and applications.

3 THE NEXT GENERATION OF INTERNETWORKING

In this section we present our design of the next generation internet abstraction, called the very high speed internet (VHSI) abstraction. It resembles the existing internet abstraction only in rudimentary ways. For example, the internet level protocol in the VHSI also interfaces with applications and underlying networks, uses transport facilities of the networks to forward packets, and uses gateways to switch packets between networks. The VHSI abstraction, which includes a number of significant improvements over the current internet abstraction, is shown in Figure 3. 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. A host may include a simple internet router (default next hop for nonlocal hosts), a resource server pointing to a gateway, a network interface, and an interface to a number of transport protocols. A gateway may include inter and intra domain routing, resource servers for multiple networks, and interfaces to several networks.

It is important to point out that the VHSI may also include a connectionless service, a modified version of the current datagram IP. However, we have intentionally focused on only the congram-oriented service, because this is the most novel aspect of the VHSI. There are also a number of issues

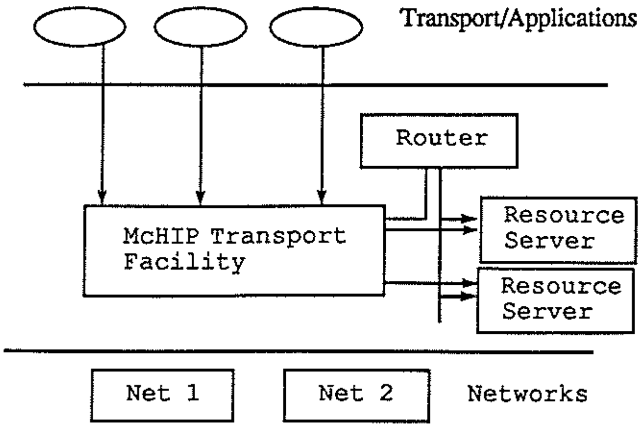


Figure 3: VHSI Internet Abstraction

related to the interoperability of connectionless and congram-oriented services at the internet level, but we have postponed their treatment until after we have acquired sufficient understanding of the proposed congram-oriented service.

The details of the VHSI abstraction are presented in terms of the discussion on the component networks, a brief description of the multipoint congram-oriented internet protocol, resource management strategies for diverse networks, design of gateway architectures, and functionality expected of internet routing protocols. We have included the discussion on the internet routing for the sake of completeness, but it is not within the scope of our research program.

3.1 Component Networks

The VHSI, as 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. Of course, some of the existing networks such as Ethernet and proprietary networks (e.g. SNA and DECNET) will be part of the VHSI because of their continued usefulness in some environments and because of their large penetration. 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. In other words, the network level diversity implies some cost to the component networks in terms of having to provide certain functionality to the internet level. We believe that without imposing such constraints, making end-to-end performance guarantees to applications is very difficult, if not impossible.

Note that the imposition of requirements on the component networks is a departure from the current Internet philosophy. The current Internet model does not expect anything from the component networks, except for best effort delivery of datagrams. This means that a network can lose, duplicate, and resequence packets arbitrarily often. In the following paragraphs we summarize the diversity permitted and the minimum functionality expected of the component networks in the VHSI.

Network Diversity

In terms of diversity, networks can have different speed, packet size, packet format, resource management policies, access protocols, routing capabilities, and access constraints. Figure 4 shows a small part of a hypothetical VHSI, and Table 1 lists important attributes (in some cases assumed) of various networks to get a feel for the diversity. For example, it is assumed that the backbone network of the future ScienceNet (network n3) provides a datagram service at the network level and has mechanisms for its

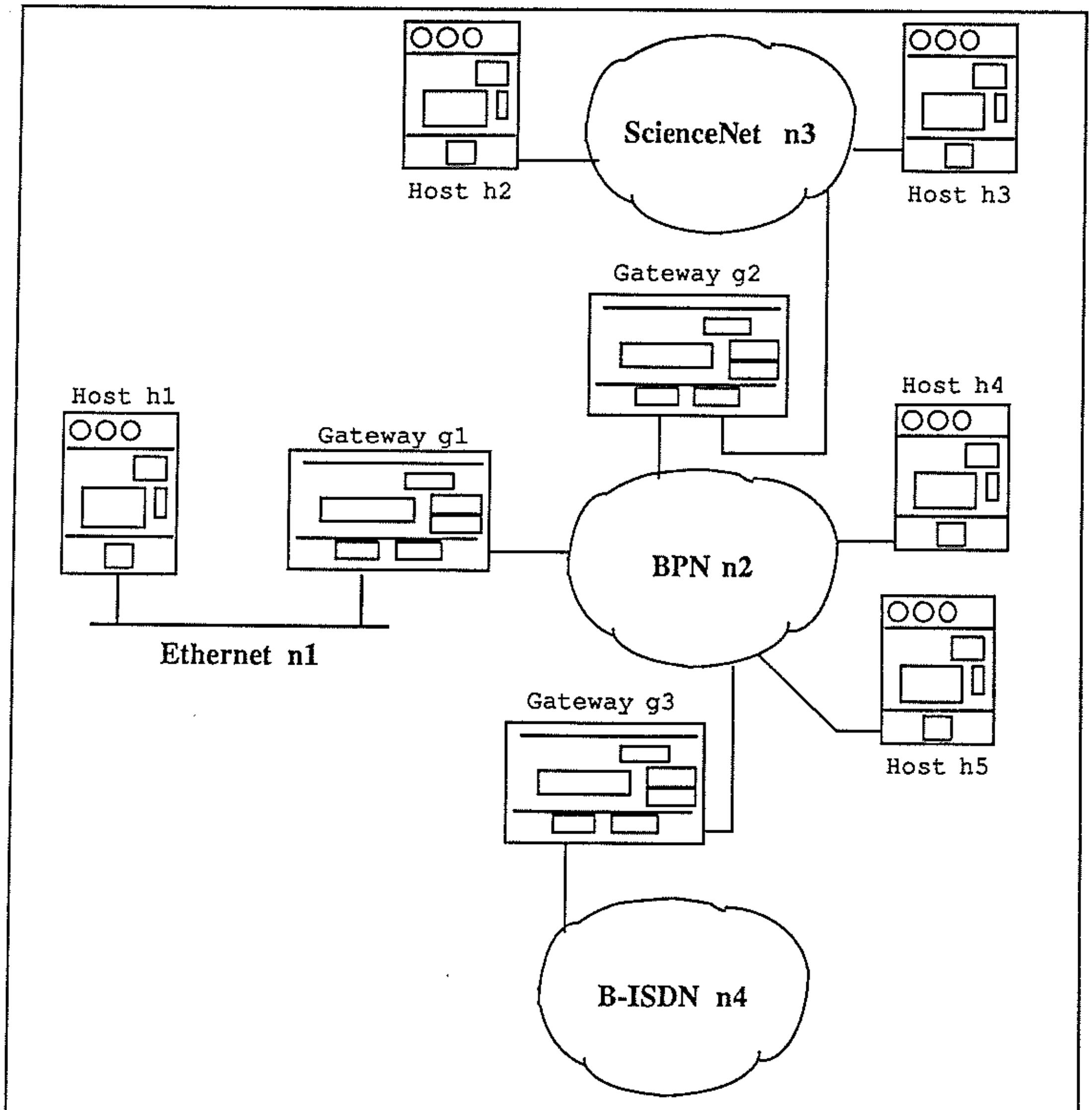


Figure 4: A Small Cross-section of VHSI

gateways (such as g2) to do limited resource allocation and management. Note that the networks in this table are carefully chosen for their diversity. For example, at the national level, an ATM based broadband integrated service digital network (BISDN) is included because telephone companies have strong interests in supporting ATM based public networks; BPN is included because it is an example of the high speed networking technology that a private corporate network may adopt; and ScienceNet exemplifies networks which are appropriate for the scientific community and supported by federal agencies. Similarly, there will be great diversity in local area networks providing differing capabilities. Examples of local area networks that the VHSI would include are Ethernet, FDDI, and PBXs.

It is important to note that the VHSI allows networks which internally support connectionless or connection-oriented access. Also, it allows connections with different semantics, e.g. BPN and FDDI connections are different, but both are permitted in VHSI. Most of the network diversity that we are concerned with is fundamental and cannot be avoided as a result of standardization. For example, it is unrealistic to assume that any standardization will result in all networks using the same packet size and format (such as ATM cells), or communication links of the same bandwidth.

