Autonomic Renumbering in the Future Internet

Frédéric Beck and Isabelle Chrisment, INRIA
Ralph Droms, Cisco
Olivier Festor, INRIA

ABSTRACT

IPv6 is an essential building block of the evolution toward the future Internet. To take the full benefit of this protocol and exploit all its features, the future Internet needs to gracefully couple it with autonomics. In this article we demonstrate through our experience with network renumbering how the coupling of both IPv6 core functionality extended with major functions of the autonomic world can lead to fully autonomous activities of main management functions. We instantiate the notions of self-configuration, self-monitoring, self-protection, and self-healing in the network renumbering process and show how all together they can make renumbering a real success. We illustrate the various functions with the tools we have implemented to support them.

INTRODUCTION

Designing the building blocks of the future Internet is today one of the most important challenges for the networking community. Visions are flourishing, encouraged by many large initiatives like GENI[1] in the United States and the Future Internet initiative[2] part of the European 7th Framework Program.

Two main approaches have emerged as the leading paths to the future Internet: one advocating a clean slate, revolutionary approach to design the future Internet, and a second one advocating an evolutionary approach extending current protocols with the required capabilities to address all issues of the future Internet.

In this article we follow the evolutionary path and address one specific management function enabled by the IPv6 protocol: renumbering. Network renumbering is a very interesting feature of IPv6, which offers very promising perspectives in terms of automated network adaptation through self-configuration; it is also one of the most risky procedures, which needs a special attention in the management plane. Despite a decade of standardization, this essential function still needs work, as reaffirmed recently by Carpenter *et al.* in [1]. To enable full automation of this function, several self-management functions have to be added to the network to complement the basic functionality offered by IPv6 for this task, mostly advertising. As described in [2], self-management involves the interaction of five functional blocks:

- Self-configuration at the device level and self-organization at the network level to adapt configuration to context
- Self-protection to recognize and avoid threats (distributed denial of service [DDOS], heat, power failure)
- Self-healing to diagnose abnormal operation and take actions to normalize behavior
- Self-optimization to continuously improve system performance
- Self-monitoring to continuously collect state and context information

Additional strata, often transversal to the above listed functions, are commonly agreed upon in the autonomic networking communities nowadays like network awareness [3], which, extended with self-discovery and analysis features, leads to autognostics.

While the network renumbering service is enabled by the Neighbor Discovery Protocol (NDP) built in IPv6, we show in this article that without adding the previously listed functions to the management plane, renumbering will not guarantee the integrity and safety of the concerned network. In addition to motivating the support provided by autonomic features in the renumbering process, we describe the services that can support them and show, through the tools we have built, how to aid IPv6 network renumbering to become really autonomic and thus an essential building block of the future Internet.

The article is organized as follows. The next section is dedicated to the renumbering service and self-configuration issues linked to it. The problems that arise if the function is not addressed in a holistic way are also presented in this section. We then address the self-monitoring part of the solution that leads to autonomic IPv6 renumbering. Self-protection and self-healing in network renumbering are the subject of the following section. The following section contains a synthetic view of the tool support that can cover the self-functions for IPv6 renumbering. The final section concludes the article and draws some directions for future work.

[1] *http://geni.net/*

[2] *http://www.future-internet.eu/*

0163-6804/10/$25.00 © 2010 IEEE

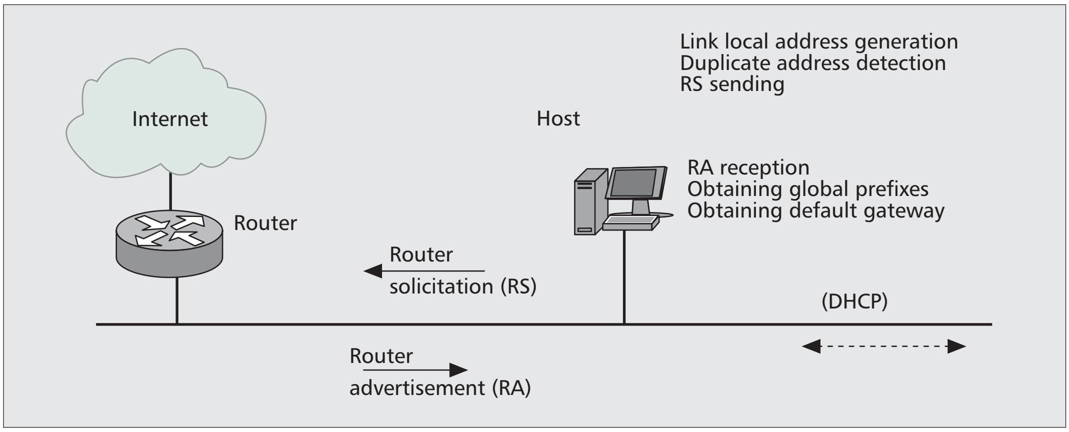
Link local address generation
Duplicate address detection
RS sending
Internet
Host
RA reception
Obtaining global prefixes
Obtaining default gateway
Router
Router
solicitation (RS)
(DHCP)
Router
advertisement (RA)

Figure 1. *Auto-configuration mechanism.*

Renumbering can be triggered by many different events like changes in the internal topology, a network merge, uplink prefix changes due to the migration towards a new provider, or to a dial on demand. The frequency of these events varies and can complicate the task of the administrator.

