An Evaluation of the NAP Protocol for IPv6 Router Auto-configuration

Guillaume Chelius
INRIA - INSA de Lyon
Villeurbanne 69621 Cedex
France
guillaume.chelius@inria.fr

Eric Fleury
INRIA - INSA de Lyon
Villeurbanne 69621 Cedex
France
eric.fleury@inria.fr

Bruno Sericola
INRIA - IRISA
35042 Rennes Cedex
bruno.sericola@irisa.fr

Laurent Toutain
GET/ENST Bretagne
35576 Cesson Sévigné Cedex
- France
laurent.toutain@enst-bretagne.fr

David Binet
Orange Labs
14000 Caen - France
david.binet@orange-ftgroup.com

ABSTRACT

This paper presents a model of the NAP protocol, dedicated to the auto-configuration of IPv6 routers. If hosts auto-configuration is defined by IPv6, IPv6 routers still have to be manually configured. In order to succeed in new networking domains, a full auto-configuration feature must be offered. NAP offers a fully distributed solution that uses a link state OSPFv3-like approach to perform prefix collision detection and avoidance. In this paper, we present a model for NAP and analyze the average and maximum autoconfiguration delay as a function of the network size and the prefix space size.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network management*

General Terms

Algorithms, Performance

Keywords

IPv6, router auto-configuration, zero-configuration

1. INTRODUCTION

Though auto-configuration has been presented as one of the attractive new features of IPv6, it cannot be assimilated as real plug and play network yet as routers still need to be manually configured. In some environments such as home networking, this is not satisfying as the complete network configuration should be hidden to the user. Even if these networks generally imply a limited number of nodes and links, the topology may remain complex (*e.g.* with redundant links or loops), in particular due to the increasing number of available link layer technologies (Ethernet, Wi-Fi, PLC, IEEE 802.16.4,...) and the lack of network culture of the network user/owner.

To offer a real plug and play networking, IPv6 still suffers from the lack of a router auto-configuration protocol. The goal of such a protocol is to allocate a unique prefix to each of the network link. This allocation should be done inside the prefix space imposed by the Internet provider. In a home network, one or several edge routers may be connected to the Internet. These edge routers establish a dialog with the provider through the DHCPv6 Prefix Delegation protocol[1] and acquire a delegated prefix. Depending on the delegated prefix size, the number of prefixes that can be allocated for the network may be very close to the number of links in the home network. The router auto-configuration protocol must perform efficiently in this condition.
Regarding interconnecting networks, three approaches can be investigated:

- bridging all layer 2 technologies together to form a single link. This allows a simple auto-configuration as IPv6 only sees one link. Edge routers announce their prefix on the bridged link and hosts auto-configure their IPv6 addresses using Neighbor Discovery Protocol (NDP). This approach implies the use of a Spanning Tree Protocol to avoid the effect of loops. Moreover, on a zero configuration network, no quality of service is possible and the traffic may be sent on low bandwidth or over overloaded links leading to bad performances.

- Trill (*Transparent Interconnection of Lots of Links*) [5] is an IETF attempt to define an hybrid component between router and bridge (called RBridge) which modifies the way IP packets are bridged. Instead of using

[1]We will make no assumption on the prefix length. IETF recommends the use of prefixes with a length of 48 or 64 bits, but we can imagine some situations where a provider may allocate a 60 bit long prefix.

IPv6'07, August 31, 2007, Kyoto, Japan.

a Spanning Tree algorithm, a routing protocol based on IS-IS is used to define the best path between two Rbridges. Frames are tunneled between them. RBrigding offers a better convergence time and a better use of the network resources compared to a Spanning Tree. RBridges have to offer broadcast or multicast capabilities among all links but to simplify ARP/ND protocols, RBridges keep a mapping between MAC addresses and IP addresses. As hosts logically belong to a same link, auto-configuration remains as simple as in a bridged network. Even if this solution works for both IPv4 and IPv6 stacks, it implies a lot of modifications in the bridge functionalities especially when interacting with a Spanning Tree Protocol or VLANs. It also implies the addition of some L3 knowledge to the L2 equipments.

