Network Working Group
Internet Draft
draft-crocker-mast-analysis-00.doc
Expires: <2-04>

D. Crocker
Brandenburg InternetWorking
September 16, 2003

CHOICES FOR SUPPORT OF MULTIADDRESSING

STATUS OF THIS MEMO

COPYRIGHT NOTICE

ABSTRACT

Classic Internet transport protocols use a single source IP address and a single destination IP address, as part of the identification for an individual data flow. TCP includes these in its definition of a connection and its calculation of the header checksum. Hence the transport service is tied to a particular IP address pair. This is problematic for multihomed hosts and for mobile hosts. They cannot use more than one, for any single transport association (context). In recent years, there have been efforts to overcome many of these limitations, through different approaches at different places in the Internet architecture. This paper reviews the requirements for support of multiaddressing (mobility and multihoming), and the efforts to support them. Barriers to adoption, administrative overhead, and operational efficiency are of particular concern.

CONTENTS

1. INTRODUCTION

Classic Internet transport protocols use a single source IP address and a single destination IP address, as part of the identification for an individual transport data flow. For example, TCP includes these in its definition of a connection and its calculation of the header checksum. Hence a classic transport association is tied to a particular IP address pair. This is problematic for multihomed hosts and for mobile hosts. Both have access to multiple IP addresses, but they are prevented from using more than one within an existing

transport exchange. For a host to use a different IP address pair, participants must initiate a new exchange. In the case of TCP, this means a new connection.

In recent years, there have been efforts to overcome many of these limitations, through different approaches at different places in the Internet architecture. Some modify the IP infrastructure, with embedded redirection services. Some define transport enhancements to support a set of addresses directly, and some define a layer between classic IP and classic transport. Each of the existing proposals has notable limitations in functionality, implementation, deployment or use.

This paper reviews the requirements for support of multiaddressing (mobility and multihoming), and the efforts to support them. Barriers to adoption, administrative overhead, and operational efficiency are of particular concern.

1.1. Terminology

This paper discusses requirements and methods for enabling an endpoint (host) to use multiple addresses during single application associations (sessions).

"Agent" refers to a forwarding service that represents an endpoint for multiaddressing. For mobility, the agent resides on the "home" network and relays datagrams to the endpoints actual location on the Internet. The endpoints are modified to support this forwarding technique. For multihoming, an agent hides the presence of multiple addresses from the endpoint located on the local network.

"Address" refers to a string that indicates a location, usually in terms of network topology. IP addresses specify a topological network access point. They usually are considered to specify an endpoint interface. However discussions about mobility are enhanced by viewing the value as belonging to the network (interface) rather than to the endpoint.

"Association" refers to a transport-level exchange context between endpoints, such as a TCP connection.

"Endpoint" refers to an end-system that participates in an association. Endpoints are distinguished from intermediate, infrastructure nodes and hosts.

"Identifier" refers to a unique label for an endpoint. The label is used simply for distinguishing one endpoint from another. If the location information in an address is ignored, it can serve as an identifier. However an address will usually suffer administrative and referential limitations as a global identifier for mobile endpoints.

"Initiator" refers to an endpoint that initiates contact with a target endpoint. In client/server architecture it is the client.

"Mobility" refers to the availability of different addresses at the same endpoint, over time. This may even include discontinuities, at times having no available addresses. It also may include overlapping availability of addresses. Interestingly, this looks the same as multihoming.

"Multiaddressing" refers to the availability of different addresses at the same endpoint. It encompasses both multihoming and mobility.

"Multihoming" refers to the availability of multiple addresses at the same endpoint, simultaneously. It is typically used to refer to multiple network attachments for a host, but works equally well for multiple upstream network attachments by the local network, when the different upstream addresses are visible to the host. Interestingly, multihomed environments often must support dynamic changes, such as when adding a new upstream provider. Therefore, multihoming can include mobility features and mobility can include multihoming features.

"Path discovery" provides a sender with the means for learning about the addresses from which they can send.

"Path selection" is required when more than one address is available to the sender. Although the sender is limited to specifying an address, rather than a path, it appears that thinking of it as path selection aids consideration of solutions. In effect, it formulates the selection task as being similar to the job of routers. Route formulation is mature technology, so that this aspect of multiaddress processing will be tractable, if not straightforward.

"Rendezvous" permits a host that is initiating an association to find the target of the association, such as a client finding a server. "Finding" means obtaining a valid address for the target. A public process is required for rendezvous. The primary Internet mechanism for rendezvous has been the Domain Name Service (DNS). The DNS uses long, variable-length strings (names) and is tailored for large-scale rendezvous with names and addresses (mappings) that change infrequently.

"Target" refers to an endpoint that receives contact from an Initiator endpoint. In a client/server architecture, this is the server.

1.2. Scenarios

What are the situations and concerns that affect design and use of a mechanism for the support of multiaddressing?

Section 3 of [MOBHOM], has an excellent discussion of these issues.

