The ISP Column
An occasional column on things Internet

Multi-Homing and Identity in IPv6

June 2004

The previous column has described the motivations for multi-homing and the ways in which multi-homing is implemented at present. The current solution imposes an incremental load on the routing system. The objective of the IETF Multi6 Working Group is to explore approaches to multi-homing in IPv6 that do not impose such a routing burden.

The Multi-Homed Scenario

The simplest formulation of the multi-homing environment is indicated in Figure 1.

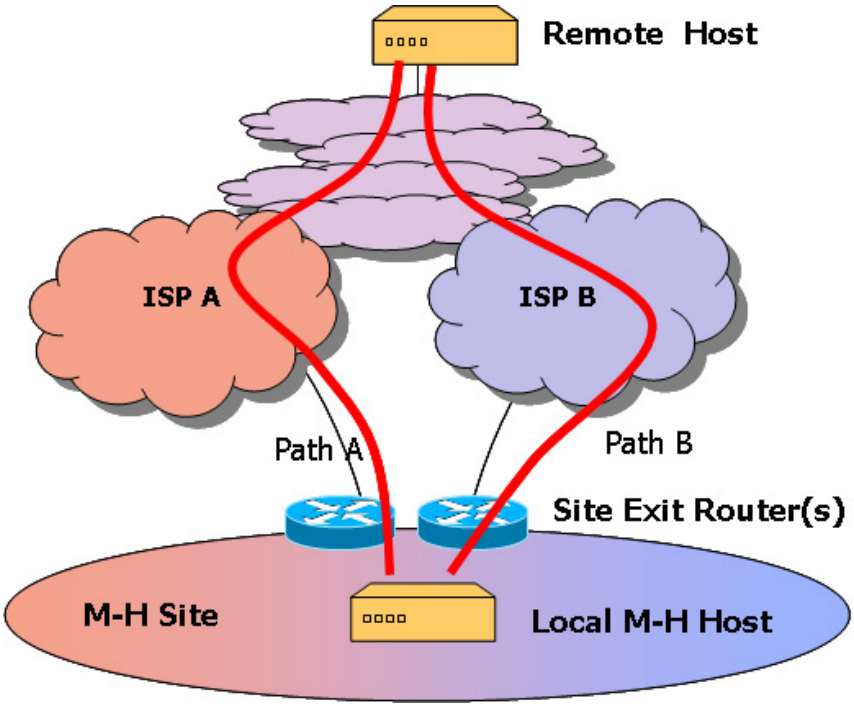

Figure 1: The Multi-Homed Domain

This figure shows a multi-homed site connected to two upstream service providers, ISP A and ISP B. To illustrate some of the features of various approaches to this topic some further detail is shown for the multi-homed site. This detail includes the edge routers that connect the site to the service providers, and a host within the site. Also a remote host is shown, that may or may not be multi-homed itself.

So what we want is to allow the local host to exchange IP packets with the remote host, such that this exchange of packets is to be seamlessly supported across dynamic changes in connectivity. So if the local host starts a session with the remote host over Path A, and this path breaks, then the session should be able to immediately use Path B without any disruption, and without any additional form of session re-initiation. In other words, the application session should not be required to be explicitly aware of underlying path changes at the level of packet forwarding paths chosen by the network.

This simple scenario is intended to illustrate the basic multi-homing environment. Variations of this scenario include additional external providers of transit connectivity to the local site, complex site requirements and constraints, where the site may not interface uniformly to all external transit providers and may want to spread traffic across all connections in the normal operational state, sequential rather than simultaneous external transit reachability, where external connections may not all be available at any time, communication with remote multi-homed hosts, multi-way communications of more than two parties, use of third party referrals and the imposition of policy constraints on path selection. However, the basic scenario is sufficient to illustrate the major architectural aspects of support for multi-homing.

The major characteristic of this scenario is that the address space used by, and advertised as reachable by, ISP A is distinct from the address space used by ISP B.

