Internet Engineering Task Force
INTERNET DRAFT
Expires in December 1999

Francis Dupont
GIE DYADE

June 25. 1999

Multihomed routing domain issues for IPv6 aggregatable scheme

<draft-ietf-ngtrans-6bone-multi-01.txt>

Status of this Memo

Abstract

This document exposes some issues for multihomed routing domains using the aggregatable addressing and routing scheme. A routing domain is multihomed when it uses two or more providers of the upper level. Most of these issues are not specific to IPv6 but are consequences of the addressing and routing scheme.

1. Introduction

The aggregatable addressing and routing scheme [AGGR] defines an IPv6 aggregatable global unicast address format for use in the Internet and the associated routing.

The address assignment and allocation mechanism is fully hierarchical, a prefix of a given level (ie. of a given length) denotes all the destinations in the prefix ie. aggregates them. The customers of an Internet service provider are in its prefix (as a consequence a multihomed routing domain has several prefixes).

The routing is standard datagram routing, hop by hop, on destination address only (as in IPv4). But it is a prefix routing, ie. forwarding decisions are based on a "longest prefix match" algorithm on arbitrary bit boundaries without any knowledge of the internal structure of addresses.

When there are two routes for the same prefix with the same length then the best is caught for the inter-domain routing protocol [BGP]:

- o policy rules;
- o shortest path, the path being the list of routing domains to cross;
- o protocol metric.

The aggregation idea is the bet that in most of the cases a single-homed Internet service provider at a given level should know (ie. has routes to) only:

- o its upper provider (ie. a shorter prefix, used as a default) if it is not a top-level provider;
- o its customers (ie. longer routes in its prefix);
- o some routes to other customers of its upper provider (ie. sibling prefixes, at the same level).

With addresses this gives (with P1:P2/x for the concatenation of prefixes P1 and P2 with the length x):

- o T/t for the upper provider;
- o T:P/t+p for the provider itself;
- o T:P1/t+p1, T:P2/t+p2, ..., T:Pn/t+pn for siblings;
- o T:P:C1/t+p+c1, T:P:C2/t+p+c2, ..., T:P:Cn/t+p+cn for customers.

The routing information for siblings is only needed for top-level providers. For an other provider it is only an optimization (ie. a backdoor) because any destination, including sibling, not in its own prefix, is reachable through the upper provider.

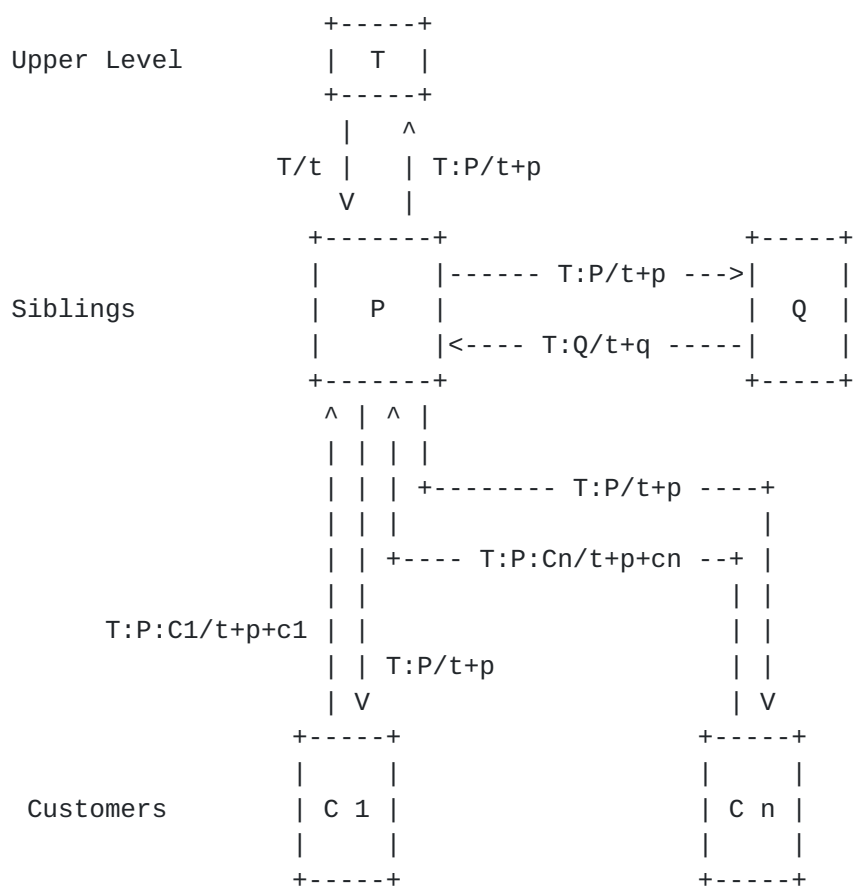
Usual routing exchanges for P at prefix T:P/t+p are:

- o from the upper provider the route to T/t which can be used as a default (ie. <>/0);
- o from a customer the route to T:P:C/t+p+c;

o from a sibling the route to T:Q/t+q;

o to anybody the route for T:P/t+p (and nothing else).

The scheme is with arrows for route (and traffic) exchange:



The aggregation is shown by the fact one announces only the route to its own "aggregated" prefix and masks routes to longer prefixes. Upper levels should not know the details of lower levels, this transparency property should be kept.

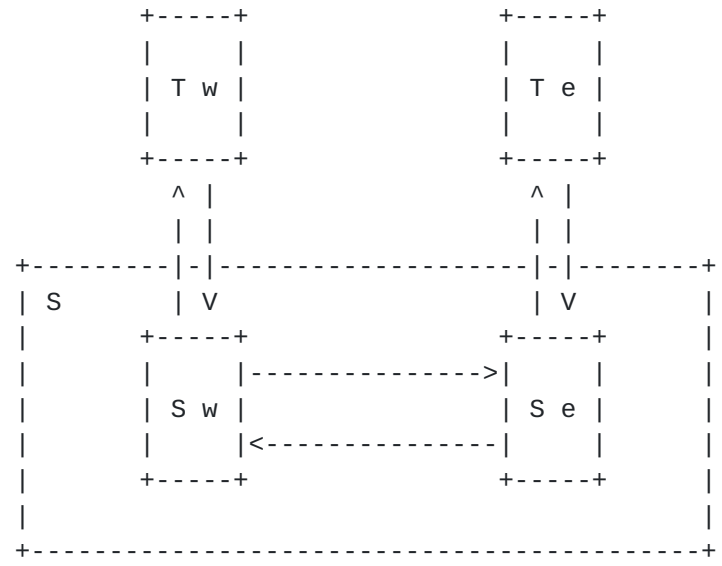
A top-level provider has no upper provider (ie. no default) and must exchange routes with all the other top-level providers (ie. full routing with its siblings is mandatory). In order to avoid routing table explosion, the length of top-level prefixes is bounded (therefore the number of top-level providers is bounded too).

2. Multihomed Routing Domains

A multihomed routing domain has more than one provider then it has more than one prefix (usually a prefix per provider).

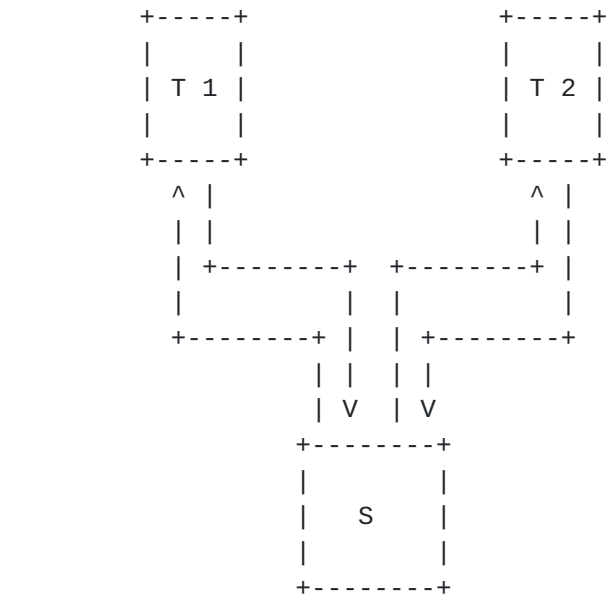
There are several reasons to be multihomed:

- o the "two coasts" case where the routing domain is split into sub-domains in different locations, each domain using a local provider:



But in fact this comes down to two routing domains with a backdoor between them. The extra routes can be hidden and there is no further matter.