It is included here by reference without section 3.2. Section 3.2 covers an interesting topic that appears to be independent of multiaddressing.

The included text comprises the following sub-sections:

3. Usage scenarios
3.1 End-host mobility
3.2 Location privacy …
3.3 End-host multi-homing
3.4 Site multi-homing
3.5 Combined mobility and multi-homing
3.6 Network renumbering
3.7 Combined all

1.3. IETF Background

Historically, IETF focus on mobility has split between initial attachment configurations, into an otherwise static environment such as by using DHCP, versus forwarding mechanisms, such as by modifying the IP infrastructure with Mobile IP. Multihoming has largely been ignored, except in routing protocol work. Recent efforts are pursuing direct enhancements to transport or insertion of a mapping layer between IP and transport. There has also been adjunct activity, relevant to this topic.

The following summary of IETF activities relies on text from the Abstracts of documents for those activities. Analysis of the different architectural and protocol efforts is in Section 3, "Internet Stack Placement".

The Name Space Research Group [NSRG] considered modifications to the Internet architecture, including whether an additional level of naming above layer 3, but below the application layer, is needed. Purpose-Built Keys [PBK] specifies a template for the use of specially generated public/private key pairs, to provide assurance that successive messages in the communication come from the same source. This is accomplished without the use of external certification authorities.

Stream Control Transmission Protocol [SCTP] is a reliable transport protocol for multiplexed data streams. It includes modern mechanisms for safe initiation of a connection, as well as the necessary tools for reliability and congestion control. It also has a mechanism for communication access to multiple IP addresses between the participation host pair. [TCP-MH] uses TCP options to support multihoming. Datagram Congestion Control Protocol [DCCP] is a proposal for a network-friendly, unreliable transport-level datagram delivery service.

Mobile IP [MIP] provides an agent service to allow transparent routing of IP datagrams to mobile nodes in the Internet. Host Identity Protocol [HIP] is used to

establish a rapid authentication between two hosts and to provide continuity of communications between those hosts independent of the networking layer. The [LIN6] protocol defines a layer that supports multiple addresses, between IPv6 and transport. Multiple Address Service for Transport [MAST] supports association of multiple IP addresses during the life of any transport instantiation, by defining a layer between IP and transport. It operates only in the endpoints and works with IPv4 and IPv6.

1.4. Discussion Venue

Discussion and commentary are encouraged about the topics presented in this document. The preferred forum is the <mailto:multi6@ops.ietf.org> mailing list, for which archives and subscription information are available at <http://ietf.org/html.charters/multi6-charter.html>.

NOTE: The early drafts of a review document, like this, are certain to have significant errors. The author strongly requests guidance for clarifying and correcting any problematic text.

1.5. Document History

-00 Derived from draft-crocker-mast-proposal-00. Extended discussions about alternative proposals and architectural issues, separated from the -proposal- draft.

NOTE: The author has put forward the MAST proposal. Clearly that colors the perspective in this discussion paper.

2. REQUIREMENTS AND CONSTRAINTS

2.1. Mobility

Mobility is time-varying access to multiple addresses for the same endpoint. Key parameters to mobility are scope of change, rate of change and source(s) of the change. Over what portion of the Internet topology might a change take place; how often will changes occur; and which of the participants will change their addresses?

It is generally accepted that rapid, local changes should be handled by a layer below IP and therefore should be invisible to IP. For initiator endpoints that are subject to occasional detachment, with eventual reconnection, the current set of technologies is probably sufficient.

What is missing is support for initiator and target systems that move over the course of minutes or hours and need to maintain existing transport associations or need to maintain their availability for new associations. There are no IP-related standards for maintaining associations during mobility. For maintaining target availability, DNS dynamic update [DNSDYN] is plausible; however it is not widely deployed and the typical DNS record lifetime settings and client caching behaviors suggest that existing DNS use is better tailored for changes over days, rather than shorter times. Separately the core role of DNS for Internet infrastructure operations suggests avoiding major changes to its operational model. Supporting potentially high volumes of rapid changes probably require very different software and administration than are used for the current DNS.

The difference between mobility prior to initial contact and mobility during an association is significant. In the latter case, the mobile host can use the association state when needing to inform the other endpoint about the change. Prior to an association -- or when both endpoints are mutually mobile -- an independent rendezvous venue is required.

The difference between initiator mobility and target mobility is also significant, with respect to initial contact. In particular the initiator needs to be able to find the target. Again, this requires a rendezvous mechanism, such as having the routing system map from identifiers to routes, rather than addresses to routes. Either it must be provided implicitly within the network or there must be an external "rendezvous" mechanism. For static servers, the DNS already provides this rendezvous quite well. However current DNS use does not support frequent address changes over short periods. Hence enhancements are needed to support rendezvous with a mobile target.

2.2. Multihoming

The Internet already supports a number of types of "indirect" multihoming. The core of dynamic packet-switched routing is exploitation of alternative routes, so that the path between endpoints might vary considerable over the course of an association. For networks with multiple attachments to a backbone, external routing technology already permits propagation of alternate routing information. Further a domain name may have multiple address records that point to the same network. (However there is no indication whether the same records are, instead, pointing to different, redundant systems; on the other hand the importance of this ambiguity is not clear.)

