Network Working Group
Internet-Draft
Expires: March 17, 2004

E. Lear
R. Droms
Cisco Systems, Inc.
September 18, 2003

What's In A Name: Thoughts from the NSRG
draft-irtf-nsrg-report-10.txt

Status of this Memo

Copyright Notice

Abstract

Over the last few years, the use of IP addresses for Internet connectivity has changed dramatically. The Name Space Research Group (NSRG) was chartered by the IRTF to review these changes, and make recommendations on whether or not remediation within the protocol stack is necessary. This document reports the outcome of some of the discussions within the research group.

One of the questiones addressed by the NSRG is: Does the TCP/IP protocol suite need an additional level of naming above layer 3 but below the application layer? There was no consensus on the answer. This document reviews the motivation for an additional naming mechanism, reviews related work, proposes a straw man "stack name" and discusses the structure and use of those names.

1. Introduction

The use of IP addresses in the Internet has changed over time. While routing of IP packets has remained largely unchanged, the use and assignment of IP addresses has changed considerably.

For many years, hosts could be named by applications either by their mnemonic domain name or their IP address. The static binding between name and address allowed the interchangeable use of either to identify a host. Indeed for quite some time, as Internet name service matured it was necessary for applications and end users to have a valid fallback method in the case of a name server failure.

However, several new developments have changed the nature of addressing in the Internet:

- dynamic IP addressing, as provided, for example, through PPP [1] and DHCP [2]
- private network address space and network address translators (NAT) [3]
- virtual hosts, where one host is assigned multiple IP addresses
- load sharing or load balancing, where one IP address is shared by multiple hosts, so the services at that address can be provided by multiple hosts

The overall addressing model on the Internet has shifted to one of dynamic binding between a host and its address. A host is assigned an address from place to place, or from time to time, when the host needs to assert a location in the network topology. In addition, a single IP address can now be shared by multiple servers to represent a single logical end point. The converse is also true - a single server can represent multiple logical endpoints, and not even have to use multiple addresses [4].

This is not the first document to point out the differences between names, addresses, and routes. Shoch delineated those differences as early as 1978 [5]. Saltzer, et al., have also written about the nature of naming and addressing [6][7]. Research into the nature of names, addresses and routes can help provide insight into the current situation, in which the function of IP addresses is overloaded to serve the function of a location in the network, an interface, a host name, and a portion of that which identifies a TCP connection.

Given the changing nature of the use of IP addresses for end point identification on the network, is something more than IP addresses and domain names needed to identify hosts? What functionality would that something bring to the table?

1.1 Security Considerations

Communications today are secured through one of several means. For strongest protocol security, the communication is encrypted and the ends are identified with verifiable public keys. Several systems are available today to do this, including SSH [8], the IPSEC mechanisms of ESP [9] and IKE [10], TLS [11], and PGP [12].

The absence of a name space that uniquely identifies a host has created problems in the design of ESP and AH (IPsec). ESP and AH should bind security associations to a name for a host that is distinct from either a domain name or an IP address, because both the DNS entry and the IP address can change, for example when a host moves form one point in a network to another, while the security association could remain valid. Another advantage to a name space that is independent of the network topology is that ESP and AH could use such names for security associations that traverse NAT devices. In the absence of a persistent name in the Internet Architecture, IP addresses are used for the binding of security associations. This is an architectural shortcoming, not a feature.

At a different level, there is an expectation that the routing system guides a packet toward the destination end point indicated in the IP destination address. Until a few years ago, this would not have been an unreasonable assumption. Today there are exceptions, particularly transparent web proxies and firewalls.

With some of the currently contemplated changes, the risk of a transport connection being hijacked changes. Instead of having to intercept every packet, an attacker may only need to forge a rebinding message to one end or the other of a connection.

1.2 How Things Have Changed

As mentioned earlier, the nature of addressing in the Internet has changed. One important change in the Internet addressing model comes from the use of NAT [13][14][15]. When a web client contacts a server to request a web page, it is quite likely that the remote address and TCP port, as it appears to the web server, will not be the same as the host's source address and port used by the web client. Furthermore, it is likely to be difficult for the web client to determine the address received by the server as the client's address. And, a host that has a NAT device between it and the

Internet cannot become a server because other clients have no way to address it.

Another change to the addressing model in the Internet is that computers are far more mobile than they were just a few years ago. When a host moves from one location to another, its address changes to reflect the change in its point of attachment to the Internet. Because TCP bases its transport connection state on IP addresses, any connections to the old address are lost (but see below).

One of the largest changes in the character of Internet usage involves the resources people access and how they access them. Whereas in the past one intended to access a particular host with a particular IP address, today one is likely more interested in accessing a service, such as a news service, or a banking service, and one is less interested in the host upon which the service sits. An industry has built up around the notion these so-called content delivery or overlay networks. The IP address of the web server provides an ephemeral point of contact for a particular web page. In particular with secure services, what matters most to the user is that a particular trusted company has verifiably provided the service.

1.3 Why Things Have Changed

The most important change the Internet has undergone is spectacular growth. The result of the growth has been shortages in address space and routing resources.

