

Network Working Group
INTERNET-DRAFT
Category: Informational
Expires: April 11, 2008
[11](#) October 2007

B. Aboba
D. Thaler
Microsoft Corporation
Loa Andersson
Acreo AB

Principles of Internet Host Configuration
[draft-aboba-ip-config-05.txt](#)

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 11, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document describes principles of Internet host configuration. It covers issues relating to configuration of Internet layer parameters, as well as parameters affecting higher layer protocols.

INTERNET-DRAFT

Principles of Host Configuration

11 October 2007

Table of Contents

1.	Introduction.....	3
1.1	Terminology	3
2.	Principles	6
2.1	Minimize Configuration	6
2.2	Less is More	6
2.3	Diversity is Not a Benefit	7
2.4	Lower Layer Independence	8
2.5	Configuration is Not Access Control	9
3.	Additional Discussion	10
3.1	General Purpose Mechanisms	10
3.2	Service Discovery Protocols	10
3.3	Fate Sharing	11
4.	Security Considerations	12
4.1	Configuration Authentication	13
5.	IANA Considerations	14
6.	References	14
6.1	Informative References	14
	Acknowledgments	16
	Authors' Addresses	17
	Full Copyright Statement	18
	Intellectual Property	18

INTERNET-DRAFT

Principles of Host Configuration

11 October 2007

1. Introduction

This document describes principles of Internet host configuration. It covers issues relating to configuration of Internet layer parameters, as well as parameters affecting higher layer protocols.

In recent years, a number of architectural questions have arisen, for which we provide guidance to protocol developers:

- o What protocol layers and general approaches are most appropriate for configuration of various parameters.
- o The relationship between parameter configuration and service discovery.
- o The relationship between network access authentication and host configuration.
- o The role of link-layer protocols and tunneling protocols in Internet host configuration.

The role of the link-layer and tunneling protocols is particularly important, since it can affect the properties of a link as seen by higher layers (for example, whether privacy extensions specified in "Privacy Extensions for Stateless Address Autoconfiguration in IPv6" [[RFC3041](#)] are available to applications).

1.1. Terminology

link A communication facility or medium over which nodes can communicate at the link-layer, i.e., the layer immediately below IP. Examples are Ethernets (simple or bridged), PPP links, X.25, Frame Relay, or ATM networks as well as internet (or higher) layer "tunnels", such as tunnels over IPv4 or IPv6 itself.

on link An address that is assigned to an interface on a specified link.

off link The opposite of "on link"; an address that is not assigned to any interfaces on the specified link.

Internet layer configuration is defined as the configuration required to support the operation of the Internet layer. This includes IP address(es), subnet prefix(es), default gateway(s), mobility agent(s), boot service configuration and other parameters.

IP address(es)

Internet Protocol (IP) address configuration includes both configuration of link-scope addresses as well as global addresses. Configuration of IP addresses is an important step, since this enables a host to fill in the source address in the packets it sends, as well as to receive packets destined to that address. As a result, the host can receive unicast IP packets, rather than requiring that IP packets be sent to the broadcast or multicast address. Configuration of an IP address also enables the use of IP fragmentation, since packets sent from the unknown address cannot be reliably reassembled (since fragments from multiple hosts using the unknown address might be reassembled into a single IP packet). Configuration of an IP address also enables use of Internet layer security facilities such as IPsec specified in "Security Architecture for the Internet Protocol" [[RFC4301](#)].

Subnet prefix(es)

Once a subnet prefix is configured, hosts with an IP address can exchange unicast IP packets with on-link hosts.

Default gateway(s)

Once a default gateway is configured, hosts with an IP address can send and receive unicast IP packets from off-link hosts, assuming unobstructed connectivity.

Mobility agent(s)

While Mobile IPv4 [[RFC3344](#)] and Mobile IPv6 [[RFC3775](#)] include their own mechanisms for locating home agents, it is also

possible for mobile nodes to utilize dynamic home agent configuration.

Other parameters

Internet layer parameter configuration also includes configuration of per-host parameters (e.g. hostname) and per-interface parameters (e.g. IP Time-To-Live (TTL), enabling/disabling of IP forwarding and source routing, and Maximum Transmission Unit (MTU)).