Network	R	Packet Length			RM	DGRAM	CO	PTP	MTP	BDCAST	ACCON
		Fixed	Var	Max							
Ethernet	10		•	1500	none	•			•	•	No Transit
BPN	100	•		72	internal		•		•		none
ScienceNet	100		•	1000	none	•		•			none
Corporate	45	•		200	none		•	•			No Transit
Regional	45	•		200	none		•	•			none
FDDI	100		•	4600	limited	•	•		•	•	none

R	Maximum Transfer Rate in Mbps
Max	Maximum Packet Length in Bytes
RM	Resource Management
DGRAM	Datagram Communication
CO	Connection Oriented Communication
PTP	Point-to-point Communication
MTP	Multipoint Communication
BDCAST	Broadcast Communication
ACCON	Access Constraints

Table 1: Component Network Diversity

Network Requirements

Parametric Description of Network Capabilities. Given the variety of capabilities of the component networks in VHSI, it is essential that the internet protocol include mechanisms for describing the capabilities of networks and for exchanging these descriptions among the gateways to guide the routing decisions. For example, when selecting a route for a connection requiring a particular bandwidth (say 1 Mbps), it is essential that the route not traverse subnetworks incapable of supporting that bandwidth. Similarly, connections requiring low packet loss rates should not be routed through networks that lose packets frequently.

The following list gives a few of the parameters that might be included as part of a network description. A few of these parameters are given relative to a *standard reference path,* which, for example, might be a path carrying heavy traffic between endpoints that are geographically distant.

- *Bandwidth options.* This specifies the various connection bandwidths that the network can support. It may be specified as a single value, a few discrete values, or a range of values.
- *Bandwidth allocation option.* This specifies the type of bandwidth allocation that the network can support. Options include peak bandwidth allocation, statistical allocation (which allows connections with varying instantaneous data rates to statistically share the bandwidth, but allows explicit allocation to ensure predictable performance), and no bandwidth allocation (which provides no performance guarantees).
- *Packet loss rate.* This specifies the frequency of packet loss on a standard reference path.
- *Packet misordering separation.* This specifies the time between transmission of packets on a standard reference path at which the likelihood of packet misordering exceeds some threshold.

- *Packet delay.* This specifies delay on a standard reference path (perhaps average and ninety-ninth percentile).
- *Multipoint capability.* This specifies the ability and characteristics of the multipoint connections that can be supported across the network. Characteristics include number of endpoints and various operations permitted on the connection.
- *Routing Constraints.* This specifies any access constraints that the network may have. For example, a network can decide not to route traffic originating from certain hosts or networks or traffic to have traveled over some networks. Also, some networks can specify not to route any transit traffic through them.

The parameters listed above are envisioned as static and can be obtained from the network operator or the network operations center. It may also be useful to allow more dynamic traffic information to be included and updated periodically. Using parameters such as these, along with the knowledge of individual connection requirements, it is possible for the internet protocols to make better decisions about routing of new connections.

Resource Management. We have argued that in a high speed environment it is necessary to do resource management on a per application congram/connection basis to make performance guarantees to applications. This requires the network to either do its own resource management, or allow its directly connected gateways to provide this functionality. The purpose is to allow every component network to ensure that if an application is using its specified share of resources, the network can meet application's performance constraints, such as end-to-end delay, throughput, and packet loss rate.

Obviously, not all of the existing and the emerging networks have this functionality. However, we believe that suitable mechanisms can be designed which would allow either networks or their gateways to do appropriate resource management while requiring minimal changes to their internal structure. Details of how to do this are presented in Section 3.3.

3.2 Multipoint Congram-oriented High Performance Internet Protocol

Congram Justification

As mentioned earlier, one of the strengths of the VHSI is its multipoint congram-oriented internet service, which can provide variable grade service with performance guarantees to applications. The protocol primarily responsible for providing this service is called the Multipoint Congram-oriented High performance Internet Protocol (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 [6,35]. 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 strengths of both the connection and datagram.

Connection vs. Connectionless

The issue of connection vs. connectionless service is at least as old as computer communications and has been a continual source of religious debates. The reason for its persistence 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" as indicated by the following paragraph from this report [24]:

> Current connectionless services create significant overhead per packet, and this overhead implies severe limitations on packet rates in the switch. The source of the current connectionless world in which our networks and gateways find themselves was the tradeoffs among bandwidth, storage, and switch complexity. However, these tradeoffs are changing dramatically, and certainly there will be a significant change by the time the phase 3 networks arrive.
>
> The information available with connection-oriented communication can be extremely valuable in simplifying the processing requirements of the switch; this is especially important since the switch is likely to become the bottleneck in phase 3.

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" [12].

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 call this connection abstraction 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[3]. 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 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 Overview

The purpose of this subsection is to give an overview of McHIP. 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 5. 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). The details of tracks B and C are left for the future reports, but the motivation for them is presented later in this section.

The details of track A are presented in terms of three phases of a congram: congram set up, operation, and termination. Figure 6 shows messages exchanged between the transport/application protocol and McHIP during these three phases. The internet is considered as a black box which can route a multipoint congram and can transport data among the endpoints. The attributes of a congram are also shown. They fall in four categories: bandwidth, delay, reliability, and access permissions. The bandwidth

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

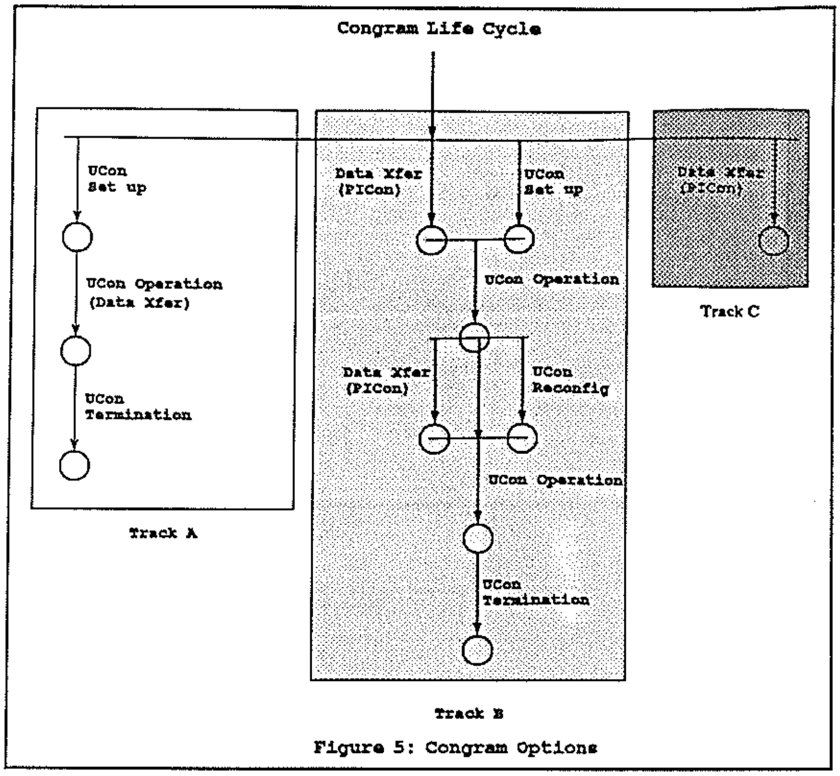


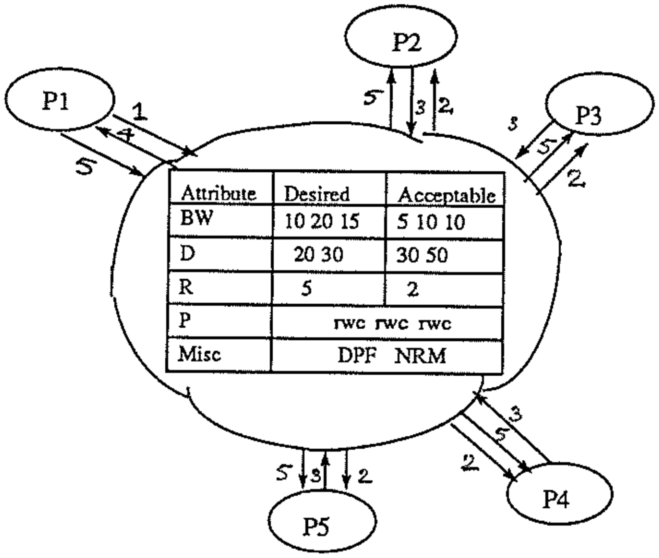
Figure 5: Congram Options

attributes include values for peak bandwidth, average bandwidth, and burst factor, at the desired and minimum acceptable levels. The degraded performance flag (DPF) indicates if the application can tolerate performance degradation in case of network congestion. Thus, this flag allows an application to specify its willingness to let the network reduce its resources during the life of the congram. The delay attributes (D) include minimum and maximum delay for any packet at the desired and minimally acceptable levels. The access permissions (P) for the congram are defined for the owner, group, and others, and within each class, an endpoint can have read as well as write permissions. The network route mask (NRM) is a set of predicates that specify the routing constraints of the congram. It is important to note that all applications need not specify all attributes, and that what constitutes the appropriate list is a subject of further research. Now let us consider each phase of the congram in detail.