As the growth of the Internet exploded so did address space utilization. A combination of measures, including the introduction of private address space, NATs, and a tightening of policy by addressing registries reduced the risk of the Internet running out of allocatable addresses until the 2010 time frame (or later). As a result, however, the unique identification of a host and the universal ability to reach it was lost.

At the same time, Internet routing tables exploded in size. To reduce routing tables routes, classless routing [16] was developed and deployed to aggregate routes on bit boundaries, rather than on old classful boundaries. Next, the IANA discontinued its policy of allocating addresses directly to end users and instead allocated them hierarchically to providers, requiring providers to show sufficient allocation and utilization to justify further assignments. This retarded for a time the explosion in routing, but it did not eliminate growth. While work continues in this area, it is important to understand that as of this writing the aggregation of routes through CIDR is the most efficient way to route Internet traffic,

given its current design goals.

There is a natural conflict between the above two policies. If one allocates addresses in small chunks, more routing entries will result. Periodically providers will renumber to get larger blocks, at the inconvenience of all of their customers.

In summary, the Internet has exploded in size, NATs are now widely present, and the use of mechanisms such as PPP and DHCP are widely deployed. In addition, services are now as much or more of interest than are individual hosts. Given all of these changes, is it possible to add a new name space that will make connectivity more stable and allow us to establish some new operating assumptions, such as the ones that these complications broke?

2. Related Work

There exists a large body of work on name spaces and their bindings. The work discussed below primarily relates to the binding of stacks to IP addresses, with an eye toward mobility or transience.

2.1 Mobile IP

Mobile IP addresses the problem of having a stable host identifier on mobile hosts. As a host changes its connection point to the network, it updateds a home agent with the mobile host's new address. The home agent represents a static point through packets can be exchanged with the mobile host. Mobile IP provides a different solutions for IPv4 [17] and IPv6 [18]. In IPv4, Mobile IP is a tunneling mechanism. In IPv6, mobile hosts make use of destination options. A mobile host uses its home address to create transport connections and communicate with other hosts. Datagrams exchanged with an IPv4 mobile host are tunneled through a home agent and optionally a foreign agent, so that the mobile host's can be found in the routing system without additional global routing overhead. In IPv4 the home agent is separate from the other end of a transport connection, and packets take a triangular route. In IPv6, support of mobility is required, and the likely non-mobile host, the correspondent node, is aware that the other end is mobile. Therefore, once the mobile host and remote host establish communications they can "short circuit" to remove the home agent. This is key because, while the foreign agent is likely to be near the mobile host, the home agent is unlikely to be near anybody.

