Zero Configuration Networking
Internet-Draft
Expires: March 20, 2003

A. Williams
Motorola

September 19, 2002

Requirements for Automatic Configuration of IP Hosts
draft-ietf-zeroconf-reqts-12.txt

Status of this Memo

Copyright Notice

Abstract

Many common TCP/IP protocols such as DHCP [RFC2131], DNS [RFC1034][RFC1035], MADCAP [RFC2730], and LDAP [RFC2251] must be configured and maintained by an administrative staff. This is unacceptable for emerging networks such as home networks, automobile networks, airplane networks, or ad hoc networks at conferences, emergency relief stations, and many others. Such networks may be nothing more than two isolated laptop PCs connected via a wireless LAN. For all these networks, an administrative staff will not exist and the users of these networks neither have the time nor inclination to learn network administration skills. Instead, these networks need protocols that require zero user configuration and administration. This document is part of an effort to define such zero configuration

(zeroconf) protocols.

Before embarking on defining zeroconf protocols, protocol requirements are needed. This document states the zeroconf protocol requirements for four protocol areas; they are: IP interface configuration, translation between host name and IP address, IP multicast address allocation, and service discovery. This document does not define specific protocols, just requirements. The requirements for these four areas result from examining everyday use or scenarios of these protocols.

Table of Contents

1. Introduction

A zeroconf protocol is able to operate correctly in the absence of configured information from either a user or infrastructure services such as conventional DHCP [RFC2131] or DNS [RFC1034][RFC1035] servers. Zeroconf protocols may use configured information, when it is available, but do not rely on it being present. For example, the use of MAC addresses (i.e. layer two addresses) as parameters in zeroconf protocols is generally acceptable because they are globally unique and readily available on most devices of interest.

The benefits of zeroconf protocols over existing configured protocols are an increase in the ease-of-use for end-users and a simplification of the infrastructure necessary to operate protocols.

This document discusses requirements for zeroconf protocols in four areas:

- IP interface configuration
- Translation between host name and IP address
- IP multicast address allocation
- Service discovery

Security considerations are also discussed.

1.1 Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2 Reading This Document

Introduction, Scenarios, Requirements, and Security Considerations are the major sections of this document.

The Scenarios & Requirements sections walk through protocol scenarios and then lists the requirements of the protocol needed to accomplish the scenario.

The Security Consideration section lists security issues with zeroconf protocols.

Requirements dispersed throughout this document begin with the text "Requirements:" or "Requirement:" on a single line, which is followed

by a bulleted list of requirements.

1.3 Zeroconf Coexistence

It is not necessary to simultaneously use zeroconf protocols in all four areas (i.e. IP interface configuration, translation between host name and IP address, IP multicast address allocation, service discovery). For example, it may make sense on some networks to provide a DHCP server for configured IP interface configuration, but perform translation between host name and IP address using a zeroconf protocol.

Given that zeroconf protocols may be deployed on existing configured networks, care must be taken in their design to ensure minimum disruption to existing networks and applications. Particular consideration should be given to the security implications of deploying zeroconf protocols in conjunction with standard configured network protocols.

Requirements:

- Zeroconf protocols MUST minimise their impact on existing networks.
- Zeroconf protocols SHOULD minimise their impact on existing applications.
- Zeroconf protocols MUST NOT be any less secure than related current IETF-Standard protocols.

1.4 Scalability

The primary reasons to deploy Zeroconf protocols are simplicity and ease-of-use. Scalability is important but it is a secondary goal. Thus, scalability should not detract from the primary goals of simplicity and ease-of-use.

1.5 Routable Protocol Requirement

Zeroconf protocols are not inherently limited to a single IP subnet. If a protocol is intended to span multiple IP subnets it MUST NOT use broadcasts or link-local addressing.

Requirement:

- Protocols intended to span multiple IP subnets MUST NOT use broadcasts or link-local addressing.

1.6 Maintaining Consistent Network State

Many networks undergo change during their lifetime. For example, hosts may be added and removed, network segments may be re-arranged, and devices may change names or run different services. In a configured network an administrator ensures that protocol parameters are updated to reflect these changes and is responsible for ensuring network consistency.

In contrast, zeroconf protocols must adapt to changing network conditions. Zeroconf protocols must be able to resolve conflicts and return the network to a consistent state after changes in network topology or other events.

Requirement:

- Zeroconf protocols MUST restore the network to a consistent state in a timely fashion when changes in network topology or other events occur.

