

Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation

Raj Jain, Fellow of IEEE
Department of Computer Science and Engineering
Washington University in Saint Louis
Saint Louis, MO 63130
jain@cse.wustl.edu

Abstract—The basic ideas of the Internet architecture were developed 30+ years ago. In these 30 years, we have learnt a lot about networking and packet switching. Is this the way we would design the Internet if we were to start it now? This paper is an attempt to answer this question raised by US National Science Foundation, which has embarked on the design of the next generation Internet called GENI.

In this paper, we point out key problems with the current Internet Architecture and propose directions for the solutions. We propose a general architectural framework for the next generation Internet, which we call Internet 3.0.

The next generation Internet should be secure. It should allow business to set their boundaries and enforce their policies inside their boundaries. It should allow governments to set rules that protect their citizens on the Internet the same way they protect them on other means of transports. It should allow receivers to set policies for how and where they receive their information. They should have freedom to select their names, IDs and addresses with as little centralized control as possible. The architecture should be general enough to allow different governments to have different rules. Information transport architecture should provide at least as much control and freedom as the goods transport networks provide.

We propose the framework of an architecture that supports all these requirements.

I. INTRODUCTION

Internet has changed the way we work and live and has contributed positively to the growth of business and defense. Nonetheless, many part of the Internet architecture were developed 30+ years ago. In these 30 years, we have learnt a lot about networking and packet switching. Is this the way we would design the Internet if we were to start it now? This paper is an attempt to answer this question which has been raised by US National Science Foundation, which has embarked on the design of the next generation Internet called Global Environment for Network Innovation (GENI) [1].

In this paper, we point out key problems with the current Internet architecture and propose directions for the solutions. In particular, the next generation of Internet has to be commerce friendly. It has to be designed to meet the needs of businesses, organizations, and governments. The first generation was designed by researchers for research. The design team did an excellent job resulting in its adoption by the masses. The next generation Internet should build on this success, keep the best ideas of the past and add features that will help businesses, organizations, and governments utilize it in the same way they

utilize other methods of communication and transport and have the same or superior level of flexibility.

We coined the term Internet 3.0 to denote the next generation of Internet. This naming is along the lines of current fascination or networking industry with Web 2.0. National Science foundation is currently planning for this next generation of Internet under its GENI program. With several hundred millions of dollars investment planned in this program, this will be one of the biggest projects undertaken by the NSF. In the coming years, most networking researchers will be working on projects related to this program.

Our proposal is cumulative. Our goal here is to start with the best ideas from all known sources, extend them and put them together in a coherent, interoperable, realizable framework. So while there are many new ideas in this proposal, there are many ideas that have been presented before. In fact, we have borrowed heavily from current internetworking research as well as from other means of transporting information and goods such as telephone networks, airlines, railroads, highways, walkways, and postal services.

The next generation Internet should be secure. It should allow business to set their boundaries and enforce their policies inside their boundaries. It should allow governments to set rules that protect their citizens on the Internet the same way they protect other means of transports. It should allow people to set policies for how and where they receive their information. They should have freedom to select their names, IDs and addresses with as little centralized control as possible. The architecture should be general enough to allow different governments to have different rules. Information transport architecture should provide at least much control and freedom as the goods transport networks provide.

The next generation Internet should be designed for mobile objects. People, computers, laptops, palm tops are mobile. The naming, addressing architecture has to allow so that these objects can move and decide how and where they want to receive their Internet traffic with full rights of privacy of their location if desired.

Our architectural framework is called "Generalized Internetworking Architecture (GINA)". The key feature of GINA is that it is very general. The next generation Internet, like the current Internet, will be used with a variety of applications over a variety of link technologies. Therefore, this proposal

does not limit itself to a particular set of applications or a particular set of link technologies, such as wireless or optical networks. This is an architecture framework and, therefore, it allows numerous flexibilities that may not be present in any one implementation of it. The implementers of this framework are expected to limit the choices to keep the cost of implementing too many alternatives. For example, GINA allows unlimited levels of routing hierarchy. Implementations may constrain themselves to two levels, which like the current Internet may consist of inter-domain and intra-domain routing. Network administrators may further limit the choices offered by a particular implementation.

The purpose of this research proposal is to help develop the overall network architecture for Internet 3.0. We seek to design a next-generation Internet for security, robustness, manageability, utility, social and other needs. The proposal identifies a number of requirements that should be satisfied by the next generation Internet. We then present the outline of an architecture, right here in this proposal, that satisfies most of these requirements.

There are two key parts of this paper. First we explain what is Internet 3.0 and motivate why the industry, governments, and other organizations should be involved in the development of Internet 3.0. We then point out the areas where the current internet can be improved. Finally we present the framework of an architecture to provide these improvements.

II. RELATED PRIOR WORK

The problem of improving networking architecture is not a new one. The bibliography lists a number of papers on various architectural issues. Most of these papers address one or two aspects of networking architecture.

