UPGRADE
The European Journal for the Informatics Professional
http://www.upgrade-cepis.org
Vol. VI, issue No. 2, April 2005

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www. upgrade-cepis.org/>

UPGRADE is the anchor point for **UP**ENET (**UP**GRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
• **Mondo Digitale**, digital journal from the Italian CEPIS society AICA
• **Novática**, journal from the Spanish CEPIS society ATI
• **OCG Journal**, journal from the Austrian CEPIS society OCG
• **Pliroforiki**, journal from the Cyprus CEPIS society CCS
• **Pro Dialog**, journal from the Polish CEPIS society PTI-PIPS

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by **Novática** <http://www.ati.es/novatica/>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática* <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by **Novática**, and in Italian (summary, abstracts and some articles online) by the Italian CEPIS society ALSI <http://www.alsi.it> and the Italian IT portal **Tecnoteca** <http://www.tecnoteca.it/>

UPGRADE was created in October 2000 by CEPIS and was first published by **Novática** and **INFORMATIK/INFORMATIQUE**, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

UPGRADE **Newslist** available at <http://www.upgrade-cepis.org/pages/editinfo.html#newslist>

ISSN 1684-5285

Monograph of next issue (June 2005):
"**Free Software Engineering**"
(The full schedule of **UP**GRADE is available at our website)

Monograph: IPv6 - More than A Protocol
(published jointly with Novática*)
Guest Editors: *Jordi Domingo-Pascual,Alberto García-Martínez, and Matthew Ford*

MOSAIC

UPENET (UPGRADE European NETwork)

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>, and in Italian (online edition only, containing summary, abstracts, and some articles) by the Italian CEPIS society ALSI (*Associazione nazionale Laureati in Scienze dell'informazione e Informatica*) and the Italian IT portal Tecnoteca at <http://www.tecnoteca.it>.

Tools for IPv6 Multihoming

Marcelo Bagnulo-Braun, Alberto García-Martínez, and Arturo Azcorra-Saloña

The availability of two or more connectivity providers (a configuration known as multihoming) allows improvements in failure tolerance and enables traffic engineering capabilities. Current IPv4 (Internet Protocol version 4) multihoming solutions suffer from scalability limitations, or are partial solutions based on NAT (Network Address Translation) technologies. In this article we present a set of tools that allow IPv6 (Internet Protocol version 6) networks to benefit from multihoming, taking advantage from the fact that each provider delegates its own set of addresses. We distinguish two cases: establishing new communications either with or without failures, and maintaining previously established communications in case of failure.

Keywords: Communications Security, Fault Tolerance, IPv6, Multihoming.

1 Introduction

In the current IPv4 (Internet Protocol version 4) infrastructure, configurations in which a network obtains connectivity through two or more providers are gaining momentum [1]. These networks are known as *multihomed*. The main advantage provided by this configuration is fault tolerance, although additional capabilities such as traffic engineering can also be obtained.

The most popular multihoming configuration for IPv4 relies on interdomain routing, relaying on the injection into the BGP (Border Gateway Protocol) routing system of the prefix or prefixes of the multihomed network [2], as it is shown in Figure 1. If a failure occurs and affects one of the paths in the meshed topology, the routing system will find a valid alternative path, if one exists.

Providing multihoming benefits for a network implies adding to the global routing table new entries that correspond to the prefixes of the multihomed networks. These entries were not needed before becoming multihomed, because reachability was propagated through an aggregated prefix of a larger provider. The addition of new entries in the global routing table is deemed as undesirable, since it results in scalability problems for the BGP route processing that yields to increases in the convergence time [3] among other inconveniences. To limit the number of entries propagated into BGP, some providers impose *de facto* restrictions by filtering overly specific prefixes, precluding small networks and residential users from the benefits of multihoming. Despite this, BGP configuration hassles and stringent requirements for the stability in the operation would result in too many obstacles for these users. For these cases it is still possible to deploy solutions based on Network Address Translation (NAT) to obtain limited fault tolerance and traffic engineering capacities [4], as it is shown in Figure 2. However, besides the problems inherent in NAT [5][6], previously established communications are not preserved in case of a failure, because the change of the addresses being used would result in an interruption of the communication (e.g. consider a TCP, Transmission Control Protocol, connection).