- Routing seems the more generic solution. By definition, it is totally transparent to any layer 2 technology supporting IP (e.g. different L2 address spaces or frame sizes) and permits routing optimizations such as load balancing. The drawback of this approach comes from the fact that the internal routers need to be configured. A simple solution is possible if all interfaces are directly connected to a single router. This router acquires the global prefix from the ISP and automatically appends a Subnet-ID value based on the interface number. This model is very restrictive in terms of topology and multi-homing capabilities as different ISPs will have to share the same CPE.

In this paper we focus on a distributed method to configure routers (i.e. automatically allocate a /64 prefix to each link). When this configuration is done, other already standardized mechanisms such as IPv6 Stateless Address Auto-Configuration and DHCPv6 can be used without any modification.

2. CONTRIBUTIONS

In our approach, we suppose that providers connected to the Home Network announce their Global Prefixes (GP) using a protocol such as DHCPv6 Prefix Delegation [6] to the edge routers (CPE: *Customer Premise Equipment*). Several CPE can be located in the Home Network and each CPE can receive several GP. The prefix length of a GP can be different for each allocation. We make no assumption on the home network topology which can be viewed as an arbitary graph.

In [1], we studied the different ways Subnet-IDs may be dynamically allocated by a centralized approach using the DHCPv6 Prefix Delegation of /64 based on the received GP or through a distributed approach. This former approach is multi-homing-compliant and offers more reactivivity to network topology changes.

In this paper, we describe in more details the NAP (*No Administration Protocol*) protocol. The main idea is to propagate prefix informations inside the whole network and establish a consensus on auto-configured unique Subnet-ID values. Our approach consists in modifying current routing protocols to include a new network auto-configuration mechanism. We describe an implementation of this algorithm based on OSPFv3 extensions [4] and we study analytically the auto-configuration speed as well as the stability of automatically allocated prefixes/addresses in an arbitrary network topology using this protocol.

Section 3 focuses on the NAP implementation based on a Link State approach. Section 4 presents an analytical model of our solution and finally, Section 5 concludes the article.

3. NAP PROTOCOL

3.1 NAP Algorithm

The NAP algorithm is very simple. The auto-configuration of a complete network roughly works as follows. In a first step, one or several border gateways (or CPE) acquire global prefixes from a provider. This information is flooded in the network by the CPEs. Upon reception of these prefixes, the designated router of each link chooses locally Subnet IDs to reach the /64 prefix limit. The chosen SID values are flooded to all other routers of the network in order to detect collisions. In case of a collision on a SID value, the router with the smallest ID keeps its SID value and the other colliding routers draw a new value. This new value is drown with the knowledge of already allocated SIDs in order to prevent new collisions. This step is repeated until each link in the network is allocated a unique SID value for each of the delegated GP.

Our implementation of the NAP algorithm is based on OSPFv3. This routing protocol is from the Link State family which requires a database synchronization between all routers. In OSPF, all routers describe their local topology and propagate this information in a reliable manner to the other databases. A router acquires a topological view of the network and computes the shortest path to reach each prefix located in the network. The address of the first router on the path is stored together with the prefix in the Forwarding Information Base (FIB). For use by other applications, OSPF proposes also some opaque types that may be propagated inside the network in the router databases without interacting directly with the routing process. The flooding may be limited to three available scopes: the link, the area and the network. In our NAP implementation, OSPFv3 is mainly used for this reliable flooding mechanism and the resulting database synchronization [2].

The other native property of OSPFv3 that is used to implement NAP is the election of a designated router on each link. On a broadcast media, routers discover themselves using a Hello protocol. Hello messages are also used for the election of a Designated Router (DR) (and a Backup Designated Router). In NAP, the DR is responsible for choosing the SID values for the link, checking its uniqueness and guaranteeing it during the network lifetime. Other routers on the link just follow the configuration messages given by the DR to configure their interfaces.