This is a general requirement applicable to all zeroconf protocols. It should be kept in mind when considering the scenarios in Section 2 and will be applied to derive requirements for each zeroconf protocol area considered in Section 3.

2. Scenarios

The scenarios described in the next few sections highlight the general characteristics of the zeroconf protocol environment. Each zeroconf protocol needs to deal with the following aspects of the zeroconf environment.

2.1 Addition and Removal of Devices

Zeroconf protocols are expected to be useful in networks where hosts come and go. Hosts using zeroconf protocols must not assume that network resources assigned to them (e.g. address assignments, names, etc) will be appropriate for networks they subsequently join. In addition, network resources allocated to a host should be reclaimed once it leaves the network.

Requirements:

- Zeroconf protocols MUST support mechanisms to probe whether a network resource is currently in use.

- Hosts using zeroconf protocols MUST validate allocated network resources when moving to a new network or powering up.

- Zeroconf protocols MUST support timely reclamation of any network resources they allocate.

Implication:

- The information needed to allocate network resources must arrive in the network along with the host.

2.2 Network Coalescing and Partitioning

Inevitably, two or more operational networks using zeroconf protocols will be connected together, creating a single merged network. Prior to the merge, each zeroconf network has independently allocated resources (e.g. addresses, names, etc). After merging, two hosts in the merged network may end up using the same network resource, thus potentially creating conflicts.

In general a network merge "event" cannot be detected. For example, the installation of a layer-2 bridge between two zeroconf networks does not result in network interfaces going up and down on the hosts, which would be an indication that they should re-validate or reconfigure the network resources they are using.

Implication:

- It is not sufficient to rely on hosts detecting conflicts during power on or movement from network to network. Rather, detection and resolution of network conflicts is an ongoing part of zeroconf protocol operation.

Requirement:

- Zeroconf protocols MUST resolve network resource conflicts in a timely manner and on an ongoing basis.

Zeroconf protocols that detect and resolve network resource conflicts on an ongoing basis will benefit from increased robustness in the face of poor implementation, and varying network conditions.

A zeroconf network may also be split into two or more smaller independent networks. The requirement from Section 2.1 that network resources be reclaimed in a timely fashion also applies in this case. Since network merging increases the potential for network conflicts, it may be prudent to ensure that network resources associated with hosts are not immediately re-claimed for re-use. Any network which periodically joins and partitions with another zeroconf network will benefit from this behaviour. An example is an IP network in a car

joining with the home network whilst the car is parked in the garage and partitioning when it is driven away.

Requirement:

- Zeroconf protocols SHOULD NOT immediately reuse network resources as soon as they become available.
- Network resources SHOULD be allocated in a way that minimises the probability that two hosts will be allocated the same resource.
- Network resources SHOULD be allocated in a way that increases the chances of a particular host being allocated the same resource should it leave and rejoin the network.

3. Requirements

This section contains a subsection for each of the four protocol areas. Within each subsection, terms and assumptions are followed by specific zeroconf protocol requirements in that area. Each subsection ends with IPv6 considerations.

3.1 IP Interface Configuration

In this document, IP interface configuration always includes the configuration of an IP address and netmask; it may include some routing information (such as default route). IP interface configuration is needed before almost any IP communication can take place.

Terms:

IP subnet: A single logical IP network that may span multiple link layer networks. All IP hosts on the IP subnet communicate without any layer 3 forwarding device (e.g. router).

internetwork: Multiple IP subnets connected by routers.

network: a context sensitive term that may apply to one or more of the terms: a link layer network, an IP subnet, or an internetwork.

bridge: a networking device that connects two link layer networks by using only link-layer protocols (e.g. Ethernet).

IP interface configuration must take place before an IP packet can be sent from one host to another. This section requires that sufficient information be provided by a zeroconf interface configuration

protocol to allow IP packets to be sent to a unicast destination IP address on the same subnet as the sender, and on a different subnet to the sender.

Requirements: A zeroconf IP interface configuration protocol

- MUST configure an appropriate netmask.
- MUST allocate unique IP addresses within an IP subnet.
- MUST allow configuration of zero or more gateways (for the internetwork scenario).
- MUST have unique IP subnets within an internetwork (This is only for the case when there is one or more router attached in the network where autoconfigured hosts. How routers obtain their configuration is beyond of the scope of this document.)

The following requirements are derived from applying Section 2.1 and Section 2.2 to IP interface configuration.