For IPv6 (Internet Protocol version 6) deployment there is a consensus in keeping the number of entries in the routing table as low as possible, so it is not acceptable in an IPv4-like multihoming solution. Medium and small IPv6 networks will receive addresses from address blocks assigned to large providers. These large providers exchange routing information through BGP. A medium or small network that seeks multihoming benefits, i.e. a residential user with two connectivity providers, receives different address blocks from each provider. Therefore, the end systems of this network will be configured with addresses built upon the prefixes of each one of the providers. This configuration is known as *multiaddressing*.

There are several problems for a system with addresses

Marcelo Bagnulo-Braun is a Teaching and Research Assistant in the Telematics Engineering Department of the *Universidad Carlos III de Madrid*, Spain. He is currently a PhD student at the same university, designing solutions for IPv6 multihoming. He participates in several research projects related to IPv6, both international (6LINK) and national (OPTINET6, SAM). He has published several papers and internet drafts on the subject. <marcelo@it.uc3m.es>

Alberto García-Martínez is an Associate Professor in the Telematics Engineering Department of the *Universidad Carlos III de Madrid*, Spain. He received his PhD. in Telematics Engineering from the *Universidad Politécnica de Madrid* (UPM) in 1999. He has participated in several national and international research projects on IPv6 and QoS, and has published several papers on the subject. <alberto@it.uc3m.es>

Arturo Azcorra-Saloña is a Professor in the Telematics Engineering Department of the *Universidad Carlos III de Madrid*, Spain. He has led several research projects on next generation protocols and services, addressing IPv6, mobility, QoS, active networks, etc. He also coordinates the Network of Excellence E-NEXT, funded by the 6th Framework Programme of the European Union. He has authored more than 100 national and international articles. <azcorra@it.uc3m.es>

from several providers. First, the possibility of establishing a communication with a remote element is not guaranteed even in the absence of failures. Some providers perform *ingress filtering* for security reasons, filtering those packets that are sent by their clients with source addresses that do not belong to the address block assigned to the clients. If a system in a multiaddressed network sends through provider *A* packets that carry addresses obtained from provider *B*, these packets will be discarded by provider *A* due to ingress filtering. Even if this problem is solved, there still remain some adjustments to current networking functionalities to assure that a valid path is found for a communication that is to be initiated after a network failure. Finally, more complex modifications are required for the preservation of communications that were established prior to a failure.

In this article we present tools for handling the establishment of new communications when multiaddressed networks are considered, and additional tools for preserving previously established communications in case of failure. For establishing new communications, we propose the deployment in the multihomed networks of routing systems that rely on source address for packet forwarding in addition to the common use of the destination address. We also propose the exploration of different pairs <source address, destination address> when a failure occurs. For the preservation of established communications, a new sublayer within the network layer is proposed for dynamically managing, by means of a specific protocol, the multiple locators available in a multiaddressing scheme. The following tools required for solution are currently been discussed by the IETF (Internet Engineering Task Force) [7]: tools for locator management, tools for failure detection, tools for the determination and enforcement of valid alternative paths, and tools for flow identification regardless of the selected path.

The article is structured as follows: in the next section the tools required for the proper establishment of new connections in a multiaddressing environment are discussed, considering scenarios with and without failures. In Section 3 we present the tools that allow the preservation of a communication in case of a failure. Finally, conclusions are presented.