Requirements and Considerations

The Multi6 Working Group has studied aspects of this problem, and has published RFC 3582, a summary of some of the requirements that a multi-homing approach should attempt to address. These requirements include:

- redundancy
- load sharing
- traffic engineering
- policy constraints
- simplicity of approach
- transport-layer survivability
- DNS compatibility
- packet filtering capability
- scaleability
- legacy compatibility

That's a long list, and a pretty tough one if you want to meet all requirements simultaneously. There is also work in progress in looking at additional considerations, including the interaction with routing, changes to packets and protocol behavior, and issues relating to a possible split between identity and locator specification.

The major constraint here is that these approaches are attempting to avoid placing incremental load on the routing system. This implies that the simple approach, described in the previous column, of the multi-homed site announcing an address prefix to all upstream providers simultaneously, and letting the routing system mend any changes in connectivity is not an option here.

This implies that the site can only make use of addresses provided by its upstream ISPs, and cannot 'cross-announce'. By this it is meant that the site cannot announce the address prefix it obtained from ISP A to ISP B, and vice versa.

This is starting to get intriguing. Under these constraints the remote host, R, when it wants to send a packet to the multi-homed host via path A must use an address for the host drawn from ISP A's aggregate address block, and when it wants to send a packet via path B, then the destination address in the packet header is drawn from the ISP B's address block. How can host R know that the two addresses actually refer to the same remote host? How can R know when to switch from one address to the other? How can these in-session address changes be supported in a manner that prevents session hijacking by various forms of man-in-the-middle attacks?

It seems that as we attempt to provide solutions to these questions we have to look at something more fundamental in networking, that of identity and location.

Identity and Location

In any technology development exercise, the design process appears to require some level of compromise between competing requirements, and the decision process often has to make some assumptions about the environment in which the technology will be deployed. In some cases these assumptions may be minor in nature, while in other cases they have far-reaching implications.

In the case of IP it appears that one of the assumptions in the original design was that of a relatively static network topology with host systems that were located in a fixed point within this topology. The design trade-off was that of combining an object's identity and its location into a single protocol element, the address. The alternative, that of a split between identity and location with its overhead of an additional layer of indirection in resolving identity into a location did not appear to be warranted. So in the IP architecture your identity is your address is your location.

It is a matter of longstanding study that continues today as to the merits of delineating these two roles of identity at the IP level, creating one identity realm as a means of uniquely identifying an instance of a protocol stack within an end device (variously called a "stack identifier" or "endpoint identifier" in previous studies) and a second identity realm that is used to identify the current location of the identity element within the network (typically called a "locator" identity)

> See Saltzer's write up of this in RFC 1498, *On the Naming and Binding of Network Destinations*, and Schoch's paper in the Proceedings of the 17th IEEE Computer Society International Conference of December 1978, *Internetwork Naming, Addressing, and Routing*.

What we appear to be asking for in the example above is for the remote host to have a constant concept of the identity of the multi-homed host throughout its session, and to be able to use a number of different locators to address packets to that host, responding to changes in connectivity as required. In other words we appear to be looking at making a quite fundamental change to the IPv6 protocol architecture by adding a capability to split apart the concepts of identity and location, and manage each separately within a session.

The implication here is that there is an additional layer of indirection in the protocol behavior. While the upper layers of the protocol stack need to maintain a peering based on some concept of an identity that is persistent across at least the session (and possibly longer), the lower levels need to be able to direct packets to each other based on a locator value. This implies that somewhere within the protocol stack there is a need to map between identities and locators.

If multiple forwarding paths are to be supported for a single transport session, and path selection is to be decoupled from the functions of transport session initiation and maintenance, then the corollary of this requirement in architectural terms appears to be that some changes are required in the protocol architecture

to decouple the concepts of identification of the endpoint and identification of the location and Associated path selection for the endpoint. This change in the protocol architecture would permit a transport session to use an invariant endpoint identity value to initiate and maintain a session, while allowing the forwarding layer to dynamically change paths and associated endpoint locator identities without impacting on the operation of the session, nor would such a decoupled concept of identities and locators add any incremental load to the routing system.

