A QoS-Driven ISP Selection Mechanism for IPv6 Multi-Homed Sites

Marcelo Bagnulo, Alberto Garcia-Martinez, David Larrabeiti, Arturo Azcorra

Departamento de Ingeniería Telemática
Universidad Carlos III, Madrid. Spain.
Email: {marcelo, alberto, dlarra, azcorra}@it.uc3m.es

A global solution for the provision of QoS in IPng sites must include ISP selection based on per-application requirements. In this article we present a new site-local architecture for QoS-driven ISP selection in multi-homed domains, performed in a per application basis. This architecture proposes the novel use of existent network services, a new type of routing header, and the modification of address selection mechanisms to take into account QoS requirements. This proposal is an evolution of current technology, and therefore precludes the addition of new protocols, enabling fast deployment. The site-local scope of the proposed solution results in ISP transparency and thus in ISP independency.

1. Introduction[1]

As more organizations depend on critical applications built over the Internet, access links are becoming a vital resource for them. Consequently, many organizations are improving fault tolerance and QoS over their Internet connection through *multi-homing*, i.e. the achievement of global connectivity through several connections, possibly supplied by different Internet Service Providers (ISPs). Focusing on QoS, some mechanism is required to allow QoS policies to gain control over the path followed by packets, enabling for example, to route traffic generated by critical applications (i.e. real time or multimedia applications) through links providing a proper service in terms of bandwidth, delay, etc., while non critical traffic (i.e. ftp, www) is routed without interfering premium traffic.

IPv6 aggregatable global unicast address delegation rules [1] constrains multi-homed sites to obtain one prefix per connecting ISP when current provider-based aggregation, aimed to provide routing scalability is used. Therefore, the site will have as much prefixes as ISPs. Address selection mechanisms impact on ISP selection and thus on the QoS obtained since source address selection will determine the return path of sent packets.

In this article, we will present an architecture designed to provide QoS-driven ISP selection on a per-application basis, taking into account coherence between forward

[1] This research was supported by the LONG (Laboratories Over Next Generation networks) project, IST-1999-20393.

and backward paths. The solution presented relies on a host mechanism that builds packets appropriately to enable the network components to route it through the selected ISP.

The remainder of the paper is structured as follows: in section 2 an introduction to IPv6 multi-homing is presented, along with a discussion of multi-homing requirements. In section 3, the problem addressed in the article is thoroughly characterized and the proposed architecture is detailed, describing the network and host mechanisms, and their interaction for achieving QoS-enabled ISP selection. Finally, section 4 is devoted to conclusions and future work.

2. Introduction to IPv6 Multi-homing

Multi-homing deployment has been fostered since Internet connectivity has become a critical resource. Increasing QoS requirements, including those regarding to reliability, and the proliferation of Internet providers, are raising demand for appropriate multi-homing architectures.

In IPv4 solutions, ISPs propagate reachability routes to the multi-homed sites over the network. However, the actual IPv6 framework cannot directly apply the IPv4 solution, since one of the main concerns of IPng is routing system scalability, based on hierarchical routing, which restricts route propagation. Therefore, the IPv6 community has addressed the problem of defining specific mechanisms to allow IPv6 site multi-homing, taking into account the routing particularities above mentioned. Some of the results of this effort are described in this section, and include a draft detailing the requirements that IPv6 multi-homing proposals should fulfill [2] (issued by a recently created IETF group on IPv6 multi-homing, *multi6*), and a set of different architectures that has been proposed for IPv6.

2.1. Multi-homing Requirements

The initial requirements identified for multi-homing are [2]:

- *Redundancy and reliability*: A multi-homing architecture is built to improve Internet connection reliability, so fault tolerance is a key issue in any proposed mechanism. Failures that should be coped with include physical and logical link breakdowns, routing protocol malfunction, and ISP or exchange crash.
- *Load sharing*: Distribution of load among available links is another key issue in a multi-homed environment. Distribution criteria include:
  - Performance improvement in situations such as long-term congestion.
  - Cost optimization: Depending on the SLA agreed, cost and quality of different connections may vary among ISPs. Cost-aware selection mechanisms are needed to fulfill the requirements of multi-homed organizations.
- *Simplicity* is a relevant condition if the architecture is meant to be deployed. The fewer the changes in existing protocols and mechanisms that are introduced, the faster the solution would be developed.