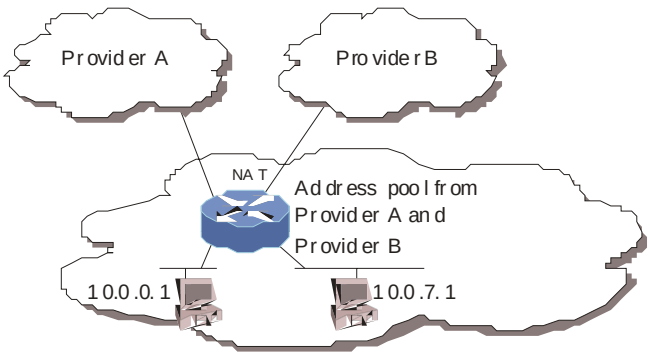

Figure 2: NAT-based Multihoming in IPv4 Networks.

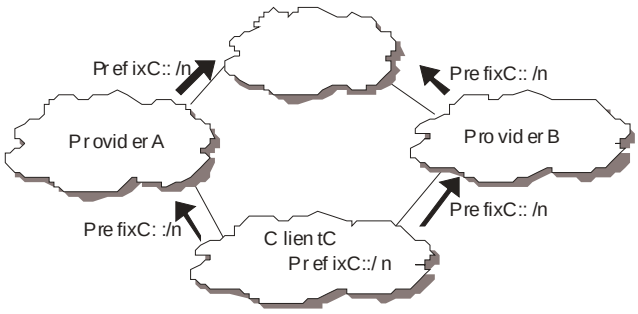

Figure 1: BGP Route Injection for the Provision of Multihoming in Ipv4 Networks.

2 Initiating Communications in A Multiaddressing Environment

As it has been presented in the introduction, a multiaddressing environment can suffer from connectivity problems without the occurrence of any failure. We can illustrate this by the case study depicted in Figure 3, in which Host1 is going to initiate a communication with Host2. In a common situation, Host1 would access the DNS to obtain the IPv6 addresses in which Host2 is accessible. Host1 will perform the Default Address Selection procedure [8], taking as inputs all Host2 addresses that have been acquired from the DNS, and all Host1 addresses locally available, to obtain a <source address, destination address> pair to be used in the communication.

Even though there is no failure in the network infrastructure, it may happen that ingress filtering [9] may preclude the possibility of communication. This filtering is applied to avoid the malicious usage of addresses that are not owned by the users, i.e. *address spoofing*. The motivation is as follows: If a user spoofs an address, in particular an address that does not belong to the topological IP region in which the user is placed, it could perform attacks anonymously, since the real location of the attacker could not be easily obtained. Note that if the attacker is not placed in the path between the attacked node and the network from which it has spoofed the address, the attacker may send packets, but could not receive the responses, since these responses will be sent to the network that owns the address. However, Denial of Service attacks can be issued in this situation, by flooding the network to which the attacker's packets are directed, or by flowing the network from which the source addresses have been spoofed (if the response traffic generated upon the attacker's request is high). A protection measure against address spoofing is to deploy ingress filtering in the providers. Ingress filtering consists of filtering those packets generated by a client with source addresses that does not belong to the address range assigned to that client. Coming back to the multihoming scenario, a packet generated by Host1, with a legitimate address obtained from ProviderA, will be discarded if the packet is forwarded through ProviderB. Note that when Host1 chooses a source

address for the packet, it selects the provider through which packets coming from Host2 are received (and therefore, part of the reception path). As a consequence of the deployment of ingress filtering, source address selection also defines the provider through which packets can exit without being filtered. Analogously, the destination address also establishes a particular path through which packets will arrive to Host2, and the path that Host2 has to use for outgoing packets. Therefore, we can state that in a multiaddressing scenario the node that initiates the communication defines the ingress and exit path that will be used by the packets of the communication.

To avoid packet losses due to ingress filtering, we propose the deployment in the multihomed network of an intra-site routing scheme in which the source address is also taken into account for forwarding [10][11]. For this purpose, the routers in the multihomed networks will have, in addition to the default routing table, as many routing tables as different providers for the multihomed networks. In the packet forwarding process, the source address of the packet considered will determine the routing table to use, that will result in forwarding outgoing packets through the provider that has delegated the address block used in the source address (or the default routing table if the packet is an incoming one and the source address does not match with any local prefix). An especially simple configuration occurs when a single router is responsible for connecting to all the providers, because it is only required to configure a single router. This may be a typical case in residential networks.

