Easing the Transition to IPv6

Miguel Ángel Díaz, Jordi Palet
CONSULINTEL, San José Artesano 1, Alcobendas, 28108, Madrid, SPAIN
{miguelangel.diaz, jordi.palet}@consulintel.es

Abstract

This paper presents the "auto-transition" concept which tries to ensure that any network device can obtain IPv6 connectivity at any time and whatever network is attached to, even if such network is connected to Internet only with IPv4. The algorithm looks for the best possible transition mechanism according to performance criteria as well as the scenario where the device is located. By implementing such auto-transition algorithm in either or both end-nodes or middle-boxes (CPEs), users could always obtain IPv6 connectivity with no human intervention. The document do not actually provides a complete solution, just an evaluation of the problem and the requirements towards a future documented solution.

1. Introduction

Lots of devices and applications around us will benefit obtaining IPv6 connectivity everywhere: home automation, wearable devices, cars, PDAs, mobile phones, peer-to-peer applications, remote control applications, etc. IPv6 is suitable to solve the network requirements that those devices/applications will need: addressing space, end-to-end secure peer-to-peer communication, auto-configuration features and so on. The main goal of the "auto-transition" concept is to facilitate the IPv6 deployment in a seamless way for such devices and applications because native IPv6 connectivity is not always possible and users need to use an IPv6 transition mechanism in a seamless way.

The "auto-transition" concept addresses the need to fill a gap in transition mechanisms: while IPv6 provides auto-configuration features, enabling devices to work according to the plug-and-play philosophy, (i.e. with no manual intervention), they only can be applied once the device has obtained IPv6 connectivity. As consequence if native IPv6 connectivity is not available, users need to have technical knowledge to choose and to configure manually the adequate transition mechanism that fits in the network scenario where the user is. This can become an unacceptable brake to the IPv6 deployment, mainly in home and SOHO environments where users usually do not have any networking knowledge.

The auto-transition algorithm deals with all the tasks required to configure automatically the best IPv6 connectivity at anytime, in any network scenario, which include native IPv6 connectivity detection and if this is not available, transition mechanism selection. It can be implemented either in stand-alone devices (host, PDA, etc.) or middle boxes like CPE routers of any kind.

2. Auto-Transition overview

When the device is attached to the network the auto-transition algorithm must check if native IPv6 connectivity is available by means of Router Advertisements (RA) [1] or DHCPv6 [2]. Otherwise, the algorithm should try to obtain IPv6 connectivity by using the best transition mechanism according to the specific network configuration where the device is attached, which we call scenario.

Whether the conditions of the network change or the user/device changes the network attachment location while moving, the auto-transition algorithm has to monitor periodically the network parameters (i.e. IPv4 address, loss, delays, etc.) in order to detect those changes and to decide if another transition mechanism different to the one currently being used is more convenient, for instance in case is providing better performance on the new environment.

Users could introduce some parameters by means of a wizard during the installation of the application that implements the auto-transition algorithm, but the default configuration should be enough for the majority of the users. Once this wizard is up and running, all the tasks should be made automatically by the system and no manual intervention is required. Of course, an experienced user could still make certain

improvements to the default wizard configuration, in specific situations or scenarios.

This approach should be available at least in two kind of platforms: **End-devices** which do not intend to provide IPv6 connectivity to others (hosts, PDAs, mobile phones, home automation devices, white goods, consumer electronics, etc.) and **CPE devices** which are located between two different networks to provide native IPv6 connectivity by means of RA [1] or DHCPv6 [2] (typically routers, IPv4 NAT boxes, etc).

2.1. Selection of the proper transition mechanism

The best IPv6 connectivity, in principle, is obviously the native one, if available, since it should not add extra delays in the communication neither introduce more complexity to the networks, as for example, the packets will not be tunneled or encapsulated through other protocols. Consequently the auto-transition algorithm first must check if IPv6 native connectivity is available. On the contrary the auto-transition algorithm must choose the right transition mechanism to be used to ensure the IPv6 connectivity.

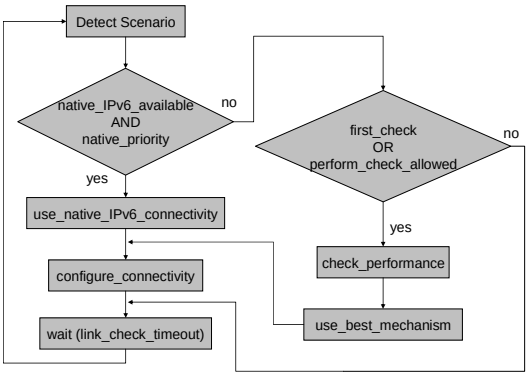

Figure 1. Auto-Transition Algorithm

A few scenarios with particular network requirements had been defined already ([4], [5], [6]), but not all the transition mechanisms fit in such network scenarios, as being evaluated at [7], which is trying to make the best fit to each scenario. The auto-transition algorithm should take into account the results shown in [7] to select a list of candidate mechanism to be checked on the scenario where the user is located. Finally, given the fact that the end user always demands the best performance on the IPv6 connectivity, it should be the main criteria to choose the right transition mechanism from the candidate list.