Self-Configuration and Renumbering

IPv6 network renumbering is a procedure in which all devices located on a subnet change their IPv6 prefix addresses. The effects of renumbering can be reduced by replacing an old prefix by a new prefix without a flag day, permitting a period of time where both prefixes are in use. Services transit from the old prefix to the new prefix by simultaneously using the old prefix for existing service connections while using the new prefix for any new service invocation. The first function to ease this feature is to automate IP address configuration. In this section we show how the self-configuration building block is applied through the NDP [4]. We also point out that self-configuration does not avoid some problems that can lead to the failure of the renumbering procedure.

IPv6 Network Renumbering

Renumbering can be triggered by many different events like changes in the internal topology, a network merge, uplink prefix changes due to the migration toward a new provider, or dial on demand. The frequency of these events varies and can complicate the task of the administrator. Several investigations have already been made on this topic within numerous projects such as 6net[3] in Europe. The major challenge is to achieve transparent renumbering and avoid disruptions for users. To reach this aim, a procedure in eight steps has to be followed [5]:

1. There must be a stable and working situation with an existing prefix (old prefix).
2. Obtain the new prefix and new reverse zone from the delegating authority.
3. Set up a parallel routing architecture for the new prefix.
4. Hosts' addressing: the new prefix is announced.
5. Have a stable configuration where the network is multihomed (with two or more addresses). IPv6 explicitly allows multiple prefixes to be assigned to a link simultaneously.
6. The old prefix is obsolete (lifetimes set to zero). The transition from the old to the new prefix for services can be made.
7. Remove the old prefix. Addresses for the old prefix are deleted from the hosts' interfaces.
8. This is equivalent to the first state, but using the new prefix.

One cause of renumbering is a change in the network numbering architecture, which results from moving to a new service provider. Multihoming in IPv6 is a useful feature to minimize the impact of a renumbering event, but is not sufficient to solve the renumbering issue. Multihoming enables multiple connections to different providers but has important issues to deal with like ingress filtering and induced explosion of routing tables sizes. When a network is renumbered, the multihoming phase is used for a very limited time (as short as possible) as the IPv6 prefix has to be changed.

Anycast can also be considered as a useful service to ease renumbering. This, however, does not work since either anycast addresses are taken from the unicast address space and, as such, depend on renumbering; or, if their prefix is not bound to the unicast space, they must be announced as a separate route, and thus lead to an unwanted increase in routing tables. Even if it would be useful on a single site, its use for host address assignment is prohibited in RFC 4291 [6].

The IPv6 NDP is used to announce the assignment of IPv6 prefixes in a network, replacing IPv4 Router Discovery [7] and Address Resolution Protocol (ARP) [8]. It also adds new functionalities such as IPv6 stateless address auto-configuration [9]. Neighbor Discovery allows a network administrator to specify the prefixes assigned to a link, and enables a node to automatically configure the addresses and routes, simply by being connected to the network link.

As illustrated in Fig. 1, the router solicitation (RS) message is sent by a node to ask for information about the on-link routers and prefixes. The router replies with a router advertisement (RA) providing the default gateway's address, the router's validity, the list of IPv6 prefixes the router handles, maximum transfer unit (MTU), Mobile IPv6 options, and a lot of other information. Additionally, the router can send an unsolicited RA message, which can be used to announce new prefixes or deprecate existing prefixes, as in steps 4 and 6 of the renumbering procedure. Using RS and RA messages, a node can

[3] *http://www.6net.org*

dynamically configure addresses and other information about its interface.

A host using the stateless auto-configuration facility has to control whether the obtained address is already used in the network through a duplicate address detection mechanism. If stateful auto-configuration is preferred, a Dynamic Host Configuration Protocol (DHCP) server will be contacted.

A host using the stateless auto-configuration facility has to control whether the obtained address is already used in the network through a duplicate address detection mechanism. If stateful auto-configuration is preferred, a DHCP server will be contacted.

ISSUES

To successfully complete the renumbering process, more changes are required in both nodes and routers, including the following.