Recently, NSF has conducted several workshops on the research required in various important areas of networking such as wireless [2], optical [3], distributed systems [4], and virtualization [5]. The reports of these workshops are good sources of information for what is missing in the current Internet and what is required in the next generation. Stoica et al [6] presented an architecture for addressing for mobile objects. Most general results so far are in the final report of DARPA project [7] and in papers by Balakrishnan et al [8].

In the past, most of the research was devoted to how to improve the current architecture and there was little thought about how would one do it right if it was possible to develop a new Internet now. NSF's FIND and GENI programs provide the first opportunity to researchers to think freely and the proposal in this paper makes the most of this opportunity.

III. INTERNET GENERATIONS

Internet is now almost 40 years old. The first RFC from the Internet Engineering Task Force is dated April 1969. The actual ARPAnet program started a couple of years earlier. Since its beginning, Internet has gone through two major generations each lasting about 20 years. During the first two decades, Internet was mostly a research project. Industry itself was divided and was busy developing competing networking

technologies: IBMs SNA, Digitals DECnet, Xerox's XNS and AppleTalk to name a few. The standards groups were busy developing the Open System Interconnection (OSI) protocols. This phase lasted till about 1989 and can be called Internet 1.0 or the research Internet.

Beginning with 1989, Internet entered a new phase with the industry starting to adopt it for commerce. A number of issues that were not considered important till then began to surface as a result of this adoption. The first RFC on security is dated 1989. The scalability issues required dividing routing into domains. Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP) were developed as a result. The shortage of IP addresses led to the development of a number of solutions including private addresses, network address translation (NAT), and IPv6. Traffic management, congestion control, and quality of service issues became important. We call this as Internet 2.0 or the commercial Internet.

Now we are entering a new phase, where Internet has become an integral part of our lives, our businesses, our government, and our defense. We have learnt a lot about networking in the past 40 years. This knowledge should be the basis for designing the next generation of Internet: the Internet 3.0.

IV. TOP TEN FEATURES REQUIRED IN THE NEXT GENERATION INTERNET ARCHITECTURE

In this section, we list the top ten features that would help remove some of the problems faced by current Internet users.

A. *Energy Efficient Communication*

Current Internet architecture requires both source and destination end-systems to be up and awake for the communication to take place. All packets received when the destination is down are dropped. With wireless devices, this restriction is being relaxed by allowing base stations to store the packets while the subscriber device is sleeping. For energy efficient communication, this should be generalized to wired devices as well.

B. *Separation of Identity and Address*

In current Internet a system is identified by its IP address. As a result, when a system changes its point of attachment, the address changes. This makes reaching mobile systems difficult. This is a well-known problem and a number of attempts and proposals have been made in the past to solve this problem - including Mobile IP, Internet Indirection Infrastructure [6], Host Identity protocol [9], [10] and others [8].

C. *Location Awareness*

IP addresses are not related to geographical location. This can be considered strength of IP. However, a big share of information transfer applications, like any other transport system, requires finding the nearest server. Also, mobile nodes need to know their location. Next generation Internet should let the receiver decide about their location privacy.

D. Explicit Support for Client-Server Traffic and Distributed Services

A big share of current Internet traffic is client-server traffic. A web user trying to reach Google is an example of client-server traffic. These users are trying to reach "Google," which is not a single system. It is a distributed service with hundreds of systems in hundreds of location. The user is interested in the communicating with the nearest instance of this service. In current Internet, the name Google is resolved to a single IP address and so directing users to the right server is unnecessarily complex.

E. Person-to-Person Communication

The internet was designed for computer communication. But the real target of communication is often a human being. A person may be reachable by a desktop computer, a laptop, a cell phone or a wired phone. The goal is to reach the person and not the desktop computer, the laptop, or the phones. Since the person does not have an IP address, we the users are forced to select one of these intermediate stops as the destination for our communication instead of the real destination the person. If each person had an address, the network could decide the right intermediate device or the person could dynamically change the device as appropriate.

F. Security

Security issues of current Internet are well known. It is necessary that the next generation allow the option of authentication of sources/destinations/intermediate systems, privacy of location, privacy of data, and data integrity guarantees.

G. Control, Management, and Data Plane separation

In the current Internet, control, management, and data planes are intermixed. Control messages (e.g., TCP connection setup messages) or management messages (SNMP messages) follow the same links as the data messages. Control signals are also piggybacked on the data packets. This introduces significant security risk as evidenced by all the security attacks on the Internet. The telephone network, on the other hand, uses a separate control network, and is generally considered more secure than Internet. Generalized Multiprotocol Label Switching (GMPLS) is one attempt to separate control and data planes. One advantage of this separation is that it allows data plane to be non-packet oriented such as wavelengths, SONET frames, or even power transmission lines. This separation should be integral part of the next generation architecture.