- *Scalability* has been a major concern in IPv6 definition so proposed multi-homing architectures must adhere to this policy. In particular, routing scalability provided by ISP-based aggregation must be preserved. Other scalability issues such as the manageability of the solution should also be considered.

Note that the requirement of QoS-driven ISP selection is included among the multi-homing requirements, in the *load-sharing* item discussion.

2.2. Review of Proposed Solutions for Multi-homing Issues

We can find several proposals aiming to solve different multi-homing challenges. We will introduce some of them that we have considered close to our study.

2.2.1. IPv4 Multi-homing Proposals

Initial IPv4 approaches to the multi-homing problem were based on the ability of CIDR for prefix propagation [3]. The most common solutions are based on the injection of routing information regarding the site prefix into the Internet. Since this approach does not impose the addition of new prefixes, address scarcity is not fostered. The main drawback of these approaches is that they contribute to the undesirable explosion of the number of entries in the routing tables of the default free zone (DFZ) of the Internet [4].

A more scalable approach is presented in [5], with several prefixes assigned to the multi-homed site, one per each ISP. In normal operation no extra routing information is advertised to the DFZ, and non-hierarchical route injection only occurs in case of link failure. The main drawback of this mechanism, given the lack of IPv4 addresses, lies in the requirement of assigning several prefixes to each site.

2.2.2. IPv6-compatible Multi-homing Proposals

As we have seen, IPv4 solutions, that are based on routing information injection, collide with the IPv6 provider-based aggregation paradigm. We will briefly present some new IPv6-oriented solutions.

Multi-homing support at exit routers, presented in [5] and developed for IPv6 in [6], is an optimization of the last IPv4 mechanism described, eliminating the need for route information injection even in the case of link failure. This solution is aimed to provide link fault tolerance through multi-homing. Provider-based aggregation is assumed, so each ISP delegates one prefix to the considered site. Link fault tolerance is achieved through tunnels between site egress routers (RA, RB) and ISP border routers (BRB, BRA) as it is depicted in figure 1. In normal operation tunneled paths are advertised with a low priority, to guarantee that traffic is routed through direct links whenever possible. In case of link failure, the link that is down is no longer advertised as a valid route, so tunneled path becomes the preferred route (in spite of its low priority).

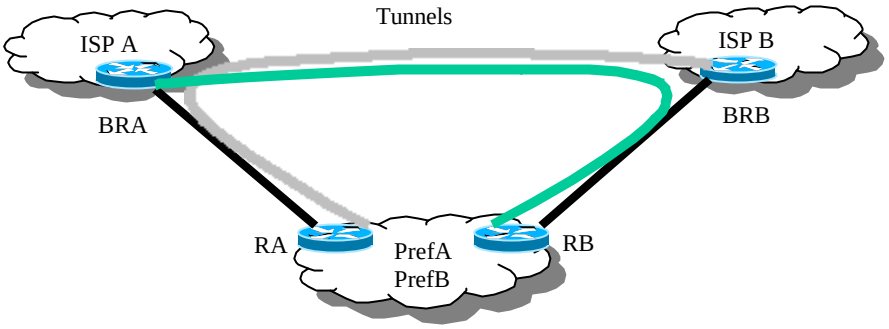

Fig. 1. Multi-homing at exit routers

Multi-homing with router renumbering [7] is intended to avoid the usage of ISP delegated prefixes when the delegating ISP is down. The proposal uses the Router Renumbering protocol [8] as a mechanism to deprecate addresses delegated by the unreachable ISP in routers. Routers perform deprecation of host addresses using Router Advertisement. While this solution is a good option for long-term failures, established communications cannot be preserved after the failure event. A better approach could be an hybrid one, i.e. to use Router Renumbering for long-term failures and a tunneling scheme, such as the one described before, for established communications.

There are other proposed solutions, which involve the usage of mobile IP protocols [7], the modification of the TCP handshake protocol to include a set of possible source and destination addresses [9] or the report of link failure through explicit routing information [10].