There appears to be three generic forms of architectural approaches to this problem, namely:

New Protocol Element:
: Insertion of a new element in the protocol stack that manages a persistent identity for the session.

New Transport Protocol:
: Specify a new transport protocol that includes the functionality of an interface to the upper level protocol of an identity value, while allowing the interface to the IP layer to use addresses in the context of locator values.

Modify a Protocol Element:
: Modify the Transport or IP protocol stack element in the host in order to support dynamic forwarding locator change

Locator Considerations

Within the conventional IP architecture, if a host is reachable through multiple network paths, there is still a single address for the host, and the selection of viable forward and reverse paths is the task of the routing system.

In a split identity / locator architecture where multi-homing is expressed by the presence of multiple locators for a single identity routing will find a 'best' set of paths for each locator pair, but it cannot resolve a 'best', or even a viable path pair when there is a choice between multiple locators.

If remote host R is to initiate a communication with the local multi-homed host, it would normally query the DNS for the locator for the local host. In this context the DNS would return 2 locators (One using the A prefix and the other using the B prefix). The remote host would select one of these locators and send a packet to this destination locator. This would direct the packet to the local host along path A or B, depending on the selected locator. If the path between the local site and the transit provider fails, then the address prefix announced by the transit provider to the inter-domain routing system will continue to be the provider's address prefix. The remote host will not see any change in routing, yet packets sent to the local host will now fail to be delivered. The question posed by the multi-homing problem is: "If the remote host is aware of the multi-homing environment, how could it switch over to using the equivalent locator for the local multi-homed host that implicitly causes a path change?"

If the local multi-homed host wishes to initiate a session with remote host R, it needs to send a packet to R with a valid source and destination locator. While the destination locator is that of R, what source locator should the local host use? There are two implications for this choice. Firstly the remote host will, by default, use this source locator as the destination locator in its response, and hence this choice of source locator will direct the reverse path from R to the local host. Secondly, the ISPs A and B may be using reverse unicast address filtering on source addresses of packets passed to the ISP, as a means of prevention of source address spoofing. This implies that if the multi-homed host selects a source locator from address prefix A, and the local routing to R selects a best path via ISP B, then ISP B's ingress filters will discard the packet.

Within this addressing structure there is no form of routing-based repair of certain network failures. If the link between the local site and ISP A fails, there is no change in the route advertisements made by ISP A to its external routing peers. Even though the multi-homed site continues to be reachable via ISP B, packets directed to the site using ISP A's prefix will be discarded by ISP A as the destination is unreachable. The implication here is that if the local host wishes to maintain a session across such events it needs to communicate to remote host R that it is possible to switch to using a destination address for the multi-homed host that is based on ISP B' address prefix.

In an aggregated routing environment multiple transit paths to a host imply multiple locator prefixes for the host, where each possible transit path is identified by a specific locator value for the host. The implication of this constraint on multi-homing is that paths being passed to the local multi-homed site via transit provider ISP A must use a forwarding-level destination IP locator drawn from ISP A's advertised locator prefix set that maps to the multi-homed host. Equally, packets being passed via the transit of ISP B must use a destination locator drawn from ISP B's locator prefix set. The further implication here is that path selection (ISP A vs ISP B transit for incoming packets) is an outcome of the process of selecting a locator for the destination host.

A new Identity Protocol Element

One approach to this objective is to add a new element into the model of the protocol stack.

The presentation to the upper level protocol stack element (ULP) would use endpoint identifiers to uniquely identify both the local stack and the remote stack. This is intended to provide the ULP with stable identifiers for the duration of the ULP session.

This stack element would be responsible for changing locators in the event of path failure, as well as allowing additional locators to be added, or locators to be removed in the course of a session. All this would be in the context of a session, as defined by a peering of identities.