H. Isolation

For many critical applications, users demand "isolation in a shared environment." Isolation means that the performance of one application is not affected by other applications sharing the same resources. One alternative is to provide dedicated resources to such applications. This is the reason for popularity of virtual private lines (T1/E1 lines) from the telecommunications companies to form private networks. The next generation networks should provide a programmable mix of isolation and sharing.

I. Symmetric/Asymmetric Protocols

Most current Internet protocols are symmetric since they were designed for end-systems with similar capabilities. In sensor networks and also when communicating with palm devices, one end-system may be significantly resource constrained compared to the other end. So in some instances it is justifiable to allow asymmetric protocols.

J. Quality of Service

Quality of service, by its name, belongs to a service, which in turn relates to the groups of packets used in that service. Users are normally interested in receiving some guarantees about the delay and throughput of their flows. The stateless nature of IP makes it difficult to guarantee QoS. Next generation Internet should allow a variety of QoS guarantees including total isolation, if desired. Also QoS has to be related to economics. QoS techniques with no relationship to charging policies have not been successful in the past.

V. ADDITIONAL FEATURES

In addition to the above ten features, there are several other desirable features. We list them here.

A. Global Routing with Local Control of Naming and Addressing

Originally, IP required each system to have a globally unique address. This led to the problem of IP addresses shortage, which has been solved partly by private addressing and IPv6 addressing. Each of these solutions has their own issues. For example, nodes with private addresses are not easily reachable from outside. Next generation Internet should allow organizations the flexibility of deciding which of their local objects are accessible from outside and which are not.

B. Real Time Services

Today many of the emergency and important protection services run on Internet. These services need real-time guarantee. Often, separate dedicated/private networks are used to guarantee the required performance. The next-generation Internet should make this possible on the shared internet.

C. Cross-Layer Communication

In the current Internet, medium-specific details are hidden from transport and applications. There are no inherent architectural interfaces for applications to find that they are going over a particular medium and, therefore, can take advantage of its specific properties or change their characteristics based on it. For example, applications do not know and cannot easily adopt for Ethernet (free multicast), wireless (low speed, high loss rate), or satellites (long-delay).

D. *Manycast*

Many of the real-time systems follow a publisher-subscriber model, in which the data monitoring devices act as publishers and are subscribed by controllers that gather and analyze the data to make control decisions. For reliability reasons, multiple redundant monitors and controllers are used. This requires an n-by-m communication, where data can come to each of m subscribers from any one of n redundant publishers. This we call "Manycast." Anycasts and multicasts are special cases of manycast.

E. *Receiver Control*

Receivers have little control over the rates, priorities, and other attributes of packets coming through the line that they pay for. A communication involves three entities - sources, networks, and receivers. Of these, sources have most control in terms of setting the rate and priority of packets. The network owners then have the next level of control in the form of packet classification and rate throttling. Receivers need a way to indicate their preferences and policies for traffic coming through their link, which is currently missing in the current Internet.

F. *Support for Data Aggregation and Transformation*

The next generation network should provide facilities to aggregate, consolidate, and transform data. This is often necessary to accommodate a variety of end systems. In many sensor network applications, it is necessary for the intermediate systems to summarize the data. Video transcoding and compression are required to support a variety of video presentation standards (NTSC, PAL,...) on a variety of screen sizes (theatre screens, cell phones, palm devices,...).

G. *Support for Streaming Data*

Many of the real world applications are stream-oriented requiring a fixed or minimum throughput guarantee. A simple dedicated wire provides this guarantee. The next generation Internet should provide support for such applications.

H. *Virtualization*

One of the key requirements set for GENI is virtualization. The next generation architecture should allow multiple virtual meta-networks on the top of a base substrate. These virtual networks require isolation and link attributes that are not affected much by other meta-networks on the same substrate.

VI. THE GINA FRAMEWORK: KEY FEATURES

The GINA framework has been designed to address the issues identified above. The details of the framework are described in detail in the next few sections. In this section, we list the key features and their benefits:

A. *Mobility*

Each GINA object has separate ID and address. The addresses are dynamic and depend upon the current location of the object. While ID is more stable and do not change as the object moves.

B. *Role or Service based Communication*

GINA allows objects that are distributed and have multiple addresses. For example, Google is a service that may have servers all over the world. GINA hosts can reach the nearest server by design. Similarly, it is possible to address an object by its role, e.g., a manager. This helps in client-server traffic, which is becoming a large part of the Internet traffic today.

C. *Hybrid (Packet and Stream based) Communication*

GINA allows both packet-based and circuit-based traffic. This helps enforcing strict real time constraints and in virtualization.

D. *Enforcement of Organizational Policies*

GINA has clear organizational boundaries as part of the architecture. Each organization and sub-organization can enforce policies on packets leaving or entering the organization. This is possible by ID hierarchy and realms.

E. *Enforcement of Service Provider Policies*

GINA distinguishes network connectivity from organizational ownership. Network service providers can enforce their own policies as the packets leave from their network into other service provider or customer networks. This is possible by an address hierarchy and zones.