Three messages (or LSA : Link State Announcement) have been defined to implement NAP over OSPF:

- SID_Link : This message is issued by the designated router to inform other routers of the selected prefixes. The scope is the link. Each DR generates this message containing a set of structures with 4 elements : the GP, its length, the selected SID and its length[3].

[2]Interactions with the routing protocol are also necessary when a consensus is reached on the chosen prefixes to offer a connectivity between the different links.

[3]The SID length is necessary if prefixes are not aligned on

- SID_Area : This message is issued by a designated router to inform all other DRs of its selected prefixes. The scope of this message is the area. Only DR will process this message. In case of conflicts, some DRs will have to renumber their prefixes. It contains the same information as the SID_Link.

- GP_LSA: This message is issued by a border router (CPE) to inform all the routers in the domain of the prefixes learned from the provider. It contains the list of GP values and their lengths given by the provider .

An edge router learn a Global Prefix from a provider, generally through DHCPv6 Prefix Delegation, and floods a GP_LSA to all the routers. If no prefix is available on the network, a default prefix based on the ULA prefix is used and all the routers use this well-known prefix. This default prefix can be the same for all Home Networks, as if two Home Networks merges, NAP renumbers automatically the SIDs to avoid collisions.

Upon reception of a new prefix in a GP_LSA (or not receiving any GP at all), a designated router draws an available random number (*i.e.* a random number that does not already appear in the LS database for that GP) and informs all the routers in the area of the selected value to detect collisions. Collisions occur if two or more DRs draw the same available random number. If no collision appears , the DR announce the selected SID value to the link routers using a SID_link message.

The information carried in SID_Link and SID_Area are the same and the messages differ only in their scope. It is however difficult to merge these two messages. SID_Area is to inform other DRs of SID values that are selected to allow collision detection and avoidance. SID_Link is sent only when the SID value is guaranteed to be unique. A router (including the DR) receiving a SID_link configures its interface using an Interface ID value based on the announced prefix and MAC/EUI-64 address of this interface.

The SID_link is also used to inform other routers of the selected prefix. In case of a DR death, a router is elected as new DR and continues to propagate the same SID value on the link. The new designated router generates a new SID_link and SID_area to communicate the change.

The OSPF flooding and DR election help to detect network partitioning or merging:

- When two links are merged, one of the DRs looses its status and deprecates the prefix by sending a SID_link and SID_area with the maximum age to remove this information from the routers database. Routers deprecate the prefix on hosts using NDP.

- If the link is split in two networks there are two possibilities. Either both sub-links keep an indirect connectivity through other links; in that case, the DRs of the sub-links will receive an SID_area corresponding to the other sub-link and one of the routers will have to renumber (this will be processed as a SID collision). Or one of the sub-links is totally isolated from the network. In that case, it will not receive GP_LSA and will renumber using the ULA prefix.

/64 limits.

3.2 NAP model

We present here an algorithmic description of the NAP protocol. Let L be the number of links to be numbered in a site and M the number of available SID values for the network. Of course we must have $L \leq M$. To each network link, we associate its LinkID value which is the concatenation of the link Designated Router ID and its link-connected interface number. A lexical order may be set among the links using their LinkID value. Let NCL (Non Configured Links) be the set of links with no valuable attributed prefix. This set is ordered using the lexical order defined by the LinkID values. The NAP auto-configuration process can be sequentially described using algorithm 1.