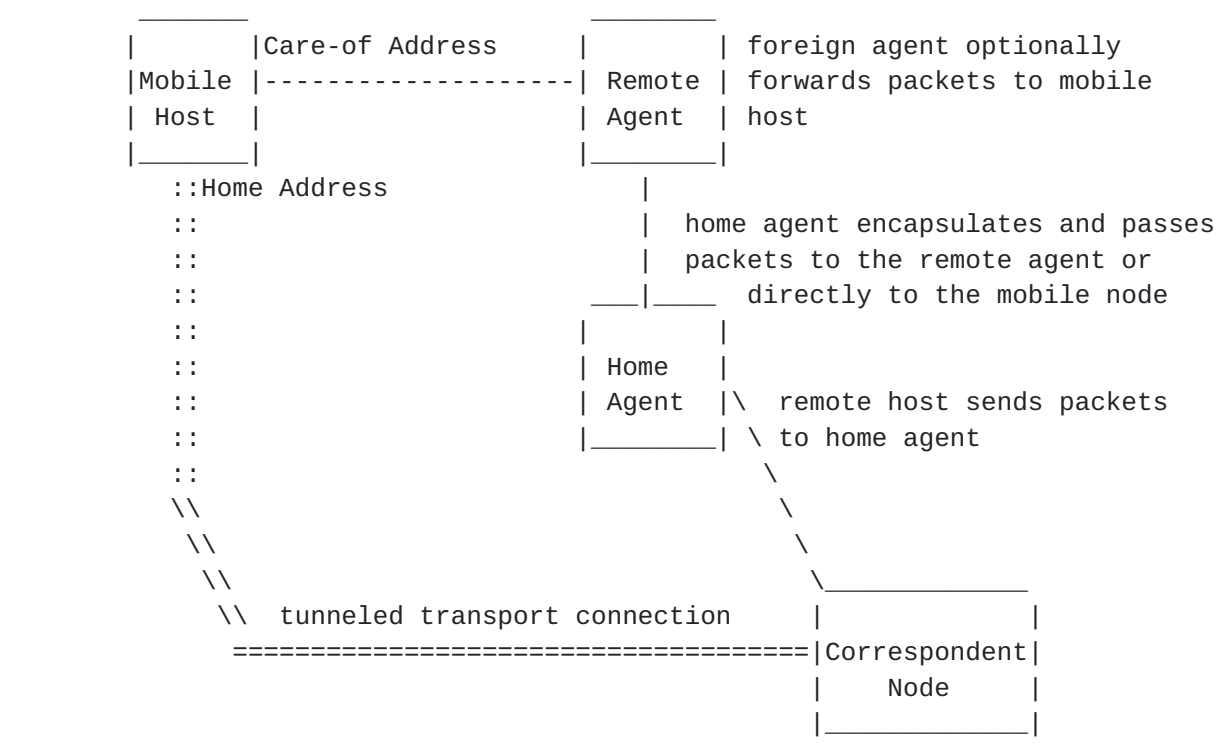

Figure 1: Mobile IPv4

In effect, Mobile IP turns the mobile node's IP address into a host identifier, where the "care of" address is the host's current location. The way Mobile IP succeeds is that it uses tunneling within the topology to represent an address at one location when it is in fact at another. However, a route to the mobile node's address itself must be available within the topology at all times. In an IPv4 world this would be untenable because of constraints on both the addressing and routing systems. With IPv6, the addressing pressures are off, and so each host can have a unique end address. However, problems remain with the routing system. In addition, there is a class of devices for which there may be no "home", such as devices in airplanes, mobile homes, or constant travelers. Additionally, there is a desire within some of the mobility community to have "micromobility" mechanisms that enable faster movement than envisioned by Mobile IP. The Routing Research Group (rrg) is currently investigating this area.

Most importantly, a mobile device can't withstand the loss of the home agent, even if the mobile device is still connected to the network. With the home agent offline, no incoming connections can get to them, and long-lived communications cannot be re-established. If the identity wasn't overloaded on the home address, it might be possible to work around such a failure.

2.2 Stream Control Transport Protocol (SCTP)

Many of the problems raised have to do with the overloading of layer 3 location information at higher layers, such as the use of an IP address in the pseudo-header in TCP. SCTP [19], an alternative to TCP, uses IP addresses in a more dynamic way as the identifiers for connection end points. TCP transport connection end points are named by IP addresses, and there are precisely two end point addresses, one for each end. SCTP allows multiple addresses per end, nominally for redundancy of applications that require high availability. However, it is possible to move a connection as a host moves from one location to another, or as its address changes due to renumbering (for whatever purpose). Work has progressed within the IETF to introduce a new capability to SCTP, that allows connection end points to change the set of IP addresses used for a connection [20].

There are three limitations to this idea. For one, it only affects those hosts that use SCTP, and therefore the idea is not sufficiently general.

The second problem is that, as contemplated in the draft, the risk of an attacker hijacking a connection is elevated. This same problem exists within Mobile IP, and may similarly be mitigated by purpose built keys (see below).

Finally, because SCTP does not have a home agent, SCTP does not handle the case when two nodes change their location at the same time, a case some would argue is a corner case.

2.3 Host Identity Payload (HIP)

Host Identity Payload (HIP) is a new approach to the problem of naming end points [21]. It inserts an additional "name" between layer 3 and layer 4, thus becoming layer 3.5. The goal is to decouple the transport layer from the Internet layer, so that changes in the Internet layer do not impact the transport layer, and the benefit is shared by all mechanisms atop transport protocols that use HIP.

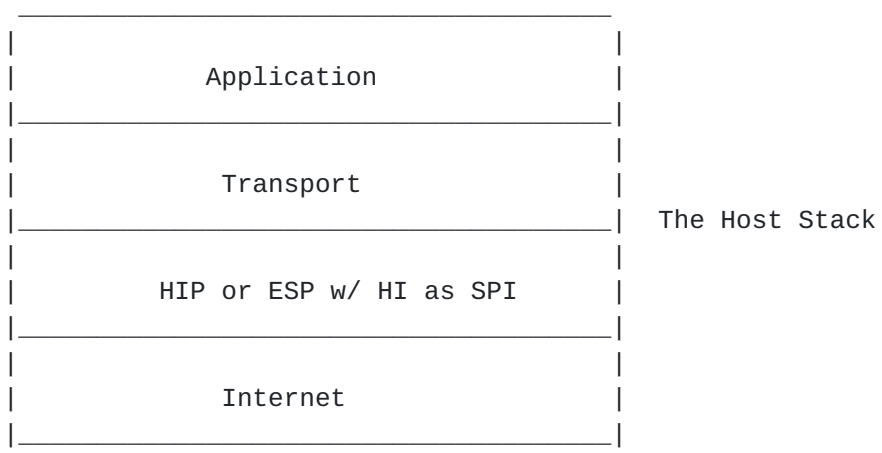

Figure 2: Host Identity

HIP itself relies on a cryptographic host identity (HI) that is represented in a Host Identity Tag (HIT) of various forms. One is a hash of the public key host identity, another is an administratively assigned value coupled with a smaller hash of the public key host identity. Host identities can be public or anonymous, the difference being whether or not they are published in a directory.

Whereas today one binds the transport layer to an IP address, HIP proposes that the transport layer binds to a host identity tag (HIT). The DNS is used to determine the HI and HIT, or to validate via reverse lookup an HIT. Further, the DNS continues to be used to get an Internet address.