Requirements: A zeroconf IP interface configuration protocol

- MUST be capable of discovering whether an IP address is currently in use.
- Hosts using a zeroconf interface configuration protocol MUST validate allocated IP addresses when moving to a new network or powering up.
- MUST support timely reclamation of allocated IP addresses.
- MUST resolve IP address conflicts in a timely manner and on an ongoing basis.
- SHOULD NOT immediately reuse IP addresses as soon as they become available.
- IP addresses SHOULD be allocated in a way that minimises the probability that two hosts will be allocated the same address.
- IP addresses SHOULD be allocated in a way that increases the chances of a particular host being allocated the same address should it leave and rejoin the network.

3.1.1 IPv6 Considerations

IPv6 provides a mechanism that allows a host to generate a link-local IP address Autoconfiguration[RFC2462]. Thus a zeroconf IP interface configuration solution for generating link-local addresses already exists for hosts using IPv6.

3.2 Translation between Host name and IP Address

A zeroconf name resolution protocol allows users to refer to their devices by name rather than IP address. Host names are more user friendly than IP addresses and host names have a greater chance of remaining unchanged over time. A zeroconf device connected to different networks is quite likely to use different IP addresses, however a host name may stay the same. For applications like web browsers which store bookmarks and histories, use of names rather than IP addresses is beneficial.

In a zeroconf network, the information required to construct a host name must enter the network with the device. It is expected that users will configure names into devices, or that devices will come with a pre-configured name. Devices may also construct a name from a MAC address or serial number. How this is to be achieved is outside the scope of this document.

Terms:

host name: A textual name that allows a user to refer to a host by name rather than IP address.

Requirements:

- A zeroconf name resolution protocol MUST allow host names to be mapped into IP addresses.

- A zeroconf name resolution protocol SHOULD allow IP addresses to be mapped back to names.

- Because hosts can connect and disconnect from a network at any time, the failure to resolve a name at some time MUST NOT be taken as an indication that the name will remain invalid for any length of time.

- A zeroconf name resolution protocol SHOULD support resolution of names on multiple IP subnets connected by a router.

The following requirements are derived from applying Section 2.1 and Section 2.2 to zeroconf name resolution.

Requirements:

- A zeroconf name resolution protocol MUST support mechanisms to probe whether a host name is currently in use.
- A host moving to a new network or powering up MUST ensure that all names it will respond to do not conflict with names already in use.
- Zeroconf name resolution protocols MUST allow timely re-use of hostnames.
- Zeroconf name resolution protocols MUST resolve host name conflicts in a timely manner and on an ongoing basis.
- Conflict detection procedures (such as probing for the existence of a desired host name) MUST NOT prevent valid hostnames from being resolved.
- Zeroconf name resolution protocols SHOULD NOT immediately reuse host names as soon as they become available.
- Host names SHOULD be chosen in a way that minimises the probability that two hosts will use the same name. Note that this is out of scope of the name resolution protocol itself.

3.2.1 IPv6 Considerations

Protocols to perform translation between host name and IP address have no known zeroconf-related differences for IPv4 and IPv6.

3.2.2 Relationship to the DNS

Zeroconf name resolution protocols cannot be directly equated with the DNS even though they may have a number of similarities. For example, the DNS protocols as deployed today rely on a server infrastructure that may not be present in a zeroconf environment. Host names used in zeroconf networks are inherently locally scoped whereas DNS names are global and unique by design.

At the time of writing, consensus on how zeroconf name resolution protocols should interact with the DNS has not been reached. The next sections will attempt to capture the flavour of the different approaches that have proposed.

3.2.2.1 Close coupling

In this approach an application may look up a DNS name (e.g. "www.someco.com") using an existing API and receive and answer from a zeroconf name resolution protocol. The zeroconf name resolution protocol makes use of existing on-the-wire DNS formats, resource record definitions, and namespace. Some names may have a DNS suffix that identifies them as being local in scope.

Issues yet to be resolved with this approach relate to security and consistency. If the zeroconf name resolution involves multicasting the request on a local network then the risk of spoofed responses to global DNS names like "www.someco.com" is increased. If the namespace is the same, then doing a zeroconf name lookup should return results consistent with DNS lookup for the same name. What is meant by this consistency is not agreed. Should the zeroconf lookup only be used when the DNS lookup has failed? Should that lookup reflect what would have been returned by the DNS? How should the probing for uniqueness of a zeroconf name relate to updating of a DNS record?

3.2.2.2 Completely orthogonal

Another approach is to ensure that the zeroconf namespace and the DNS namespace are completely orthogonal. There is therefore no possibility of any application using the DNS via existing APIs behaving differently after a zeroconf name resolution protocol is deployed. Applications would need to explicitly use a zeroconf name lookup API in order to resolve names using a zeroconf lookup protocol. Conversely, existing applications will derive no benefit from zeroconf protocols unless they are re-written. Deployment of a zeroconf name resolution protocol would necessitate application upgrade.