Algorithm 1 NAP

while $(NCL \neq \emptyset)$ **do**
  for $(i \in NCL)$ **do**
    $v(i)$ **= random(SID);**
  Broadcast;
  for $(i \in NCL$ **such that** $v(i) \in SID)$ **do**
    $NCL = NCL - \{i\}$**;**
    $SID = SID - \{v(i)\}$**;**

4. MODEL AND ANALYSIS

Given the very simple algorithm described above, we want to compute the stabilization speed of the NAP protocol in a network, that is, the mean number of steps that are required for each link to be attributed a unique SID value when the network is bootstrapped. The stabilization speed is thus directly correlated to the number of collisions that will occur and must be resolved, possibly in chain.

We model the NAP protocol by using a successive set of random drawns. The main goal is to assign a prefix to each link. A way to model this problem is to consider that the prefix interval is represented by a set of M urns in which one must randomly distribute L balls which are going to represent the links. As stated before the goal is to have only one ball (link) associated to a given urn (prefix). We thus have $L \leq M$.

The algorithm can be model in terms of urn/balls as follow:

Algorithm 2 NAP_Process(M, L)

▷ *Input: M urns and L balls*
▷ *Pre condition: $M \geq L$*
if $(L \neq 0)$ **then**
  Randomly throw the L balls into the M urns;
1 **Let $c \leq M$ denote the number of urns containing at least one ball;**
2 **Keep aside these c urns and for each of them one of the balls contained inside;**
3 **Call NAP_Process($M - c$, $L - c$);**
end

By repeating the procedure NAP_Process, eventually every ball will be stored in an urn and every urn will contain at most one ball. Let us denote by N the random variable counting the number of iterations needed to reach such a configuration. N is the number of calls to the recursive procedure NAP_Process including the first one. We are interested in computing the distribution and the expectation of the random variable N.

Let us consider the homogeneous discrete-time Markov chain $X = \{X_n,\ n \in \mathbb{N}\}$ on the state space $\mathcal{S} = \{0, 1, \ldots, L\}$ where the event $\{X_n = i\}$ represents, for $n \geq 1$, the fact that, after n transitions, or calls to the procedure NAP_Process,

the procedure NAP_Process(L-i,M-i) is currently being executed. The Markov chain starts in state 0 with probability 1, which means that the first call to the procedure is made by NAP_Process(L,M). We denote by $\mathbb{P}(L,M) = (p_{i,j}(L,M))_{(i,j)\in\mathcal{S}^2}$ the transition probability matrix of the Markov chain X. X is clearly acyclic and state L is the absorbing state of X. This means that for every $i \in \mathcal{S} - \{L\}$ and $j \in \mathcal{S}$, we have $p_{ij}(L,M) = 0$ for $i \geq j$ and $p_{L,L}(L,M) = 1$. The random variable N can thus be defined more formally, for $L \geq 1$, as

$$N = \sum_{n=0}^{L} \sum_{i=0}^{L-1} 1_{\{X_n=i\}}.$$

N is the number of transient states visited before absorption. Since the Markov chain is acyclic, we have $1 \leq N \leq L$, with probability 1.

In the next subsection, we derive the transition probability of matrix $\mathbb{P}(L,M)$ which has the following form, where we write $p_{i,j}$ instead of $p_{i,j}(L,M)$.

$$\mathbb{P}(L,M) = \begin{pmatrix} 0 & p_{0,1} & p_{0,2} & \cdots & \cdots & p_{0,L} \\ 0 & 0 & p_{1,1} & p_{1,2} & \cdots & p_{1,L} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & p_{L-1,L} \\ 0 & 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

4.1 Computation of the transition probability matrix

We are now going to derive the transition probability $p_{ij}(L,M)$ of the transition matrix $\mathbb{P}(L,M)$ for all $i,j \in \mathcal{S}$ such that $i < j$. For $i < j$, the transition probability $p_{i,j}(L,M)$ is the probability to obtain exactly $j-i$ non empty urns when throwing $L-i$ balls into $M-i$ urns.

For all $i < j$, we thus have:

$$p_{i,j}(L,M) = p_{0,j-i}(L-i, M-i).$$

We obtain these transition probablities using the following theorem.