Congram Set up

The congram set up phase typically consists of identifying a path, allocating resources on this path, and initializing appropriate tables and hardware mechanisms to ensure that subsequent packets will be switched with minimal processing. As an example, consider Figure 4 and assume that host h1 on Ethernet n1 is trying to initiate a multipoint congram with hosts h2–h5.

Host h1 (McHIP-h1) needs to query the resource server on gateway g1 to check if a congram of given attributes can be supported on network n1. Assuming there is enough bandwidth available on network n1, the resource server will allow the congram. Subsequently, McHIP-h1 will learn from its routing server that hosts h2–h5 can be reached via g1. Hence McHIP-h1 sends the inet-open-con() request to McHIP-g1, which consults its routing server to decide that h4 and h5 are directly reachable on network BPN and that h2 and h3 are reachable via g2. McHIP-g1 decides to open a multipoint congram on BPN with g2, h4, and h5 (which, for BPN consists of opening a congram and adding each endpoint singularly). Subsequently, McHIP-g2 forwards inet-open-con() to McHIP-h2 and McHIP-h3. The names and sequence of the message exchange are documented in Figure 7.



Attribute	Desired	Acceptable
BW	10 20 15	5 10 10
D	20 30	30 50
R	5	2
P	rwc rwc rwc	
Misc	DPF NRM	

1) inet-open-con sent
2) inet-open-con received
3) inet-open-con-ack sent
4) inet-open-con-ack received
5) Data transmission begins

BW -- Bandwidth
D -- Delay
R -- Reliability
P -- Access Permissions
NRM -- Network Route Mask
DPF -- Degraded Performance Flag
rwc -- read/write/check

Figure 6: Multipoint Congram in McHIP

In any case, every gateway agrees to route a multipoint congram only if the corresponding network has enough resources for the congram. In the case of networks such as Ethernet and ScienceNet, which use the datagram approach, suitable mechanisms are provided for gateways to monitor the resource usage within the network. Gateways are authorized to block congrams if they determine that enough resources are not available to meet the performance needs of the application.

During the congram set up, gateways have to do various table initializations to simplify the per packet processing for the subsequent packets. There are two important points to note. First, every gateway may perform these functions differently, depending on the type of networks that it is connected to. Second, congram set up operations, though similar to their counterparts in other connection-oriented protocols, are additionally complex because of the diversity of underlying networks. For example, gateway g2 in our example has to remember that when a packet is received from gateway g1, it needs to send copies of the packet to both hosts h2 and h3. Additionally, g2 may initialize some fragmentation and reassembly logic to ensure that packets coming on ScienceNet get fragmented into fixed size packets before being sent on to BPN.

Congram Operation

During the life of a congram, there are primarily two types of operations: data transfer and modifications of congram attributes.

Data transfer operations involve the transfer of data packets from the source to various endpoints in a multipoint congram. The per packet processing for a data packet typically involves matching the congram id of the input packet in a table to decide on which output link(s) to send the packet. However, the internet per packet processing in a gateway is more involved and depends on the component networks. For example, ScienceNet (n3) does not support multipoint congrams, and therefore, gateway g2 must maintain multiple point-to-point congrams, and make copies of input packets and send them

```
inet-open-con()                  P1 ⟹ McHIP-h1
    inet-resource-request()      McHIP-h1 ⟹ RS-g1
    inet-resource-ack()          RS-g1 ⟹ McHIP-h1
inet-open-con()                  McHIP-h1 ⟹ McHIP-g1
        bpn-open-con()           McHIP-g1 ⟹ BPN
        bpn-open-con-ack()       BPN ⟹ McHIP-g1
        bpn-add-ep()             McHIP-g1 ⟹ BPN for h4
        bpn-add-ep-ack()         BPN ⟹ McHIP-g1
        bpn-add-ep()             McHIP-g1 ⟹ BPN for h5
        bpn-add-ep-ack()         BPN ⟹ McHIP-g1
        bpn-add-ep()             McHIP-g1 ⟹ BPN for g2
        bpn-add-ep-ack()         BPN ⟹ McHIP-g1
inet-open-con()                  McHIP-g1 ⟹ McHIP-h4,McHIP-h5,McHIP-g2
inet-open-con-ack()              McHIP-h4 ⟹ McHIP-g1
inet-open-con-ack()              McHIP-h5 ⟹ McHIP-g1
inet-open-con()                  McHIP-g2 ⟹ McHIP-h2
inet-open-con()                  McHIP-g2 ⟹ McHIP-h3
inet-open-con-ack()              McHIP-h2 ⟹ McHIP-g2
inet-open-con-ack()              McHIP-h3 ⟹ McHIP-g2
inet-open-con-ack()              McHIP-g2 ⟹ McHIP-g1
inet-open-con-ack()              McHIP-g1 ⟹ McHIP-h1
```

Figure 7: Congram Setup

to appropriate endpoints on n3. Also, gateway g2 interfaces a connection oriented network (n2) and a datagram network (n3). Thus, while forwarding packets from n2 to n3, g2 has to translate the connection id in packets to network addresses of destinations on n3. Similarly, when g2 forwards packets from datagram network n3 to connection-oriented network n2, it has to do some sequencing of packets arriving from n3. Also, we expect that the connection oriented networks and datagram networks will have drastically different packet lengths, and therefore, a gateway such as g2 must perform packet fragmentation and reassembly. We believe that it is better to do reassembly at the intermediate gateways than to send very short packets into a datagram network, resulting in inefficient use of network resources.

McHIP must also make performance guarantees to applications and avoid congestion using tight monitoring and control of resource usage. This implies that gateways act as check points to ensure that the traffic characteristics of a congram do not change drastically at the network boundaries. If they do, the gateway must perform appropriate traffic smoothing and sequencing. Section 3.4 describes the per packet processing at a gateway in more detail.

In addition to data transfer operations, we wish to allow congram modification operations which include adding a new endpoint, deleting an endpoint, changing attributes of the congram, and re-routing part of the congram in the case of a network failure. Most of the congram modification operations are done in the control processor of the gateway, and are therefore, slow compared to packet forwarding. However, while the modification is being executed, the application can continue to use the congram at its old specifications. For example, an application may request an increase in its bandwidth, and while the internetwork is attempting to allocate more bandwidth, the application would be able to send data at its previous rate. Again, diversity of networks and elaborate modifications of a congram make the McHIP more complex and challenging to design.

Congram Termination

Congram termination is relatively simple and essentially involves making sure that the data in transit is taken care of, and the resources of the congram are deallocated. We allow only the originator of the congram to close a multipoint congram. Thus, in our example (refer to Figure 4), the application at h1 may decide to close the congram, which results in McHIP-h1 sending a inet-close-con() to McHIP-g1. The McHIP-g1 forwards this inet-close-con() to h4, h5, and g2 and waits for acknowledgements. McHIP-h4 and McHIP-h5 notify their respective applications of the event and send an inet-close-con-ack() to McHIP-g1. McHIP-g2 forwards the inet-close-con() to h2 and h3 and also waits for acknowledgements. When McHIP-g2 gets the acknowledgements, it notifies its resource server to deallocate resources of the congram, and it forwards the acknowledgements to McHIP-g1. Similarly, McHIP-g1, after receiving acknowledgements from McHIP-h1, McHIP-h2, and McHIP-g2, closes the multipoint connection on BPN, sends the acknowledgements to McHIP-h1, and finally also deallocates the resources on Ethernet n1.