- o reliable service: to be able to use another provider in case of a connectivity problem. Of course the purpose is to limit trouble to the only case when all the providers fail (and NOT when at least one fails!).

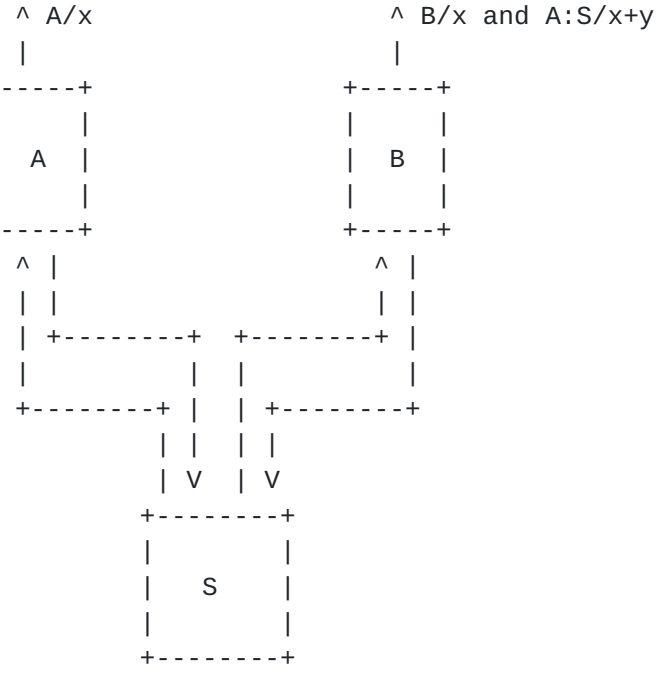


A given host of a such routing domain may (and should if reliable connectivity is needed) have two different addresses, one for each prefix (T1:S1:H in T1:S1/t1+s1 and T2:S2:H in T2:S2/t2+s2).

This document mainly covers this case.

3. The Transparency Issue

If a domain prefix is announced at an upper level, it has to be announced to this whole level.

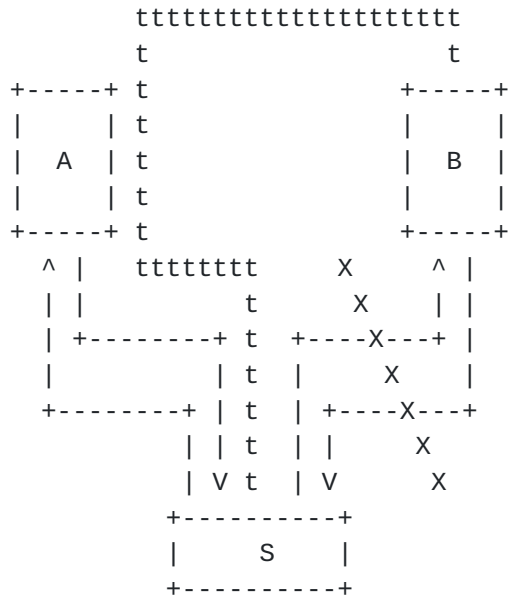


If the provider B tries to announce the prefix A:S/x+y in order to be able to route the traffic for S with both prefixes A:S/x+y and B:S/x+y then B will catch the whole traffic for S because the prefix A:S/x+y is longer than the prefix A/x (x+y > x) so it is a better match...

In this case the only solution is that both A and B announce routes to prefixes A:S/x+y and B:S/x+y which breaks the transparency property and obviously does not scale.

The [MULT] document proposes to announce the prefix A:S/x+y by B only when the path through A (then announces by A) is not available. This makes transparency problems less important but a route for a long prefix is liable to filtering or flap damping mechanisms and should be avoid.