Theorem 1. *The number of ways of throwing r different balls in n different urns such that exactly m urns are not empty is*

$$\binom{n}{m} \sum_{k=0}^{m} \binom{m}{k} (-1)^k (m-k)^r.$$

Proof. See for instance [2]. ∎

The number of ways of throwing r different balls in n different urns being equal to n^r, we obtain, for $i < j$,

$$p_{i,j}(L,M) = \binom{M-i}{j-i} \sum_{k=0}^{j-i} \binom{j-i}{k} (-1)^k \times \left(\frac{j-i-k}{M-i}\right)^{L-i}.$$

As expected, we have $p_{L-1,L}(L,M) = 1$ and the case $j = L$ yields to the well-known birthday problem and thus we have, for $i < L$,

$$p_{i,L}(L,M) = p_{0,L-i}(L-i, M-i) = \frac{(M-i)!}{(M-L)!(M-i)^{L-i}}.$$

4.2 Distribution of N

Let Q be the submatrix obtained from matrix $\mathbb{P}(L,M)$ by deleting the last line and the last column which correspond to the absorbing state L. We denote by α the row vector containing the initial probability distribution of the transient states of X. The dimension of vector α is thus equal to L and since $X_0 = 0$ with probability 1, we have $\alpha = (1, 0, \ldots, 0)$. By using classical results on Markov chains, the distribution of N is given, for $n \geq 1$, by

$$\Pr\{N = n\} = \alpha Q^{n-1}(I-Q)\mathbb{1},$$

where I is the identity matrix of dimension L and $\mathbb{1}$ is the column vector of dimension L with all its entries equal to 1. The matrix Q being acyclic, we have $Q^L = 0$. The expected value of N is given by

$$\mathbb{E}(N) = \sum_{n=0}^{\infty} \Pr\{N > n\} = \sum_{n=0}^{\infty} \alpha Q^n \mathbb{1} = \alpha (I-Q)^{-1} \mathbb{1}.$$

To compute $\mathbb{E}(N)$ we first introduce the column vector $V = (V_i)_{0\leq i\leq L-1}$ of conditional expectations $V_i = \mathbb{E}(N|X_0 = i)$, which is given by

$$V = (I-Q)^{-1}\mathbb{1}$$

Our goal is to compute $V_0 = \mathbb{E}(N)$. The vector V is solution to the linear system $(I-Q)V = \mathbb{1}$, which can also be written as $V = \mathbb{1} + QV$, that is, for every $0 \leq i \leq L-1$,

$$V_i = 1 + \sum_{j=0}^{L-1} p_{i,j}(L,M) V_j.$$

The matrix $\mathbb{P}(L,M)$ being acyclic, we get, as expected, $V_{L-1} = 1$, and for every $0 \leq i \leq L-2$,

$$V_i = 1 + \sum_{j=i+1}^{L-1} p_{i,j}(L,M) V_j.$$

The algorithm to compute $\mathbb{E}(N)$, which is equal to V_0, is the following :

Algorithm 3 Mean number of iterations(M, L)

▷ *Input: M urns and L balls*
▷ *Pre condition: $M \geq L \geq 1$*
$V_{L-1} = 1$;
for $i = L-2$ **downto** 0 **do**
  $V_i = 1 + \sum_{j=i+1}^{L-1} p_{i,j}(L,M) V_j$;
end

4.3 Convergence speed

The table 1 gives the average NAP convergence time for different network sizes and standard lengths of allocated GPs. It is important to note the rapid convergence of the algorithm even when SIDs are a scarce resource. For instance when a /60 is given to a network containing 15 links, it takes a little bit more that 3 rounds to reach a consensus between routers on a unique SID value for each link. When limitations on numbering are not so scarce, for instance when an /48 is allocated to the Home Network, SID numbering is done in one round. This result is very encouraging as it shows that the distributed allocation of prefixes is a viable and efficient strategy for router autoconfiguration, even in critical cases where the prefix space is small compared to the number of links.