Motivation for Perpetual Internet Congram (PICon)

Two major concerns, which may also be associated with congrams, with the connection-oriented approach are the following:

- Connection set up overhead, in terms of latency, may not be acceptable to applications that have small amounts of data to send or simply cannot wait for the connection set up time.
- Connection reconfiguration, which may be necessary due to network failures, is a high overhead operation, involving identification of a new path and set up of new tables (modification of state information). During this time, either service is disrupted or packets are lost. Service disruptions and lost packets are difficult to deal with in a traditional connection-oriented approach, because the connection must provide a perfectly reliable service to its higher level protocols. In other words, the connection oriented approach is less robust to network failures than the datagram approach.

It is important to note that McHIP provides only a plesio-reliable service, and therefore, its higher level protocols would include appropriate functionality to deal with the lost packets and service disruptions [31,32]. Thus, the VHSI abstraction is inherently robust (as the datagram model), and allows McHIP to deal with network failures by doing nothing, and letting the higher level protocols take care of them. Note this is the same approach as that of datagram IP and has worked very well in the existing Internet. Of course, it is important to note that the VHSI expects network failures to be rare. Also, reconfigurability due to network congestion is an unlikely event, because McHIP emphasizes congestion avoidance by explicit resource allocation. Thus, in the normal course of operation, McHIP expects and delivers high performance with high predictability to its higher level protocols, but in the case of rare network failures, it is acceptable for McHIP to let the higher level protocol try to recover from failures. However, we want to provide mechanisms which can reduce the impact of congram set up and reconfiguration on higher level protocols.

We deal with these issues by allowing two types of congram at the internet level: user and perpetual internet congrams. A congram can be in either the transient or established state. A user congram is set up, used, and terminated by an application. An established user congram has endpoints, path, and resources associated with it, otherwise, it is in the transient state. Thus, a congram that is being set up or reconfigured is in the transient state.

A perpetual congram is a long lived congram between McHIP entities at some subset of gateways and hosts. Perpetual congrams are to be configured such that all of them together cover most of the internet topology. The purpose of perpetual congrams is to carry data for congrams that are in the transient state (which do not yet have all resources allocated to them). It is possible that one transient congram may use the concatenation of (segments of) multiple perpetual congrams. Thus, the perpetual congrams provide temporary resources to congrams in the transient state in order to allow applications to send data during congram set up and reconfiguration. Under this scheme, if an application has a small amount of data to send and does not want to set up a congram of its own, it can send this data on a perpetual congram.

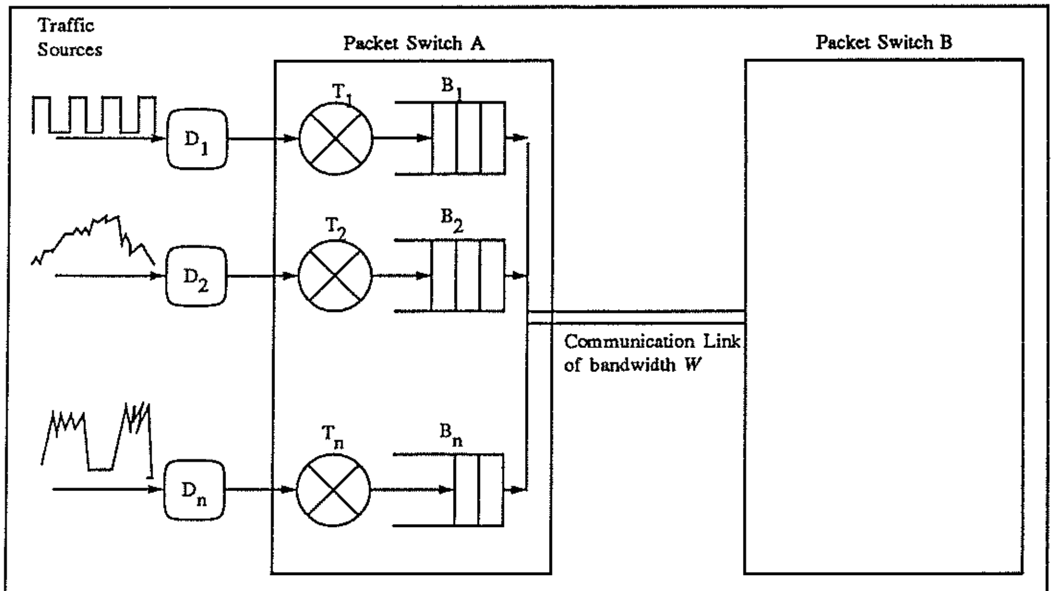


Figure 8: Basic Resource Management Model

This idea of multiplexing data from user congrams in the transient state onto perpetuai congrams has a lot of promise, because with this functionality, we can have a congram abstraction which provides variable grade service and performance guarantees, and which also allows efficient reconfigurability. Thus, this kind of congram abstraction can have advantages of both congram-oriented and connectionless approaches.

For the sake of clarity, this brief description of McHIP has not included any discussion on how the protocol deals with error conditions and with link and gateway failures. The full specification of the protocol, which is nearing completion for a preliminary version, accounts for these situations [20].

3.3 Internet Resource Management

One of our major goals is to 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 do monitoring and control of resource allocation and usage on a per congram/connection basis. Figure 8 shows the basic model for resource management on a point-to-point channel connecting two packet switches. The bandwidth of the channel is W bps and each packet switch has B packet buffers. Each of m congrams supported on this channel is statistically allocated some fraction w_i of the bandwidth W (called effective bandwidth) and is also allocated b_i buffers. Whenever a new congram is to start, it specifies its resource and performance needs using an *application description model* (ADM). The ADM is specified as a number of parameters, such as $D = p_1, p_2, \cdots p_n$, where p_1 may be the peak bandwidth and p_2 the average bandwidth requested by the congram. D is used to decide the congram's effective bandwidth and buffer requirements, and the congram is allowed to start only if these resources are available. Otherwise, the congram is blocked. Once a congram is established, suitable enforcement mechanisms are provided in terms of traffic valves (t_i) to ensure that the congram is not using more than its specified share of resources. Of course, the basic purpose of all this is to select D and compute w_i and b_i such that we can maximize the channel utilization, minimize blocking, and still meet the performance needs of applications with high probability.

Only a few networks do such elaborate resource management. For example, BPN uses $D =$ *(peak-bw,* *average–bw, burst–factor)*, and some other networks use a simplistic $D = (peak\text{–}bw)$ [1]. 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 gateways 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 example, a datagram network does not do any explicit resource management. Our scheme suggests that all or a subset of its directly connected gateways act as resource servers, and thus keep track of all active congrams and available resources in the network. Every time a new internet congram is set up within or across this network, one of these gateways is consulted to check if appropriate resources are available to support this congram. Various gateways communicate with each other and possibly with packet switches within the network to ensure that their view of resource availability in the network is consistent.

In the case of broadcast local area networks (such as Ethernet), this scheme can be easily implemented with good results. The gateway keeps a record of all active congrams with their resource needs and also monitors traffic on the broadcast channel. This furnishes an accurate state of resource availability to the resource server, and thus allows the gateway to make decisions about new congram requests. Of course, if the network carries datagram traffic, suitable mechanisms are provided to ensure that it does not affect the congram traffic.

In the case of a wide area point-to-point network, design of the resource servers is a relatively complex problem. Our strategy is to require every packet switch to periodically report availability of resources on its output links to one of the gateways. The gateway compiles this information along with the static parametric description of the network and resource usage of all active congrams in a resource database. Gateways also periodically exchange appropriate information from their resource databases with each other to ensure that their view of resource availability in the network is consistent. Obviously, the actual resource availability is constantly changing in a real network, especially in the presence of datagram traffic, and gateways have to make the decisions based on outdated information. We believe we can design update mechanisms which are robust to the short term perturbations resulting from information not yet reported to gateways. For example, gateways can maintain a multipoint congram among themselves with sufficient resources allocated to the congram. Thus, the resource updates in the network are likely to be propagated promptly without being discarded, even in the case of temporary overload. Also, the resource update information contains short term as well as long term usage patterns, which help gateways to make more accurate judgements about the state of resource availability in the network.

Another issue to be considered in the case of wide area datagram networks is that packets or datagrams of an internet congram may travel on different routes, and therefore, gateways need to consider alternate paths and have to allocate resources on those paths. In datagram networks that allow source routing, this is not a problem, because the gateway can specify a source route and allocate resources only on this path. It is important to note that ANSI and other routing standards are moving towards supporting the source route option for other reasons, but it is also useful for resource allocation [5]. Without such an option, a gateway must allocate resources on alternate paths based on the expected fraction of traffic on each path. Feasibility and effectiveness of such resource allocation methods on datagram networks is one of the topics of the research in progress.