Whether one should want to decouple the transport layer from the Internet layer is a controversial question. After all, that coupling has for many years provided the barest bones of the security of knowing that the packets that make up the connection are being guided through the network by routing tables in Internet routers that are owned by people and organizations whose intent is to get one's packets from source to destination. If we divorce the transport layer from the Internet layer, we introduce another way for an attacker to potentially hijack connections. HIP attempts to address this through the use of public key verification.

Additionally, HIP raises an issue regarding other uses for aggregation of IP addresses. Today, they are not only aggregated for purposes of reduced routing, but also for reduced administration. A typical access list used on the Internet will have some sort of a mask, indicating that a group of hosts from the same subnet may access some resource. Because the value of a HIT is a hash in part,

only the administratively assigned value can be aggregated, introducing an allocation limitation and authorization concerns.

On the other hand, there is the old computer science saying, any problem can be fixed by an additional layer of indirection It should be possible to administratively aggregate on groupings that are made at higher layers.

An alternative approach would be to aggregate based on domain names, rather than HI values. draft-moskowitz-hip-arch-02.txt [21] describes this approach in more detail.

A key concern with HIP is whether or not it will work in a mobile world. If the DNS is involved, or if any directory is involved, will caching semantics eventually limit scalability, or are there mobility mechanisms that can be employed to make use of directories feasible?

2.4 Purpose Built Keys

Purpose built keys (PBKs) are temporary end point identifiers that are used to validate a given endpoint during a communication [22]. Rather than attempting to build an infrastructure to validate the end points, however, PBK's sole purpose is to ensure that two hosts that originate a communication may continue that communication with the knowledge that at its conclusion each end point will be the same end point it was at the start. Thus, even if one's address changes it is still possible to validate oneself to the other side of the communication.

PBKs make no claim as to who the parties actually are. They make no use of public key infrastructures. PBKs are themselves ephemeral for the duration of a communication.

PBKs take the form of ad hoc public/private key pairs. When a node wishes to validate itself to another node it signs a piece of data with its private key that is validated by the other end with the public key. The public key itself becomes an end point identifier.

PBKs might be instantiated in several different places in the stack; for example, they may be carried in an IPv6 header extension or used by an application protocol.

2.5 RSIP and MIDCOM

Two related efforts have been made to stitch together name spaces that conflict. One is Realm Specific IP (RSIP) [23], which allows the temporary allocation of address space in one "realm" by a host in another realm, not unlike the way an address is gotten via DHCP. The

benefit of RSIP is that it allows the end point to know what address it is assigned, so that it may pass such information along in the data path, if necessary. The problem with RSIP is that host routing decisions are very complex. The host makes decisions based on destination, a process that requires a fair amount of configuration, and lacks certainty as it is based on a non-unique IP addresses. Because RSIP borrows public addresses it must relinquish them as quickly as possible, or the point of NAT is negated. In order to make better use of the scarce public resource, RSIP implementations would need to route not just on destination address, but on application information as well. For example, internal hosts would probably not need external addresses merely to browse the web.

MIDCOM, an architecture for middlebox communication, is a similar approach [24]. However, rather than tunneling traffic, an agreement between an end point and its agent and a "middle box" such as a NAT or a firewall is made so that the end point understands what transformation will be made by the middle box. Where a NAT or a web cache translates from one name space to another, MIDCOM enables end points to identify that translation.

MIDCOM is contemplated for use by specific applications, and thus it avoids the problems associated with RSIP. However, neither MIDCOM nor RSIP resolve how to discover such middle boxes. Nor do they provide a unique way for a host behind a NAT to identify itself in an enduring way. Finally, they both run into problems when multiple NATs are introduced in a path.

2.6 GSE or "8+8"

One proposal attempts to ease the conflict between the end systems' need to have a fixed name for themselves, and the routing systems' need for address assignments that minimize the overhead of routing calculations [25]. The clash between these two needs produces either the inconvenience (for the end systems) of renumbering, or routing inefficiency and potentially poor address space utilization as well.

Known as 8+8, Global Site End system (GSE) would have split the IPv6 address into two parts: a routing system portion that would be assigned and managed by service providers that would change based on routing system requirements, and a locally managed portion that would be assigned and managed by terminal autonomous systems. While each portion is globally unique, there are in effect two addresses, one to get a packet to an autonomous system and another to get to the host. Further, end hosts might not be aware, at least initially, of their routing portions. It was envisioned that the renumbering of the routing portion could be done as a matter of

signaling, with little

administrative involvement from the end point. Another goal of GSE was to eliminate additional routing overhead caused by multihomed end systems, whose information must today be carried throughout the routing system. By allowing end enterprises to have multiple global parts for purposes of multihoming, the terminal ASes would become what are today's last-hop ISPs.

Separation of transport names and internet names could also occur by having transports only use the local portion of the IP address in their pseudo-header calculatioons. There are a number of challenges that GSE would have to overcome. For one, how does one glue together the provider portion of an address with the more local part, and how would one accomplish the task securely? Would doing so eliminate the need or interest in adding other additional name spaces?