We should stress that the proposed solutions focus on fault tolerance, taking into account even long term failures and the preservation of established TCP connections in case of failure. Load balancing could be enabled in some cases, but QoS driven ISP selection has not been considered.

3. QoS-Driven ISP Selection

We will present now a proposal for an ISP selection mechanism based on QoS criteria. First the problem is delimited, and the intended solution scope is stated. Later the proposed architecture is described and finally the main components are detailed.

3.1 Problem Delimitation

The topology we are going to consider for illustrating the problem is depicted in figure 2.

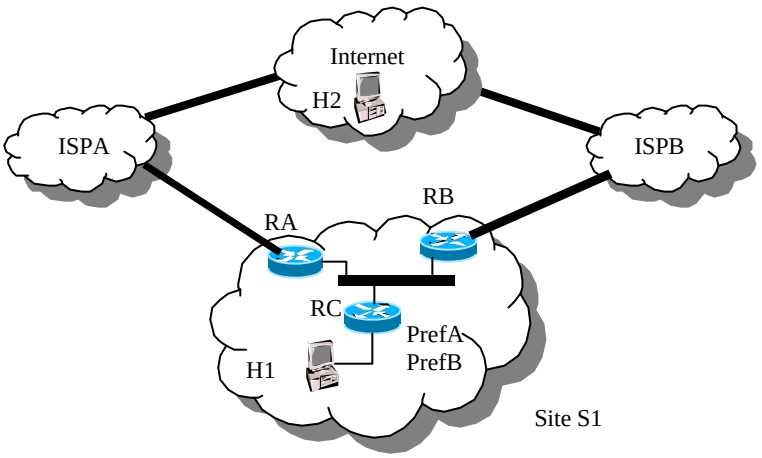

Fig. 2. Problem topology

As we can see in the figure above, we are considering a site (S1), composed of several sub-networks, that obtains external connectivity through two different ISPs, ISPA and ISPB. Note that even if we are considering the case of only two ISPs, the solution presented is valid for more ISPs, and this limitation is introduced only to facilitate comprehension.

Each one of the connected ISPs, delegates one prefix to S1, so that the hosts belonging to S1 may have addresses containing prefix PrefA::/nA and prefix PrefB::/nB. In order to enable the reception of packets from a given ISP, any host must have one address with the corresponding prefix of that ISP.

To properly delimitate the problem, we could divide the problem into the following two cases:

- *Internally initiated traffic:* Suppose that a host (H1) that belongs to S1 needs to communicate with a remote host (H2). Based on site-local routing policies, the host

and/or the network should be able to decide which ISP will carry the outbound packet, and this selection should be based on QoS requirements that could be specified on a per application basis.
Note that the route from the Internet towards the destination host is determined by the destination address returned by DNS, and by the external routing policies[2]. However, intra-site mechanisms can influence on intra-site routing and ISP selection.
The ISP used in the return path for the outbound traffic is determined by the source address included in the outbound packet, more specifically by the prefix used (PrefA::/nA or PrefB::/nB), so there should be a relation between the selected exit path and the source address included in the packet, in order to provide a coherent path for outgoing and incoming packets belonging to the same data stream.

- *Externally initiated traffic:* The ISP included in the path for inbound connections is determined by the prefixes of the addresses returned by DNS, PrefA::/nA or PrefB::/nB. This issue is out of the scope of this work.

3.2. An Architecture for a QoS-based ISP Selection Mechanism

In this paper we present a solution that allows the selection of the ISP used for coursing locally initiated traffic (i.e. traffic initiated by H1). The ISP selection is based in the QoS policies defined within a site and it is done on a per application basis (H1 can select ISPA for some applications and ISPB for others).

The solution for the stated problem should involve both host and network. A basic justification for this statement is the following: Intra-site route selection is performed by network devices, because the host does not have all the routing information needed for that decision (and it is not desirable that it had it). However, the host is involved in the route selection, since the return path is determined by the source address included by the host. So we will next present the two main components of the solution: network elements and host mechanism.