3.4 Gateway Architecture

As mentioned earlier, a gateway in the VHSI has to implement the MCHIP protocol and provide the functionality to be a network resource server and an internet route server. The gateway architectures must be such as to allow efficient MCHIP implementation. Thus, the important design goals for a gateway

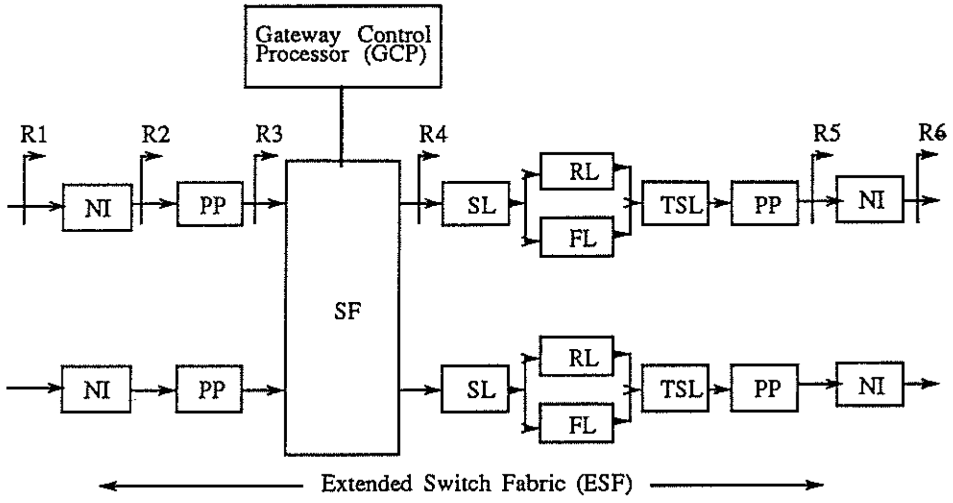


Figure 9A: Proposed Gateway Architecture

R1, R6	Network Header	McHIP Header	Data	McHIP Trailer	Network Header
R2	McHIP Header	Data	McHIP Trailer		
R3,R4	SF Control	McHIP Header	Data	McHIP Trailer	
R5	NI Info	McHIP Header	Data	McHIP Trailer	

Figure 9B: Packet Formats in a Gateway

architecture in the VHSI include the ability to interface with variable number of input ports (2–64), to interface with networks that support data rates up to a few hundred Mbps with high link utilization, and to switch input packets with latency less than a few milliseconds.

A gateway, to the first approximation, is a switch and can be designed using a switching fabric of a fast packet switch as shown in Figure 9. The figure also shows the internal encapsulations and decapsulations of a packet at important reference points in the gateway. All the per packet processing is done in hardware using an extended switching fabric (ESF), and the McHIP congram, resource, and route management are implemented on a gateway control processor (GCP), connected to the ESF. Thus, the GCP receives McHIP requests such as `open_con()`, `close_con()`, as well as resource and route requests and updates. It processes these requests with the help of the resource server and routing server and initializes the appropriate logic in the ESF to facilitate the subsequent packet forwarding. An important aspect of this architecture is that the critical path, consisting of per packet processing, is implemented in hardware for high speed operation, and the non-critical path, consisting of congram, resource, and route management, in software on the GCP for reasons of flexibility and economy.

Operation of the ESF is described next. The network interface (NI) does the network specific encapsulation and decapsulation. The input packet processor (PP) does error checking, table lookup(s) on congram id to decide output port(s) and the number of copies to be made, and encapsulation of the packet for internal use. The switching fabric (SF) is responsible for routing packets from an input port to the appropriate output port or to a set of output ports which means making the required number of copies and routing each copy to an appropriate output port. Then on the output side, the packet may undergo operations such as sequencing (SL — sequencing logic), fragmentation (FL), reassembly (RL), and traffic smoothing (TSL — discussed in Section 4.3). The output PP adds some network specific control information (for example, a source route) to be used by the NI and makes additional copies of the packet to be sent on the corresponding network. Note that a gateway has to send multiple copies of a packet on the same output port in the case of a multipoint congram whose endpoints are on a network that does not internally support multipoint communication. These copies are not made within the ESF, but are made by the output PP. In the following paragraphs we discuss the design of various building blocks of the gateway.

Switching Fabric. As mentioned earlier, the gateway can use the same type of switching fabric as used in fast packet switches. However, two new issues arise in this environment that need special consideration: switch size and packet length. Switch size has to do with the number of ports in the switch fabric, which can range from 2 to 64. For example, a gateway connecting a LAN to a backbone wide area network requires only two ports, which is quite common in the existing internet. In VHSI, however, we also expect to see larger gateway implementations which will interconnect several networks with 5–10 ports per network. For example, a gateway may connect 10–20 LAN segments of a campus network directly to a backbone network, in order to avoid the performance penalties resulting from the hierarchical structure of the campus network. Another example is that of two high speed public networks that have enough traffic between them to require approximately ten communication lines interconnecting them via a gateway. In these cases, the gateway obviously needs at least tens of ports.

When the number of ports is small (e.g. ≤ 8), multistage switch fabrics are unnecessarily complex, and a simpler crossbar implementation can be satisfactory. As an extreme, a bus is adequate for a gateway connecting only two ports. Thus, the selection of a switch fabric for a gateway depends on the gateway connectivity and its expected growth.

The second issue has to do with the packet length for the switching fabric in a gateway. Most high speed packet switches use fixed length packets, but a gateway invariably has to interface to networks with different packet lengths and to networks that allow variable length packets. Thus, the switch fabric in a gateway must either switch variable length packets or fragment input packets internally to fixed length packets, switch them using a fixed length packet fabric, and reassemble them at the output as needed by the next hop network[4]. Clearly, both approaches have their relative advantages.

Our aim in the design of a VHSI gateway is to use an existing switch fabric, without modifications, and design additional hardware around it to do the internet specific tasks. Two switch fabrics that we are interested in are the Knockout and BPN. The Knockout switch allows variable length packets and is optimized for the number of ports in the range of 0–64 [36,37]. The BPN switch is attractive because it is being locally developed as part of another related research project, and thus, would be readily available for the prototype effort. The BPN switch is also ATM compatible, and thus, the gateway design based on the BPN switch fabric would allow future VHSI gateways to use a wide variety of ATM compatible switching fabrics.

The most important part of the high speed gateway architecture is the hardware implementation of the per packet processing operations such as sequencing, fragmentation, and reassembly. Implementation of these operations in hardware is discussed next.

[4] Note that this internal fragmentation and reassembly is in addition to the internet level fragmentation and reassembly.

Fragmentation and Reassembly. Considering the increasing discrepancy in packet lengths, in only a few cases can congrams be established to use paths which avoid fragmentation and reassembly without any performance penalty. Thus, fragmentation and reassembly at network boundaries will be a necessity on most of the internet paths. Note, for example, that sending 64 byte long ATM cells on a datagram network with a maximum packet size of 2000 bytes would result in unacceptable inefficiency. Assuming we do the fragmentation and reassembly at network boundaries, the important question is: should it be done at the internet level within the gateway, or should it be left to individual networks? Though it is tempting to leave this functionality to individual networks, and thus, simplify the internet level processing, we argue that it is a bad idea. A gateway is responsible for setting up internet congrams, and for making performance guarantees to applications, and therefore, it is the most appropriate entity to have control over fragmentation and reassembly. Also, the basic purpose of gateways is to deal with network diversity and relieve the individual networks and applications from the complexity (such as fragmentation and reassembly) arising from this diversity. In the following paragraphs we show how fragmentation and reassembly can be implemented in hardware along with other internet functionality.

The fragmentation of internet packets is relatively straight forward. The fragmentation logic has to receive a stream of input packets (long packets) with MCHIP header, divide each packet into smaller fragments, copy the appropriate header information to each fragment from the original packet (with some modifications), and send the fragments out. The fragmentation can be done on the fly without having to ever buffer more than one packet. Clearly, fragmentation logic is a good example of a *synchronized streams processor* (SSP), which takes a stream of packets, performs a relatively simple and prespecified transformation on them, and outputs the transformed stream. As part of the BPN project, a high level SSP silicon compiler (SSPC) has been developed which can take the functional description of a SSP and generate a VLSI design for SSP implementation [29].