F. *Energy Conservation*

GINA allows functions such as security, storage, reception and transmission to be delegated to servers. This allows objects to be accessible even when they are sleeping or away resulting in battery savings.

G. *Non-Packet Based Data-Planes*

GINA has clear separation of control, data, and management planes. The data plane can be non-packet based, such as SONET streams, wavelength, or electric power lines. The control and management planes in these cases are packet based.

These are just some of the key features of GINA. Actually our goal is to satisfy all the requirements identified earlier in this paper. The rest of this paper is organized as follows. We first define objects in GINA and then explain how objects acquire Names, addresses, and IDs. We then introduce the concept of realms and explains how the GINA objects follow organizational boundaries.

VII. GINA ARCHITECTURE OUTLINE

A. *GINA Objects*

Each addressable unit in GINA is called "Object." Examples of objects are computers, routers, firewalls, and proxy servers. What we call end-systems, middle-boxes, or intermediate systems in current Internet will all be objects in GINA. However, the concept of objects is more general than these systems in two aspects. First, objects include non-computing entities such as humans, companies, departments, cities, and countries. Anything that can be addressed is an object. Thus, in GINA it

is possible to send packets to a person, say, John. John may not have an electronic connection to GINA Internet but will have a voice connection to his cell phone, a visual connection to his laptop monitor or palmtop monitor. When someone wants to contact John, they are not interested in contacting the laptop or the palmtop, or the cell phone. In current Internet, the sender has to make the choice of the three connections that John has. In GINA Internet, the sender, the network, and the receiver can jointly decide the best path to John. For example, the sender can simply send the packets to John and the network's responsibility then is to find the best path from the sender to John. John may instruct that the packets be delivered to his palmtop. These instructions from John will, of course, be very dynamic and will change by the time of the day.

GINA also allows the possibility of John carrying a certificate in the form of a "SIM" card (as in GSM phones) that when inserted in to any computer will allow that computer to as John's computer. The point is that in all these examples, the destination of Internet traffic is John and not the computer. Therefore, John is a valid GINA object and needs a GINA Name, ID, and address.

The second way GINA object concept is different from current Internet is that it is recursive. A group of objects can also be treated as an object. So a network is one object, a network of networks is an object. A department (with multiple objects inside) is also an object. A company (with multiple departments) is also an object.

Note that the connection between GINA object and the GINA Internet does not have to be electronic. Audio, visual connections are allowed.

B. Attributes of GINA Objects

Each object in GINA has a set of names, IDs, addresses, security keys, certificates, and other attributes that are registered with the "local" registry. The names and IDs are similar in the sense that names are ASCII strings for human use while IDs are corresponding binary strings that are used by computers and are part of the packet headers. The addresses relate to the physical connectivity and are very dynamic. When the object moves, the address changes and so we do not require correspondents to know addresses. The correspondents always send packets to names, which are then translated to corresponding IDs. It is network's job to translate IDs to addresses. The exact method of ID to address translation is one of the research problems that we will handle during this project. There are already several known ways to do this. For example, Indirect Internet Infrastructure (I3) [6] provides one way of assigning IDs and relating them to addresses by careful global allocation of IDs. Balakrishnan et al [8] suggested using distributed hash tables. The host-identity protocol (HIP) working group selected public key as ID and uses DNS to bind it to an address [9], [10]. It is clear that more work needs to be done in this area.

An object may have multiple names. Each name may translate to a set of IDs. Each ID may translate to a set of addresses as discussed later in this proposal.

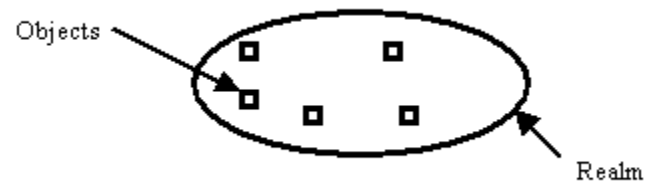


Fig. 1. GINA Objects

C. Object Names

Each object in GINA can have multiple names and these names are valid in a local context, which we call "realm" (see Fig. 1). For example, a person's home is one realm. The home may have multiple people, computers, and other GINA objects. The realm manager has complete control over assignment of names. The same names can be used in other realms by their managers. Even in one realm, two objects may have the same name. For example, if two objects have name "printer" this will resolved to two IDs and either the sender, the network, or receiver policy will help decide whether the packet is sent to any or all of the IDs. The packet will be delivered via anycast or multicast accordingly.

The printer example brings out another attribute of GINA names. Printing is a service and each service has a name and since there can be multiple objects that provide that service, the names need not be unique. It is the job of the realm manager to properly assign names so that the names have some sensible meaning for use by other humans. Also, the names in some large realms may have to follow the copyright, trademark, and other restrictions. For example, while one can name a computer in one's home as IBM. However, it would not be a meaningful name for a business in a city unless it has some relationship to IBM.