What is notably missing is a means for an existing association to directly use multiple paths, in particular when the paths terminate at one of the endpoints. Here, the fact that classic Internet transport services rely on single, specific IP addresses is the barrier.

Support of multihoming can be useful for robustness and throughput. The former makes loss of a path transparent to the association. The latter increases the effective bandwidth for an association. In general, the former goal is dominating current work. At the least, using multiple paths for increased bandwidth ensures a high degree of out-of-order arrivals. This usually reduces target endpoint performance, rather than increasing it.

2.3. Security

The level of security built into IP is minimal. Some would say it is non-existent. However classic transport services rely on having a significant degree of correlation between the IP address in the source field of an IP datagram and the likelihood that the IP datagram came from that address. The context of repeated exchanges between source and destination addresses is taken as a validation of this correlation. Permitting the IP address of a source to vary during an association is an invitation to connection hijacking. Hence, any support for multiple addresses must contain a strong anti-hijacking mechanism.

All other security concerns are independent of multiaddressing; and they are probably best handled by additional mechanisms, such as IPSec and TLS. There is no indication that any of these other mechanisms need to be changed, so support multiaddressing.

Once there is an effort to design protection against hijacking, it is easy to consider adding more protections, such as privacy or, perhaps, other kinds of authentication. Although such mechanisms obviously would be useful, they are not essential to the basic requirements of multiaddressing. Further, they might be redundant with mechanisms provided elsewhere in the architecture.

Any effort related to multiaddress support, which goes beyond preventing hijacking, needs to have explicit discussion about its relationship to other security mechanisms and the need for attaching these additional capabilities to multiaddress support. As with any opportunity for adding features to a design effort, there should be concern about causing unnecessary design complexity, delays to the specification effort, and difficulty in implementation.

2.4. Implementation

The software that supports IP and classic transport services is mature. Usually it is highly tuned and highly robust. Often it is also complex. Hence it can be risky to introduce modifications to one or more of these modules. On the other hand, attempting to introduce multiaddress support through additional modules runs the risk of being awkward and inefficient.

2.5. Deployment and Use

However difficult it is to have vendors make major modifications to mature software, it is far more difficult to deploy the changes to a global installed base of hundreds of millions of platforms. Changes to support multiaddressing need to consider barriers to adoption by users and operators, both ISPs and enterprises. What is the effort needed to deploy the changes? What is the effort needed to use it? How broad must the adoption be before users can obtain benefit? What dependencies do the changes have on existing or new services?

Making one new service depend upon the reliable performance of another new service greatly increases the riskiness of the effort. Making a change require modification to the Internet's infrastructure typically creates a long delay before it is useful. In particular, early adopters gain no immediate benefit from their efforts; this acts as a disincentive for adoption. Everyone waits for others to take the first step.

2.6. Matters of State

Support for multiple addresses requires adding a conceptual layer of referential indirection. Beyond simple use of the DNS, endpoints currently use individual endpoint addresses within an association. In order to use multiple addresses, to refer to the same endpoint, some type of aggregation and mapping mechanism must be added. The mechanism defines a relationship between the referenced endpoint and a set of addresses. Where should this state information be placed in the Internet architecture?

If the major lesson of the Internet is scaling, the major embodiment of that lesson is to place complexity in the edges, rather than the infrastructure. Generally, this does not mean that there is a balanced debate between the choices. Rather, there is an assumption that a change should be made to the edges rather than the infrastructure. It is made in the infrastructure only when there is a clear agreement that doing otherwise will seriously reduce the utility of the change.

This methodology can even be applied to some infrastructure changes. A change that will clearly have an infrastructure impact might be introduced incrementally, via endpoint modifications. Two major examples of this are DNS and MIME. Both were added to operational, infrastructure services (the IP internet and the Internet mail service, respectively) but were added in a fashion that made no immediate changes to the existing services. Rather, edge systems independently chose to adopt the changes. Any two endpoints wishing to exploit the change, for interacting with each other, immediately benefited from the adoption. Over time, adoption became sufficiently broad-based to make the change effectively part of the infrastructure service. Although the IP network works well without the DNS, end-user utility of the Internet, without the DNS, would be nil. Similarly the ability to use attachments has become a fundamental part of the Internet mail experience.

Addition of support for multiaddressing faces a similar type of choice. Should the change be made above the transport layer, in the transport layer, in the IP layer, or perhaps between IP and transport? How is the aggregation established and how is it maintained? Do IP (or TCP, or...) packets contain the mappings or are they maintained in the endpoints or, perhaps, in the IP infrastructure?

The answers to these questions need to be determined by their effect on barriers to adoption, operational overhead, and administrative convenience.