Implementation of the reassembly logic in hardware for the general case is considered complex and harmful [22], but it can be simplified in the VHSI environment because gateways have more knowledge about the underlying networks via the parametric description, and because packets belonging to a congram normally travel on the same path. For example, gateways know the packet misordering probability and misordering separation for a given network. Thus, the reassembly logic can determine how many buffers to allocate and how long to wait for an out of sequence packet. The reassembly essentially involves sequencing input packets, copying data parts into bigger segments, and generating the right MCHIP header for the reassembled packet. The packet sequencing logic is the most complex component and is discussed later. Once the packets are in the right sequence, the reassembly is again easy to implement as an SSP. Note that for fragments that arrive too late or do not arrive at all, the reassembly process times out and either sends the partially assembled packet or discards it. Both approaches are useful for different applications.

It is important to note that the gateway has to concurrently do reassembly of packets on multiple channels for a given output port. The maximum number of concurrent reassemblies is equal to the number of possible logical channels, which is very large (e.g., 2^{16} for 16 bit long logical channel numbers). However, realistically there are only a small fraction of logical channels in use at any time, and only a subset of these channels may require reassembly. Thus, it is reasonable to assume that there are 16–32 concurrent reassemblies. Of course, if all reassembly pipelines are busy, and a new congram request is received which requires reassembly, the gateway can deny this request.

Packet Sequencing. The internet gateways can do limited packet sequencing to compensate for networks that may misorder packets. The idea here is not to guarantee perfect sequencing, because this functionality belongs to transport and application protocols and should not be duplicated at the internet level for efficiency reasons. However, we argue that because packet sequencing is included in reassembly logic, which is implemented in hardware, this functionality can be made available to other congrams, without significant penalty in terms of delay. Packet sequencing within the gateway does help in making better performance guarantees to the application. Figure 10 shows the block diagram design of the packet sequencing logic. The major components include

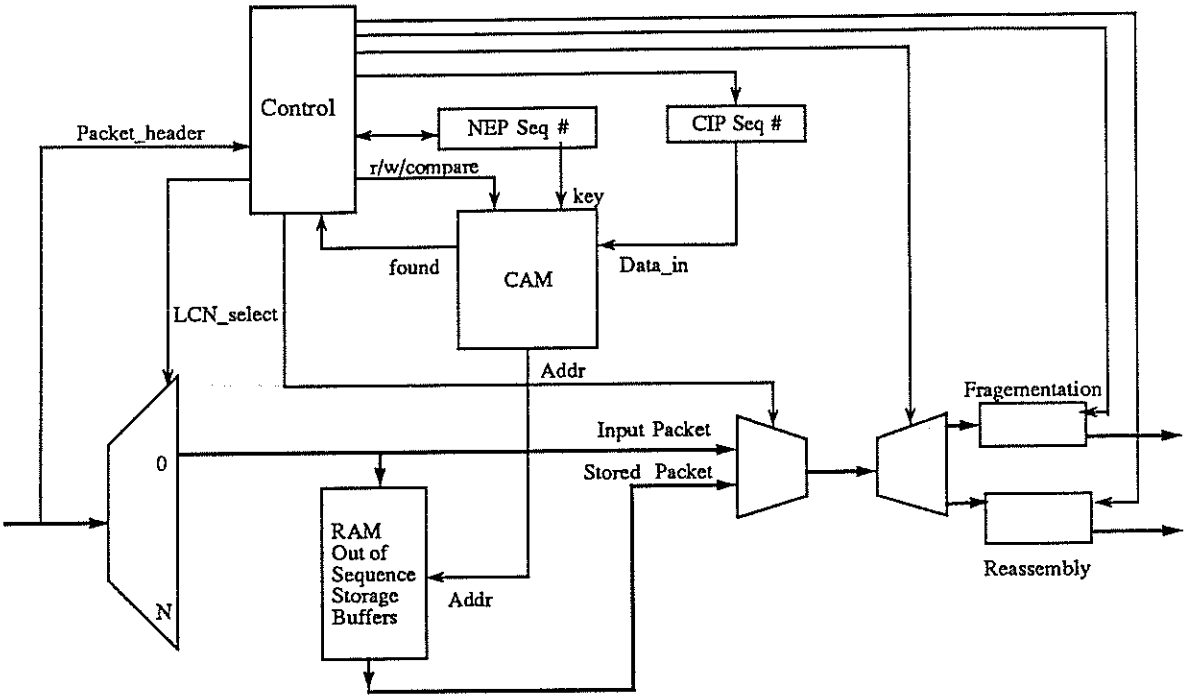


Figure 10: Packet Sequencer for a VHSI Gateway

- a RAM to provide storage for the out of sequence packets
- a CAM (content addressable memory) to store the sequence numbers of out of sequence packets waiting in RAM
- a register (NEP) to keep track of the sequence number of the next expected packet in sequence
- a register (CIP) to contain the sequence number of the current input packet
- multiplexers and demultiplexers
- control logic

At the time of the congram set up, the NEP register is loaded with the expected sequence number of the first packet and a suitable timeout value is selected which decides how long to wait for an out of sequence packet before considering it lost. When a packet is received, its sequence number is compared with the next expected sequence number, and if it matches (NEP = CIP), the packet is routed directly to the output. This is the normal case. If CIP and NEP do not match, the packet is stored in RAM with its sequence number and a pointer in CAM. For every packet cycle, the next expected sequence number is also matched with CAM keys to check if the packet to transmit has previously been received. If it has, the CAM gives the address of the packet in the RAM which is used to read out the next packet onto the output. The control logic is responsible for enabling the appropriate multiplexer and demultiplexer selects and also for keeping track of timeouts. This scheme can be implemented using two chips: one for the control and CAM and the other one for the RAM. However, it requires the RAM access times to be twice as fast as the input/output data rate, because in a given packet cycle, it is possible that the output packet is read out from RAM and an out of sequence input packet is being copied into RAM. We could reduce the speed requirements on RAM by using parallel or interleaved memory structures but that would increase the memory chip count.

As mentioned earlier, a gateway has to do concurrent fragmentation and reassembly for multiple logical channels, which means we need those many copies of the sequencing, fragmentation and reassembly logic. It is reasonable to assume that one pipeline consisting of fragmentation and reassembly can be implemented on three custom chips, and thus, for N concurrent pipelines, we need $3N$ chips.

It is important to note that we have simplified the packet forwarding in a gateway by using a congram-oriented protocol, but have made it more complex by introducing packet sequencing, fragmentation, reassembly, and traffic smoothing. We believe that this is consistent with our objectives of achieving higher throughput from the gateways and making performance guarantees to applications.

3.5 Internet Routing

The purpose of this section is to summarize the functionality expected of the internet routing protocols in the VHSI. Although internet routing problems are not within the scope of this research, there are other research groups that are actively pursuing these issues. We summarize the requirements for the internet routing within the VHSI in the following paragraphs:

Multipoint routing. An important aspect of the VHSI abstraction is a multipoint congram-oriented service, which requires an internet routing protocol to take a set of endpoints and decide the next hop for each in order to build a corresponding congram spanning tree. Of course, the congram tree should be such as to optimize an appropriate internet cost function.

Routing based on resource requirements. In order to make performance guarantees, a congram is routed such that its resource needs can be met with high probability. Thus, the internet routing protocol has to account for the resource requirements of the congram (or the quality of service it needs) and for the resource availability of the underlying networks. Note that the resource requirements of a congram may be specified using a set of parameters as described in Section 3.2.

Routing based on access constraints. As mentioned earlier, a network in the VHSI can specify certain access constraints to ensure that its resources are used to carry only authorized traffic. Similarly, an application can specify routing constraints with respect to the subnetworks and nodes traversed. Clearly, the internet routing protocol has to account for these access/routing constraints while establishing congrams.

Standard requirements. In addition to above requirements, the internet routing protocol has to meet a number of other *standard* requirements. For example, it should be stable and quickly converge to a solution in the event of topological changes, should allow load balancing on multiple paths, should dynamically adjust to significant network changes, and should allow the internet to be hierarchically organized in domains and subdomains.

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 [9]. For example, the IETF (Internet Engineering Task Force) working group on Open Routing and Internet Task Force on Autonomous Networks are developing routing models which would include support for policy based routing and type of service routing across a large internet of diverse networks [15]. Similarly, NBS and ANSI have proposals which also include support for policy based routing [23,10].