2.7 Universal Resource Names

Universal resource names (URNs) do not provide us a mechanism to resolve our naming concerns [26]. Rather, they may provide us the form of the name to use, and perhaps a framework for resolution. For instance, a host identity may conceivably be represented as a URN. URNs further the notion of defining a binding and boundaries between the name of an object and its location.

3. Discussion: Users, Hosts, Entities and Stacks

The original addressing architecture of IP and TCP assumed that there is a one-to-one relationship between an IP address and a communicating "entity." By "entity," we mean an identifiable participant in an Internet communication. Examples of an entity include a host, a user, a client program or a service. This one-to-one relationship between IP address and entity was assumed to exist throughout the duration of a "session" (usually a TCP connection); that is, all of the IP datagrams exchanged during a session would share the same endpoint identifiers, and the endpoint identifiers in those datagrams would not be altered as the datagrams traversed the Internet.

There is also an assumption that the binding between an entity and an IP address would vary only infrequently over time. The DNS allows the binding between a domain name for a host and its IP address to vary over time, but changes in those bindings may propagate slowly and do not accommodate frequent changes.

As explained in section 1, the underlying addressing architecture of the Internet has changed, leading to the need for new naming mechanisms that function with host mobility, the instantiation of multiple entities on a single host and the instantiation of a single

entity across multiple hosts, and that can provide security independent of IP addressing.

When a host moves from one location to another, or when a host receives a new address for some other reason, its identity has not changed, nor has that of the person using it. That entity may well be in communication with other computers and have access rights to network resources. Indeed, multiple entities may be represented by a single computer.

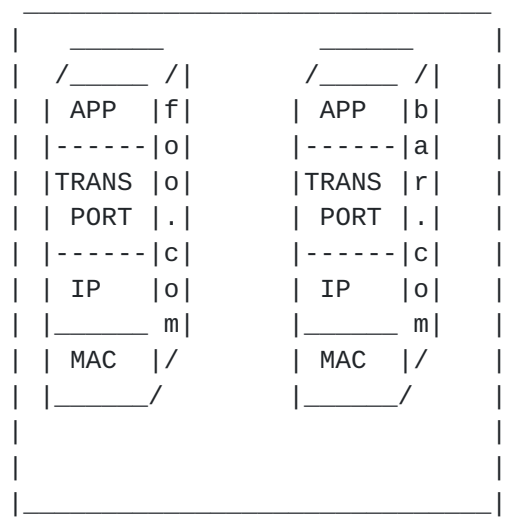

Figure 3: One application: multiple stacks on a single device

Today, a host may represent multiple entities. This happens when a service provider hosts many web sites on one server. Similarly, a single entity may be represented by multiple hosts. Replicated web servers are just such an example. These entities are "protocol stacks" or simply "stacks", instantiations of the TCP/IP model, be they across one or many hosts. A stack is defined as one participant or the process on one side of an end-to-end communication. That participant may move and may be represented by multiple hosts.

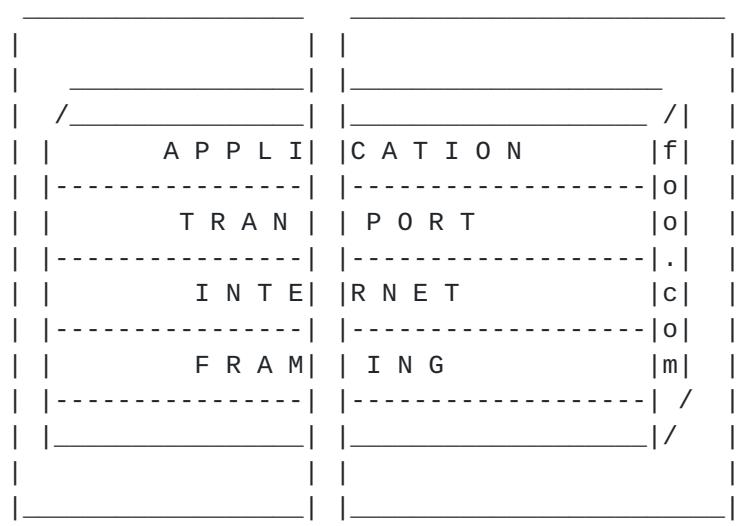

Figure 4: Another application: single stack represented by multiple hosts

Each instance of a stack has a name, a "stack name". At an architectural level the Name Space Research Group debated the value of such names, and their associated costs. Forms of this name are used in numerous places today. SSH uses public/private key pairs to identify end points. An HTTP cookie anonymously identifies one end of a communication, in such a manner that both the connection and the IP address of the other end point may change many times. Stack names are intended to identify mobile nodes, devices behind NATs, and participants in a content delivery or overlay network.

When two devices represent a single end point they must share state in order to keep the other end from becoming confused (to say the very least). When doing so, such stacks may indeed consist of multiple processes on each end. One view is that such processes can theoretically be named independently of the Internet layer, allowing for sessions to migrate at the behest of applications. However, it is not possible to standardize migration independent of applications that retain state in all manner of places, from configuration files to memory. Additional names of such processes serve only to identify those who are authorized somehow to represent the end point, and do not themselves alleviate effort required to ensure application consistency.