Boot service configuration

Boot service configuration is defined as the configuration necessary for a host to obtain and perhaps also to verify an appropriate boot image. This is appropriate for diskless hosts looking to obtain a boot image via mechanisms such as the Trivial File Transfer Protocol (TFTP) [[RFC1350](#)], Network File System (NFS) [[RFC3530](#)] and Internet Small Computer Systems Interface (iSCSI) [[RFC3720](#)][[RFC4173](#)]. It also may be useful in situations where it is necessary to update the boot

image of a host that supports a disk, such as in the Preboot eXecution Environment (PXE) [[PXE](#)][[RFC4578](#)]. While strictly speaking boot services operate above the Internet layer, where boot service is used to obtain the Internet layer code, it may be considered part of Internet layer configuration.

Higher-layer configuration is defined as the configuration required to support the operation of other components above the Internet layer. This includes, for example:

Name Service Configuration

The configuration required for the host to resolve names. This includes configuration of the addresses of name resolution servers, including IEN 116, Domain Name Service (DNS), Windows Internet Name Service (WINS), Internet Storage Name Service (iSNS) and Network Information Service (NIS) servers, and the setting of name resolution parameters such as the NetBIOS node type, the DNS domain and search list, etc. It may also include the transmission or setting of the host's own name. Note that Link local name resolution services (such as NetBIOS [[RFC1001](#)], LLMNR [[RFC4795](#)] and mDNS [[mDNS](#)]) typically do not require configuration.

Once the host has completed name service configuration, it is capable of resolving names using name resolution protocols that require configuration. This not only allows the host to communicate with off-link hosts whose IP address is not known, but to the extent that name services requiring configuration are utilized for service discovery, this also enables the host to discover services available on the network or elsewhere.

Time Service Configuration

Time service configuration includes configuration of servers for protocols such as the Simple Network Time Protocol (SNTP) and the Network Time Protocol (NTP). Since accurate determination of the time may be important to operation of the applications running on the host (including security services), configuration of time servers may be a prerequisite for higher layer operation. However, it is typically not a requirement for Internet layer configuration.

Other service configuration

This can include discovery of additional servers and devices, such as printers, Session Initiation Protocol (SIP) proxies, etc.

[2.](#) Principles

This section describes basic principles of Internet host configuration.

[2.1.](#) Minimize Configuration

Anything that can be configured can be misconfigured. [RFC 1958 \[RFC1958\] Section 3.8](#) states: "Avoid options and parameters whenever possible. Any options and parameters should be configured or negotiated dynamically rather than manually."

That is, to minimize the possibility of configuration errors, parameters should be automatically computed (or at least have reasonable defaults) whenever possible. For example, the

Transmission Control Protocol (TCP) [[RFC793](#)] does not require configuration of the Maximum Segment Size, but is able to compute an appropriate value.

Some protocols support self-configuration mechanisms, such as capability negotiation or discovery of other hosts that implement the same protocol.

[2.2.](#) Less is More

The availability of standardized, simple mechanisms for general-purpose Internet host configuration is highly desirable. [RFC 1958](#) [[RFC1958](#)] states, "Performance and cost must be considered as well as functionality" and "Keep it simple. When in doubt during design, choose the simplest solution."

To allow protocol support in more types of devices, it is important to minimize the footprint requirement. For example, Internet hosts span a wide range of devices, from embedded devices operating with minimal footprint to supercomputers. Since the resources (e.g. memory and code size) available for host configuration may be very small, it is desirable for a host to be able to configure itself in as simple a manner as possible.

One interesting example is IP support in pre-boot execution environments. Since by definition boot configuration is required in hosts that have not yet fully booted, it is often necessary for pre-boot code to be executed from Read Only Memory (ROM), with minimal available memory. In PXE, prior to obtaining a boot image, the host is typically only able to communicate using IP and the User Datagram Protocol (UDP). This is one reason why Internet layer configuration mechanisms typically depend only on IP and UDP. After obtaining the boot image, the host will have the full facilities of TCP/IP

available to it, including support for reliable transport protocols, IPsec, etc.

In order to reduce complexity, it is desirable for Internet layer configuration mechanisms to avoid dependencies on higher layers. Since embedded hosts may wish to minimize the code included within a boot ROM, availability of higher layer facilities cannot be guaranteed during Internet layer configuration. In fact, it cannot

even be guaranteed that all Internet layer facilities will be available. For example, IP fragmentation and reassembly may not work reliably until a host has obtained an IP address.

[2.3.](#) Diversity is Not a Benefit

The number of host configuration mechanisms should be minimized. Diversity in Internet host configuration mechanisms presents several problems:

Interoperability

As configuration diversity increases, it becomes likely that a host will not support the configuration mechanism(s) available on the network to which it has attached, creating interoperability problems.