Once the basic connectivity issues have been solved, we should address the initiation of new connections in case of failures. If the failure arises in one of the providers or links close to Host2 (for example in ProviderK), and Host1 starts the communication, it can happen the following: Host1 accesses DNS to obtain the addresses of Host2. With the received addresses, the Default Address Selection procedure [8] selects a <source address, destination address> pair. Suppose that the destination address belongs to the address range delegated to the network of Host2 by ProviderK. The application tries to establish the communication using these parameters, but it is not possible, and detects the communication problems (the transport connection establishment process indicates a failure, or a timer in the application expires). In this case, the behaviour recommended to the applications is to retry with another destination address from the set obtained from the DNS [8]. Repeating this process, connectivity failures close to Host2 can be solved if at least one valid path to the destination exists, with a time penalty of the sum of the time required for the timer expiration for all the explored paths.

If the failure occurs close to Host1, the exploration of alternative paths is achieved by the variation of source addresses. This would require modifications in the Default Address Selection mechanism, modifications that are currently under discussion. Note that in the proposed scheme, failure detection at the endpoints relies only on the information provided by timer expiration, and this information is not enough to determine if the failure has occurred in the destination endpoint (for example, because the destination has been turned off), in a location close to the destination endpoint (requiring changes in the destination address), or close to the source (requiring changes in the source address). Additionally, multiple failures may arise, restricting the combinations of source and destination addresses that are valid. Therefore, only the exploration of all the possible pairs <source address, destination address> assures connectivity if there is at least one valid path among the endpoints. The sequential exploration of these alternatives can be slow, while parallel exploration can be expensive in bandwidth and may require changes in the applications or in the network stack interface.

3 Preservation of Communications in A Multiaddressing Environment

After the discussion in the previous section of basic issues about the establishment of communications in multiaddressing multihomed environments, we approach in this section the problem of preserving established communications (such as a TCP connection - but not exclusively) in case of a failure. It would be desirable that this functionality were provided to the applications in a transparent fashion, i.e. without requiring changes in the applications for benefiting from the service. We can also consider that it could also be beneficial if the service is also provided transparently to the transport layer (TCP, UDP - User Datagram Protocol - and other protocols). To fulfil both requirements, it has been proposed that the multihoming functionality can be provided at the network layer. The following considerations are developed from the information available in several working documents ([12][13][14][15]).

A network layer multihoming solution would be in charge of changing the IP addresses used for packet forwarding, to ensure that addresses that define valid paths for a considered communication are used. The addresses included in the actual IP packets, addresses that are used for packet

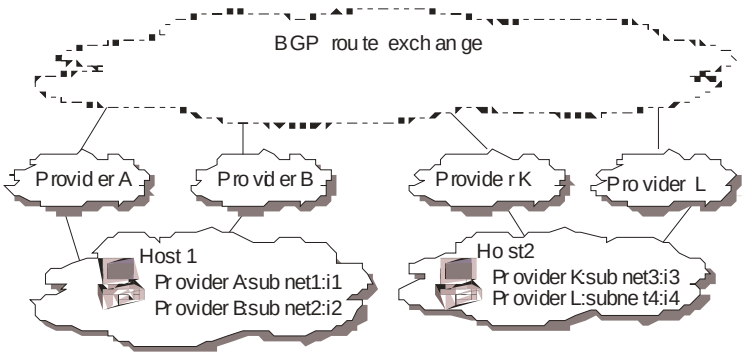

Figure 3: Multiaddressing-based Multihomed in IPv6.

forwarding, are known as *locators*. However, the transport layer usually employs IP addresses as part of the basic parameter set that identifies a communication. TCP, for example, uses the source address and destination address, apart from source port and destination port, to uniquely identify a connection. These addresses are usually provided by the application of the host that is starting the communication, and they can also be used by the applications for identifying the communication. When we use the IP addresses this way, we call them *identifiers*. After establishing a communication in a similar fashion to the one commented in the previous section, a multihoming solution will be in charge of managing different locators to ensure that connectivity between endpoints is preserved if there is at least one valid path between them, while presenting a single identifier to the upper layers.