The local registry helps resolve the names to IDs. The IDs are returned with other attributes (such as location, if it were known) that can be used by the requester to narrow down the possible set of IDs.

D. Object IDs

GINA objects IDs are arbitrary binary strings that are arbitrarily assigned by the realm manager. For example, five computers in a single household may have IDs of 001, 010, 111, 100, 110, respectively. Since a group of object is also an object, a group of objects with a common attribute may have a name and an ID. For example, the group "printers" may have an ID of 111. While each printer may have an individual name, ID, and location attributes.

Since GINA separates the concept of addresses into IDs and addresses, we have to also decide which attributes of current Internet addresses belong to GINA IDs, which to GINA addresses, and which to both. In general any attribute that does not change as the object moves, belongs to GINA IDs.

In current Internet, we have unicast and multicast addresses. Correspondingly GINA has unicast and multicast IDs. The ad-

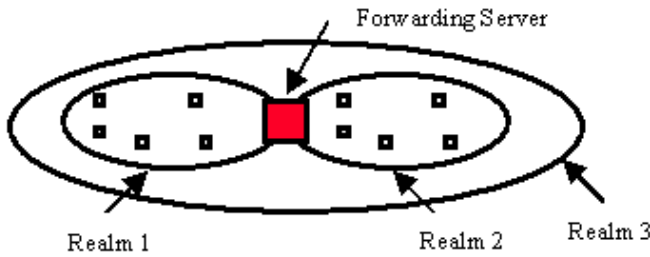


Fig. 2. Forwarding Servers

addresses are related to the points of attachment and connectivity. Several objects that share a point of attachment and so may have a "multicast" address. The multicast IDs and multicast addresses have different purposes and different meanings.

E. GINA Realms

It has already been pointed out that GINA object names and IDs that are valid within a realm. Each realm has a manager that controls the assignment and resolution of names, IDs, and addresses. Since Realm is a single administrative domain, the objects within a realm can easily communicate with each other. Objects in one realm wishing to communicate with objects in another realm send the packets through forwarding servers, which connect two or more realms as shown in Fig. 2. When a packet crosses a realm boundary, it is handled specially according to the policies set by the managers of the two realms at the transit point.

Like the concept of object, the concept of realms is also recursive. For example, a group of realms can also form a realm. The group need not be physically contiguous. For example, Department of Computer Science is one realm; Washington University is a realm, which is a group of several department realms. All the universities in Midwest could form a "Midwest Universities" realm and so on.

Membership in a realm is controlled by the realm manager and provides certain rights and privileges to the members, while requiring certain responsibilities and rules of trust from them.

Notice that the realm is an organizational concept and is very different from "Administrative domain" in current Internet, which are related to connectivity.

F. Realm Hierarchy

GINA universe is organized as a hierarchy of realms. Each realm in this hierarchy has a number of parents and a number of children as shown in Fig. 2. Note that the hierarchy is not a binary tree since a realm can have two or more parents, i.e., an organization can be part of several higher-level organizations and can have several lower level sub organizations.

Each realm is a GINA object and has names and IDs. Any path from the root of the universe to an object in the ID hierarchy gives the universally unique ID of the object. The ID is represented in the root-to-leaf order. Names of the object can similarly be concatenated to form a universally unique

name. For example, the object 1 in the bottom left corner has a name of R.L2.L1.1. Here, R, L2, and L1 are names of the root and lower level realms as shown in the figure.

When two objects communicate, it is not necessary to know the universally unique name or ID of the other object. It is sufficient to know the names up to the level at which they have a common parent. So for example, when object 1 and 2 communicate, they just use their given names, L1.1 and L1.2, since they are in the same realm L1. However, when object 1 communicates with 4, the names of 1 and 4 are L2.L1.1 and L2.L3.4, respectively. The common ancestor is L2.

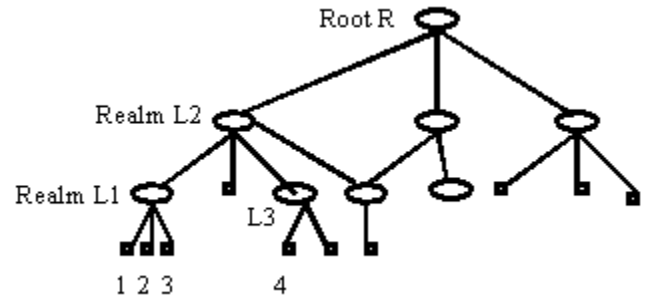


Fig. 3. GINA Realm Hierarchy

G. Object Addresses

Unlike the names and IDs, which are somewhat arbitrarily assigned, the address of an object relates to its connectivity. An object that provides hundreds of services may have hundreds of IDs but if has only one attachment, it will have only one address.

H. Address Hierarchy and Zones

In terms of addresses, the universe is organized as a hierarchy, which we call "zones" (see Fig. 4). While realm hierarchy indicates organizational membership of objects. The zone hierarchy indicates connectivity of resources. For example, a Sprint Cell phone subscriber working for Washington University is a part of the Washington university realm but its address belongs to Sprint Zone. Note that there are many similarities between zones and realms. Both are objects that have their own IDs and addresses. Both have managers that set policies for packets entering/leaving or moving in their part of the network.