In order to make the mechanism as simple as possible only delay and packet loss should be actually considered for knowing the link performance that each evaluated transition mechanism presents. According to this philosophy the auto-transition algorithm could operate by means of the simple algorithm shown in Figure 1. The meaning of each task or parameter is as follows:

detect_scenario: Task to deal with detecting the scenario where the device willing to have IPv6 connectivity is attached. It checks if native IPv6 is available, if a public IPv4 address is available, if a NAT is being used and what type, if there is a proxy or firewall, or if other protocols can be operated.

native_IPv6_available: Indicates if native IPv6 is available.

native_priority: Indicates that IPv6 connectivity has no higher priority if other transition mechanism offers better performance.

use_native_IPv6_connectivity: Task to configure the interface to use native IPv6 connectivity, using stateless or stateful auto-configuration, upon their availability.

first_check: Defines if this is the first time this check is being done after an interface reset.

performance_check_allowed: Indicates if the performance of the selected mechanism must be measured after selected, for instance, to avoid traffic being generated in non-flat rate links (3GPP, ISDN, etc.).

check_performance: This task checks the performance that each transition mechanism presents, including native IPv6 if available, by measuring delays and losses.

use_best_mechanism: According to the measurement results, the best mechanism is selected.

configure_connectivity: Either native IPv6 connectivity or the best available transition mechanism is configured.

link_check_timeout: Once the IPv6 connectivity is obtained, the auto-transition algorithm periodically monitors the link status. The delay between consecutive checks is defined by this parameter.

A possible list of mechanisms to be checked, sorted by preference could be: Native IPv6 Connectivity, 6in4 with proto-41[3], ISATAP, 6to4, TSP, AYIYA and TEREDO.

2.2. Change of transition mechanism

Change of transition mechanism refers to the task to abandon the transition mechanism that is actually being used and start to use another one that presents better performance. This is not an easy task at all, since it involves at least two important issues:

1) To maintain the current IPv6 address. This is very important in some circumstances, since otherwise applications with communications opened will not work. Especially important is the case when the auto-transition algorithm is implemented in border devices that provide native IPv6 connectivity to the whole network by mean of RA [1] or DHCPv6 [2], because they should try to keep always the IPv6 addressing space. To do that it is still necessary to define a method that solves this issue. MIPv6 concepts/solutions could be applied and possibly also those related to multihoming.

2) User authentication without human intervention. The philosophy of the auto-transition algorithm is that all the processes are done automatically, with no human intervention. As some of the transition mechanisms may require user authentication, the auto-transition algorithm needs to store the authentication parameters (maybe configured through the wizard during the installation process), so they are automatically used when changing to a different Tunnel End Point (TEP). Also AAA mechanisms could be used.

2.3. New transition mechanisms

A number of devices do not allow tunnel-based transition mechanisms to work properly. Examples of those devices are NAT boxes, proxies or firewalls. Even building IPv6 tunnels over UDP is not always possible since some middle boxes might filter those packets. When this happens it is required that the auto-transition algorithm make usage of a method that cannot be filtered by the middle box.

The following solutions are being considered: Layer 2 VPNs (L2TP, PPTP, PPPoE), Layer 3 VPNs or Layer 4 tunnels (TLS/SSH, HTTP, SSH). The last type of tunnels (layer 4) is the only one that can always ensure the traversal of any middle-box, but it also offers the lower performance, so it should be chosen only as the very last resort.

2.4. Discovery of the IPv6 End Point

Devices running the auto-transition algorithm need to know where to find the IPv6 Tunnel End Point (TEP), which provides the IPv6 connectivity, just in case native IPv6 connectivity is not available. Having in mind that users want plug-and-play devices/services and that most of them do not have any knowledge about how the transition mechanisms works or where the nearest TEP is located, it is required to consider the auto-discovery of the IPv6 TEP (which could also include the tunnel setup handshake), so devices can find it automatically.

To achieve these goals, a solution is proposed [8] which does not imply any new protocol but making usage of the current DNS along with standardized anycast (shared unicast) addresses for each transition mechanism. The ideal situation is to implement on the ISP side all the following requirements in order to get the auto-discovery mechanism more functional. However, it is not mandatory and at least only one of them should be selected:

(1) DNS server with SRV RR support [9]. The service name for the auto-discovery purpose should be standardized for each transition mechanism in the following form:

_transition-mechanism_srv._protocol.ispname.com

One important advantage of this method is that load balancing can be done easily and efficiently by means of priority and weight parameters defined in SRV RR. More details can be found in [9].