Manually Configured Hosts/Hard Coded Addresses — Addresses often appear to be hard coded parameters in configuration files and even in some applications' source code. These addresses thus have to be manually updated to move from the old prefix to the new prefix because address auto-configuration will not be performed on these services when a new prefix is announced. After such a modification, the service or tool usually must be restarted to reload the proper configuration file. This can lead to situations where the host becomes unreachable, and all the associated services are down.

DNS — As hosts renumber their interfaces, their corresponding resource records in the DNS database must be updated. Specifically, as the new prefix is enabled in step 4 of the renumbering process, the new IPv6 addresses assigned by each host must be added to the hosts' IPv6 records (type AAAA). Similarly, as the old prefix is deprecated in step 6, the addresses from the old prefix must be removed from the hosts' AAAA records. The time to live (TTL) for the deprecated addresses must be coordinated so that these addresses expire from DNS caches at the appropriate time.

Similarly, problems can appear for services that make a single resolution (usually at startup), or for applications that bind to a specific address which is subsequently modified during renumbering. In such cases the service or application must be restarted to become operational again after renumbering.

Data Continuity Problems — Some monitoring applications store information on the evolution of a host in order to update statistics. After renumbering, if monitored hosts are not properly identified (e.g., when the address is used for this purpose), data continuity problems arise.

Security — When the routing architecture for the new prefix is being set up, the network parts running the routing protocol are vulnerable to attacks, as the access control lists (ACLs) are not set for the new prefix yet. For example, when routing is activated on the data management zone (DMZ) and the servers are renumbered, the web or mail servers will still be protected against illegal access for the old address, but if ACLs for the new address are not set at the right time, the servers become vulnerable for this newly assigned address. The routing infrastructure itself can be compromised even before the hosts are renumbered. When routing is activated for the new prefix, the routers have already assigned addresses corresponding to the new prefix. Thus, they are in the same situation as the servers mentioned in the previous example, and are likely to suffer various types of attacks or intrusions. To protect them, the first thing to do before advertising the prefix is to update both ingress and egress access lists by including rules for the new prefix.

Given the many specific cases in which renumbering affects network operation that are not solved by automated address assignment, self-configuration is not sufficient to maintain a coherent and operational network through a renumbering event. In order to prevent as many problems as possible, the network administrator must, prior to finalizing the renumbering, achieve some tasks like performing the renumbering of all the routers and switches or adjusting the prefixes' lifetimes in router advertisements, DHCP leases, DNS entries validity in caches, and so on. To automate these operations, strong coordination among the entities is required, and a global view of the network is necessary so as to adapt the configuration to the global policy of the organization.

SELF-MONITORING

IPv6 network renumbering is a complex procedure that has to follow several steps. As shown in the previous section, applying self-configuration is only part of the renumbering process. Each step must be successful and validated before moving to the next one. Following the procedure manually is fastidious, especially when the renumbered network is composed of an important number of end hosts. Hence, a tight connection between self-configuration and self-monitoring is required. Through self-monitoring, a network view is continuously provided, and allows an autonomic manager to orchestrate and optimize the whole procedure.

The autonomic manager collects information about individual hosts, and can infer the global state of the renumbering procedure as well as the state of services running on the renumbered network to ensure that those services are functioning properly as the renumbering takes place. This information can be collected from an agent on each host that communicates with a renumbering manager, as described later. Based on the manager's knowledge of the preconditions and post-conditions of each step in the renumbering procedure, the autonomic manager can monitor the progress through each renumbering step and validate the transition from each step to the next. In the occurrence of problems or because of local policies, the autonomic manager can request operator intervention and/or administrator confirmation before proceeding to the next step.

Self-monitoring becomes extremely important in step 5 of the renumbering procedure. In this step the network is in a stable situation where both the old and new prefixes are available. This step marks the beginning of the transition phase between the two prefixes. The transition phase ends when all devices stop using the old prefix for both their outgoing and incoming communications. Self-monitoring helps to determine exactly the end of this transition, ensuring that all the hosts do not use the old prefix anymore, thus avoiding service outage. Self-monitoring optimizes the transition phase duration and contributes to reduction in cost of the maintenance of both Internet connections by enabling the old one to be released earlier.

During the whole procedure, the autonomic system must check that the running services on the network remain accessible in accordance with the procedure. For example, during steps 1 to 4, a web server should answer only to its old address, whereas during steps 5 to 7, until the old address is suppressed, it should respond to both addresses, and finally, in step 8, only work with the new address. Self-monitoring allows the participating entities to test among themselves the availability of the services and to ensure the accessibility of services before validating the transition to a new step. This monitoring can be done in a fully distributed way.