The management of identifiers and locators will be performed in an entity at the endpoints, included in the network layer as an *identification sublayer*. This sublayer will exchange the locators that are available for a given communication in each of the endpoints, to make the locators ready to be used in case of failures. Following the example presented in the previous section, once established the communication between Host1 and Host2, either host (for example Host2) could initiate an exchange in which it could inform the other about all the locally available locators, triggering the corresponding answer. This task would be performed by a specific protocol for multihoming, and a new state will be generated for the participants in the information exchange. From now on, if a problem is detected when a given pair is used in the communication, any host can modify the pair to find a valid path.

Tools for identifying flows after a change in the locator pair are required. It can be necessary to include in the packets a flow identifier that should have been pre-accorded between both endpoints by means of the multihoming specific protocol. This identifier can be included in the packets in an IPv6 header extension.

To detect a failure, there are several alternatives [14]. First, there are mechanisms that allow the detection of local problems, such as an interface that is no longer operational. Additionally, we can rely on the information provided by upper layers to be able to detect problems in communications from explicit notifications of failure, or from the absence of positive confirmations. TCP, for example, could inform the network layer about communication problems if a TCP confirmation has not been received within a specified time. Finally, we can add specific signalling procedures for multihoming, sending packets that could check reachability between a given pair of locators. This procedure can be analogous to an ICMP (Internet Control Message Protocol) ping (an echo request with its corresponding response), and it can be used for checking the state of the locators currently being used for the communication, in parallel to the data exchange, i.e., out of band. A timer will check if the responses are received within the appropriate period, and if this is not the case, a process for the selection of a new locator pair will be started. The selection process will include a reachability test for different locator pairs. Once selected a new pair will be used for subsequent data packet exchanges. When the remote host receives packets with the new locators, this host will start a reachability test using as the candidate pair the locators received from the host that has initiated the change.

The ability to change locators while a communication is being held introduces security problems. As a criterion for the analysis of the security offered by the new multihoming solutions, it is usually required that the new mechanisms should not enable vulnerabilities that are not possible in the current IPv4 infrastructure [16]. With the tools that have been presented so far, we can think of new attacks that are not possible when the identifier and locator functions are integrated in a single IPv4 address, as it is the current case. A *redirection attack* consists in abusing of the multihoming mapping mechanisms to create false identifier-to-locator mapping. For example, this can be used to induce a victim to associate one of the attacker's locators to the target identifier. Consequently, when the victim sends packets to the target identifier, he will actually be sending packets to the attacker.

The main obstacle in defining a mechanism for the management of multiple locators in multihoming environments has been the provision of the appropriate security level. Mechanisms based on cryptography and cryptographically generated addresses have been proposed to avoid identity theft [17][18], but these solutions suffer from the high computational cost of performing asymmetric key operations, cost that can be unacceptable in scenarios such as a server with a large number of requests per second. Recently it has been proposed a mechanism, named Hash Based Addresses (HBA) [19], which protects the relationship between a set of locators with an identity without incurring large computational costs. In this proposal, a multihomed host Host1, located in a network with different prefixes corresponding to different providers, generates interface identifiers (the 64 less significant bits of the IPv6 address) for its own addresses by performing a hash of all the available prefixes. In this way, a 'signature' obtained from the prefixes assigned to the host is included in all its addresses. When a corresponding host Host2 establishes a communication using a particular address of Host1 (obtained**,** for example**,** from the DNS), and Host2 receives by means of the multihoming protocol the alternative locators of Host1, Host2 can check that the received locators are legitimate. To do so, Host2 performs a hash of the prefixes of the locators, which should generate the interface identifier of the address originally used for establishing the communication. An attacker would require in the order of 2^63 operations (due to the number of bits of the hash) to obtain a set of prefixes different from the initially specified that fulfil the hash check and at the same time include a locator for the attacker.