The most logical place to insert this additional protocol stack element is between the transport and internet protocol stack elements, so that the transport layer would function using endpoint identifiers, and maintain a coherent transport session using these endpoint identifiers. The IP or internetwork layer would function using locators, and the mapping from endpoint identifier to locator is undertaken within the new stack element. This is shownm in Figure 2.

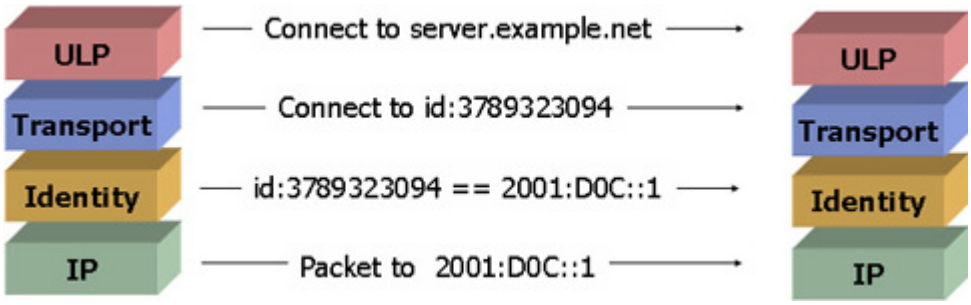

Figure 2: Identity Protocol Element

Its not quite as simple as its sounds, as there is now a need for additional signalling within the protocol stack, including explicit signalling of session start and closure from the transport layer into the identity element, as well as relevant ICMP-based signals from the IP layer. The advantage of this type of approach is that the incremental change to the protocol is relatively minor, and in terms of implementation there is the potential to insert this element into a code base with relatively small impact on the remainder of the implementation.

Modify a Protocol Element to include Identity

As an alternative to insertion of a new protocol stack element into the protocol architecture, an alternative approach is to modify an existing protocol stack element to include identity functionality. This modification could be undertaken within the transport protocol stack element, or within the internetworking stack element. The functional outcome from these modifications would be to create a mechanism to support the use of multiple locators within the context of a single endpoint-to-endpoint session.

Within the transport layer, this functionality can be achieved, for example, by the binding of a set of locators to a single session, and then communicating this locator set to the remote transport entity. This would allow the local transport entity to switch the mapping to a different locator for either the local endpoint or the remote endpoint while maintaining the integrity of the ULP session.

Within the IP level this functionality could be supported by a form of dynamic rewriting of the packet header as it is processed by the protocol element. Incoming packets with the source and destination locators in the packet header are mapped to packets with the equivalent endpoint identifiers in both fields before being passed to the transport layer. The reverse mapping is performed to outgoing packets passed from the transport layer to the IP layer. Mechanisms that support direct rewriting of the packet header are potential candidates in this approach, as are various forms of packet header transformations of encapsulation, where the original endpoint identifier packet header is preserved in the packet and an outer level locator packet header is wrapped around the packet as it is passed through the internetworking protocol stack element.

In these scenarios, there are common issues of what state is kept, by which part of the protocol stack, how state is maintained with dynamic additions and removals of locator bindings, and does only one piece of code have to be aware of the endpoint / locator split or do multiple protocol elements have to be modified? For example, if the functionality is added at the internetworking (IP) layer, there is no context of an active transport session, so that removal of identity / locator state information for terminated sessions needs to be triggered by some additional mechanism from the transport layer to the internetworking ayer.

Approaches to Endpoint Identity

Both of the above mechanisms assume some form of exchange of information that allows the parties to the communication to be aware of the remote endpoint identity and the associated mapping to locators. There are a number of choices in terms of the way in which this information exchange can be implemented.

The first such possible approach is termed here a 'conventional' approach, where the mode of operation is in terms of encapsulating the protocol data unit (PDU) passed from the ULP with additional data elements that specifically refer to the function of the endpoint identity protocol stack element. The compound data element is passed to the LLP as its PDU. The corresponding actions on receipt of a PDU from a LLP is to extract the fields of the data unit that correspond to the identity function, and pass the reminder of the PSU to the ULP. The identity protocol operates in an "in-band" mode, communicating with its remote peer entity through additional information wrapped around the ULP PDU.