As used above, "sessions" are a mechanism that the current IP stack does not formally provide. If a session layer existed in the classic sense it might sit above the transport layer, and a session could consist of more than a single transport layer connection. If the session layer appears below the transport layer, then transport layer

connections can be bound to a name, such as a session identifier, something other than that of the IP address, and transport layer connections could persist across IP address changes.

3.1 Requirements, desirable features and design decisions

Stack names are defined to be a new naming structure integrated into Internet addressing, which provide a solution to several problems in the current addressing architecture. We have identified several requirements for this naming structure. Stack names will allow continuation of sessions independent of host mobility or other host renumbering. A stack that spans several hosts is identified by a single stack name, and multiple stacks on a single host are unambiguously identified by separate stack names. Stack names allow authentication of stack identity, authentication of the origin and contents of messages and privacy for message contents. Finally, the stack name architecture will interoperate with existing Internet infrastructure, including existing host implementations and core routing, for backward compatibility.

Stack names are intended to address as many of the problems in the current Internet addressesing as possible, including: NAT, mobility, renumbering, multiple entities on one host and entities that span multiple hosts. Stack names should be globally unique, so that state about stack names, such as mapping information, need not be kept in the network. Stack names should also provide anonymity, so that users or other entities cannot be easily identified through a stack name.

These requirements and features lead to several design decisions:

- Internal structure: opaque/structured, fixed-length/variable-length, universally-unique/random-unique
- Position in stack
- Mapping to mnemonic name (are stack names ever visible to humans?)
- Relationship between stack names and routing system

Each of these design points is discussed below.

3.2 What do stack names look like?

Names may be structured or unstructured. If they are structured, what encoding do they use, and what is their scope? Is the length of such a name fixed or variable? Are stack names unique across the Internet? If so, are they guaranteed unique through some sort of a registry or are they statistically unique? If it is a registry, is

it centralized or distributed, such as the DNS? The remainder of this section summarizes the discussion within the NSRG on these questions.

Again, one possibility is that stack names could be represented as Mobile IP home addresses. The benefit of this idea is that one might well derive a large benefit without having to incur any additional protocol engineering, at least initially. By representing stack names in this way the architectural distinction between stack name and location is somewhat muddled. If the goal is to separate location and entities, where an IP address represents location, a Mobile IPv6 answer doesn't get us to the goal.

3.2.1 Uniqueness

The reason this document exists is that uniqueness is desirable. Uniqueness offers certainty that a name represents exactly one object. A records from the DNS never were intended to have uniqueness. IP addresses, particularly in a V4 environment, no longer have uniqueness. Uniqueness allows people and programs to build operating assumptions about the other end of a communication. TCP was designed with such an assumption.

Being uniquely identified also raises security concerns. What if you don't want to be uniquely identified by generators of SPAM or by powerful entities such as governments? Note that we refer to the uniqueness of the object referenced by the identifier. An object itself might have multiple names.

3.2.2 Statistical uniqueness versus universal uniqueness

The classic way the model ensured uniqueness of names and addresses on the Internet was to have those names and addresses assigned by central authorities through a distributed tree-structured database. The overhead for name assignment may be distributed through delegation of authority. While this mechanism for name assignment guarantees uniqueness to the level of competence of those authorities, such delegation introduces overhead, artificial markets, trademark concerns, and other problems.

Some members of the NSRG are concerned that any new registry for stack names would bring unwelcome and burdensome administrative costs to connecting to the Internet, either as a service or a user. One could envision a very large reverse lookup domain that contains all host identities, leaving little room for decentralization.

In particular two problems have cropped up with centralized name spaces. The first is that of domain squatting, where people buy a

name simply for its usefulness to others. The second problem lies with IP addresses, that are allocated and sold by providers. Those providers may choose to make a "service" out of making addresses available to customers. When designing a new name space, one should introduce no artificial scarcity.

One way to avoid a new administrative overhead would be for individuals to be able to generate statistically unique names. However, statistically unique names can easily be mapped TO, but they are less easily mapped FROM. This is because it is difficult to establish trust relationships necessary to make changes to the mapping. For instance, if a central authority controls the name space, there must be some sort of authentication and authorization model in place for the change to be allowed. If such a mechanism is in place, one has to wonder (a) why the names used for that infrastructure are not used and therefore (b) why statistically unique names would be of any advantage.

There was a consensus that if we were to introduce a new name space it should not be mnemonic in nature. The DNS exists for that purpose today, and while others have recently identified a need to revisit the DNS, that was not the purpose of this effort.

3.2.3 Mapping

This brings into question several related concerns with naming: what, if any, mapping mechanisms exist? Should stack names map to IP addresses, to domain names, or for that matter, to anything? Do domain names, X.509 distinguished names, or other names map to stack names? Each is a separate question. A name on its own is of very limited value. The mappings infer how the name will be used. Is a stack name just something that sits in a transport control block on a device? In effect purpose built keys could accomplish that task.