Some problems can automatically be detected during the renumbering procedure since the autonomic manager infers a global state of the procedure and orchestrates the transitions between the different steps. For example, the manager validates the transition between steps 4 and 5 once all the monitored hosts are in a multihoming situation. If some hosts do not begin to use an address for the new prefix, it usually means that they are not using stateless auto-configuration. If they are using stateful auto-configuration with DHCPv6, the manager may trigger a reconfiguration, by either telling the DHCP server to send a reconfigure message to the host or directly triggering the process on the host via the monitoring agent. If the host is using static addressing, the agent may also reconfigure the addressing locally, while following the directions given by the manager. If the problem concerns a service that is not available for the new address, or any long-term session, such as SSH or NFS, which prevents the transition step to end, the system may not always be able to react automatically, as the problems are too specific or human related. In this case an alert and guidelines for solving the problem should be sent to the administrator, who would then take the appropriate measures. If the issue is blocking, the autonomic system should be able to revert to a previous working and safe configuration upon decision of the administrator.

Self-Protection and Self-Healing

So far, we took the assumption that all hosts on our network were healthy and behaving as they are supposed to. In order to reach our goal of autonomic renumbering, we now have to consider that some hosts on the network can be malicious. To avoid these nodes from harming the network, it must be protected, and the troubles created by malicious nodes must be corrected. This process commonly comprises both self-protection and self-healing functions.

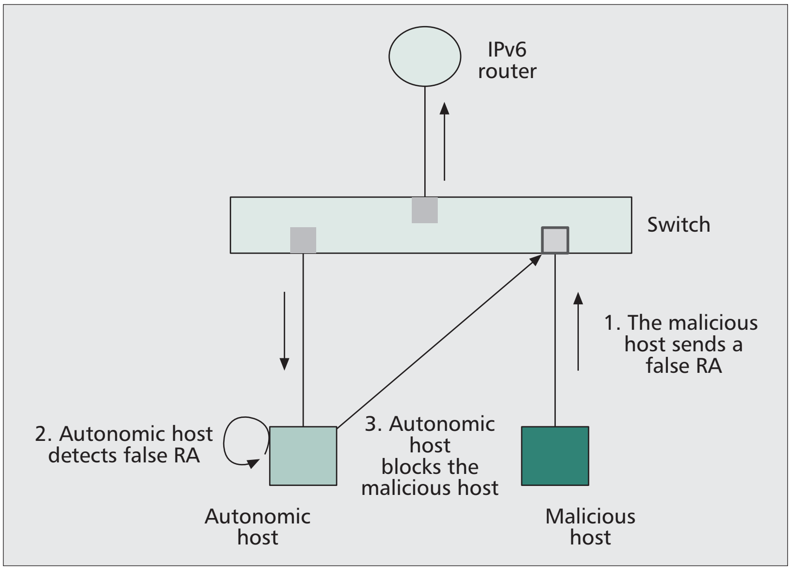
IPv6 router
Switch
1. The malicious host sends a false RA
2. Autonomic host detects false RA
3. Autonomic host blocks the malicious host
Autonomic host
Malicious host

Figure 2. *Self-protection in network renumbering.*

Self-Protection

In the context of renumbering, self-protecting means that functions in the network should detect attacks or misconfigurations that compromise the network and the renumbering procedure. Typically, an attack consists of sending a bogus RA during step 4, 6, or 7 of the renumbering procedure, causing trouble and errors in the addressing of devices or their routing tables. This bogus RA may contain false lifetimes for a prefix involved in the renumbering or a fake prefix.

As shown in Fig. 2, an autonomic device should detect this bogus RA and identify the source of the message. This detection can be done via a tool like NDPMon, described in the next section, which uses knowledge of the legitimate RA and routers. Another option is to use signed RAs to avoid this problem (as defined in the Secure Neighbor Discovery [SEND] protocol), but this option requires the deployment of a complex architecture, including certificate authorities and usage of compliant nodes and routers. Moreover, the usage of IPSec as a building block of SEND introduces additional processing of the RA, which may be harmful for real-time communications. In our architecture we opted for the first option, because we believe that this is the situation we will encounter in most cases. After detecting a bogus RA, the autonomic system, via the autonomic manager, or via any other device upon decision of the manager, as shown in Fig. 3, will reconfigure the network entity/ies in order to place an ACL directly in the network component to block malicious traffic from the bogus system on the network, to stop the attack and prevent any further ones originating from the same device.

Self-protection during a renumbering procedure not only addresses misconfiguration and attacks against the renumbering procedure. As this process is based on the NDP, all attacks against this protocol, as described in [10], have potential to harm the procedure and the network.