4 WORK IN PROGRESS

We have presented a novel internet abstraction, called the VHSI abstraction, as a candidate for the next generation of internet. The description of the VHSI abstraction in the previous section has established its viability. However, there are still a number of research questions to be addressed, and a number of design exercises to be undertaken to resolve the associated tradeoffs, and demonstrate its feasibility in a realistic environment. We divide this into three areas: multipoint congram-oriented service, resource management across diverse networks, and gateway architecture.

4.1 Multipoint Congram-oriented Internet Service

There are two important components to this aspect of the research: design and development of a prototype multipoint congram-oriented service, and evaluation of tradeoffs associated with congram sophistication vs. performance of congram management.

Specification and Prototype Implementation of McHIP

We have been working on the design and specification of a multipoint congram-oriented high performance internet protocol. The current version of the protocol includes the following features [20]:

- The protocol allows an application to request the grade of service it needs by specifying bandwidth, delay, and reliability attributes, and also the routing constraints that the application may have.
- The protocol includes a simple multipoint congram-oriented service which is easy to implement and prove correct. This version does not use PICons and does not permit additions and deletions of endpoints once the congram has been established.
- The protocol works with resource servers to perform resource management on a per congram basis.

The specifications includes details of the service primitives that higher level protocols can request, descriptions of various packet types and formats, details of how the protocol provides various services, and also a number of representative scenarios of its operation. Once the specifications are complete, we will implement a prototype of McHIP. The plan is to implement all of McHIP, including packet forwarding, in software, and to thoroughly test the protocol and its implementation. Subsequently, we plan to implement all the per packet processing in hardware as part of the prototype gateway implementation.

A very important aspect of the prototype effort is to deploy and test these implementations in a small scale experimental internet, consisting of a BPN (four 16 port switches) and a few segments of Ethernet and FDDI. This experimental internet is a part of a joint project between the Southwestern Bell Telephone (SWBT) and Washington University (WU), and is planned to demonstrate the feasibility of BPN (or high speed packet switching) technology for visual/image communications.

Congram Sophistication

The introduction of a congram abstraction in a protocol raises a number of issues concerning the state information and resource binding associated with the congram. The two most important reasons for using a congram in high speed networks are assistance in resource management and simplification of per packet processing. It is important to note that the congram abstraction is also beneficial to a number of applications, because it relieves them of considerable complexity associated with the communication. In fact, applications could take advantage of even more sophisticated congram abstractions, but this leads to increased complexity in congram management with corresponding performance degradation. Thus, there exist a number of interesting tradeoffs that deal with the sophistication of congram abstraction vs. the performance and complexity of congram management. The goal is to determine how much and what kind of sophistication can be permitted without compromising the performance and simplicity of congram management.

A relatively fundamental issue is what the performance metric should be for congram management. In high speed networking efforts, the emphasis has been on the performance characterization of the data path, and not of congram management. We argue that as we define more sophisticated congram abstractions, we must also concern ourselves with the performance of congram management to prevent it from becoming the performance bottleneck. We propose the tuple $(\nu, \tau, \kappa, \phi)$ as the performance metric for congram management, where the elements of the tuple have the following meaning:

- ν is the number of congram management operations per second, including congram set up, additions and deletions of endpoints, and change of attributes of a congram. This parameter is useful for gateway and host implementations. Clearly, ν depends on the mix of congram operations, and is specified in those terms. ν should be maximized.
- τ is a vector of times it takes to execute each type of congram operation. For example, in the case of adding an endpoint to a congram, $\tau_{\mathrm{add\_endpoint}}$ is the time between requesting the add endpoint operation and the time that the endpoint is on the congram. Its value depends, among other things, on other operations in progress at the same time. τ should be minimized.
- κ is a vector of the number of messages to be processed and exchanged with other gateways or hosts to execute each type of a congram operation. κ should be minimized.
- ϕ is the amount of information to be stored as part of the state of a congram. ϕ should be minimized.

Estimates of these parameters will also depend on the size and dynamics of the congram in general. Note that these parameters can also characterize the cost associated with the congram reconfigurations. For a given congram abstraction, we can get estimates of these parameters by simulations, analytical modeling, or in some cases by simpler mean value analysis.

In order to provide low overhead congram set up and reconfiguration, we have proposed the idea of perpetual congrams, which are used to carry data for congrams in the transient state. This idea of multiplexing data from user congrams in the transient state on to perpetual congrams has a lot of promise, because with this functionality, we can have an abstraction which provides variable grade service and performance guarantees, and which also allows efficient reconfigurability. Thus, this kind of congram abstraction can have advantages of both connection-oriented and connectionless approaches. However, we need to address a number of other interesting questions: what is the proper topology, bandwidth, and number of perpetual congrams? Under what circumstances does an application use only the perpetual congram(s) rather than creating its own congram? How is congestion avoided on perpetual congrams, and should they change their path/topology and characteristics in order to dynamically adapt to the demands of congrams in the transient state? The effectiveness of perpetual congrams will be high if the congrams in the transient state require a small fraction of resources allocated on perpetual congrams, and have good statistical averaging properties. We are starting to address these questions and evaluate the appropriateness of perpetual congrams in the McHIP environment.

4.2 Resource Management

We have argued that resource management on a per congram basis is the key to making performance guarantees to applications. The effectiveness of such resource management depends on factors such as the average size, duration, and burstiness of the congram, and on multiplexing with other congrams. There is an obvious need to quantify the effectiveness of this strategy in terms of these parameters. As mentioned earlier, a number of networks do not do any internal resource management, and in such cases we have proposed that the gateways provide this functionality. Gateways can keep track of all active congrams and their resource usage and also monitor resource availability in the network. We propose to develop simulation models which will evaluate the resource management strategy and associated tradeoffs for two cases: broadcast LANs and connectionless WANs.

Resource Management on Broadcast LANs

In the case of broadcast LANs, the strategy of gateways acting as resource servers can be conveniently implemented, because a gateway can monitor network traffic while keeping a record of active congrams and their resource needs. We are developing simulations to systematically study the effectiveness of this approach on LANs, such as Ethernet and FDDI. The simulation consists of a number of controlled sources and a resource server. The sources are programmed to generate both congram-oriented and connectionless traffic of specified input parameters, such as the average bandwidth, peak bandwidth, and burst factor. Before establishing a congram, the source first consults the resource server to determine if enough resources are available to support the given congram. The simulation allows the user to specify the fraction of network resource to be allocated for the congram-oriented traffic. The remainder is used by the connectionless traffic, and the resource has the capability to choke connectionless sources if they start using more than their share of resources. In the case of token networks, priorities can additionally be assigned to different sources to give them appropriate shares of network resources. Thus, the user of the simulation decides as input parameters the number of sources, characteristics of sources, resource management policies, and other network parameters.

As output parameters, the simulation will measure channel utilization, blocking of congram-oriented traffic, and end-to-end delay for packets of different sources. Clearly, the most desired operation point would be to provide a tight bound on packet delay, and still achieve high channel utilization and low blocking. Note, however, that these objectives are conflicting in the sense that a tighter bound on packet delay suggests lower offered load which may result in lower channel utilization. The purpose of this simulation study is to show that the proposed resource management scheme can provide performance guarantees in terms of bandwidth and bounded delay to various applications without compromising channel utilization and blocking.

Resource Management on Connectionless WANs

The basic objective of resource management on a per congram basis is still the same, that is, to achieve high performance (high throughput and low delay) with high predictability (tight bound on delay and low packet loss rate) without compromising network utilization and blocking. However, in the case of a WAN, which does not do resource management on a per congram basis, using gateways as resource servers is more difficult. The fundamental constraints are that gateways cannot easily monitor traffic throughout the entire network, and because the gateways have to work with outdated resource availability information due to the latency in obtaining information from packet switches. We plan to design two mechanisms (protocols) to help gateways do resource management in such a WAN environment: First, allow packet switches to periodically send resource availability information on its output links to one of the gateways; second, allow gateways to compile this information in a resource database and exchange this information with each other to keep the resource database consistent. The first mechanism is similar to a routing information exchange protocol, whereas the second is a form of maintenance of a distributed database. The real challenge is how to engineer these mechanisms so that gateways have sufficiently accurate resource availability information without making the resource update mechanisms too responsive and the update overhead too high to be acceptable.