3.2.1. Network Elements for QoS-based ISP Selection

A mechanism intended to force routing through a specific ISP in multi-homed sites is suggested in [11]. The main idea is to include a routing header with an intermediate anycast address identifying selected ISP routers, in order to force the packet path through the chosen ISP. In this section we describe a different approach, based in a mechanism for path selection that achieves ISP transparency through the usage of a self-destructive routing header. We will end with a comparison between the mechanisms proposed and other possible approaches.

---

[2] The Routing Header option can be used to specify routes, but global topology information is required for making routing decisions, information that is not generally available in hosts.

ISP transparent path selection. ISP selection can be performed by means of selecting the appropriate exit router, in the case that each ISP connection were supported in a different router. Note that supporting both ISP connections on a single router introduces a single point of failure, which precludes the fault tolerance ability of multi-homing architectures, so we will focus on the first case. Besides, when all the packets are routed to a single exit router, ISP selection can be implemented locally in the border device.

When different routers support ISP connections, exit router selection can be done using a Routing Header that includes a site-local address assigned to one of the router interfaces. In the case that there was more than one exit router connected to the same ISP, this address would become an anycast address, because we should assign it to all the connecting routers.

Self-destructive routing header. Routing header information is no longer useful once the exit router is reached. Furthermore, considering that the address included in this particular routing header has site-local scope, it becomes meaningless once the packet is out of the site. So a new type of routing header is proposed, possibility that is considered in [12]. This type of routing header is self-destructive, meaning that once it has reached the intermediate destination (exit router in our case) this particular routing header is removed from the packet. This reduces the overhead introduced by the ISP selection mechanism in the most critical links in terms of bandwidth, the WAN links.

One concern about the stated solution could be the overhead introduced by routing headers, even considering that routing headers are only present inside the local site. Note that the packets routed through one of the ISPs do not need to include the routing header, since it could be the site's default exit router.

The main advantages of the proposed solution are described next:

- ISP transparence: In previous proposals, the routing header was used to force a path including the ISP routers anycast address; as a consequence, processing of the routing header requires ISP routing resources. In this proposal, the processing of the routing header is done by the site exit router, which combined with the self destructive routing header makes path selection completely transparent to the ISP. This presents several benefits:
  - Improved scalability: as one ISP connects many sites with the Internet, interpretation of routing headers could be a heavy task. If it is done at site exit routers, scalability is preserved.
  - No bandwidth overhead caused by ISP selection on WAN links, because the self-destructive routing header ensures that no useless routing header information is transmitted over these links.
- ISP independence. There is no need for ISP cooperation, such as support for the all routers anycast address in the ISP network.
- This solution is based in existing protocols, so it is fast and easy to implement.
- This solution is deployed completely at network level, without NAT or proxy services that could compromise performance.

3.2.2. Host Mechanism for QoS-based ISP Selection

As we have seen before, once the ISP to be used is decided, the host must include in all packets the appropriate routing header, in order to force routing through the selected ISP, and the according source address to ensure a coherent return path. In the following paragraphs we are going to detail how source address and routing header are determined. We first introduce the non QoS-enabled source address selection algorithm described in [13] and then we present a novel extended mechanism performing the desired behavior.

3.2.2.1 Source Address Selection Algorithm

The source address selection mechanism relies on a set of rules to obtain the most appropriate source address for a given destination address and a defined policy. For further reading on this issue the reader is referred to [13].

The policy table is a longest prefix match table that takes an address (source or destination) as input, and returns two values: a label value and a precedence value. The label value is used to match destination addresses with source addresses. The precedence value is used to select destination address among a set of available destination addresses, and it is not needed for our study so it will not be introduced. The suggested default policy table is included in Table 1

Prefix	Precedence	Label
::1/128	50	0
::/0	40	1
2002::/16	30	2
::/96	20	3
::ffff:0:0/96	10	4

Table 1. Default policy table

The process of source address selection is as follows: Once a packet is to be sent to a destination address, the host routing mechanisms will select the interface used for delivering the packet. After this, source address selection is started. The source address selection algorithm has as inputs a destination address (D) and the first two source addresses (SA and SB) from a proposed candidate source address set, and it returns the source address that fits best with the destination address. Successive pair-wise comparisons are performed throughout all addresses in the candidate set to obtain the best one. The algorithm is implemented as an ordered set of rules; if a rule selects one of the two addresses, no further rules are processed. Here we list the proposed rules (Sx refers to any SA and SB)

- Rule 1: Prefer same address: If Sx=D then prefer Sx
- Rule 2: Prefer appropriate scope: If SB has a larger scope than SA (i.e. SA is a site local address and SB is a global address) and D has a larger scope than SA then prefer SB; otherwise, prefer SA.