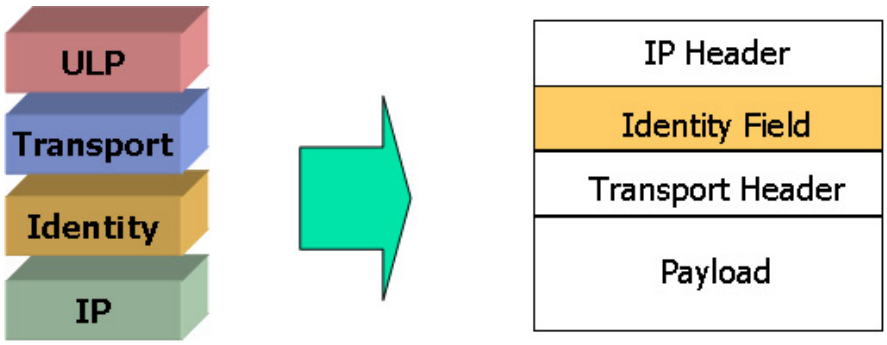

Figure 3: Identity Protocol through Encapsulation

Another approach is to allow the identity stack element to communicate using a separate communications channel, where the identity element generates dedicated messages that are directed to its peer, and passesthese PDUs to the LLP independently of the PDUs that are passed to the identity protocol element from the ULP. This allows the identity element to exchange information and synchronize state with the remote identity element semi-independently of the ULP protocol exchange. As a part of the identity function is to transform the ULP PDU to include locator information, there is an associated requirement to ensure that the identity peering state remains synchronized to the exchange of ULP PDUs, so that the remote identity element can correctly recognize the locator to endpoint mapping for each active session.

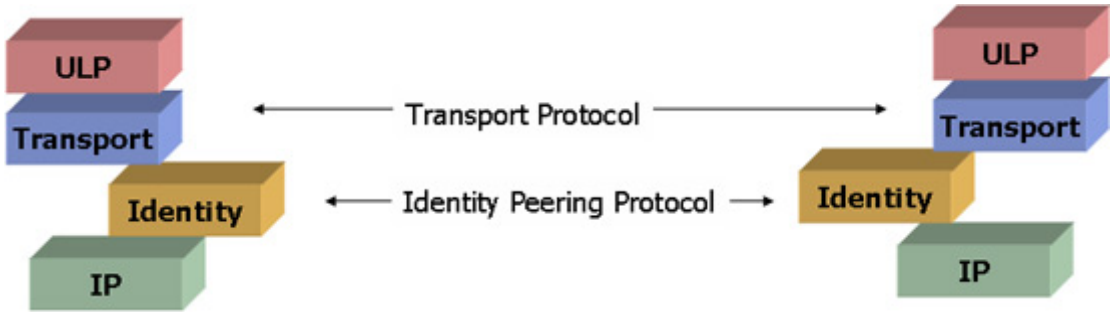

Figure 4: Identity Protocol as an out-of-band protocol

Another potential approach here is to allow the endpoint to locator mappings to be held at a third party point. This model is already used for supporting the name to IP address mappings performed by the Domain Name system, where the mapping is obtained by reference to a third party, namely a DNS resolver. A similar form of third party mapping between endpoints and a locator set could be supported through the use of the DNS, or a similar third party referential mechanism. Rather than have each party exchange endpoint to locator mappings, this approach would see this mapping being obtained as a result of a lookup for a DNS Endpoint to Locator set map contained as DNS Resource Records, for example.