Footprint In order to interoperate, hosts need to implement all configuration mechanisms used on the link layers they support. This increases the required footprint, a burden for embedded devices.

Redundancy

To support diversity in host configuration mechanisms, operators would need to support multiple configuration services to ensure that hosts connecting to their networks could configure themselves. This represents an additional expense for little benefit.

Latency As configuration diversity increases, hosts supporting multiple configuration mechanisms may spend increasing effort to determine which mechanism(s) are supported. This adds to configuration latency.

Conflicts Whenever multiple mechanisms are available, it is possible that multiple configuration(s) will be returned. To handle this, hosts would need to merge potentially conflicting configurations. This would require conflict resolution logic, such as ranking of potential configuration sources, increasing implementation complexity.

To limit configuration latency, hosts may simultaneously attempt to obtain configuration by multiple mechanisms. This can result in increasing on-the-wire traffic, both from use of multiple mechanisms as well as from retransmissions within configuration mechanisms not implemented on the network.

Security Support for multiple configuration mechanisms increases the attack surface of the host.

2.4. Lower Layer Independence

[RFC 1958](#) [[RFC1958](#)] states, "Modularity is good. If you can keep things separate, do so."

It is becoming increasingly common for hosts to support multiple network access mechanisms, including dialup, wireless and wired local area networks, wireless metropolitan and wide area networks, etc. As a result, it is desirable for hosts to be able to configure themselves on multiple networks without adding configuration code specific to a new link layer.

As a result, it is highly desirable for Internet host configuration mechanisms to be independent of the underlying lower layer. That is, the link layer protocol (whether it be a physical link, or a virtual tunnel link) should only be explicitly aware of link-layer parameters (although it may configure link-layer parameters - see [Section 2.1](#)). Introduction of lower layer dependencies increases the likelihood of interoperability problems and adds to the number of Internet layer configuration mechanisms that hosts need to implement.

Lower layer dependencies can be best avoided by keeping Internet host configuration above the link layer, thereby enabling configuration to be handled for any link layer that supports IP. In order to provide media independence, Internet host configuration mechanisms should be link-layer protocol independent.

While there are examples of IP address assignment within the link layer (such as in the Point-to-Point Protocol (PPP) IPv4CP [[RFC1332](#)]), the disadvantages of this approach have now become apparent. The main disadvantages include the extra complexity of implementing different mechanisms on different link layers, and the difficulty in adding new parameters which would require defining a mechanism in each link layer protocol.

For example, PPP IPv4CP and Internet Protocol Control Protocol (IPCP) extensions for name service configuration [[RFC1877](#)] were developed at a time when the Dynamic Host Configuration Protocol (DHCP) [[RFC2131](#)]

had not yet been widely implemented on access devices or in service provider networks. However, in IPv6 where link layer independent mechanisms such as stateless address configuration [[RFC2462](#)] and DHCPv6 [[RFC3736](#)] are available, PPP IPv6CP [[RFC2472](#)] instead simply configures an Interface-Identifier which is similar to a MAC address. This enables PPP IPv6CP to avoid having to duplicate DHCPv6 functionality.

In contrast, Internet Key Exchange Version 2 (IKEv2) [[RFC4306](#)] utilizes the same approach as PPP IPv4CP by defining a Configuration Payload for Internet host configuration for both IPv4 and IPv6. As pointed out in [[RFC3456](#)], leveraging DHCP has advantages in terms of address management integration, address pool management, reconfiguration and fail-over. On the other hand, the IKEv2 approach reduces the number of exchanges.

Extensions to link layer protocols for the purpose of Internet, transport or application layer configuration (including server configuration) should be avoided. Such extensions can negatively affect the properties of a link as seen by higher layers. For example, if a link layer protocol (or tunneling protocol) configures individual IPv6 addresses and precludes using any other addresses, then applications that desire privacy extensions [[RFC3041](#)] may not function well. Similar issues may arise for other types of addresses, such as Cryptographically Generated Addresses [[RFC3972](#)].

Avoiding lower layer dependencies is desirable even where the lower layer is link independent. For example, while the Extensible Authentication Protocol (EAP) [[RFC3748](#)] may be run over any link satisfying the requirements of [[RFC3748](#)] [Section 3.1](#), many link layers do not support EAP and therefore Internet layer configuration mechanisms with EAP dependencies would not be usable on all links that support IP.