- Rule 3: Avoid deprecated addresses.
- Rule 4: Prefer home address: This applies to mobile IP environments so it will not be discussed here.
- Rule 5: Prefer outgoing interface: If SA is assigned to the interface that will be used to send the packet towards D, and SB is not, then prefer SA.
- Rule 6: Prefer matching label. Obtain label for SA, SB and D from policy table and compute the following condition: if label(SA)=label(D) and label(SB)<>label(D) then prefer SA.
- Rule 7: Prefer public address: if SA is a public address and SB is a temporary address, then prefer SA (public and temporary addresses are discussed elsewhere [14])
- Rule 8: Use longest matched prefix of Sx with D.

3.2.2.2 Proposed Host Mechanism

The only modifications required to allow QoS-enabled source address selection are related with the policy table. In particular, port information (to identify the application) and an intermediate address (to force routing path) are included in the table. Also the policy table search algorithm must be adapted to included the added information. The address selection rules, however, do not have to be modified.

Policy table modifications. Port information must be included in order to be able to make application based source address selection. Besides, we also include exit router addresses in another column to be used as intermediate addresses in the routing process. The new policy table will then have the aspect illustrated in Table 2.

To perform a lookup in the table, an address and optionally a port are required as inputs, and the outputs are a label value, a precedence value and an intermediate routing address when available. The lookup algorithm first performs longest prefix match among the matching port entries, and, if no matching port is found, longest prefix match is performed with entries that do not have a specified port.

Port	Prefix	Precedence	Label	Intermed. Add.
P1	0::/0	60	100	
	::1/128	50	0	
	::/0	40	1	
	2002::/16	30	2	
	::/96	20	3	
	::ffff:0:0/96	10	4	
	PrefA::H1/128	5	100	RA

Table 2. QoS-enabled policy table

Resulting behavior. Consider that a multi-homed site (figure 2) is connected to ISPA through exit router RA, and to ISPB through exit router RB. The site obtains prefix PrefA::/nA from ISPA and prefix PrefB::/nB from ISPB. Suppose that a host (H1) of the considered site needs to send QoS-demanding traffic, with destination port P1, to

several destination addresses through ISPA, and best-effort traffic through ISPB. In order to ease the explanation, suppose that host H1 has only one interface with two assigned addresses, PrefA::H1 and PrefB::H1, besides link-local and site-local addresses. The host policy table included in table 2 accomplish the desired behavior:

Applying the defined policy table, when an application needs to communicate to a remote host D with an application listening at port P1, the policy table will return the following values:

- For destination address D and port P1, it will return a label value equal to 100, because destination address and entry ports match.
- For source address PrefA::H1, it will return a label value equal to 100, and RA as the intermediate address for the routing header.
- For source address PrefB::H1, it will return a label value equal to 1, because longest prefix match is applied.

Final label matching leads to a resulting packet with PrefA:H1 as source address, RA as destination address and PA as destination port, and a routing header containing address D. This packet will then exit through ISPA, and response packets will also be routed through ISPA because their destination address will be PrefA:H1.

Note that all the previously described process is carried out only if the first five rules specified in the source address selection algorithm do not apply. We will next justify that rule 6, where the host part of the QoS-driven ISP selection mechanism resides, is reached when it is needed.

- Rule 1: Same address only applies when the target host is the local host, so no ISP selection is needed.
- Rule 2 assures that ISP selection is aimed only to traffic addressed to destinations outside the site; therefore no ISP selection is performed with site local or link-local connections. Mechanism will only work for global addresses, which is the intended behavior. To avoid being routed through exit routers when systems of the same site use ports associated with QoS delivery service, site-local addressing should be deployed.
- Rule 3 enables compatibility with the fault tolerance features provided by multi-homing through the Router Renumbering mechanism. So when there is a failure in an ISP, address deprecation through Router Renumbering and Router Advertising precludes the selection of addresses delegated by crashed ISP, without need to modify existing policy tables on hosts.
- Rule 4 does not applies.
- Rule 5. The interface selection in the host routing mechanism for a given destination could force selection in rule 5 if the chosen interface has not been assigned at least one address from all ISPs. Then, if one interface has an address with prefix PrefA::/nA, it must also have an address with prefix PrefB::/nB and vice versa to ensure proper functioning.