An object's universal address or address at any level is obtained by prefixing its address with those of successive ancestors.

An object can reside in multiple zones at the same time. For example, a person may have a home address and an office address. These represent two connections that the person has.

I. Mobility and Addresses

When an object moves from one zone to another, it gets a new set of addresses. It can keep or renounce the old address. Keeping the old address allows for a smooth handover.

J. Server Objects

Each realm has a set of server objects that can perform services for the objects in the realm. Examples of server objects are forwarding servers, route servers, authentication servers, encryption servers, proxy servers, etc. Forwarding servers forward the packets; Route servers provide routes to distant objects; Authentication servers authenticate the source realm of the arriving packets and add their signatures to packets leaving their realm; Proxy servers act as source or destination for objects that may be sleeping or are away.

Objects in the realm as well as the realm manager rely on these servers. The objects can either perform these services themselves or delegate to one or more of such servers.

Each object registers its delegations with the local registry.

K. Routing in GINA

Routing is based on connectivity and consists of finding a path through the zone hierarchy. Based on connectivity, zones are organized as a multi-level hierarchy as shown in Fig. 5. Each ellipse represents a zone at a particular level. Objects that are in two different levels act as transit points for the traffic leaving that zone. The packets are forwarded towards the destination address one level at a time.

GINA routing is analogous to the routing we use when going from one place to the next. For example, to go from my home in Saint Louis, MO to Frankfurt, Germany, I need to cross a walking zone and reach my car. Then I drive to the airport using an auto-zone. At the airport I switch to the airplane zone and take multiple flights that optimize the path through the airplane zone. Once in Germany, I follow the downward journey through the auto zone and the walking zone.

The key point is that while the path in each zone may be optimal, the end-to-end path is not necessarily optimal. But this is the price we pay for the scalability and simplicity. The routing databases in each zone are small enough and somewhat related to the number of objects in the zone. Routing table

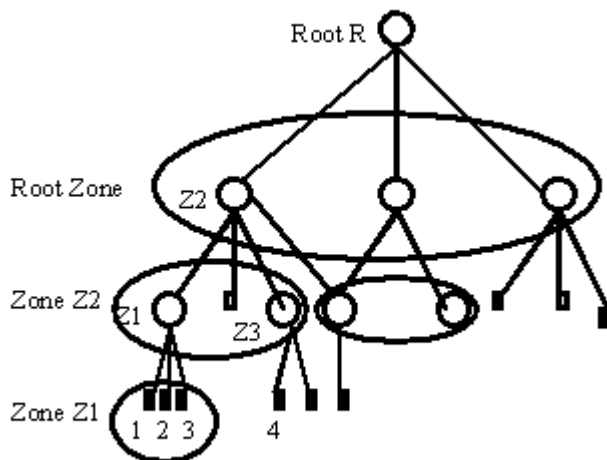


Fig. 4. GINA Address Hierarchy

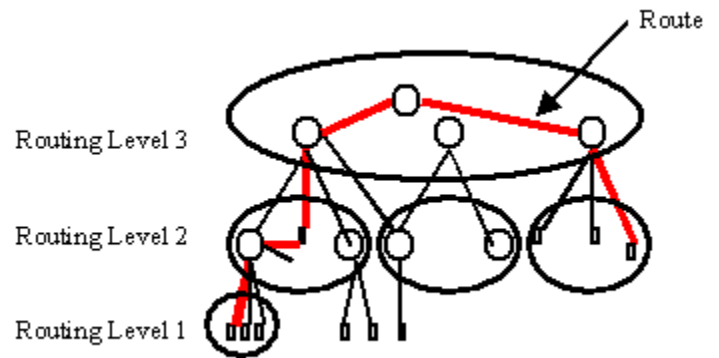


Fig. 5. Routing in GINA

exchanges are limited to those between forwarding servers in the zone. Only summaries of routes are exchanged with higher and lower layers. At each level, packets are sent to the "optimal" forwarding server or to "default" forwarding server. Exits from the zone are to higher levels or lower levels. Entry forwarding server puts the route on that zone in the packet.

L. GINA Packets

In order to communicate with an object, the source object has to know the name of the destination object. The name has to be up to the common ancestor. The names can be translated to IDs using registries at the appropriate levels. The packets contain IDs of the source and destination. The IDs are replaced by addresses by a combination of "knowledge" and "necessity." This late binding is helpful for mobile objects. The top level ID is translated to address and is replaced by a loose source route in the packet.

M. Channels

When the Internet was invented, most communication was via circuits. One of the key contributions of the Internet was to introduce the datagram concept where each packet is handled individually. The datagram and circuit camps have since debated the merits and demerits of the two approaches. It turns out that it is not necessary to support just one. It is possible to support both. Many of the recent wireless standards support both circuits and datagram traffic. GINA borrows these concepts from those standards and applies it to wired networks as well.