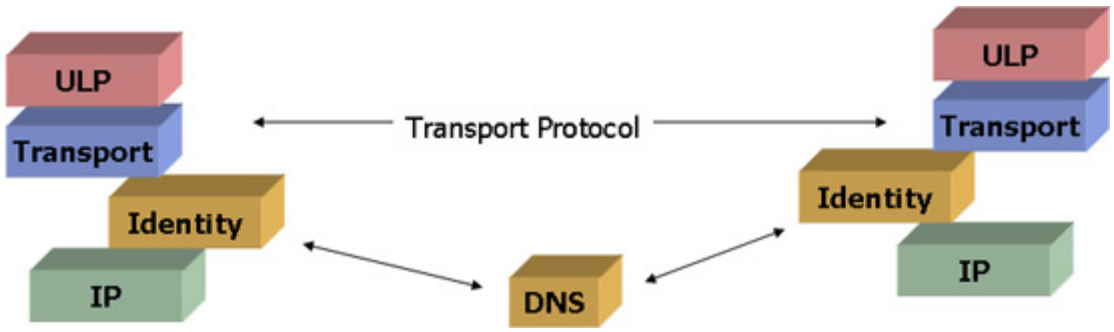

Figure 5: Identity Resolution by 3rd party reference

Endpoint Identity Structure

So far we've used the term "endpoint identity" without examining what form this identity may take. There are a number of considerations regarding the structure and form of this identity.

One possible form of an identity is the use of identity tokens lifted from the underlying protocol's "address space". In other words an endpoint identity is a special case instance of an IPv6 protocol address. There are a number of advantages in using this form of endpoint identity, not the least of which is minimal change, as the suite of IP protocols and associated applications already manipulate IP addresses. The essential difference in a domain that distinguishes between endpoint identity and locator is that the endpoint identity parts of the protocol would operate only on those addresses that assume the role of endpoint identities, and the mapping function would undertake a mapping from an endpoint "address" to a set of potential locator "addresses", and also undertake a reverse mapping from a locator "address" to the distinguished endpoint identifier "address". The address space is hierarchically structured, permitting a suitably efficient mapping to be performed in both directions, and the underlying semantics of addresses in the context of public networking includes the necessary considerations of global uniqueness of endpoint identity token values.

In this approach the endpoint identifier be a valid locator. This would imply the existence of a 'distinguished' or 'home' locator, and other locators could be dynamically mapped to this initial locator peering as required. The drawback of this approach is that the endpoint identifier is now based on one of the transit provider's address prefixes, and a change of transit provider would necessarily require a change of endpoint identifier values within the multi-homed site. An alternative approach for address-formatted identifiers is to use address values which are not part of the global unicast locator space, allowing applications and protocol elements to distinguish between endpoint identity values and locators based on address prefix value. It is also possible to allow the endpoint identity and locator space to overlap, and distinguish between the two identity realms by the context of usage rather than by a prefix comparison.

Alternatively, it is also feasible to use the fully qualified domain name (FQDN) as an endpoint identity, undertaking a similar mapping as described above, using the FQDN as the lookup "key". The implication here is that there is no default 'address' that is to be associated with the endpoint identifier. The syntactic properties of these two different identity realms have obvious considerations in terms of the manner in which these identities may be used within PDUs, and the additional reliance on the DNS does bring up the topic of ensuring that there are no circular dependencies. If starting a multi-homed session involves resolving a DNS name, which in turn involves a multi-homed session initiation which in turn involves a DNS name resolution, and so on, then we are well on the way to such a dependency.

It is also an option to consider a new structured identity space which is not generated through the reuse of IPv6 address values nor drawn from the FQDN. Given that the address space would need to be structured in

such a fashion that permits it to be used as a lookup key to obtain the corresponding locator set, the obvious question in such an option is what additional or altered characteristics would be used in such an endpoint identity space that would distinguish it from either of the above approaches?

Instead of structured tokens that double as lookup keys to obtain mappings from endpoint identities to locator sets, the alternative is to use an unstructured token space, where individual token values are drawn opportunistically for use within a multi-homed session context. Here the semantics of the endpoint identity are subtly changed. The endpoint identity is not a persistent alias or reference to the identity of the endpoint, but a means to allow an endpoint identity protocol stack element to confirm that two locators are part of the same mapped locator set for an endpoint. In this context the unstructured opportunistic endpoint identifier values are used in determining locator equivalence rather than in some form of lookup function.