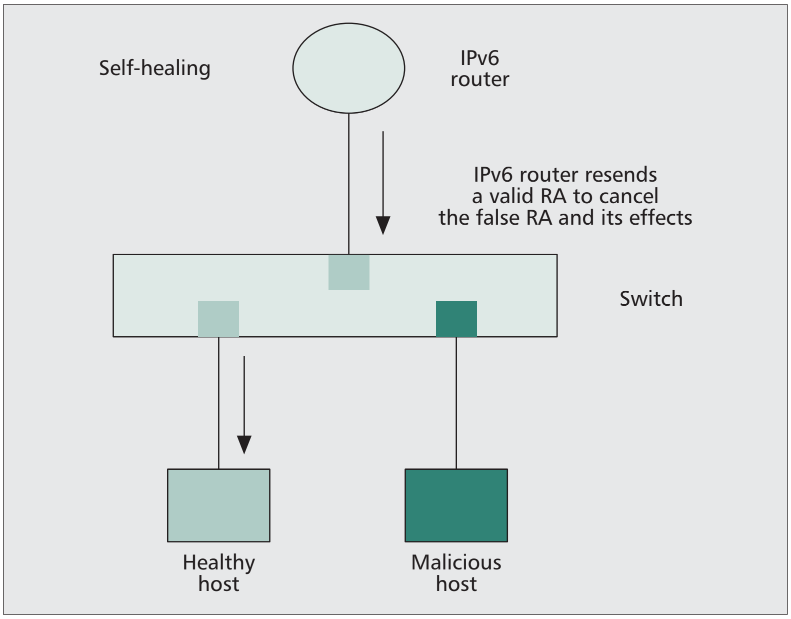
Self-healing
IPv6 router
IPv6 router resends a valid RA to cancel the false RA and its effects
Switch
Healthy host
Malicious host

Figure 3. *Self-healing during network renumbering.*

SELF-HEALING

Once a problem has been detected and the attacked device has protected itself, the system must have the ability to recover from the attack by either finding an alternative way of using resources or reconfiguring itself to keep functioning smoothly.

If we keep the same example as for self-protection, the damages the network underwent must be corrected so that the network reverts to its original and valid state. As the issuance of an invalid RA was the cause of the trouble, the healing procedure consists of sending a new valid RA, which will annihilate the effects of the bogus one. If the RA contained wrong information about a valid prefix, resending the valid RA is sufficient, as shown in Fig. 3.

If the RA did advertise a false prefix, we need to forge an equivalent RA, but with modified lifetimes (preferably very short ones), to ensure that all traces of the false prefix disappear as fast as possible.

The countermeasure to be taken while performing self-healing depends on the problem detected by the self-protection operation. These two aspects of an autonomic system are closely linked in the autonomic renumbering process.

Moreover, all problems or potential problems detected and solved should be logged, and the administrator should be warned via an alarm system. He will thus be able to validate the changes, correct the misconfiguration, and take further measures to ensure that the malfunctions do not reappear, especially if they are caused by the intrusion in the network of an attacker.

TOOLS SUPPORT

All components of the autonomic renumbering engine were implemented in an integrated architecture. In this section we present our architecture and the tools that can be combined to build the autonomic renumbering solution. We also address the necessary interoperability issues.

MONITORING THE RENUMBERING PROCEDURE

The monitoring part of the architecture is composed of three entities: management agents, one renumbering manager, and service renumbering probes.