4. Conclusions and Future Work

We have presented a QoS-driven ISP selection mechanism, which allows to force routing through a selected ISP of both inbound and outbound packets of on a per application basis. The selection of ISP is based on site local policies implemented in hosts through a modified policy table. The overall behaviour of the mechanism can be summarized as follows: When an application running on a local host needs to communicate with another application listening in a particular port of the remote host, the source address selection algorithm uses the policy table to provide an intermediate address and a source address. The intermediate address forces outbound packets to pass through the ISP assigned for that application, and the corresponding source address ensures that response packets will be routed across the same ISP. Once the host transmits the packet, the network routes it to the selected ISP exit router. When the router receives the packet, it updates destination address accordingly and removes the routing header, according to the behaviour of a newly defined routing header type.

A key advantage of the presented mechanism is the provision of QoS on a per application basis, without introducing a new protocol, based only on the modification of an existing mechanism. Changes only affect hosts, so deployment is eased. Since only network-level processing is performed, performance will not be impaired by this mechanism. Another significant advantage is ISP transparence, which allows enabling ISP selection without ISP cooperation as a result of the site-local nature of the proposed solution. Moreover, ISP performance is not affected by site multi-homing. It is also important to note compatibility with existing multi-homing mechanisms, such as *multi-homing support at exit routers* or *multi-homing with Router Renumbering*, allowing a complete solution providing a fault tolerant and QoS aware multi-homing architecture. We want to stress that this is the first proposal achieving per application QoS-driven ISP selection. Moreover, this capacity is achieved in an ISP transparent manner.

Further work remains to be done. Manual policy table configuration has a high management cost, and opposes to the IPv6 autoconfiguration-enabling spirit. Therefore, automatic distribution mechanisms should be developed in order to provide convenient policy management. Another future working issue is the evaluation of a possible optimization for the overhead in bandwidth and processing introduced by the routing header usage. A possible improvement is the inclusion of a routing header only in the first packet of a connection, along with an IPv6 header flow label. Routing of subsequent packets of the connection will be based on the flow label.

References

[1] Deering, S., Hinden, R., O'Dell, M.: RFC 2374 - An IPv6 Aggregatable Global Unicast Address Format. July 1998.

[2] Black, B., Gill, V., Abley, J.: Requirements for IP Multihoming Architectures, draft-ietf-multi6-multihoming-requirements-00, February 2001.

[3] Fuller, V., Li, T., Yu, J., Varadhan, K.: RFC 1519 - Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy. September 1993.

[4] Huston, G.: Analysing the Internet BGP Routing Table. Internet Protocol Journal. Cisco 2001.

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

[6] Hagino, J.: IPv6 multihoming support at site exit routers. draft-ietf-ipngwg-ipv6-2260-01. April 2001.

[7] Dupont, F.: Multihomed routing domain issues for IPv6 aggregatable scheme. draft-ietf-ipngwg-multi-isp-00. September 1999.

[8] Crawford, M.: RFC 2894 - Router Renumbering for IPv6. August 2000.

[9] Tattam, P. Preserving active TCP sessions on multihomed IPv6 networks. IPng Working Group Meeting Minutes. Tokio, September 1999.

[10] Bragg, N.: Routing support for IPv6 Multi-homing. draft-bragg-ipv6-multihoming-00. November 2000.

[11] Johnson, D., Deering, S.: RFC 2526 - Reserved IPv6 Subnet Anycast Addresses. March 1999.

[12] Deering, S., Hinden, R.: RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification. December 1998.

[13] Draves, R.: Default Address Selection for IPv6. draft-ietf-ipngwg-default-addr-select-04. May 2001.

[14] Narten, T., Draves, R.: RFC 3041 - Privacy Extensions for Stateless Address Autoconfiguration in IPv6. January 2001.