2.7. Endpoint Identifiers

Historically, IP addresses have served the dual role of network interface locator and endpoint identifier (EID). Adding support for multiaddressing serves to highlight the need for splitting these two roles. IP addresses work quite well as network interface locators. However their topological dependence makes them work poorly as identifiers, in the richer world of multiaddressing.

Does an EID need to be assigned by a registry or can it be dynamically computed? Does it need to be publicly visible across the Internet or can it be kept private to individual associations? Does it need to be used frequently, such as in every datagram, or is it needed only for specific transactions, such as initiating or recovering an association?

It is appealing to define an EID to be publicly registered and carried in every datagram. This provides the maximum amount of decoupling from addressing and appears to offer an especially clean modification to the transport layer interface. Transport header calculation merely needs to switch to use of the EID, rather than the address. With sufficiently strong protection against hijacking, this approach can almost make the address irrelevant to the transport layer.

However there still must be a mapping between EID and addresses, so the IP service knows where to send the datagram. Hence, the state information of an EID/addresses "routing" table must reside somewhere. Unless the IP infrastructure is modified to directly support EIDs, this state information is most probably in the endpoints.

Having a public EID means that a new, global registration service must be developed and operated. Some believe network operators will not mind this additional work; others disagree.

Having an EID in every datagram means that the string must be as short as possible. Even then it will add significant overhead to datagram header size. However given the need to process multiaddressing, having the EID in every datagram probably will not alter datagram processing overhead, in the endpoints, from any other approach to using EIDs.

If an EID is used only occasionally, one candidate is a domain name. Domain names already have an administrative structure, and they are well engrained into Internet use.

Their length is not a problem, when they are need only periodically. One objection to using domain names is that they are already used in a number of ways that do not suit the role of EID. It is unclear how the fact that domain names serve multiple roles prevents their serving the role of EID.

2.8. Signaling

How does an endpoint learn its addresses? The notable challenge is when a NAT modifies the address an endpoint uses directly, to a different address that is visible to the rest of the network.

How does an endpoint communicate its available set of addresses to another endpoint?

DNS is currently useful for registering essentially static sets. More dynamic or tailored communication requires a signaling exchange between endpoints. This can be done through a distinct signaling protocol, such as is done with MAST, or inline -- that is, as a sub-exchange -- within an existing protocol, such as is done with TCP-MH.

2.9. Operation Through NATs

A Network Address Translation (NAT) device maps between one set of addresses, and another. In typical cases, addresses from the interior of a network are mapped to different ports on a single, public address on the outside of the network.

This mapping task must be performed with knowledge of transport protocol details because it must adjust transport headers, as well as IP-level addresses.

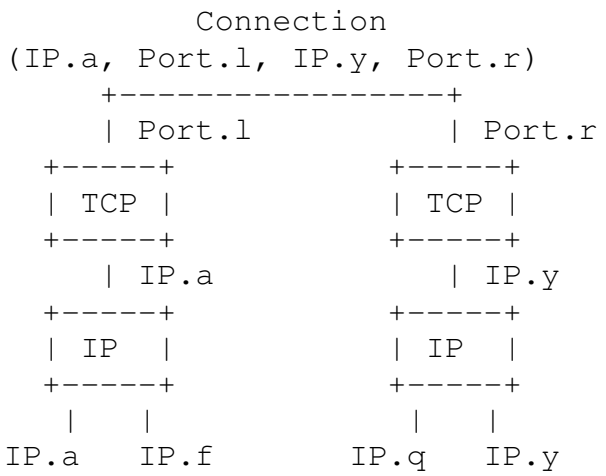
Stateless NATs are likely to work with most multihoming solutions and some mobility solutions. The NAT will simply do its usual task of replacing IP addresses and adjusting dependent transport headers accordingly. However, there is the basic question of whether a multiaddressed initiator correctly knows its own addresses. Typically it will not. Given the prevalence of NATs, a solution to multiaddressing needs to deal with this scenario.

Some solutions require that NATs be upgraded to support the solution. This is another example of an infrastructure dependency.

3. INTERNET STACK PLACEMENT

From a purely technical standpoint, multiaddressing can be supported through a number of different mechanisms. This section discusses the possible venues within the Internet stack, and existing efforts that are pursuing these choices.

The current architecture for transport use of IP addresses makes a direct linkage to a specific IP address pair:

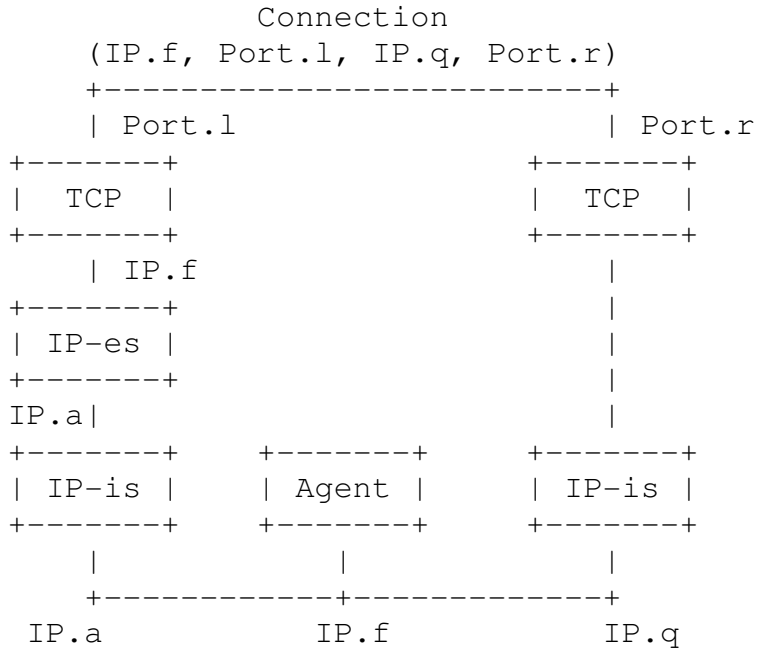


This example shows each host being multihomed. However a given association must choose a single IP address, at each end, and bind the connection to it.

3.1. IP Infrastructure

In the classic Internet infrastructure model, a datagram contains topological references to the source and destination network interfaces. The network knows nothing about higher-level issues, such as whether two interfaces are attached to the same endpoint. This design derives from the explicit desire to keep the Internet infrastructure as simple as possible, by putting as much functionality as possible into the endpoints rather than in the Internet's switching devices.

The Mobile IP [MIP] effort provides an encapsulation-based forwarding service. An agent intercepts datagrams using an original destination IP address, and then forwards the datagram to the destination's new IP address. An optimization may (later) permit direct transmission to the new venue. This is achieved by use of datagram encapsulation -- tunneling the original IP datagram inside a new one -- and by having datagrams carry both an address and an end-point identifier. [HOWIE] provides an interesting discussion of MIPv6 adoption and use issues.



Conceptually, the biggest problem with this approach is that it attempts to take topology-related information -- the IP address -- and use it as the basis for contacting an endpoint non-topologically.

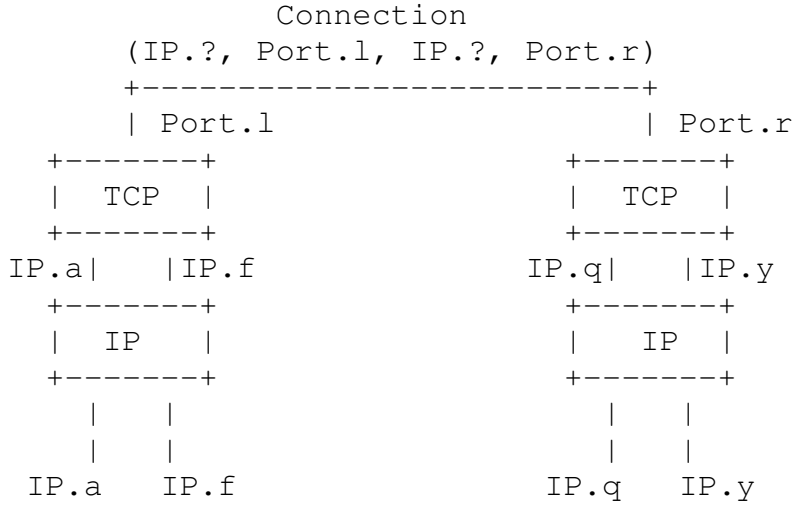
Operationally, the biggest problems with this approach are that forwarding services are inefficient, multi-layer encapsulation adds complexity, and the service requires infrastructure change.

Therefore, this approach changes the infrastructure and changes the IP datagram. Hence it changes several different aspects of the Internet architecture, with each change constituting a significant barrier to adoption or efficiency.

3.2. Transport-Level

Recent transport protocols, such as [SCTP], [TCPMH] and the proposal for [DCCP], use multiple IP addresses directly in the transport association. These efforts have primarily focused on multihoming, with the time-varying nature of mobility being ignored or retrofitted. TCP-MH notably uses TCP options for inline signaling of multihoming information between the endpoints; its current specification appears to have weak protection against hijacking but this can be remedied.

A transport-level approach has the benefit of placing the necessary functionality only in end-systems and avoiding possible address translation problems.