[2.5](#). Configuration is Not Access Control

Network access authentication is a distinct problem from Internet host configuration. Network access authentication is best handled independently of the configuration mechanisms in use for the Internet and higher layers.

For example, attempting to control access by requiring authentication in order to obtain configuration parameters (such as an IP address) has little value if the user can manually configure the host. Having an Internet (or higher) layer protocol authenticate clients is appropriate to prevent resource exhaustion of a scarce resource on

the server, but not for preventing rogue hosts from obtaining access to a link. Note that client authentication is not required for

Stateless DHCPv6 [[RFC3736](#)] since it does not result in allocation of any limited resources on the server.

[3.](#) Additional Discussion

[3.1.](#) General Purpose Mechanisms

Protocols should either be self-configuring (especially where fate sharing is important), or use general-purpose configuration mechanisms (such as DHCP or a service discovery protocol, as noted in [Section 3.2](#)). The choice should be made taking into account the architectural principles discussed in [Section 2](#).

Given the number of Internet host configuration mechanisms that have already been defined, there is no justification for hard coding of service IP addresses or domain names. Taking into account the problems resulting from the proliferation of these mechanisms, there is no apparent need for the development of additional general-purpose configuration mechanisms.

When defining a new host parameter, protocol designers should first consider whether configuration is indeed necessary (see [Section 2.1](#)). If configuration is necessary, in addition to considering fate sharing (see [Section 3.3](#)), protocol designers should consider:

1. The organizational implications for administrators. For example, routers and servers are often administered by different sets of individuals, so that configuring a router with server parameters may require cross-group collaboration.
2. Whether the parameter is a per-interface or a global parameter. For example, most standard general purpose configuration protocols run on a per-interface basis and hence are more appropriate for per-interface parameters.

[3.2.](#) Service Discovery Protocols

Higher-layer configuration often includes configuring server addresses. The question arises as to how this differs from "service

discovery" as provided by Service Discovery protocols such as the Service Location Protocol Version 2 (SLPv2) [[RFC2608](#)].

In general-purpose configuration mechanisms such as DHCP, server instances are considered equivalent. In service discovery protocols, on the other hand, a host desires to find a server satisfying a particular set of criteria, which may vary by request.

Service discovery protocols such as SLPv2 can support discovery of

servers on the Internet [[RFC3832](#)], not just those within the local administrative domain. General-purpose configuration mechanisms such as DHCP, on the other hand, typically assume the server(s) in the local administrative domain contain the authoritative set of information.

For the service discovery problem (i.e., where the criteria varies on a per-request basis, even from the same host), protocols should either be self-discovering (if fate sharing is critical), or use general purpose service discovery mechanisms.

In order to avoid a dependency on multicast routing, it is necessary for a host to either restrict discovery to services on the local link or to discover the location of the Directory Agent (DA). Therefore the use of service discovery protocols beyond the local link is typically dependent on a parameter configuration mechanism. As a result, service discovery protocols are typically not appropriate for use in obtaining basic Internet layer configuration, although they can be used to obtain higher-layer configuration for parameters that don't meet the assumptions above made by general-purpose configuration mechanisms.

[3.3.](#) Fate Sharing

If a server (or set of servers) is needed to get a set of configuration parameters, "fate sharing" ([RFC1958](#) [Section 2.3](#)) is preserved if the servers are ones without which the parameters could not be used, even if they were obtained via other means. The possibility of incorrect information being configured is minimized if there is only one machine which is authoritative for the information (i.e., there is no need to keep multiple authoritative servers in sync). For example, learning default gateways via Router

Advertisements provides perfect fate sharing. That is, gateway addresses can be obtained if and only if they can actually be used. Similarly, obtaining DNS server configuration from a DNS server would provide fate sharing since the configuration would only be obtainable if the DNS server were available.

While fate sharing is a desirable property of a configuration mechanism, in many situations fate sharing is imperfect or unavailable. When utilized to discover services on the local link, service discovery protocols typically provide for fate sharing, since hosts providing service information typically also provide the services. However, where service discovery is assisted by a DA, the ability to discover services is dependent on whether the DA is operational, even though the DA is typically not involved in the delivery of the service. Since the DA and service agents (SAs) can be out of synchronization, it is possible for the DA to provide user

agents (UAs) with service information that is no longer current. For example, service descriptions provided to the DA by SAs might be included in response to service discovery queries sent by UAs even after the SAs were no longer operational. Similarly, recently introduced SAs might not yet have registered their services with the DA. Thus, fate sharing can be imperfect.