4.3 Gateway Architecture

The important design goals of a gateway architecture in the VHSI include the ability to interface with diverse networks that support data rates up to a few hundred Mbps with high link utilization, to switch input packets with latency less than a few milliseconds, and to interface with a variable number of input ports (2–64). We are working on developing a prototype gateway based on the proposed design and the evaluation of associated tradeoffs as described in the following paragraphs:

Prototype Gateway Effort

The most important aspect of the gateway design, given a switch fabric, is the design of additional building blocks, such as packet sequencing, fragmentation, and reassembly logic. We presented a high level functional design of the building blocks in Section 3.4, and we have undertaken the design of custom VLSI chips to implement them in hardware. As mentioned earlier, fragmentation and reassembly can be implemented as a SSP (synchronous streams processor) using the SSP compiler. Furthermore, we plan to design a packet processor (Figure 9) for the gateway, which will do error detection, address translation, and encapsulation of packets for internal use. Clearly, the exact design of the packet processor depends on the characteristics of the networks to be connected.

In a generic datagram environment, no assumptions can be made about the underlying networks, path of packet traversal, and packet delay distribution, and therefore, the hardware implementation of these functions is complex and expensive. However, we claim that in the VHSI environment, we can considerably simplify fragmentation and reassembly to make their hardware implementation practical. Thus, the purpose of these design exercises is to demonstrate feasibility of these functions in hardware as well as to estimate their complexity.

To demonstrate the feasibility of this architecture as a whole, we plan to build a couple of small, two port gateways, based on the proposed design, and deploy them in the experimental SWBT-WU internet (refer to Section 4.1). The prototype gateway will include network interfaces for BPN, FDDI, and Ethernet, and the McHIP implementation in software on a gateway control processor (GCP) for congram, resource, and route management.

Tradeoffs in Gateway Architecture

We are interested in evaluating the following tradeoffs associated with the gateway architecture:

Traffic Smoothing. One of the functions that a gateway can provide in the VHSI environment is that of restoration and/or imposition of rate specification on the packets of a congram. The traffic pattern (or packet flow) of a congram may deviate from the rate specification stipulated by the endpoints as the packets travel across packet switches and networks. The reasons for such deviations include inter-congram interference resulting from multiplexing, and a series of fragmentation and reassembly operations. The change in the traffic pattern of a congram means a change in the resource needs of the congram, which, if not accounted for, can lead to undesirable overload conditions. There are two ways to deal with changes in the traffic pattern of a congram. First, while establishing congrams, gateways (or McHIPs) can try to predict these changes and to allocate resources accordingly. Second, gateways can try to restore and impose the original rate specification on the packets of the congram as they are forwarded. We are working on evaluating appropriateness and effectiveness of both these approaches.

Shared Pipelines. For the sake of simplicity, we have proposed to use multiple copies of the packet sequencing, fragmentation, reassembly, and traffic smoothing pipeline. Each pipeline is associated with an output port and is dedicated to a congram for its duration. Thus, during the life of a congram, the hardware associated with the pipeline cannot be used by other congrams. The number of pipelines per output port is equal to the average number of active congrams that require services of the pipeline. If this number if large, the amount of hardware required to implement the pipelines could be prohibitively large. We want to consider two possible extensions: first, sharing a pipeline for multiple congrams, and second sharing a pool of pipelines among all output ports.

We plan to undertake the detailed design of such a pipeline and quantify the associated tradeoffs. This includes investigating architectural alternatives to satisfy buffering, speed advantage, and complexity requirements.

Fragmentation and Reassembly Overhead. An important aspect of our gateway architecture is the implementation of packet fragmentation and reassembly in hardware in gateways. We have argued that in the VHSI, the packet lengths are drastically different for different component networks, and therefore, it is essential that gateways reassemble packets in order to avoid sending very short packets on networks that support large packets. However, the performance gains must be compared against the fragmentation/reassembly overhead and added complexity due to their hardware implementation.

5 CONCLUSION

We have presented a very high speed internet (VHSI) abstraction that can help efficiently support guaranteed levels of performance for a variety of applications, and can cope with the ever increasing diversity of underlying networks with rapidly growing user population and needs. The important aspects of this abstraction are the following:

- A novel plesio-reliable multipoint congram-oriented service which we claim has the advantages of both classical connection and connectionless approaches.
- A gateway architecture that can support data rates of a few hundred Mbps, can interface with diverse networks, and can implement the congram-oriented service without becoming a performance bottleneck.
- A resource management strategy across diverse networks to provide predictable performance to congrams.

We have included design of various mechanisms (internet protocols and gateway building blocks) that would enable us to achieve the required functionality at the internet level. Work is in progress to evaluate important tradeoffs associated with the design of a congram-oriented protocol, resource management on diverse networks, and the design of new gateway architectures.

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] Callon, R., "A Proposal for a Connection-Oriented Internetwork Protocol," *ACM Computer Communications Review,* July 1983.

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

[4] Coudreuse, J. P. and M. Servel. "Prelude: An Asynchronous Time-Division Switched Network," *International Communications Conference*, 1987.

[5] Digital Equipment Corporation, "Information processing systems – Data communications – Intermediate System to Intermediate System Intra-Domain Routing Protocol," October 1987.

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

[7] De Prycker, M., Bauwens, J., "A Switching Exchange for an Asynchronous Time Division Based Network," *International Communications Conference*, 1987.

[8] Dieudonne, M., Quinquis, M., "Switching Techniques Review for Asynchronous Time Division Multiplexing," *International Switching Symposium*, 3/87.

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

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

[11] FCCSET Committee on Computer Research and Applications, "A Research and Development Strategy for High Performance Computing," Executive Office of the President, Office of Science and Technology Policy, November 1987.

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

[13] Haserodt, Kurt and Jonathan Turner. "An Architecture for Connection Management in a Broadcast Packet Network," Washington University Computer Science Department, WUCS-87-3.

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

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

[16] IETF Working Group on EGP "Exterior Gateway Protocol, Version 3, Revisions and Extensions to EGP," DARPA IETF IDEA 009, SRI Network Information Center.

[17] IETF Performance and Congestion Control Working Group, "Gateway Congestion Control Policies," IETF Draft.

[18] Jacobson, Van, "Congestion Avoidance and Control," *Proceedings of the ACM SIGCOMM'88,* August 1988.

[19] M/A-COM Government Systems, Inc. "Dissimilar Gateway Protocol Specifications."

[20] Mazraani, T., Parulkar, G.M., "Specifications of a Multipoint Congram-oriented High Performance Internet Protocol (MCHIP)," Department of Computer Science, Washington University in St. Louis, in progress.

[21] Mills, D.L., "Exterior Gateway Protocol Formal Specification," DARPA RFC 904, SRI Network Information Center, April 1984.

[22] Kent, C.A., Mogul, J.C., "Fragmentation Considered Harmful," *Proceedings of the ACM SIGCOMM'87,* August 1987.

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

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

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

[26] Parulkar, G.M., Turner, J.S., "Towards a Framework for High Speed Communication in a Heterogeneous Networking Environment," *Proceedings of IEEE INFOCOMM'89*, 1989.

[27] Postel, J. "Internet Control Message Protocol," DARPA RFC 792, SRI Network Information Center, September 1981.

[28] Ramakrishnan, K.K., Jain, R., A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with Connectionless Networks," Proceedings of the ACM SIGCOMM'88, August 1988.

[29] Robbert, G., "A Circuit Generator for Synchronous Streams Processors" MS Thesis, Department of Computer Science, Washington University in St. Louis, May 1988.

[30] Seamonson, L.J., Rosen, E.C., ""STUB" Exterior Gateway Protocol," DARPA RFC 888, SRI Network Information Center, January 1984.

[31] Sterbenz, J.P., Parulkar, G.M. "Axon: Network Virtual Storage Design," Technical Report WUCS-89-13, Department of Computer Science, Washington University in St. Louis, 1989.

[32] Sterbenz, J.P., Parulkar, G.M. "Axon: Application-oriented Lightweight Transport Protocol Design," Technical Report WUCS-89-14, Department of Computer Science, Washington University in St. Louis, 1989.

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

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

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

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

[37] Eng, K.Y., Hluchyj, M.G., Yeh, Y.S., "A Knockout Switch for Variable-Length Packets," *IEEE Journal on Selected Areas on Selected Areas in Communications,* vol. 5, no. 9, December 1987.