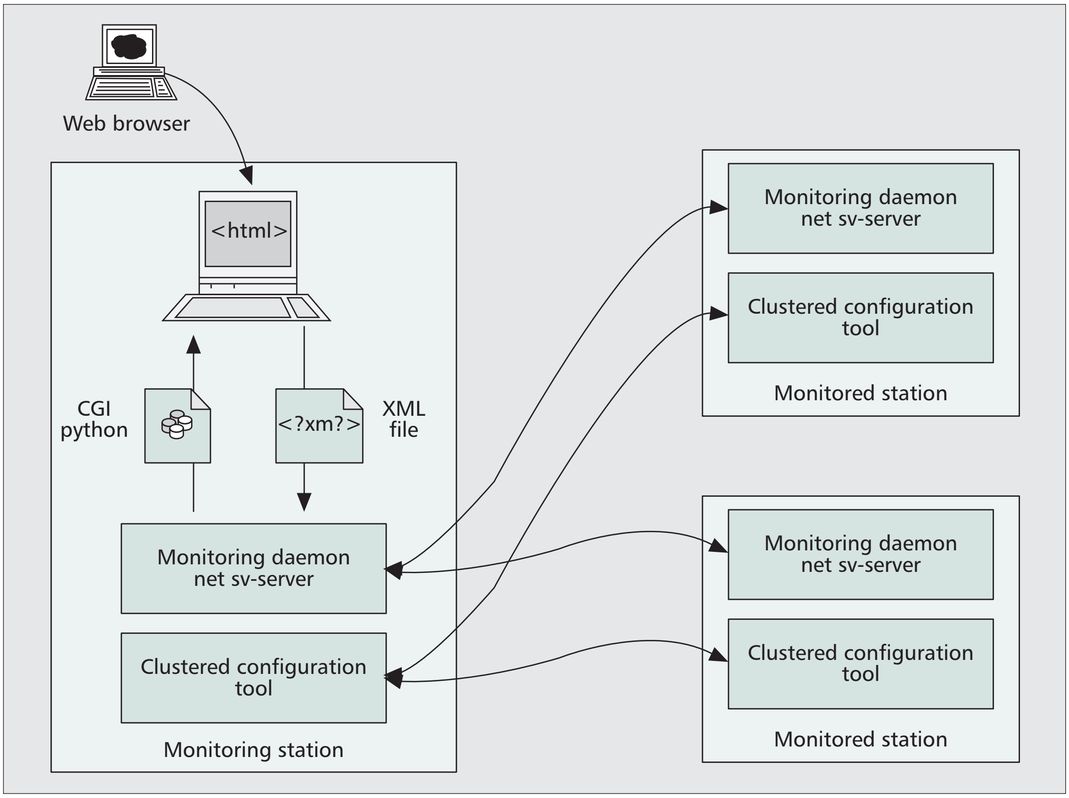
Web browser
<html>
CGI python
<?xm?>
XML file
Monitoring daemon net sv-server
Clustered configuration tool
Monitoring station
Monitoring daemon net sv-server
Clustered configuration tool
Monitored station
Monitoring daemon net sv-server
Clustered configuration tool
Monitored station

Figure 4. *NetSV architecture.*

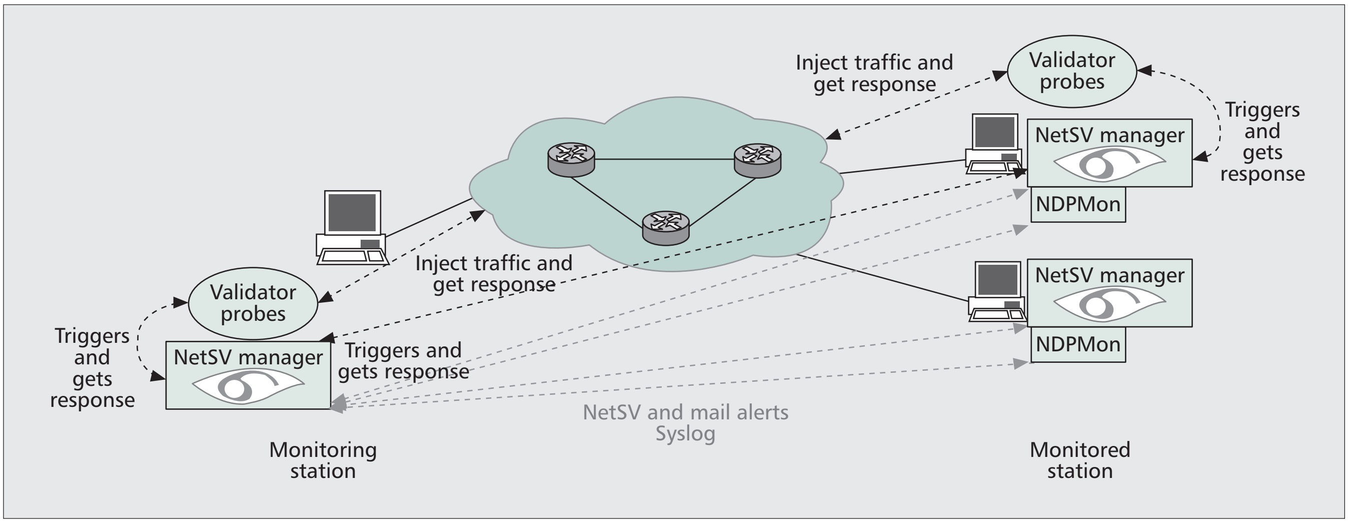
Inject traffic and get response
Validator probes
Triggers and gets response
NetSV manager
NDPMon
NetSV manager
NDPMon
Inject traffic and get response
Validator probes
Triggers and gets response
NetSV manager
Triggers and gets response
NetSV and mail alerts Syslog
Monitoring station
Monitored station

Figure 5. *Self-monitoring framework.*

Monitoring agents are deployed either within systems or at strategic monitoring points. They are able to diagnose whether the monitored devices did renumber properly or, if not, why. The availability of such agents is important since it provides the necessary semantics to help control the loops to take appropriate decisions.