Similar limitations exist for other server-based configuration mechanisms such as DHCP. Typically DHCP servers do not check for the liveness of the configuration information they provide, or do not discover new configuration information automatically. As a result, there is no guarantee that configuration information will be current.

"IPv6 Host configuration of DNS Server Information Approaches" [\[RFC4339\]](#) [Section 3.3](#) discusses the use of well-known anycast addresses for discovery of DNS servers. The use of anycast addresses enables fate sharing, even where the anycast address is provided by an unrelated server. However, in order to be universally useful, this approach would require allocation of a well-known anycast address for each service.

[4.](#) Security Considerations

Secure IP configuration presents a number of challenges. Secure configuration mechanisms include SEcure Neighbor Discovery (SEND)

[[RFC3971](#)] for stateless address autoconfiguration, or DHCP authentication for stateful address configuration. DHCPv4 [[RFC2131](#)] initially did not include support for security; this was added in [[RFC3118](#)]. DHCPv6 [[RFC3736](#)] included security support. However, DHCP authentication is not widely implemented for either DHCPv4 or DHCPv6.

PPP [[RFC1661](#)] does not support secure negotiation within IPv4CP [[RFC1332](#)] or IPv6CP [[RFC2472](#)], enabling an attacker with access to the link to subvert the negotiation. In contrast, IKEv2 [[RFC4306](#)] provides encryption, integrity and replay protection for configuration exchanges.

A number of issues exist with various classes of parameters, as discussed in [Section 2.6](#), [[RFC3756](#)] [Section 4.2.7](#), [[RFC3118](#)] [Section 1.1](#), and [[RFC3315](#)] [Section 23](#). Given the potential vulnerabilities resulting from implementation of these options, it is currently common for hosts to restrict support for DHCP options to the minimum set required to provide basic TCP/IP configuration.

[4.1](#). Configuration Authentication

In addition to denial-of-service and man-in-the-middle attacks, attacks on configuration mechanisms may target particular parameters. Since boot configuration determines the boot image to be run by the host, a successful attack on boot configuration could result in an attacker gaining complete control over a host. As a result, it is particularly important that boot configuration be secured.

The techniques available for securing Internet layer configuration are inherently limited, since classic security protocols such as IPsec [[RFC4301](#)] or TLS [[RFC4346](#)] cannot be used since an IP address is not yet available.

In situations where link layer security is provided, and the Network Access Server (NAS) acts as a DHCP relay or server, protection can be provided against rogue DHCP servers, provided that the NAS filters incoming DHCP packets from unauthorized sources. However, explicit

dependencies on lower layer security mechanisms are limited by the "lower layer independence" principle ([section 2.4](#)).

As a result, configuration security is typically implemented within the configuration protocols themselves. For example, IPv6 supports SEcure Neighbor Discovery (SEND) [[RFC3971](#)], DHCPv4 supports DHCP authentication [[RFC3118](#)], and DHCPv6 supports an equivalent facility [[RFC3315](#)].

Higher layer configuration typically does not have this problem. When DHCP authentication is supported, higher-layer configuration parameters provided by DHCP can be secured. However, even if a host does not support DHCPv6 authentication, higher-layer configuration via Stateless DHCPv6 [[RFC2462](#)] can still be secured with IPsec. Possible exceptions can exist where security facilities are not available until later in the boot process.

For example, it may be difficult to secure boot configuration even once the Internet layer has been configured, if security functionality is not available until after boot configuration has been completed. For example, it is possible that Kerberos, IPsec or TLS will not be available until later in the boot process.

Where public key cryptography is used to authenticate and integrity protect configuration, hosts need to be configured with trust anchors in order to validate received configuration messages. For a node that visits multiple administrative domains, acquiring the required trust anchors may be difficult. This is left as an area for future work.

[5.](#) IANA Considerations

This document has no actions for IANA.