4 Conclusions

In this article we have presented the mechanisms that

are currently under discussion for the support of multihoming in IPv6 networks. The large availability of IPv6 addresses allows the deployment of multiaddressing configurations, which circumvent the scalability problems that pose current IPv4 solutions. Two sub-problems have been covered: the initiation of new communications in a multiaddressing environment, either with or without failures, and the preservation of established communications in case of a failure.

For the first scenario, it is proposed that there be a modification in the routing system within the multihomed networks to avoid packet discarding due to ingress filtering, and also variations in the Default Address Selection mechanism to allow the exploration of different <source address, destination address> pairs in case of failure. For the preservation of established communications larger changes are required such as a model in which identifiers and locators are split, the definition of a new protocol, new states in the hosts, and the deployment of failure detection mechanisms in the paths. The 128 bits of the length of the IPv6 address allows the inclusion of cryptographic information or a hash of relevant information to provide sufficient security when performing locator redirections.

As opposed to the IPv4 case, IPv6 allows small networks - even residential ones - to benefit from multihoming. Even though there is not yet an outstanding IPv6 application, we expect that IPv6 multihoming support will contribute to IPv6 success.

Acknowledgements

This work was supported by the SAM (Advanced Servers with Mobility) project, funded by the Spanish National Research and Development Programme as TIC2002-04531-C04-03.

References

[1] G. Huston. Commentary on Inter-Domain Routing in the Internet. RFC 3221, December 2001.

[2] J. Abley, K. Lindqvist, E. Davies, B. Black, V. Gill. IPv4 Multihoming Practices and Limitations. draft-ietf-multi6-v4-multihoming-03, January 2005.

[3] C. Labovitz, A. Ahuja, A. Bose. Delayed Internet Routing Convergence, SIGCOMM 2000, 2000.

[4] F. Guo, J. Chen, W. Li, T. Chiueh. Experiences in Building a Multihoming Load Balancing System. IEEE Infocom 2004.

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

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

[7] Site Multihoming in IPv6. <http://www.ietf.org/html. charters/multi6-charter.html>, mail archive in <http://ops.ietf.org/lists/multi6>.

[8] R. Draves. Default Address Selection for Internet Protocol version 6 (IPv6), RFC 3484, February 2003.

[9] P. Ferguson, D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2267, January 1998.

[10] M. Bagnulo, A. García Martínez, J. Rodríguez, A. Azcorra. The Case for Source Address Dependent Routing in Multihoming, Proceeding of the First Workshop on QoS Routing, October 2004.

[11] C. Huitema, R. Draves, M. Bagnulo. Ingress filtering compatibility for IPv6 multihomed sites. Internet draft, October 2004.

[12] Multihoming L3 Shim Approach. E. Nodmark, M. Bagnulo. draft-ietf-multi6-l3shim-00.txt, January 2005.

[13] M. Bagnulo, J. Arkko. Functional decomposition of the M6 protocol. draft-ietf-multi6-functional-dec-00, December 2004.

[14] Failure Detection and Locator Selection in Multi6. J. Arkko. draft-ietf-multi6-failure-detection-00, January 2005.

[15] Multi6 Application Referral Issues. E. Nodmark. draft-ietf-multi6-app-refer-00.txt, January 2005.

[16] J. Abbley, B. Black, V. Gill. Goals for IPv6 Site-Multihoming Architectures. RFC 3582, August 2003.

[17] R. Moskowitz P. Nikander P. Jokela, T. Henderson. Host Identity Protocol. draft-ietf-hip-base-00, June 2004.

[18] T. Aura. Cryptographically Generated Addresses (CGA). draft-ietf-send-cga-06, April 2004.

[19] M. Bagnulo. Hash Based Addresses (HBA). draft-ietf-multi6-hba-00, December 2004.