A channel is a sequence of packets or bits that require certain guarantees. There are three kinds of channels: streams, flows, or multigrams (see Fig. 6). These three differ mainly in their duration and variability of guarantees. Streams consist of a constant bit rate circuit switched traffic (e.g., T1/E1) that requires strict delay guarantees. Multigrams consist of bursts of packets that have some common attribute, typically, the same exit from the current zone. Flows are longer-term sequence of packets than multigrams and may require implicit or explicit setup.

GINA streams consist of constant bit rate services and can be interspersed with packets on the same physical media. One

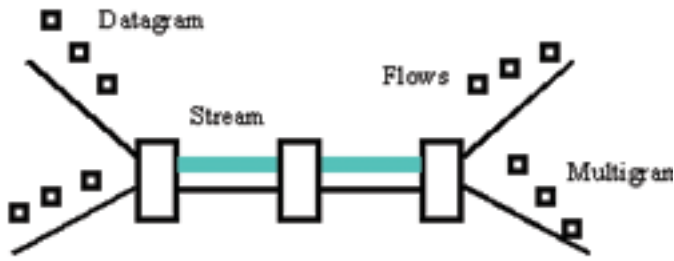


Fig. 6. GINA Channels (Streams, Multigrams, and flows)

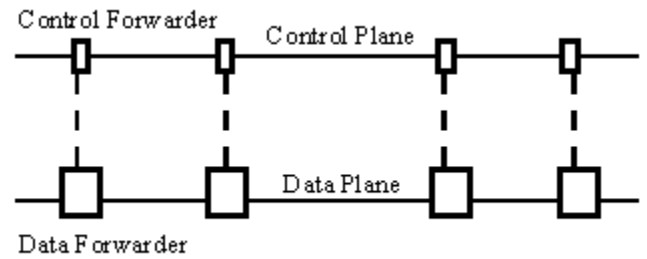


Fig. 7. Control and Data Plane Separation in GINA

way to offer these services is to have a cyclic framing structure in which some part of the cycle is reserved for streams while the remaining is used for datagrams. IEEE 802.16 (WiMAX) and IEEE 802.17 (RPR) both offer such combinations.

Setting aside the age-old religious debate about connectionless versus connection-oriented services, GINA provides both. Streams are important and natural for many applications. A simple wire, for instance, offers a stream service with a fixed bit rate and a fixed delay. When this wire is replaced by a shared wire, someone may still want to have the same fixed rate and delay guarantee. Stream is one way to offer such "Virtual wires." It is for this reason T1/E1 services are still very popular in the telecommunication market. Most VPNs are still made using private T1/E1 lines. By providing both stream and datagram services, GINA architecture does not forbid private lines but accommodates them.

Another GINA concept is that of multigram, which consists of multiple datagrams with some common attribute such as the same exit server in the current zone. In this case, the forwarding decisions made for the first packet are cached and reused for all packets of the multigram. Multigrams can also be used to represent flows that have guarantees in between those of datagram services and stream services.

N. Control and Data Plane Separation

The intermixing of control and data planes causes many security problems of the current Internet. Telephone networks use separate networks for control messages that are used to setup circuits and the circuits themselves. This is one reason why telephone networks are perceived to be more secure than Internet.

Control and Data planes are kept separate in GINA. Control messages are used to set up streams and multigrams flows. These message travel in the control plane, which is isolated from the data plane.

Rather than having a physically separate control network, GINA allows the possibility of a "virtually separate" control network in the sense that the control messages flow on a virtual wire if necessary. Of course, if more security is required a physically separate network can be used for control.

This separation of control and data is similar to the concept of GMPLS in current Internet. This allows data plane to be anything including SONET streams, wavelengths, or power lines.

O. Cross-Layer Design

In the current Internet, the feedback from lower layers to upper layers is mostly implicit. For example, when IP router drops a packet, it may at most send an ICMP message to the source IP layer but the source IP layer does not pass on this information to TCP layer. The only way TCP layer comes to know about the packet loss is by timeout. Similarly, applications have difficulty finding out different attributes of a path, e.g., available bit rate, maximum capacity, reliability, loss rate, etc.

GINA architecture will make use of cross-layer design so that upper layers can query lower layers and make use of the information that might be available locally or can be obtained by lower layers. Upper layers may also specify desired attributes of paths for their flows. Again such specifications of paths may be justified more with the use of multigrams, flows, or streams than with individual datagrams.