3.2.4 Anonymity

Related to uniqueness and mapping is anonymity. Is it possible or even desirable to have anonymous names? That is, should my computer be able to establish a communication to your computer, such that you can be assured that you are communicating with the same entity who used a particular name, without actually knowing the underlying binding between the name and the object?

3.2.5 Fixed versus variable length names

When the nature of the name is decided one must decide whether the name should be of fixed or variable length. Traditionally those fields which are found in every packet tend to be fixed length for

performance reasons, as other fields beyond them are easily indexed. The form the name takes will have some relevance to this decision. If the name appears along the lines of an X.509 distinguished name, it must be variable. If the name is otherwise fixed length and supposed to be universally unique, the field must allow for large enough numbers to not require a protocol change anytime soon. Similarly, if the name is statistically unique, the field must be large enough so that the odds of a collision are acceptably low so that the protocol needn't change anytime soon. We leave it to engineers to determine what "anytime soon" and "acceptably low" are.

A convenient feature of a variable-length name is that it allows for ease of organizational delegation. If one provides a hierarchical model such as the DNS, one can decentralize authority to get a new name or to change a name. By the same token, such structure requires a root authority from which distribution occurs. So long as the name itself is not a mnemonic, perhaps it is possible to limit problems such as domain squatting.

Ultimately, if the name is to be other than statistically unique, there will be some sort of central root service.

3.3 At what layer are stack names represented?

Where are stack names represented? Are they represented in every packet, or are they represented in only those packets that the underlying use requires? The benefit of not requiring stack names to appear in every packet is some amount of efficiency. However, the benefit of having them in every packet is that they can be used by upper layers such as ESP. In addition, end points would be able to distinguish flows of packets coming from the same host even if the IP address changes, or if the remote stack migrates to another piece of hardware. The PBK approach would alert an end point when one side knows of such a change, but as we have seen, the IP address one side sees, the other side may not, without a mechanism such as MIDCOM or RSIP. HIP and ESP solve this problem by putting an identifier (either the HIT or SPI) in every packet.

If a stable Internet layer existed it might be possible to represent stack names as IP addresses. Even if a host moved, a stack name could be represented as a Mobile IP "home" address. The PBK proposal suggests that stack names be passed as necessary as end to end options in IPv6 or simply as options in IPv4.

If a stable Internet layer doesn't exist, then stack names must appear above it. If a new mechanism were inserted between the Internet and transport layers, all end points that wish to use the mechanism would need to change.

3.3.1 A few words about transport layer mechanisms

One may not wish to completely divorce the transport layer from the Internet layer, as currently implemented. The transport layer mechanisms today are largely responsible for congestion control. If one end point moves it is quite possible that the congestion characteristics of the links involve will change as well, and it thus might be desirable for mechanisms such as TCP Slow Start to be invoked. It is also possible that codecs may no longer be appropriate for the new path, based on its new characteristics. In as much as mobile hosts change their locations and bindings with Mobile IP today, this is already an issue.

3.4 Stack names and the routing system

It would seem a certainty that the routing system would want very little to do with stack names. However, as previously mentioned, when the binding between Internet and transport layers is broken, some care must be taken to not introduce new security problems, such that a connection cannot be hijacked by another host that pretends to be authorized on behalf of an end point.

One misguided way to do this would be to enforce that binding in the routing system by monitoring binding changes. In order for the routing system to monitor the binding, it realistically must be done out in the open (i.e., not an encrypted exchange) and the binding must appear at some standard point, such as an option or at a predictable point in the packet (e.g., something akin to layer 3.5).

In other words, one would have gone all the way around from attempting to break the binding between transport and Internet layers to re-establishing the binding through the use of some sort of authorization mechanism to bind stack names and Internet addresses.

3.5 Is an architectural change needed?

The question of what level in the stack to solve the problem eventually raises whether or not we contemplate architectural changes or engineering enhancements. There can be little dispute that the topic is architectural in nature. For one, there are now numerous attempts to solve end point identification problems within the engineering space. We've already mentioned but a few. The real question is whether the existing architecture can cover the space. Here there are two lines of thought. The first is that the use of mobility mechanisms and Mobile IP will cover any perceived need to provide stack names. Assuming that it can be widely and securely deployed, Mobile IP certainly resolves many host mobility concerns. However, it remains to be seen if it can address other problems, such

as those introduced by content delivery networks.

The other line of thought is that there is an architectural distinction between names, addresses, and routes more explicit since there is otherwise an overloading of operators. Regardless of whatever tactical benefit one might gain, architectural separation should provide value in and of itself over time. The risk of this argument is that we will have introduced complexity without having actually solved any specific problem, initially.

To resolve the differences between the two schools of thought requires development of the second school of thought to the point where it can be properly defended, or for that matter, attacked.

4. Conclusions or Questions

The NSRG was not able to come to unanimity as to whether an architectural change is needed. There are two views. The first is simply that IP is just fine the way it is, and MIPv6's shortcuts allow for it to remain the end point identifier. The second view is that we could use a better architectural separation between end point identifier and locator. HIP is such an example. To better answer the question, the notion of stack names should be further developed.