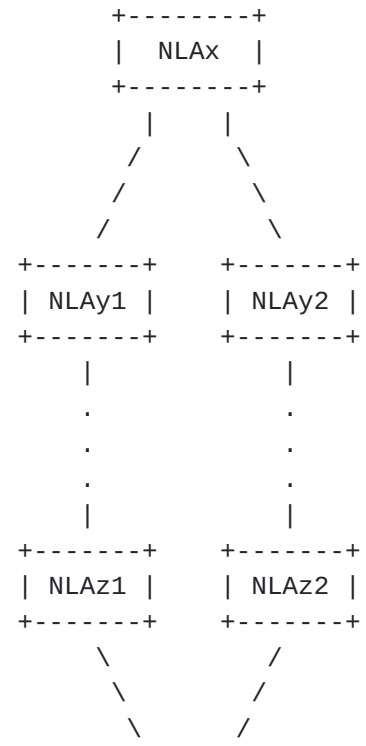
A second solution proposed by [MULT] is to use tunnels in order to keep connectivity even a path is not available:

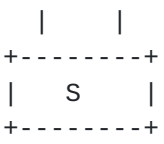


This uses a hairy configuration of EBGP and is limited by the tunnel technology. We shall try to explore other kinds of solutions.

4. Upper Level Routing

At upper levels the structure looks like:




S

For an optimal routing S should have routes for any NLAi1 or NLAi2 up to NLAx, the first common upper provider. For destinations outside the diagram any provider (NLAz1 or NLAz2) can be used, usually the choice of the provider is managed by internal policy rules.

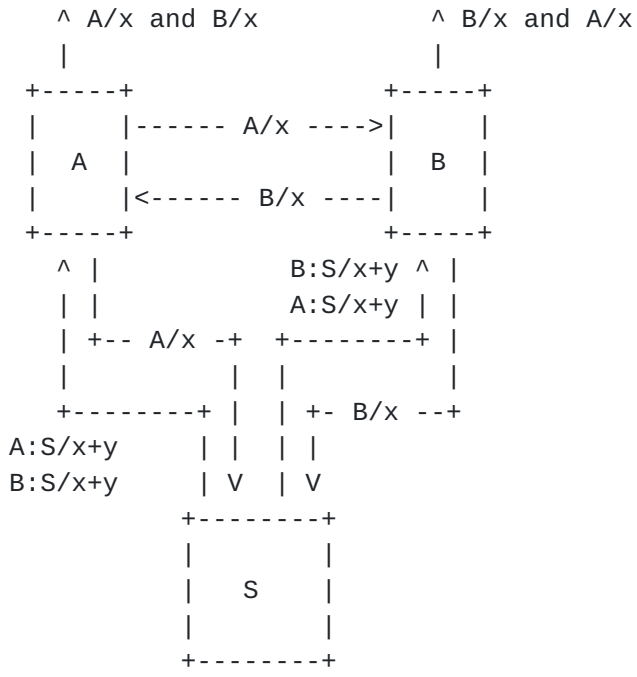
The source address selection for S nodes should be coherent with the upper level routing and the policy in order to avoid asymmetrical routing. There is some proposals [SRCA] for source address selection (and the dual problem, destination address selection) but a selection service should:

- o be synchronized with (external) routing, ie. there should be an interaction between border routers and the service;
- o be used by applications which can have more information;
- o be used as the same time than DNS resolution which makes the destination address selection easy to intergrate in the service, ie. the list of addresses returned by the resolver can be converted in a partial ordered list of source / destination address pairs.

The address selection problem should be addressed in other documents.

5. Mutual Backup

There is a case where the transparency property is kept, routing is as reliable as possible and is optimal in almost all the cases.