(2) A/CNAME RR for Unicast. A standardized A/CNAME RR for each supported transition mechanisms within the domain of the ISP. According to the same nomenclature, the DNS entries would follow the form:

transition-mechanism_srv.ispname.com

(3) Anycast (Shared Unicast) Addresses. Each transition mechanism would have an assigned anycast (shared unicast) address, such as in the case of the 6to4 transition mechanism [10]. The anycast prefix/address for each transition mechanism would be specified by IANA.

When looking for a specific TEP within the ISP the user belongs to, the user always query firstly for a DNS SRV RR to its ISP DNS server, so the ISP domain name is learned in some way (there is several ways to do that) and the DNS SRV RR query for the specific TEP is created in the form explained above. If the DNS server matches the query, it returns the proper reply with all the possible targets defined for that query and the client choose one of them according to the priority and weight parameters of each target.

If no DNS SRV RR reply is obtained, then an A/CNAME query is built by the client by appending the standardized transition service name to the ISP domain name, as explained above.

Finally if there is not a valid A/CNAME RR matching the client query, then the client will directly use the standardized anycast address. This allows the provision by third parties of the service for free, when

the own ISP does not provide it and it does not require any special deployment in the ISP infrastructure. In this point the auto-discovery function ends.

3. Network Managed Transition

The algorithm described in this paper follows an approach based on the role that the user's device plays. However the algorithm could be improved and/or even more easily managed if the ISP helps in some way to the auto-transition mechanism. Following this new approach, Policy Based Networks (PBN) [11] can offer a candidate solution to provide facilities to the auto-transition algorithm. Policies stored on the network repository might include information about the type of transition mechanisms implemented into the ISP where the user device is attached to, so the auto-transition algorithm implemented into the user's device would choose one of the mechanisms suggested/enabled by the ISP policies.

With this approach the user's device will act as a Policy Enforcement Point (PEP) [11] as well as implementing the auto-transition algorithm and it would inform the Policy Decision Point (PDP) [11] located at the ISP side about features such as type of connection, date/time, user privileges and/or whatever other relevant information. Then, the PDP might interact with other policies stored on the repository such as QoS Policies, Security Policies and so on, in order to propose the more adequate transition mechanism to be used by the device willing to get IPv6 connectivity.

Considering that most of the ISPs will not necessarily deploy transition mechanisms in the early stage, advanced IPv6 Internet Exchanges (IX) could provide this kind of services [12] and in general policy-based capabilities. The IX is not just a central peering point, which facilitates any new service deployment, but also a place where lots of useful information (routes, QoS, link conditions, etc.) about several domains is available. With this philosophy, the transition policies will be one more facility provided by this type of IXs.

Nevertheless in spite of the network approach, whether the network provides this type of transition facilities or not, the auto-transition algorithm, when present, must always work and it will provide the best possible IPv6 connectivity.

4. Conclusions

There is a need for a method to provide plug-and-play features to IPv6 transition mechanisms in the same way that the IPv6 protocol does in the local network. With this philosophy users do not have to know any technical knowledge to choose the more adequate transition mechanisms, nor to make any setup of it, nor to find out where the nearest TEP is located. They just plug their devices and they automatically become IPv6 capable whether they are in a native IPv6 environment or not, even if they are in a private IPv4 environment behind a NAT box. Some research to achieve these goals is being done and some preliminary work is presented in this paper.

5. Acknowledgements

The authors would like to acknowledge both the European Commission and Spanish Ministry of Industry support in the co-funding, respectively, of the ***Euro6IX*** and ***Auto-Transición*** projects, where this work is being developed.

6. References

[1] S. Thomson, and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.
[2] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
[3] J. Palet, C. Olvera and D. Fernández "Forwarding Protocol 41 in NAT Boxes", draft-palet-v6ops-proto41-nat-03 (work in progress).
[4] C. Huitema, "Evaluation of Transition Mechanisms for Unmanaged Networks", RFC 3904, September 2004.
[5] M. Lind, V. Ksinant, S. Park, A. Baudot and P. Savola, "Scenarios and Analysis for Introducing IPv6 into ISP Networks", draft-ietf-v6ops-isp-scenarios-analysis-03 (work in progress).
[6] J. Bound, "IPv6 Enterprise Network Scenarios", draft-ietf-v6ops-ent-scenarios-05 (work in progress).
[7] P. Savola and J. Soininen, "Evaluation of v6ops Tunneling Scenarios and Mechanisms", draft-savola-v6ops-tunneling-01 (work in progress).
[8] J. Palet, and M. A. Diaz "IPv6 Tunnel End-point Automatic Discovery Mechanism", draft-palet-v6ops-solution-tun-auto-disc-01.txt (work in progress).
[9] A. Gulbrandsen, P. Vixie and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.
[10] C. Huitema "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.
[11] R. Yavatkar, D. Pendarakis and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000.
[12] M. Morelli, J. Palet, D. Fernández and A. Gómez "Advanced IPv6 Internet Exchange model", draft-morelli-v6ops-ipv6-ix-00 (work in progress).