Specific questions that should be answered are the following:

1. How would a stack name improve the overall functionality of the Internet?

2. What does a stack name look like?

3. What is its lifetime?

4. Where does it live in the stack?

5. How is it used on the end points

6. What administrative infrastructure is needed to support it?

7. If we add an additional layer would it make the address list in SCTP unnecessary?

8. What additional security benefits would a new naming scheme offer?

9. What would the resolution mechanisms be, or what characteristics of a resolution mechanisms would be required?

Of the many existing efforts in this area, what efforts could such a

name help? For instance, would a stack name provide for a more natural MIDCOM design?

This document raises more questions than answers. Further studies will hopefully propose valid answers.

5. Further Studies

Various efforts continue independently. One outgrowth is the possibility of a HIP working group within the IETF. Although this work might occur within the IETF, it should be noted that there is a risk to attempting to standardize something about which we yet have the full benefit of having explored in research.

Work on relieving stress between routing and addressing also continues within IETF working groups.

A separate effort proceeds elsewhere in the research community to address what the Internet should look like ten years from now. That work may further conclude that stack names will play a considerably larger role.

It is possible that work will continue within the IRTF. However, that work should be conducted by smaller teams until mature proposals can be debated. Questions of "whether additional name spaces should be introduced" can only be answered in such a manner.

6. Acknowledgments

This document is a description of a review done by the Name Space Research Group of the Internet Research Task Force. The members of that group include: J. Noel Chiappa, Scott Bradner, Henning Schulzrinne, Brian Carpenter, Rob Austien, Karen Sollins, John Wroclawski, Steve Bellovin, Steve Crocker, Keith Moore, Steve Deering, Matt Holdrege, Randy Stewart, Leslie Daigle, John Ioannidis, John Day, Thomas Narten, Bob Moskowitz, Ran Atkinson, Gabriel Montenegro, and Lixia Xiang.

Particular thanks go to Noel Chiappa whose notions and continuing efforts on end points kicked off the stack name discussion. The definition of an endpoint is largely taken from Noel's unpublished draft. Thanks also to Ran Atkinson and Bob Moskowitz whose comments can be found (in some cases verbatim) in this document.

The idea of GSE or 8+8 was originally developed by Mike O'Dell. The documents in which GSE is described are not published as RFCs.

References

[1] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[2] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[3] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[4] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[5] Shoch, J., "Inter-Network Naming, Addressing and Routing", Proceedings of IEEE Compcon, pp72-97, Fall 1978.

[6] Saltzer, J., "On The Naming and Binding of Network Destinations", September 1992.

[7] Saltzer, J., Reed, D. and D. Clark, "End-To-End Arguments in System Design", ACM Transactions on Computer Systems Vol. 2, No. 4, November 1984.

[8] Ylonen, T., Kivinen, T., Saarinen, M., Rinne, T. and S. Lehtinen, "SSH Protocol Architecture", draft-ietf-secsh-architecture-14.txt (work in progress), July, 2004.

[9] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[10] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[11] Dierks, T., Allen, C., Treese, W., Karlton, P., Freier, A. and P. Kocher, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[12] Callas, J., Donnerhacke, L., Finney, H. and R. Thayer, "OpenPGP Message Format", RFC 2440, November 1998.

[13] Carpenter, B., "Internet Transparency", RFC 2775, February 2000.

[14] Hain, T., "Architectural Implications of NAT", RFC 2993, November 2000.

[15] Holdrege, M. and P. Srisuresh, "Protocol Complications with the IP Network Address Translator", RFC 3027, January 2001.

[16] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[17] Perkins, C., "IP Mobility Support", RFC 2002, October 1996.

[18] Johnson, P. and C. Perkins, "", draft-ietf-mobileip-ipv6-24.txt (work in progress), June 2003.

[19] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[20] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., Rytina, I., Belinchon, M. and P. Conrad, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", draft-ietf-tsvwg-addip-sctp-07.txt (work in progress), February 2003.

[21] Moskowitz, B., "Host Identity Payload Architecture", draft-moskowitz-hip-arch-02.txt (work in progress), February 2001.

[22] Bradner, S., Mankin, A. and J. Schiller, "A Framework for Purpose Built Keys (PBK)", draft-bradner-pbk-frame-06.txt (work in progress), June 2003.

[23] Borella, M., Lo, J., Grabelsky, D. and G. Montenegro, "Realm Specific IP: Framework", RFC 3102, October 2001.

[24] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A. and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

[25] O'Dell, M., "GSE - an alternate addressing architecture for IPv6", draft-ietf-ipngwg-gseaddr-00.txt (work in progress), 1997.

[26] Handley, M., Thaler, D. and R. Kermode, "Multicast-Scope Zone Announcement Protocol (MZAP)", RFC 2776, February 2000.

Authors' Addresses

Eliot Lear
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134

Phone: +1 408 527 4020
EMail: lear@cisco.com

Ralph Droms
Cisco Systems, Inc.
300 Apollo Drive
Westford, MA 01824

Phone: +1 978 497 4733
EMail: rdroms@cisco.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.