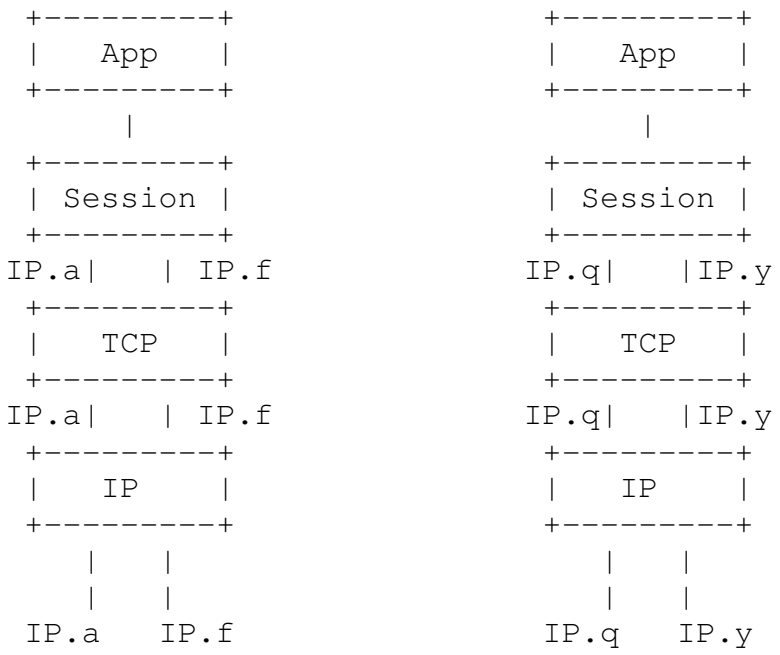
NOTE: *Given that multiaddressing is directly visible to the transport level, it is not clear how to formally define a connection. Are "virtual" addresses used? Is one of the addresses used?*

It also has the considerable benefit of leaving the IP infrastructure unchanged. Given the complexity and robustness of that infrastructure, as well as the considerable time and effort that was needed to achieve its stability, any design that avoids changing the infrastructure is to be commended.

The fact that the functionality is applicable across all transport services suggests that there might be benefit in having IP multiaddressing functionality reside in a single architectural module, separate from any specific transport service. In any case these new transport protocol efforts cannot affect the considerable installed base of services using older transport protocols, such as TCP and UDP.

3.3. Session-Level

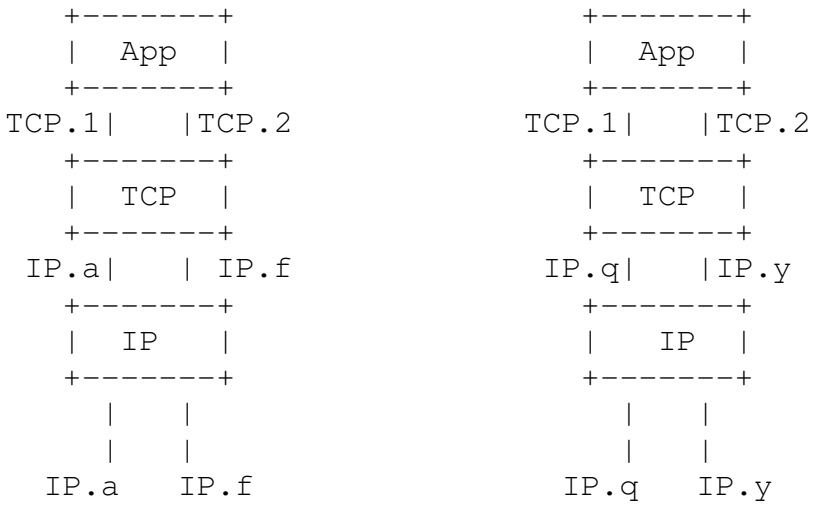
The session layer provides functionality above transport and below the application. In effect it is a way of institutionalizing application-level support. The merit of placing multiaddressing support at the session layer is that it can use multiple transport services.



The problem with this approach is that a full session layer typically replicates substantial portions of the transport service, in order to ensure reliability and in-order data sequencing. This makes the session-level approach notably complicated and inefficient.

3.4. Application-Level

Applications often provide themselves with enhanced infrastructure support services, to compensate for limitations in the lower protocol, or to optimize functionality and performance according to the peculiarities of the specific application. A typical example is with reliable data transfer, when using an unreliable datagram service. The obvious difficulty with this approach is that it burdens each new application with re-creating these (common) underlying services.



There well might be some benefit in permitting applications to discover details about multiple-address capabilities, and possibly even to specify some controls over their use, through an enhanced API. However the prevalence of multiaddressing dictate their support in lower layers.

3.5. IP Endpoint

A recent approach to multiaddressing defines a new "convergence" layer that exists only in the endpoint systems (hosts) and operates between classic IP and the transport layer. Hence these capabilities are invisible to the IP relaying infrastructure and can be invisible to the transport layer. However they may specify new or modified adjunct infrastructure services, especially to obtain full rendezvous capabilities.