A manager can be physically distributed among several entities. It is in charge of the validation of the renumbering triggers (router advertisement, administrators' orders) and orchestrates the renumbering process.

To complement the agents and the manager, we implemented a set of probes. Their task consists of permanently monitoring the health of the services while the renumbering process is going on.

This architecture (with a simplified central manager service) is implemented in a tool called NetSV,[4] which provides a monitoring service for the renumbering and validates the addressing of hosts. NetSV also provides continuous diagnosis on the monitored devices by preventing and/or detecting problems that could occur during such an operation.

NetSV is divided into three elements, as shown in Fig. 4:

- A daemon running on monitored devices (or remotely representing monitored devices) sending information on the local addressing to the monitoring host
- A daemon running on the monitoring host using the information sent by the agents to maintain a global state of the renumbering procedure
- A diagnostic tool, independent of the other blocks, taking care of service checks and diagnostics on monitored hosts

With this tool, an administrator can follow the whole renumbering procedure and validate each step.

Monitoring the Neighbor Discovery Protocol

Monitoring the devices is not sufficient. The principal renumbering triggering protocol, the NDP, also needs to be monitored. Therefore, we implemented a tool called NDPMon,[5] the NDP Monitor, similar in its basic functions to the IPv4 ArpWatch.[6] It is in charge of monitoring NDP activities and maintains an up-to-date neighbor database, which contains the correspondence between IPv6 and Ethernet addresses, alongside a timestamp. In the same way as in ArpWatch, activities and suspicious behaviors raise alerts and reports.

Alerts and reports can be sent over various transport systems such as mail and syslog. In addition to its monitoring role, NDPMon is able to detect attacks against the NDP, as defined in [10]. Misconfiguration, stack vulnerabilities, and suspicious behaviors are well addressed in NDPMon.

NDPMon operates in two phases: a learning phase and a monitoring phase. During the learning phase, NDPMon builds the neighbors database by capturing the neighbor discovery messages, and, based on the RAs received, it populates the routers list, while making the assumption that when it enters the learning phase, the network is healthy. This phase is run only once, and requires close attention from the administrator. Once this phase is over, the tool can switch to the monitoring mode.

Tools Interoperability

Although NDPMon and NetSV appear to have some similarities in host monitoring functions, they are, in fact, complementary. While NetSV monitors the renumbering procedure and validates the addressing of end hosts, NDPMon monitors the NDP activities and detects attacks against this protocol. Figure 5 shows how these tools can cooperate to enable full self-monitoring of a renumbering process.

For example, when the old prefix is supposed to be suppressed from advertisements, an attacker can still issue a forged RA after the router that managed the old prefix has stopped operating this prefix. In that case NetSV agents will not detect the forgery of the RA for the old prefix, and will not send any report to the manager. The manager will thus not detect the end of the seventh step of the procedure. With NDPMon

[4] *http://netsv.sourceforge.net*

[5] *Like NetSV, NDPMon is distributed freely. It can be downloaded in both source or binary form for various platforms and distributions from http://ndpmon.sourceforge.net*

[6] *http://www-nrg.ee.lbl.gov/*

We remain convinced that the future of autonomics in the networking sphere will focus and be successful only if applied on well identified and limited control loops like it was done here for the renumbering function.

deployed on the network, it will send an alert to the management station with information about the attacker (medium access control [MAC], IPv6 address, etc.), which can be used to neutralize it and activate the protection and healing functions.

Reconfiguring the Network

As in our study case, the autonomic system is the network itself; configuring, protecting, and healing operations mean in the end reconfiguring network equipment. Currently the platform supports Command Line Interface (CLI) configuration interfaces over Telnet or SSH and has been tested against several heterogeneous types of equipment. Future interfaces with Netconf [11] are foreseen once an integrated data model is available for this technology.

Conclusions

A major issue toward the acceptance of IPv6 as one of the key building blocks of the future Internet is its capacity to offer real autonomic behavior. In this article we focus on enabling fully autonomic network renumbering. Based on experience in various networks and development of distributed solutions for renumbering issues in the management plane, we have shown that to become fully autonomous, the current renumbering service must be backed up by distributed monitoring, orchestration, protection, and healing. To address these functions we have designed a distributed management scheme and implemented the supporting code that offers an integrated approach to the problem.

This work was illustrated with a concrete problem in one IPv6 automation feature. The architecture we developed to address the IPv6 renumbering issue, while generic in design, has not yet been extended to other application cases. As part of our future work, we will investigate how to adapt our architecture to autonomic network architectures designed from a top-down approach. We remain convinced that the future of autonomics in the networking sphere will focus and be successful only if applied on well identified and limited control loops as done here for the renumbering function. A further step is to assess the autonomic infrastructure against malicious behavior.