[6.](#) References

[6.1.](#) Informative References

- [mDNS] Cheshire, S. and M. Krochmal, "Multicast DNS", June 2005.
<http://files.multicastdns.org/draft-cheshire-dnsext-multicastdns.txt>

- [PXE] Henry, M. and M. Johnston, "Preboot Execution Environment (PXE) Specification", September 1999, <http://www.pix.net/software/pxeboot/archive/pxespec.pdf>
- [RFC793] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.
- [RFC1001] NetBIOS Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, and End-to-End Services Task Force, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods", STD 19, [RFC 1001](#), March 1987.
- [RFC1332] McGregor, G., "PPP Internet Control Protocol", [RFC 1332](#), Merit, May 1992.
- [RFC1350] Sollins, K., "The TFTP Protocol (Revision 2)", STD 33, [RFC 1350](#), July 1992.
- [RFC1661] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, [RFC 1661](#), July 1994.
- [RFC1877] Cobb, S., "PPP Internet Protocol Control Protocol Extensions for Name Server Addresses", [RFC 1877](#), December 1995.
- [RFC1958] Carpenter, B., "Architectural Principles of the Internet", [RFC 1958](#), June 1996.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", [RFC 2462](#), December 1998.
- [RFC2472] Haskin, D. and E. Allen, "IP Version 6 over PPP", [RFC 2472](#), December 1998.

- [RFC2608] Guttman, E., et al., "Service Location Protocol, Version 2", [RFC 2608](#), June 1999.
- [RFC3041] Narten, T. and R. Draves, "Privacy Extensions for Stateless

Address Autoconfiguration in IPv6", [RFC 3041](#), January 2001.

- [RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", [RFC 3118](#), June 2001.
- [RFC3315] Droms, R., Ed., Bound, J., Volz,, B., Lemon, T., Perkins, C. and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), July 2003.
- [RFC3344] Perkins, C., "IP Mobility Support for IPv4", [RFC 3344](#), August 2002.
- [RFC3456] Patel, B., Aboba, B., Kelly, S. and V. Gupta, "Dynamic Host Configuration Protocol (DHCPv4) Configuration of IPsec Tunnel Mode", [RFC 3456](#), January 2003.
- [RFC3530] Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M. and D. Noveck, "Network File System (NFS) version 4 Protocol", [RFC 3530](#), April 2003.
- [RFC3720] Satran, J., Meth, K., Sapuntzakis, C. Chadalapaka, M. and E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", [RFC 3720](#), April 2004.
- [RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", [RFC 3736](#), April 2004.
- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J. and H. Levkowetz, "Extensible Authentication Protocol (EAP)", [RFC 3748](#), June 2004.
- [RFC3756] Nikander, P., Kempf, J. and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", [RFC 3756](#), May 2004.
- [RFC3775] Johnson, D., Perkins, C. and J. Arkko, "Mobility Support in IPv6", [RFC 3775](#), June 2004.
- [RFC3832] Zhao, W., Schulzrinne, H., Guttman, E., Bisdikian, C. and W. Jerome, "Remote Service Discovery in the Service Location Protocol (SLP) via DNS SRV", [RFC 3832](#), July 2004.
- [RFC3971] Arkko, J., Kempf, J., Sommerfeld, B., Zill, B. and P. Nikander, "SEcure Neighbor Discovery (SEND)", [RFC 3971](#), March 2005.

- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", [RFC 3972](#), March 2005.
- [RFC4173] Sarkar, P., Missimer, D. and C. Sapuntzakis, "Bootstrapping Clients using the iSCSI Protocol", [RFC 4173](#), September 2005.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", [RFC 4301](#), December 2005.
- [RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", [RFC 4306](#), December 2005.
- [RFC4339] Jeong, J., "IPv6 Host Configuration of DNS Server Information Approaches", [RFC 4339](#), February 2006.
- [RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", [RFC 4346](#), April 2006.
- [RFC4578] Johnston, M. and S. Venaas, "Dynamic Host Configuration Protocol (DHCP) Options for the Intel Preboot eXecution Environment (PXE)", [RFC 4578](#), November 2006.
- [RFC4795] Aboba, B., Thaler, D. and L. Esibov, "Link-Local Multicast Name Resolution (LLMNR)", [RFC 4795](#), January 2007.

Acknowledgments

Jari Arkko, Pasi Eronen, Bob Hinden, James Kempf Danny McPherson and Pekka Savola provided valuable input on this document.

INTERNET-DRAFT

Principles of Host Configuration

11 October 2007

Authors' Addresses

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

EMail: bernarda@microsoft.com
Phone: +1 425 706 6605
Fax: +1 425 936 7329

Dave Thaler
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

EMail: dthaler@microsoft.com

Loa Andersson
Acreo AB

EMail: loa@pi.se

INTERNET-DRAFT

Principles of Host Configuration

11 October 2007

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.