P. Security in GINA

Security in GINA is handled at the realm and zone level. Whenever a packet enters a realm, the policies specified by the realm manager are enforced. Such policies may require for example, the packet source to be authenticated, authorization to be checked, packet content to be analyzed for virus, or restricted to a particular set of applications. The realm contains servers that enforce these policies. The packet has to go through these servers before it is accepted for forwarding further inside the realm. Once inside the realm, the packet moves somewhat freely without need for re-authentication at every hop. This assumes that all members of the realm have certain trust and responsibilities. As an example, consider a case where the network is organized as a set of country realm, each country consisting of city realms, each city consisting of house realms. When packets enter a country, the security policies of the country are enforced. These policies may vary from country to country. Once the packet enters the country, it enters a city realm and undergoes policies set by the city realm manager and so on. Although this example is for geographical realms, it should be easy to see that the same applies to packets flowing between companies and between departments of a company.

The realm manager may also have exit policies that are enforced on packets leaving the realm. It should be pointed

out that zone managers that manage connectivity also have policies that are enforced when packets enter/leave their zone.

Security is just one example of a policy. Other policies may relate to the setting of priorities, rates, and types of packets.

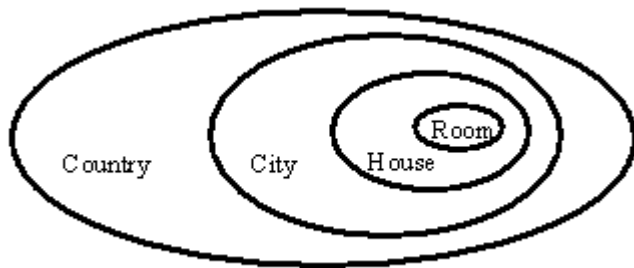


Fig. 8. Each Zone or Realm has its own Policies that are enforced at entry/exit

Q. Receiver Control

Receivers in GINA have complete control over which traffic enters their network and which packets have higher priority. This is done by setting the realm policy. This is straightforward from the policy enforcement discussion above.

For example, a person receiving video over a low-speed connection from a network provider may want to set a rate control on other traffic entering his/her realm.

R. Isolation

A strong point of GINA architecture is that it allows both channels (in the form of streams, flows, and multigrams) and datagrams. Those applications that require isolation can use streams. Streams make the resource management, allocation, and specification easier but may be wasteful if the resources are not used. Datagrams make full use of the resources but do not provide isolation between users. By providing both services and intermediate possibilities of multigrams and flows, GINA provides the best of both worlds.

Note that it is possible for datagrams to join a stream for a part of the path as shown in Fig. 6.

VIII. SUMMARY

Internet 3.0 is the next generation of internet that will result from the GENI research program being started by National Science Foundation. This paper presents several ideas about problems in the current Internet that should be fixed in the next generation. In particular, it should be energy efficient, secure, and allow mobility. It should be designed for commerce and allow governments to protect their citizens the same way they can with the other modes of communication and transportation. Active involvement of all parts of government and defense in this effort is essential. In this paper we have presented the outline of a proposed architecture that will help resolve many of the problems highlighted in the paper.

IX. ACKNOWLEDGEMENT

The author would like to thank all senior members of the Applied Research Laboratory (ARL) at Washington University in Saint Louis, who participated in several brain storming sessions and provided valuable feedback related to GINA architecture.

REFERENCES

- [1] National Science Foundation, "Global Environment for Networking Innovation," <http://www.nsf.gov/cise/geni/>
- [2] D. Raychaudhuri and M. Gerla, Editors. "Report of NSF Wireless Mobile Planning Group (WMPG) Workshop," September 2005, 48 pp, http://www.geni.net/wmpg-draft_200508.pdf
- [3] D. Blumenthal, J. Bowers, and C. Partridge, Editors, "NSF Workshop Report on Mapping a Future for Optical Networking and Communications," July 2005, <http://www.geni.net/nsf-opt-200507.pdf>
- [4] M. Frans Kaashoek, et al, "Report of the NSF Workshop on Research Challenges in Distributed Computer Systems," December 4, 2005, 13 pp., <http://www.geni.net/distributed.pdf>
- [5] T. Anderson, L. Peterson, S. Shenker and J. Turner, "Overcoming the Internet Impasse through Virtualization," Computer Magazine, April, 2005.
- [6] I. Stoica, D. Adkins, S. Zhuang, et al, "Internet Indirection Infrastructure," ACM SIGCOMM, Pittsburgh, PA, 2002, <http://i3.cs.berkeley.edu/publications/papers/i3-sigcomm.pdf>
- [7] D. Clark, et al, "New Arch: Future Generation Internet Architecture," Technical Report, Air Force Research Laboratory, Rome, NY, December 31, 2003, 76 pp., <http://www.isi.edu/newarch/iDOCS/final.finalreport.pdf>
- [8] H. Balakrishnan, et al, "A Layered Naming Architecture for the Internet," SIGCOMM 2004, pp. 343-352.
- [9] R. Moskowitz, P. Nikander, "Host Identity Protocol Architecture," Internet Draft, August 1, 2005, draft-ietf-hip-arch-03, 24 pp.
- [10] R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, "Host Identity Protocol," Internet Draft, October 24, 2005, draft-ietf-hip-base-04, 99pp.