IPv6 Ingress filtering in a multihoming environment

Etienne Gallet de Santerre, Laurent Toutain

IPv6 Ingress Filtering in a Multihoming Environment

Etienne Gallet de Santerre
Institut TELECOM ; TELECOM Bretagne ; RSM
Université européenne de Bretagne, France
Email: etienne.galletdesanterre@telecom-bretagne.eu

Laurent Toutain
Institut TELECOM ; TELECOM Bretagne ; RSM
Université européenne de Bretagne, France
Email: laurent.toutain@telecom-bretagne.eu

I. Motivations

Many companies are connected to the Internet through several Internet Service Providers (ISPs). This practice, known as multihoming, increases the communication capabilities of companies, enabling link-failure-resistant connection, load sharing and redundancy. In IPv6, multihomed sites generally have several global prefixes for their networks, which means several global addresses for each end-host of a site. When these multihomed end-hosts try to communicate with another site, the communication can be filtered by the provider edge router because of ingress filtering (the packet is dropped if the ISP has not delegated the packet's source address prefix [1], [2]). It results in an unjustified packet loss and a subsequent delay in packet transmission.

We propose to solve the ingress filtering issue in an IPv6 multihomed edge network with the Selection of the Default-route according to the Source Address (SDSA) of a packet. SDSA enhances the routing protocol in an edge network because it takes into account the source address of a packet in the routing decision. As it does not require major modifications in edge networks, SDSA is easy to deploy and brings immediate benefits. Moreover, it could be possible to couple SDSA with other solutions (*e.g.* solution providing session survival) to cover all issues that multihoming rises.

II. Related work

In IPv4, multihomed sites have a Provider Independent (PI) prefix attributed by regional registries. Those PI prefixes are advertised by ISPs to the core network, which drastically increases the core routing tables already overloaded [3].

Recently, the idea of separating the locating and the identifying role of the IP address has risen [4]. In some extent, the Loc/ID split concept can solve the ingress filtering issue. Indeed, if the Loc/ID split occurs in site border routers, the packet's source address changes (rewritten as in [5] or encapsulated) and can be one delegated by the connected ISP.

Another way to avoid the ingress filtering problem is to assure that packets going out the site through an ISP x have a source address prefix delegated by ISP x. Source Address Dependent (SAD) routing [6] proposes to have multiple routing tables in site routers, each one dependent on an ISP prefix. To be precise, a SAD router chooses the routing table in function of the packet's source address. Then, it makes the routing decision as currently. The SAD routing proposal has two disadvantages. First, it increases, proportionally to the number of ISPs, the memory space needed to store routing information in each router. Second, the construction and the update of several routing tables in each router is very time consuming; especially in a site with unreliable links (*e.g.* wireless links), which needs regular routing table recomputation.

In this paper, we show that this behaviour can be restricted at the default route.

III. A New Proposal for Selecting the Default-Route according to Source Address

Our approach to avoid the ingress filtering problem is to assure that packets, going out the multihomed site, have the correct source address, regarding to the ISP. In that purpose, SDSA takes into account the source address in the site routing decision for outgoing packets, when default-route is involved.

A. SDSA routers

Routers implied in the SDSA selection have two routing tables. The first one, very similar to actual routing tables, lists known destinations (site destinations and some specific external destinations advertised by ISPs) and the next hop to those destinations. This first table does not have any default route. We call this table the *destination table*. The second table contains all the prefixes delegated by the site's ISPs. Each prefix of this table is associated with a next hop (an SDSA router too) and drives packets along a path to the ISP, which has delegated the prefix. We call this second table the *prefix table*. The *prefix table* represents a list of different default-routes, which depend on prefixes. Such a structure has the advantage of separating the routing knowledge to the architecture knowledge. Site topology changes do not impact the *prefix table* (except possibly for next hop) and changes in delegated prefix do not imply a recomputation of the *destination table*.

B. Packet processing

An end-host in a multihomed site sends a packet to a destination in another site. On the path to a site border router, this packet is processed by an SDSA router. First, the SDSA router compares the packet's destination with *destination table*'s items. If it finds a matching entry, it forwards the packet to the next hop and the process ends.

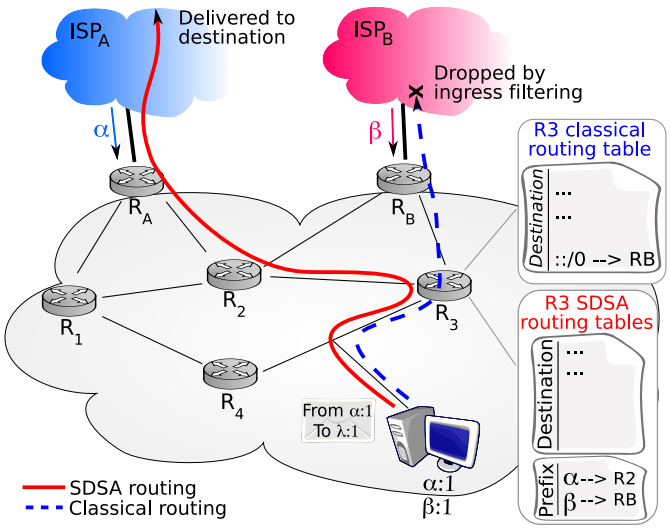

Figure 1. SDSA routing compared to classical routing

If there is no matching item, the packet's source address is compared with the *prefix table*. If no item matches, the packet is dropped, considered as address spoofing. On the contrary, if an item of the *prefix table* matches the packet's source address, the packet is forwarded to the next hop associated with the item. This next hop, an SDSA router too, processes the packet the same way. So, the packet is forwarded along a path of SDSA routers to the border router connected to the ISP, which has delegated the packet's source address prefix.