This may well be the critical decision point for this work, whether to use structured identity values that have some level of existence and validity beyond the lifetime of an individual session. I'm struck by the difference between 'absolute' and 'relative' uniqueness. With absolute uniqueness the implication is that it can support identity persistence, which in turn allows for the use of identity as a third party reference and it also implies resolution from an identity to locator sets. If a protocol stack is seeded with an identity value of a remote stack then it can initiate a packet exchange by resolving the identity into corresponding locators. With relative uniqueness the identity uniqueness is constrained temporally to the lifetime of a session (or group of sessions) and constrained in terms of a spatial realm to the communicating parties. The possibility of identity collision is admitted in a broader context. Such identity tokens can be used within the context of a session to determine if a locator is part of an existing session, but cannot be used to start a session as there is no coherent resolution operation, nor can they be used as a reference to a third party.

This classification also has some implications regarding the cost of uniqueness and the relative level of overhead in managing a long-lived unique token space and opportunistic, and probably self-generated, identity token values that have relative uniqueness..

I'm not sure I'm truly in a position to offer comment from my perspective as to which is 'better' than the other - there are aspects of value in persistence, resolvability and referential capability in structured unique identity spaces, and there are aspects of value in terms of lower overhead and higher efficiency in terms of protocol behavior and dependencies as well as reduced dependence on infrastructure activities that paint a value picture for opportunistic identity spaces.

It appears to be a Multi6 Working Group agenda item to come to some closure at some point as to where the best design point might lie in the context of multi-homing, and in the context of an evolving IPv6 protocol architecture that supports this identity / locator split, but its way to early to call the question.

Common Issues for Multi-Homing Approaches

The above overview encompasses a very wide range of potential approaches to multi-homing, and each particular approach necessarily has an associated set of considerations regarding its applicability.

There are, however, a set of considerations that appear to be common across all approaches, and they are examined in further detail in this section.

Triggering Locator Switches

Ultimately, regardless of the method of generation, a packet generated from a local multi-homed host to a remote host must have a source locator in the IP packet that is passed into the transit network. In a multi-homed situation the local multi-homed host has a number of self-referential locators that are equivalent aliases

in almost every respect. The difference between locators is the inference that at the remote end the choice of locator may determine the path used to send a packet back to the local multi-homed host. The issue here is how does the local host make a selection of the "best" source locator to use? Obviously the parameters of this selection include the objective to select a locator that represents a currently viable path from the remote host to the local multi-homed host. Local routing information for the multi-homed host does not include this reverse path information. Equally, the local host does not necessarily know of any additional policy constraints that apply to the remote host that may result in a remote host's preference to use one locator over another for the local host. Considerations of unicast reverse path forwarding filters also indicate that the selection of a source locator should result in the packet being passed to a site-exit router that is connected to the associated ISP transit provider, and that the site-exit router passes the packet to the associated ISP.

If the local multi-homed host is communicating with a remote multi-homed host, the local host may have some discretion in the choice of a destination locator. The considerations relating to the selection of a destination locator include considerations of local routing state (to ensure that the chosen destination locator reflects a viable path to the remote endpoint), policy constraints that may determine a "best" path to the remote endpoint. In such situations it may also be the case that the source address selection should also be considered in relation to the destination locator selection.

Another common issue is the consideration of the point when a locator is not considered to be viable, and the consequences to the transport session state.

A change in state for a currently used path to another path could be triggered by indications of packet loss along the current path, or by transport session timeouts, assuming an internal signalling mechanism between the transport stack element and the locator pool management stack element.

Alternatively, in the absence of local transport triggers, the site exit router could communicate failure of the outbound forwarding path in the case where the remote host is multi-homed with an associated locator set. Conventional routing would be incapable of detecting a failure in the inbound forwarding path, so there are some limitations in the approach of using routing triggers to change locator bindings.

