lnp
Internet-Draft
Intended-Status: Elective

C. Schaller
Schallsoft

February 11, 2019

Local Naming Protocol -- LNP (v.1.0)
draft-schaller-dnsop-lnp-00

Abstract

The Local (or Lightweight) Naming Protocol (LNP) is an application-level protocol for local area networks. It is a distributed, stateless protocol which intents in resolving hostnames to ip addresses without the need of any Domain Name Server. In private local area networks, ip addressses are often dynamically allocated through DHCP. The LNP can be seen as a DNS extension, which uses broadcast udp messages (similar to ARP on IP-MAC-level) to request ip addresses for hosts with a given host- or domain-name. Thus it will be possible in dynamic local area networks to access ip-based services on hosts by their hostnames, without further management.

Comments are solicited and should be addressed to the working group's mailing list at dnsop@ietf.org and/or the author(s).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

Copyright Notice

Table of Contents

1. Introduction

In private local area networks or wireless LANs, today mainly DHCP [RFC2131] managed ip address allocation is used. Since the equipment on the market does not ship with integrated DNS servers, which update their records when hosts attach to or detach from the network, there is less help for private users or user applications trying to access devices services over ip addresses. Actually no service is out of the box available for all operating systems. Microsoft Windows ships with NetBios [RFC 1002], which allows name based access, however first after activation of file-sharing. Apples MacOS is delivered with bonjour [RFC 6763], which runs out of the box. Linux and Unix systems can do both, however after explicit installation setup. All those systems are able to use DNS [RFC 1035] when connected to ip networks. So why not extend the functionality of the distributed Domain Name System, which already has the task to resolv hostnames to ip addresses.

1.1. Standalone LNP

The standalone implementation of LNP uses a UDP broadcast message to query the ip address of a host. Therefore the message contains the full qualified name of a host connected to the local network. All receivers check if the requested hostname equals their own. Every matching host then replies. In version 1.0 a replying host shall append the ip-address, bound to the interface receiving the message, as message data. The sender now has resolved the ip-address of his well known communication partner and can in turn open tcp-streams and communicate directly.

1.2. DNS extended LNP

Every networking software using sockets, that relies on name resolution to determine destinations ip-address will probably use a system call e.g. getHostByName() or getAddrInfo() to retrieve the ip-address. When every standard DNS client would be able to provide and use LNP, i.e. in case of no matching DNS record or hosts-file entry found, then no software product has to be rewritten or updated to be able to resolv hostnames in a dynamically allocated local domain as well.

2. Protocol version 1.0

The protocol relies on two types of messages, a request and a reply-message. Both messages contain two lines of human readable character-data, which end with a line-feed. The first line describes the protocol version used, i.e. "LNP v.1.0" and the second line describes the exchanged information. The request-data shall be interpreted as qualified domain name and therefore be compared by any receiver with its own hostname. Every host, that matches identically should then immediateliy reply with one line of human-readable character-data containing the desired ip-address of the interface where the request message was received on. The version of the ip-address used can be determined on either dotted-decimal-notation (IPv4) or colon-separated-hex-values (IPv6) [draft-main-ipaddr-text-rep-02]. The appended reply-data is just useful for user interaction, i.e. using the protocol on a command line interface, since the receiver of the reply gets the address information about the replyer from ip-header in binary format as well.

2.1. Wildcards

Due to security reasons, in protocol version 1.0 no wildcards are defined or accepted.

2.2. Concurrent replies

Is more than one host at a time using the same hostname in exactly the same local network, there will be multiple replies when asking for this hostname. Since it is not unique, this case must be reported, either directly to the user or at least into systems logfile. The protocol implementation 1.0 defines the following method. The ip of the first reply will be returned with error code set to NOT_UNIQUE. Further messages shall be discarded, i.e. the socket is closed.

2.3. No host responding

A timeout shall be set in case of no reply. In this case no ip-address can be returned and no statement can be made, except that the target host is not existent or not responding. The timeout can be chosen very short, since the broadcast domain is limited to the local network.

3. IANA Considerations

There has to be a well known Port number for LNP. An assignment request shall be made when this document gets accepted.

4. Security Considerations

Since there are no wildcards defined in protocol version 1.0, it is not possible to query all hosts ip-addresses at once. Furthermore the design of the protocol respects privacy, so that the name of the desired host has to be known before a valid query result can be achieved.

5. References

[RFC 1002] NetBIOS Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, End-to-End Services Task Force,"Protocol standard for a NetBIOS service on a TCP/UDP transport: Detailed specifications",DOI: 10.17487/RFC1002, March 1987, <https://www.rfc-editor.org/rfc/rfc1002.txt>.

[RFC 1035] P.V. Mockapetris, "Domain names - implementation and specification", DOI: 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/rfc/rfc1035.txt>.

[RFC 2131] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, DOI: 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/rfc/rfc2131.txt>.

[RFC 6763] S. Cheshire, M. Krochmal, "DNS-Based Service Discovery", DOI: 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/rfc/rfc6763.txt>.

Author's Address

Christian Schaller
Schallsoft
Herderstrasse 13, 08525 Plauen
Germany

Phone: +49 3741 - 554 744
Email: christian.schaller@sprintmail.de
URI: http://www.schallsoft.de