wradvs: A Didactic Server for IPv6 Stateless Autoconfiguration

Gabriel Abadi, Francisco Jimenez, and Eric Gamess
School of Computer Science
Central University of Venezuela
Caracas, Venezuela
gabriel.abadi@gmail.com, jimenezff@gmail.com, egamess@gmail.com

Abstract — In this paper we present wradvs, a free open source application developed under the GNU General Public License that provides a comprehensive solution for IPv6 stateless autoconfiguration. The application works as a service and implements RFCs 2460, 4861, 5006, 4191, 3775, and 3963. It also has an event log viewer that records detailed information of all events in real time, allowing users to troubleshoot network problems without the need of additional tools. The main goal of wradvs is to be used as a didactic application in network advanced courses at Central University of Venezuela. Thus, it has a friendly graphical user interface and presents a lot of valuable information to users while they configure all the parameters and options of the Router Advertisement messages.

Keywords: ipv6; stateless autoconfiguration; router advertisement; windows; service; daemon; didactic application.

I. INTRODUCTION

Due to the imminent IPv4 address space depletion, LACNIC (Latin American and Caribbean Internet Addresses Registry) is encouraging the adoption of IPv6 [5][13] as soon as possible. However, this adoption has been slow in Venezuela and in many other countries. One of the reasons for the delay is the lack of IPv6 network specialists. Therefore, the training of IPv6 specialists has become an important issue. In the undergraduate program of Computer Science at Central University of Venezuela (in Spanish: Universidad Central de Venezuela), some courses have been upgraded or added to the curriculum to face the problem. For example, Advanced Network Protocols (in Spanish: Protocolos Avanzados de Redes) is a new course that was introduced to the curriculum of the undergraduate program of Computer Science in 2005. Its objectives include the understanding of IPv6 standards such as the stateless autoconfiguration process described in [1].

Some daemons or servers have been developed, by the community or manufacturers, to support the stateless autoconfiguration process. However, most of these daemons do not support all the parameters and options of the stateless autoconfiguration process, and sometimes, configuration can be very difficult and the debugging system can be very poor. These deficiencies can make the teaching and learning activity process very laborious. For these reasons, we developed wradvs (Windows Router Advertisement Server). wradvs is not only one of the most complete daemon created for IPv6 stateless autoconfiguration, but it also provides a friendly graphical user interface, assists users during the configuration process with tooltips, and validates all the submitted information. Additionally, wradvs provides an Event Log Viewer with detailed information of the messages that are sent and received in the autoconfiguration process. Even if the main goal of the application is to support the teaching and learning process, it can be very useful for a network administrator that uses IPv6 servers based on Windows. The rest of the paper is organized as follows: the IPv6 Router Advertisement message is presented in Section II; IPv6 stateless autoconfiguration is discussed in Section III; related works are viewed in Section IV; wradvs is presented and justified in Section V; and finally Section VI concludes the paper.

II. ROUTER ADVERTISEMENT

A Router Advertisement (RA) message is an Internet Control Message Protocol version 6 (ICMPv6) message defined by the Neighbor Discovery (ND) protocol [2].

IPv6 routers send unsolicited RA messages pseudo-periodically and solicited RA messages in response to the receipt of a Router Solicitation (RS) message. The interval between unsolicited advertisements is randomized to reduce synchronization issues when there are multiple advertising routers on a link. RA messages contain the information required by hosts to determine default gateways, the link prefixes, the link MTU, specific routes, home agent information, recursive DNS servers, whether or not to use stateful autoconfiguration, and the duration for which addresses created through stateless address autoconfiguration are valid and preferred [5].

Fig. 1 shows the structure of the RA message which contains the following fields:

- Type: the value of this field must be 134 and identify the type of ICMPv6 message (in this case, a RA).
- Code: the value of this field must be 0.
- Checksum: stores the checksum of the ICMPv6 message.
- Current Hop Limit: indicates the default value of the Hop Limit field to be put in the IPv6 header for packets sent by hosts that receive this RA message. A value of 0 indicates that the default value of the Hop Limit field is not specified by this router.
### Managed Address Configuration Flag (M): when set, it indicates that hosts receiving this RA message must use a stateful configuration protocol, such as DHCPv6 (Dynamic Host Configuration Protocol for IPv6), to obtain addresses in addition to the addresses that might be generated from stateless address autoconfiguration.

### Other Stateless Configuration Flag (O): when set, it indicates that hosts receiving the RA message must use a stateful configuration protocol, such as DHCPv6, to obtain other configuration information.

### Home Agent Flag (H): when set, it indicates that the router sending the RA message is also functioning as a Mobile IPv6 home agent on this link [3].

### Default Router Preference (Prf): indicates whether to prefer this router over other default routers according to the level of preference. Valid values in binary are: 01 (high), 00 (medium) and 11 (low) [4].

### Reserved: this field is reserved for future use and must be set to 0.

### Router Lifetime: indicates the lifetime (in seconds) of this router as a default router. A value of 0 indicates that this router is not a default router; however, all other information contained in the RA message is still valid.

### Reachable Time: indicates the amount of time (in milliseconds) that a host should assume a neighbor is reachable after receiving a reachability confirmation. A value of 0 indicates that this router does not make any recommendation about the reachable time.

### Retransmission Timer: indicates the amount of time (in milliseconds) that a host should wait before the retransmission of a Neighbor Solicitation message. A value of 0 indicates that this router does not specify the retransmission timer.

The options that can be present in a RA message are the following:

- **Source Link-Layer Address option**: contains the link-layer address of the interface from which the RA message was sent.
- **MTU option**: contains the recommended MTU for the link. Should be sent on links that have a variable MTU.
- **Prefix Information option**: contains the prefixes that are on-link and/or used for stateless address autoconfiguration.
- **Route Information option**: contains more-specific routes that improve the ability of hosts to pick an appropriate next hop for an off-link destination [4].
- **Recursive DNS Server option**: contains one or more IPv6 addresses of recursive DNS servers [6].
- **Home Agent Information option**: contains information specific to this router’s functionality as a home agent [3].
- **Advertisement Interval option**: contains the interval at which this router sends unsolicited multicast RA messages [3].

### III. IPv6 Stateless Address Autoconfiguration

The IPv6 stateless address autoconfiguration defines the mechanism that allows a host to generate its own addresses and others network parameters using local information and information advertised by routers in RA messages. This mechanism requires no manual configuration of hosts. The address autoconfiguration process [1] for a physical interface of an IPv6 node is as follows:

When an interface becomes enabled, a tentative link-local address is generated based on the link-local prefix (FE80::/64) and an interface identifier, which is commonly derived from the link-layer address. The uniqueness of the tentative link-local address is verified using the duplicate address detection (DAD) algorithm. If the address is in use by another host, the autoconfiguration stops and manual configuration must be performed on the host.