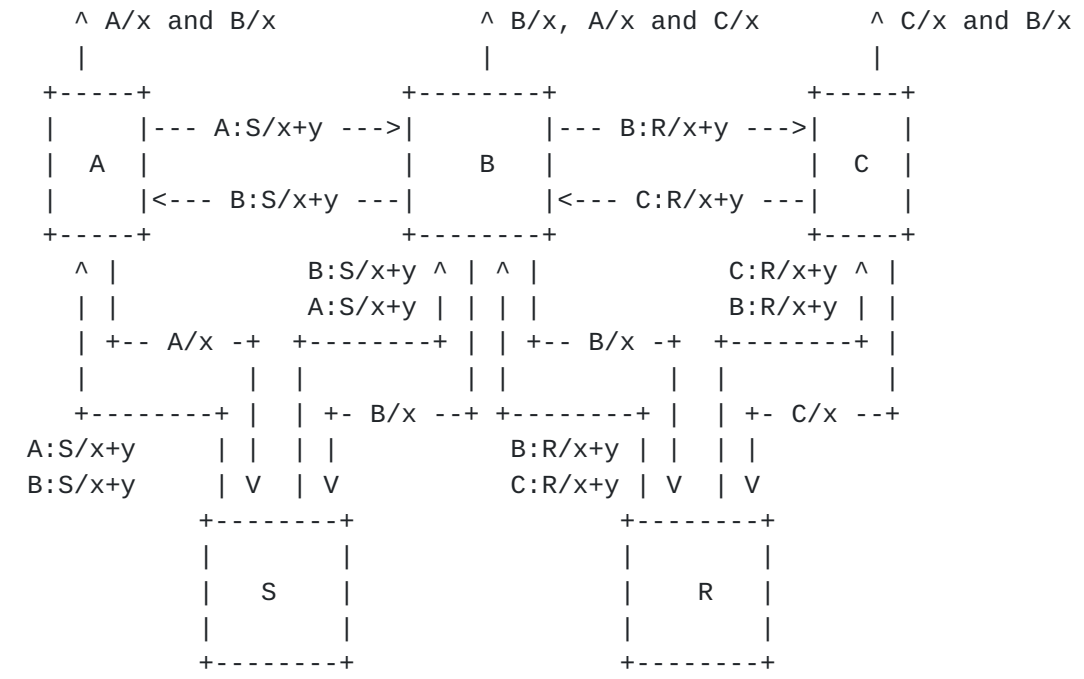
For a provider T in an upper level or the same one than providers A and B, routes for the prefix A/x are not equivalent because the prefix A/x announced by A is direct (one element (A) in the path) and the prefix A/x announced by B is indirect (two elements (B and A) in the path). Then traffic for A will go to A directly. The same thing applies for B.

The prefix A:S/x+y is longer (ie. better) than the prefix A/x then for A the whole traffic for S will go directly, same for B.

If the path through A is not available then the whole traffic for S, including the one to or from addresses in the prefix A:S/x+y will go through B.

This case supposes a mutual backup agreement between A and B which can be the case if A and B are not in competition, for instance A is a mission provider and B a geographical one. But it is a real constraint...

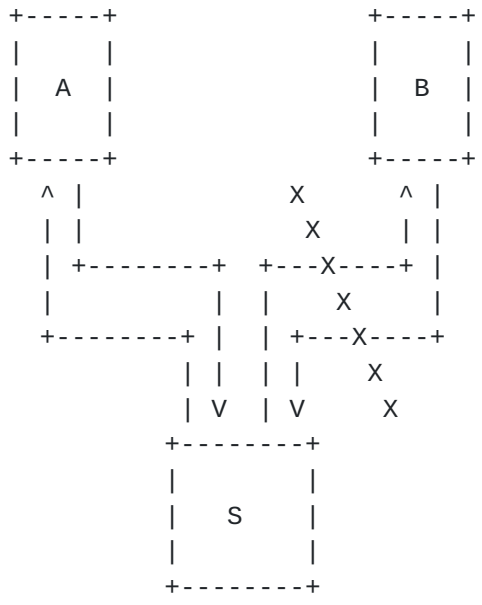
This still works if announces between A and B do not carry full prefixes (but they should include (ie. be shorter than) the prefix *:S/x+y). The backup will work only for a part of A and B (with a dark hole in case of failure for customers not implied in the backup agreement). Unfortunately this does not work in more complex cases:



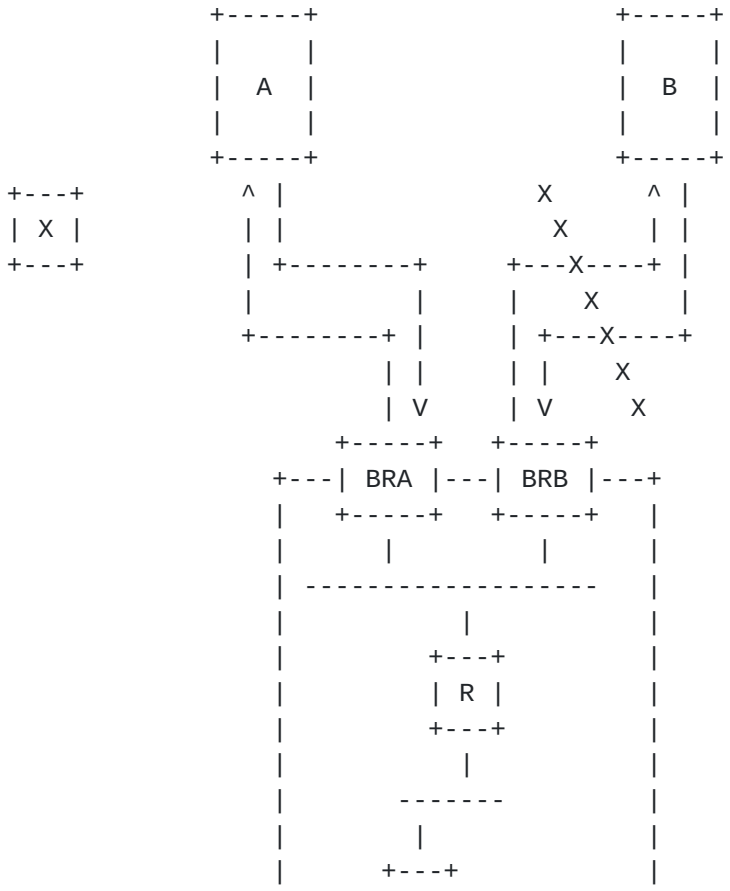
The backup is not transitive in this case, if something goes wrong in the B path for S the traffic can try to cross C which knows nothing about S and will drop packets...

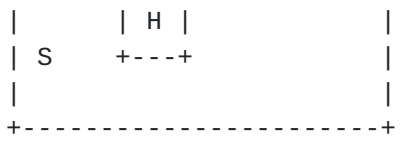
6. Broken Path

Consider the standard multihomed case when a link is broken:



If we look inside the routing domain S:




| | H |
| S +---+
|
+

The host H has two addresses, A:S:H and B:S:H, and the path through B is broken.

An external host X will use A:S:H because B:S:H does not work. The DNS will return both addresses but the applications should try all of them (on BSD 4.4 derived Unixes we have found only one standard application trying only the first returned address). We can try to play on address order in the DNS but the DNS caching mechanism makes this difficult (but it is not necessary). In conclusion new connections from X to H will work.

For new connections from H to X the problem is to force the choice of the good source address (A:S:H) by H. The proposal is to encode the "broken path" state in prefix information in router advertisement in order to inform nodes that addresses in a given prefix should not be used. The border router BRB knows there is a problem and should send this information to all the routers of S using for instance the router renumbering protocol [RENUM].

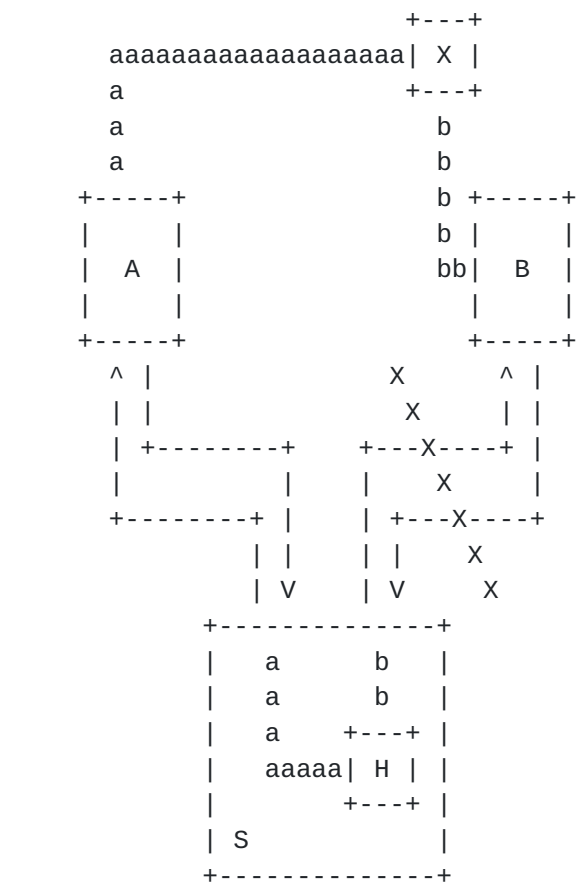
The best choice for the signaling of a "broken path" is to set the preferred lifetime of all the prefixes associated with the "broken path" to zero. This condition is very easy to recognize and its standard effect is to deprecate associated source addresses (ie. source addresses using the "broken path" are still valid but should not be used in new communications [ADDRC] what is exactly the intended behavior).