The current routing decision only depends on the destination address. In a multihomed environment, it can lead to an ingress filtering issue due to the site topology. In the example, Figure 1, router R_3 is the default gateway of the end-host. In a classical routing, R_3 has a default route to R_B. So, every packet from the end-host to an external destination, with an α source address is dropped by ISP_B. On the contrary, with SDSA routing, the packet is forwarded, depending on its source address and reaches the correct ISP network.

C. Network routing evolution

As explained before, SDSA routers have two different routing tables (*destination table* and *prefix table*). Consequently, SDSA requires the modification of the current structure of routing tables. To populate these two tables, the diffusion of routing information needs some modifications too. Currently, a border router advertises internal routes, some specific external routes and a default route. With SDSA, a border router still advertises internal and specific external routes. But instead of specifying a default route, it specifies the prefix which has been delegated by its directly connected ISP. A neighbour of this border router populates its *prefix table* with the advertised prefix and the next hop is the border router address. Other routing information is processed as it is currently done. It is important to note that the routing decision algorithm has not changed, but runs a first time for destination check and a second time, if needed, for source check.

When the routing protocol has converged, SDSA routers know all ISP delegated prefixes and advertised destinations. In that purpose, SDSA requires, at least, that all border routers are SDSA routers and that it is possible to go from one to another along a path of SDSA routers. Thus, a packet to an external destination is inevitably processed by an SDSA router.

D. SDSA Analysis

According to [7], the Buildup Time Complexity (BTC) of a table containing n entries of length k is in $O(nk)$. The Space/Memory Complexity (SMC) of such a table is in $O(n)$. Compared to a classical router, an SDSA has two tables to construct and update. We call k (resp. l) the length of a *destination* table item (resp. *prefix* table item), m the number of ISPs and n the number of advertised routes. We can make the assumption that $k \sim l$ and that $m \ll n$. For the SDSA proposal, the complexity is the sum of each table complexity. Consequently, the SDSA BTC and SMC are given by:

$$BTC_{SDSA} = O(ml + nk) = O(nk) \quad (1)$$

$$SMC_{SDSA} = O(m + n) = O(n) \quad (2)$$

The time complexity of the construction and the update of SDSA tables is equivalent to current complexity.

IV. Conclusion and Perspectives

SDSA is a very simple mechanism, capable of solving one main problem in IPv6 multihomed environment, namely, the ingress filtering issue. As it requires minor modifications to the network behaviour and does not increase the process complexity, SDSA gives immediate benefits to first adopters. It decreases the packet loss ratio, since it routes packets to the ISP which delegated their source address prefix.

Further work is intended to focus on the combination of SDSA with other mechanisms providing multihoming capabilities (session survival, redundancy) such as SHIM6 [8] or SCTP [9]. Moreover, as multihoming and mobility have similar behaviour on specific points, the study of SDSA in a mobile environment is an envisioned perspective.

References

[1] P. Ferguson and D. Senie, "Network Ingress Filtering : Defeating Denial of Service Attacks which employ IP Source Address Spoofing," May 2000, RFC2827.

[2] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks," Mar. 2004, RFC3704.

[3] T. Bu, L. Gao, and D. Towsley, "On characterizing BGP routing table growth," *Computer Networks*, 2004.

[4] D. Farinacci, V. Fuller, D. Oran, and D. Meyer, "Locator/ID Separation Protocol (LISP)," Dec. 2008, draft-farinacci-lisp-11.

[5] C. Vogt, "Six/One: A Solution for Routing and Addressing in IPv6," Nov. 2007, draft-vogt-rrg-six-one-01.

[6] M. Bagnulo, A. Garcia-Martinez, J. Rodriguez, and A. Azcorra, "End-site routing support for IPv6 multihoming," *Computer Communications*, Sep. 2005.

[7] M. Waldvogel, G. Varghese, J. Turner, and B. Plattner, "Scalable high speed ip routing lookups," in *SIGCOMM '97: Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication*, 1997, pp. 25–36.

[8] E. Nordmark and M. Bagnulo, "Shim6 : Level 3 Multihoming Shim Protocol for IPv6," Oct. 2007, draft-ietf-shim6-proto-09.

[9] R. Stewart, "Stream Control Transmission Protocol," Sep. 2007, RFC4960.