Acknowledgment

The work reported in this article was partially supported by the Cisco CARD collaborative initiatives program. The latest developments of the NDPMon toolkit were partially supported by the FP6 IST-EMANICS Network of Excellence. We would like also to thank Fred Baker and Tony Hain for their valuable feedback.

References

[1] B. Carpenter, R. Atkinson, and H. Flinck, “Renumbering Still Needs Work,” May 2009.

[2] IBM, “An Architectural Blueprint for Autonomic Computing,” White Paper, June 2005.

[3] E. Hughes and A. Somayaji, “Towards Network Awareness,” *Proc. LISA ’05*, 2005.

[4] T. Narten, E. Nordmark, and W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6),” IETF RFC 2461, draft standard, Dec. 1998.

[5] F. Baker, E. Lear, and R. Droms, “Procedures for Renumbering an IPv6 Network without a Flag Day,” IETF RFC 4192, Informational, Sept. 2005.

[6] R. Hinden and S. Deering, “IP Version 6 Addressing Architecture,” IETF RFC 4291, draft standard, Feb 2006.

[7] S. Deering, “ICMP Router Discovery Messages,” IETF RFC 1256, Sept. 1991.

[8] D.C. Plummer, “An Ethernet Address Resolution Protocol, “IETF RFC 826, STD37, Nov. 1982.

[9] S. Thomson and T. Narten, “IPv6 Stateless Address Autoconfiguration,” IETF RFC 2462, draft standard, Dec. 1998.

[10] P. Nikander, J. Kempf, and E. Nordmark, “IPv6 Neighbor Discovery (ND) Trust Models and Threats,” IETF RFC 3756, Informational, May 2004.

[11] R. Enns, Ed., “NETCONF Configuration Protocol,” IETF RFC 4741, Standards Tracks, Dec. 2006.

Additional Reading

[1] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” IETF RFC 2460, draft standard, Dec. 1998.

Biographies

Frédéric Beck (frederic.beck@loria.fr) holds an M.Sc. degree in networking and telecommunication from Université Louis Pasteur, Strasbourg, France. After working at Alcatel as an R&D engineer, he joined INRIA in Nancy as a research engineer in December 2004. His research interests are mainly centered in IPv6 (transition, renumbering, and formerly multicast and mobility) as well as network supervision and security.

Isabelle Chrisment (Isabelle.Chrisment@loria.fr) is a professor of computer science at Ecole Supérieure d’Informatique et Applications de Lorraine (ESIAL), Nancy University, France. She received her Ph.D. in computer science in 1996 from the University of Nice-Sophia Antipolis and her Habilitation degree in 2005 from Nancy University. Since 2002 she is vice-head of the MADYNES team, INRIA/LORIA Lorraine, whose research domain is the management of dynamic aspects provided by networks and services. Her research area is related to network configuration and security.

Ralph Droms (rdroms@cisco.com) is a Cisco Distinguished Engineer in the Research and Advanced Development group, where he focuses on name and address management, IPv6, and development of protocol standards for the smart grid. He supports and participates in research collaborations such as protocol modeling, automated methods for IPv6 deployment and renumbering, and IPv6 mobile ad hoc networks. He has participated in the IETF for many years, having organized the DHC working group in 1989, which he chaired until he was selected to be an Internet Area Director for the IETF in 2009. Prior to joining Cisco, he was on the faculty at Bucknell and Penn State. He has also been on the research staff at both IBM and Burroughs (Unisys). He is a co-author of *The DHCP Handbook*. His Ph.D. is in computer science from Purdue University.

Olivier Festor (olivier.festor@loria.fr) is a research director at INRIA Nancy-Grand Est where he leads the MADYNES research team. He has a Ph.D. degree (1994) and an Habilitation degree (2001) from Henri-Poincare University, Nancy, France. He spent three years at the IBM European Networking Center in Heidelberg, Germany, and one year at the EURECOM Institute in Nice, France. His research interests are in the design of algorithms and models for automated security management of large-scale networks and services. This includes monitoring, fuzzing, and vulnerability assessment. Application domains are IPv6, voice over IP services, and dynamic ad hoc networks. He has published more than 70 papers in network and service management, and serves on the technical program and organization committees as well as the editorial boards of several international conferences and journals. He was the TPC Co-Chair of IFIP/IEEE IM ’05. Since 2006 he has led the EMANICS European Network of Excellence dedicated to Management Solutions for the Future Internet and was named IFIP TC6 Working Group 6.6 co-chair in 2008.