In the last case, existing (ie. established before the failure) connections between H (using B:S:H) and X are dealt with in the next section.

7. Use Of Mobility Mechanisms

The idea is to use some mechanisms of IPv6 mobility [MOB] (home address and binding update but not home-agent nor (in fact) true mobility) in order to make critical connections resilient to provider failures.

There is a connection between H and X (using addresses B:S:H and X) with a security association for authentication (necessary for mobility and not a real constraint for a critical connection because it is easy to mess an unauthentic connection, for instance with junk RST TCP packets).



After the (used) path through B fails, the broken path condition is set in the prefix B:S information in router advertisements then H is informed of the problem.

H uses a home address B:S:H destination option in each packet for X in order to use A:S:H as the source address: for each router the source is in A's prefix and only X replaces the source address by B:S:H before looking up the PCB of the connection.

H sends a binding update with A:S:H as the care-of address to X in a packet with an Authentication Header. X receives and processes it, sends a binding acknowledgement and uses a routing header with A:S:H as the (first) destination and B:S:H as the final destination.

Summary:

- o packets from H to X:
  - source = A:S:H
  - destination = X
  - home-address = B:S:H
  - binding-update (in first packets, should be acknowledged):
    - care-of = A:S:H

```
o packets from X to H:
   source = X
   destination = A:S:H
   routing-header: one address = B:S:H
```

While X must implement the full mobile correspondent node operation, H must implement only the binding management (no movement detection, no new care-of address acquisition, no operation with a home agent). In fact H does not move, it only changes its address choice.

8. Security Considerations

A better reliability in Internet connectivity can only improve security. Critical connection should be authenticated and binding updates must be carried in authenticated packets (see [MOB] for the discussion). IPSEC is mandatory for compliant IPv6 implementations.

9. ACKNOWLEDGEMENTS

All these ideas were discussed or found at the 40th IETF meeting at Washington during lunch-time 6bone BOFs. The transparency issue was well-known (and presented by Ben Crosby). The mutual backup scheme was built by the author for a regional/organization dual-homing at a G6 meeting.
The non-transitive issue was presented by Alain Durand. The diversion of mobility mechanisms appeared in the discussion between the author and Matt Crawford who proposed the broken path stuff.
Erik Nordmark has proposed to implement the "broken path" condition as the instantaneous deprecation of addresses using the "broken path".
The author would like to acknowledge inputs of the G6, the 6bone and the RIPE communities.

10. Changes From Draft -00

- Update the "Status" section (add a reference to RFC 2026, ...).

- Add a reference about address selection problem.

- Change the "broken bit" in the "broken path" condition.

- Update the "Acknowledgements" and "References" sections.

11. References

[AGGR] Hinden, R., O'Dell, M. and Deering, S., "An IPv6 Aggregatable Global Unicast Address Format", RFC 2374, July 1998.

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

[MULT] Bates, T. and Rekhter, Y., "Scalable Support for Multi-homed Multi-provider Connectivity", RFC 2260, January 1998.

[SRCA] Draves, R., "Simple Source Address Selection for IPv6", Internet Draft, <draft-draves-ipngwg-simple-srcaddr-00.txt>, April 1999.

[RENUM] Crawford, M., "Router Renumbering for IPv6", Internet Draft, <draft-ietf-ipngwg-router-renum-08.txt>, February 1999.

[ADDRC] Thomson, S. and Narten, T., "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[MOB] Johnson, D. B., Perkins, C., "Mobility Support in IPv6", Internet Draft, <draft-ietf-mobileip-ipv6-07.txt>, November 1998.

12. Author's Address

Francis Dupont
GIE DYADE
INRIA Rocquencourt
Domaine de Voluceau
B.P. 105
78153 Le Chesnay CEDEX
FRANCE

Fax: +33 1 39 63 58 66
EMail: Francis.Dupont@inria.fr

Expire in 6 months (December 25, 1999)