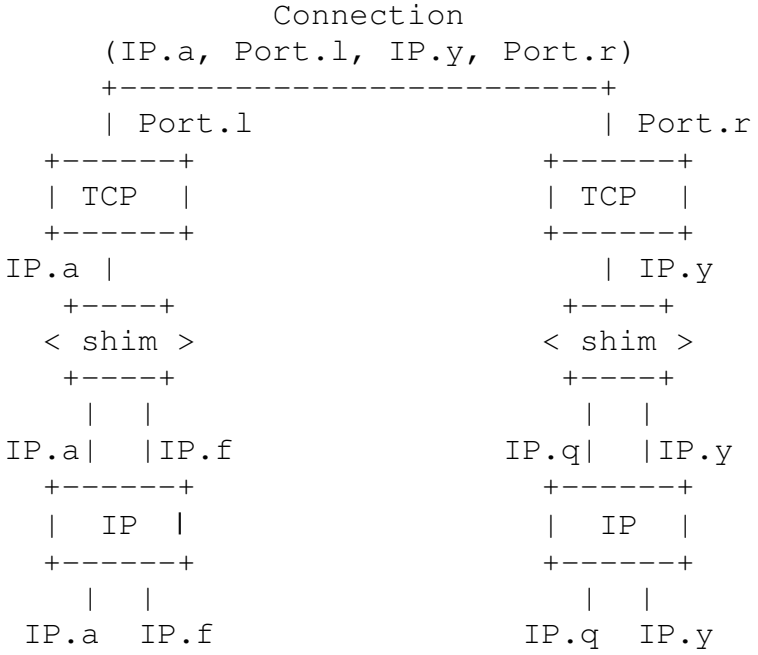
This type of approach can be viewed as using a "shim" or "wedge" partial-layer, between IP and transport, or it can be viewed as partitioning IP, between a lower, relaying module that is common to all IP nodes, versus an upper module that performs IP-related functions specific to endpoints.

The remainder of this sub-section considers these architectural views and then discusses the three IP Endpoint proposals.

3.5.1. Choosing an IP Endpoint Model

3.5.1.1. Shim Model

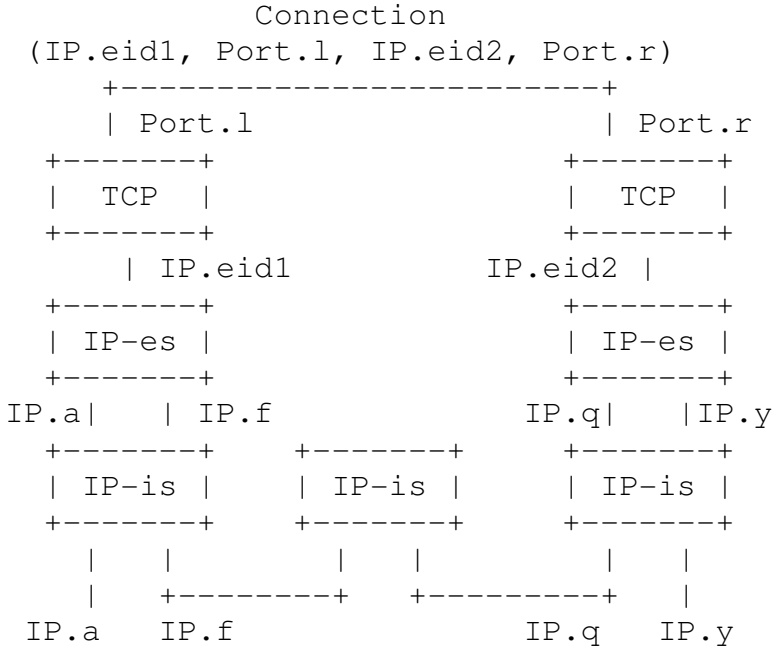
For the Shim, or wedge, approach, a portion of functionality is "intercepted" and modified by the shim module:



3.5.1.2. IP/Transport Convergence Layer Model

Rather than viewing this type of service as being ad hoc, it can be seen as an example of IP-level services that reside only in the end-systems. That is, there is a distinction between the relaying activities in every "intermediate" system (IP-is), versus IP functions that are needed only in the end-systems at the endpoints (IP-es). For multiaddressing, the architectural impact is embodied by using an "endpoint identifier" (EID) in the interface between IP-es and the transport layer, rather than using an endpoint address. Significantly, the EID might be private to the endpoint(s), rather than needing to be globally registered.

IPSec is another example of and IP-es service. Note that this architectural change also must affect the upper-layer access to DNS, since DNS address records must be converted to EIDs.



3.5.2. Host Identity Protocol (HIP)

HIP works with IPv4 and IPv6. Also, it:

* Creates a new, globally unique name space
* Uses strong, cryptographically based protocol details, overloading some HIP functionality with security functionality
* Is tied significantly to [IPSEC]
* Creates a new DNS RR entry
* Requires a Rendezvous server for mobility support
* Requires that NATs be aware of HIP

Many of the HIP features are appealing, such as the cleanliness of the architectural model depicted in Section 4 of the HIP architecture document. Were the Internet stack being created now, HIP well might be an excellent approach. However retrofitting this approach into the existing, deployed Internet entails serious barriers to adoption, such as its dependence on IPSec.

In general, addition of a DNS SRV record can be useful for achieving efficient rendezvous, with or without mobility. It permits participants to know whether a service is supported by its partner, without requiring a probe packet. While beneficial, this enhancement to DNS data structures is not required for multihoming or client (initiator) mobility.

3.5.3. LIN6

LIN6 defines a new, globally unique 64-bit end-point identifier that is used by upper layers, within an IPv6 address format. This is then mapped to one or more IPv6 IP-layer addresses.