GP length vs Network size	5 links	10 links	15 links	20 links	25 links
/60	1.5	2.19	3.06	*n.a.*	*n.a.*
/56	1.04	1.16	1.34	1.53	1.70
/52	1.00	1.01	1.03	1.05	1.07
/48	1.00	1.00	1.00	1.00	1.00

Table 1: NAP convergence time.

5. CONCLUSION AND FUTUR WORKS

In this paper, we have presented a simple protocol for router autoconfiguration. Our NAP (No Administration Protocol) protocol is simple and offers a very good behavior even when ressource are scarce. We have also introduce a model suitable to analyse analyticaly the performance of the NAP protocol. Based on this model, we have computed the convergence speed of the protocol in the average case. From these results, we can state that NAP presents a valid alternative to centralized solutions like prefix delegation with DHCP [3]. Its convergence time remains short even in critical configurations, when the number of links to configure is equal to the number of available prefixes. Moreover, its distributed architecture guarantees a high robustness and adaptability compared to centralized approaches. As described in [1], it deals efficiently with multi-homing or network dynamic compared to the classical centralized approaches.

The study of NAP and the analysis that we developed in this article could be extended to other parameters such as the number of collisions that happen during the autoconfiguration process. From this result, we can easily deduce the average number of flooding/broadcast operations required by NAP during the convergence process. This would give a good overview of the protocol impact on the network use. Other events than a network bootstrap can also be studied: the convergence time or the number of collisions after the merging of several autoconfigured networks for example.

From the implementation point of view, the NAP algorithm has been implemented in OSPFv3[4]. Even if the database synchronization and router election functionalities as well as the need for a routing protocol plead in favor of OSPFv3, some factors are also refraining its usage. OSPFv3 requires a unique 32 bits identifier for each router. In our implementation, this value is configured manually. A random number could also be used although collisions may occur in rare occasions. OSPFv3 may also be to heavy to be implemented in small devices that may act as a router in an home network.

Auto-configuration is not the only feature missing in an home environment and several important points deserve futur investigations:

- Home environment requires also multicast routing (at least to relay the DHCPv6 requests to the DHCP servers). PIM dense mode can be used but it means more code and more complexity in the router implementations.
- Multi-homing capabilities are not well-handled by current routing protocols. Packets may be routed to the wrong ISP if only the destination addresses are considered during the forward process.
- Finally, bandwidth may become a scarce resource in such an environment. Current IGP trends to aggregate the traffic through high capacity links. In a home network, such links will probably not exist. In that case, the routing protocol should be adapted to spread the traffic among the available links after consideration of the flow characteristics.

Current IGP protocols are designed from an ISP perspective, for example to be scalable. The introduction of IPv6 in the home network context, an environment with new constraints, offers the opportunity to design new routing protocols which should release the scalability property and introduce the other previously listed features to globally facilitate the network construction.

[4] `http://nap.dstm.info`

6. REFERENCES

[1] G. Chelius, É Fleury, and L. Toutain. No admistration protocol (nap) for ipv6 router auto-configuration. *Int. J. Internet Protocol Technology*, 1(2):101–108, september 2005.

[2] M. Eisen. *Introduction to Mathematical Probability Theory*. Prentice Hall, Englewood Cliffs, 1969.

[3] T. Kniveton. Mobile network prefix delegation. Internet draft, IETF, July 2005.

[4] J. Moy R. Coltun, D. Ferguson. OSPF for IPv6. IETF Request for Comments 2740, Internet Engineering Task Force, December 1999.

[5] D. G. Dutt R. Perlman, S. Gai. RBridges: Base Protocol Specification . Internet Draft draft-ietf-trill-rbridge-protocol-03.txt, Internet Engineering Task Force, March 2007.

[6] O. Troan and R. Droms. IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6. IETF Request for Comments 3633, Internet Engineering Task Force, December 2003.