An alternative to these approaches is the use of a session heartbeat protocol, where failure of the heartbeat would cause the session to seek a new locator binding that would re-establish the heartbeat.

The sensitivity of the locator-switch trigger is a consideration here. A very fine-grained sensitivity of the locator switch trigger may generate false triggers arising from short-term transient path congestion, while coarse-grained triggers may impose an undue performance penalty on the session due to an extended time to detect a path failure.

Session Startup and Maintenance

The next issue if that of the difference between the initial session startup mode of operation and the maintenance of the session state.

In a split endpoint identifier / locator environment there needs to be at least one initial locator associated with an endpoint identifier in order to establish an initial connection between the two hosts. This locator could be loaded into the DNS in a conventional fashion, or, if the endpoint identifier is a distinguished address value, the initial communication could be established using the endpoint identifier in the role of a locator (i.e. using this as a conventional address).

The initial actions in establishing a session would be similar. If the session is based on specification of a FQDN, the FQDN is first mapped to an endpoint identity value, and this endpoint identity value could then be mapped to a locator set. The locators in this set are then candidate locators for use in establishing an initial synchronized

state between the two hosts. Once the state is established it is then possible to update the initial locator set with the current set of useable locators. This update could be part of the initial synchronization actions, or deferred until required.

This leads to the concept of the use of a 'distinguished' locator that acts as the endpoint identifier, and a pool of alternative locators that are associated with this 'home' locator. This association may be statically defined, using referential pointers in a third party referral structure (such as the DNS), or dynamically added to the session through the actions of the endpoint identity protocol stack element, or both.

Security

And of course there is the ever-present issue of security. Switching locators for an active session is another name for 'session hijacking' when its done without the authorization of the original party, and there's no doubt that any additional level of indirection in a protocol stack represents another point of vulnerability.

One approach, as represented by opportunistic identities attempts to limit the realm of validity of the identity to a small scope, and even to obtain this identity through a cryptographic transformation that involves a host's private key value. As shown in the work on the 'Host Identity Protocol' this can create a reasonably robust environment that appears to fend off a number of attack vectors.

On the other hand persistent identities appear to be very tempting. They allow direct addressing of the protocol stack element without reference to its current location, they can support use through reference and can allow multi-party communications to be supported with some ease. But they pose a problem in terms of securing the identity to locator resolution, as if an attacker can disrupt or corrupt this mapping then session hijacking is an immediate consequence.

Current Status

It appears that is attempting to solve a simple problem of multi-homing in a scaleable way that does not break the routing system we've managed to re-expose one of the original design choices of the IP architecture, namely the implicit binding of identity and location.

In the IETF's Multi6 working group there are a relatively large number of proposals that attempt, one way or the other, to split these two concepts, and allow the binding between identity and location to be as dynamic operation.

Whatever approach this working group ultimately decides upon, it will have far-reaching implications for the utility of IP in the future. Persistent identity approaches will allow not only for various forms of multi-homing, but will also allow for a re-thinking of mobility, auto-configuration, service negotiation, referential persistence, multi-party communications and of course a re-thinking of security. Opportunistic relative identity is a much smaller step to take, but the lingering doubt remains that it may not be a big enough step in the long term, and we may need to retrace these steps looking at the problem in terms of persistent identity if we first opt to use an opportunistic locally scoped identity in the first place.

This may turn out to be a far tougher problem than grappling with QoS in IP!

Geoff Huston

Disclaimer

The above views do not represent the views of the Internet Society, nor do they represent the views of the author's employer, the Telstra Corporation. They were possibly the opinions of the author at the time of writing this article, but things always change, including the author's opinions!

About the Author

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also the Executive Director of the Internet Architecture Board, and is a member of the APNIC Executive Committee. He was an inaugural Trustee of the Internet Society, and served as Secretary of the Board of Trustees from 1993 until 2001, with a term of service as chair of the Board of Trustees in 1999 and 2000. He is author of a number of Internet-related books.