The LIN6 specification also provides for the rendezvous function, using DNS for basic name resolution and a separate, dynamically updated service to provide accurate information about rapidly changing addresses.

3.5.4. MAST

MAST is a control protocol for the exchange of IP address notification and authorization, to use additional IP addresses in a given host-pair context.

The primary MAST exchange transmits:

* A list of current IP addresses supported by the sender

Support exchanges:

* Establish a host-pair context
* Establish relevant authentication between the pair

MAST takes a more modest approach than HIP or LIN6. It does not define a new identifier space, has a simpler specification, permits easier implementation and adoption, and works equally with IPv4 and IPv6.

Rendezvous with a mobile target is provided as an adjunct function and relies on domain names and an existing presence service.

MAST differs from the list of HIP requirements in that it:

* Uses a name space that is transient and local to the host-pair
* Uses existing security mechanisms, limited to the sole requirement to prevent association hijacking
* Treats rendezvous as an adjunct requirement and has no special requirements on DNS, in the base service
* Is transparent to NATs

4. SECURITY CONSIDERATIONS

This is a discussion paper and specifies no actions. Hence it has no security impact, except in terms of generally discussing security issues for the IP architecture.

APPENDIX

A. Acknowledgements

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

Commenters on this text include: Marcelo Bagnulo, Iljitsch van Beijnum, Vint Cerf, Spencer Dawkins, Robert Honore, James Kempf , Eugene Kim, Eliot Lear, Pekka Nikander, Erik Nordmark, Tim Shepard, Randall R. Stewart, and Fumio Teraoka

B. References

B.1. Non-Normative

[DCCP] Kohler, E., M. Handley, S. Floyd, J. Padhye, "Datagram Congestion Control Protocol (DCCP)", draft-ietf-dccp-spec-04.txt, 30 June 2003

[DNSDYN] Vixie, P., Thomson, S., Rekhter, Y., Bound, J., Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC2136, April 1997

Wellington , B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000

[EID] Chiappa, J.N., "Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture", <http://users.exis.net/~jnc/tech/endpoints.txt>, 1999

[ETCP] Zhang, B., Zhang, B., Wu, I., "Extended Transmission Control Protocol (ETCP) Project--Extension to TCP for Mobile IP Support", <http://www.cs.ucla.edu/~bzhang/etcp/report.html>

[HIP] Moskowitz, R., "Host Identity Protocol Architecture", < http://www.ietf.org/internet-drafts/draft-moskowitz-hip-arch-03.txt >

Moskowitz, R., "Host Identity Protocol", <ietf-id: draft-moskowitz-hip-07>

Nikander, P., "End-Host Mobility and Multi-Homing with Host Identity Protocol", < http://www.ietf.org/internet-drafts/draft-nikander-hip-mm-00.txt>

[HOWIE] Howie, D., "Consequences of using MIPv6 to Achieve Mobile Ubiquitous Multimedia", http://www.mediateam.oulu.fi/publications/pdf/384.pdf

[IPSEC] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998

[LIN6] Teraoka, F., Ishiyama, M., Kunishi, M., "LIN6: A Solution to Mobility and Multi-Homing in IPv6", draft-teraoka-ipng-lin6-02.txt, 24 June 2003

[NAT] Egevang, K., and P. Francis, "The IP Network Address Translator (NAT)", RFC1631, May 1994

[NSRG] Lear, E., Droms, R., "What's In A Name: Thoughts from the NSRG", draft-irtf-nsrg-report-09.txt, March 2003

[MAST] Crocker, D., "Multiple Address Service for Transport (MAST): An Extended Proposal", draft-crocker-mast-proposal-00.txt, September 13,2003

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

Johnson, D., Perkins, C., Arkko, J., "Mobility Support in IPv6", draft-ietf-mobileip-ipv6-24.txt, June 30, 2003

Bagnulo, M., Garcia-Martinez, A., Soto, I., "Application of the MIPv6 protocol to the multi-homing problem", draft-bagnulo-multi6-mnm-00, February 25, 2003

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

[SCTP] L. Ong, and J. Yoakum "An Introduction to the Stream Control Transmission Protocol (SCTP)", <http://ietf.org/rfc/rfc3286.txt?number=3286>, May 2002

R. Stewart, et al, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", draft-ietf-tsvwg-addip-sctp-07.txt, February 26, 2003

[TCPMH] Matsumoto, A. Kozuka, M., Fujikawa, K., Okabe, Y., "TCP Multi-Home Options", draft-arifumi-tcp-mh-00.txt, 10 Sep 2003

[TLS] Dierks, T., C. Allen , "The TLS Protocol Version 1.0", RFC 2246, January 1999.

C. Author's Adress

Dave Crocker
Brandenburg InternetWorking
675 Spruce Drive
Sunnyvale, CA 94086 USA

tel: +1.408.246.8253
dcrocker@brandenburg.com

D. Full Copyright Statement