3.2.2.3 API compatible

In this approach the zeroconf namespace is distinct from the DNS namespace however zeroconf names may be resolved using an existing API. A number of operating systems use generalised name service interfaces that transparently allow a variety of name lookup protocols to be used when resolving hostnames for applications. A zeroconf name resolution protocol could be incorporated into such a system in a straightforward manner as just one more namespace and lookup protocol.

Again, there is increased risk of spoofed responses if the multicast zeroconf name resolution protocol is used to resolve "www.someco.com". One possible way of minimising the security risk

is to ensure that locally scoped names are distinguishable from DNS names, perhaps via a known reserved DNS suffix or by virtue of not containing a dot. A multicast zeroconf resolution protocol could then avoid making requests for names which look like global DNS names. Alternatively, we could require that zeroconf name lookups only be performed when the equivalent DNS lookup has failed.

3.3 IP Multicast Address Allocation Scenarios

IP Multicast is used to conserve bandwidth for multi-receiver bulk-delivery applications, such as audio, video, or news.

IP Multicast is also used to perform a logical addressing function. For example, when a host needs to communicate with local routers, it can send packets to the all-routers multicast address without having to know in advance the IP address(es) of the router(s).

IPv4 multicast addresses range from 224.0.0.0 to 239.255.255.255.

[RFC2365] defined multicast scopes are:

node-local	(unspecified for IPv4)
link-local	(224.0.0.0/24)
local	(239.255.0.0/16)
site-local	(unspecified for IPv4)
organizational-local	(239.192.0.0/14)
global	(224.0.1.0-238.255.255.255)

A relative assignment is an integer offset from the highest address in the scope and represents a 32-bit address. For example, within the local scope, 239.255.255.0/24 is reserved for relative allocations.

Source-Specific Multicast addresses are 232.0.0.0 to 232.255.255.255 [I-D.ietf-ssm-arch].

Unicast-prefix-based IPv6 multicast addresses are defined, as well as source-specific multicast addresses for IPv6[RFC3306].

Assumptions:

- The node-local and SSM addresses require no protocol or interaction between multiple hosts, thus are not mentioned further in this document.

- Global and organizational scoped addresses are meant for networks of a greater scale than zeroconf protocols, thus are not mentioned further in this document.

- Only local, link-local and site-local scopes are considered further in this document.

- If it is desirable to restrict multicast packets from entering and leaving a multicast scope boundary, it is assumed that the router at the boundary is a "boundary router" as described in [RFC2365].

Scenarios are scope enumeration, address allocation, and multiple sources.

3.3.1 Scope Enumeration

Applications that leave the choice of scope up to the user require the ability to enumerate what scopes the host is operating within. In addition, services that are assigned relative addresses require the ability to enumerate what scopes the host is within; only then will a host be able to apply the relative address to a scope.

Requirements: Application support software used to autoconfigure multicast addresses

- MUST list which of the scopes (local, link-local, or site-local) are available for hosts.

- MUST list per-scope address ranges that may be allocated.

3.3.2 Address Allocation

IP multicast address allocation (local, link-local and site-local scopes only) requires an application to be able to request the use of a suitable multicast address. Coordination among applications must occur to avoid conflicting allocations of the same address. This coordination must span the entire scope respective to the address. When an allocated address is no longer required, that address MUST become available for use again.

Requirements: A zeroconf multicast address allocation protocol

- MUST select a multicast address.

- MUST prevent conflicting allocations of the same address.

- MUST allow the multicast address to become available after the address is no longer in use.

3.3.3 Multiple Sources

An intercom system inside a home is an example of a multiple source IP multicast application. In this type of application, several sources may be sending packets destined to the same IP multicast address.

This multiple source example illustrates the problem that a particular address may continue to be valid, even after the host that initially allocated the address is no longer present; the zeroconf multicast address allocation must correctly support this type of operation. In other words, if a host allocates a multicast address, then leaves the multicast group, some other host must defend the address.

Requirements:

- A host other than the allocating host MUST be able to defend or otherwise maintain the allocation of a multicast address.

3.3.4 IPv6 Considerations

To date, no range has been reserved for dynamic allocation of Link-scoped addresses in IPv4. Hence, unless such a range is reserved, dynamic allocation of link-scoped addresses applies only to IPv6.

3.4 Service Discovery Scenarios

Service discovery protocols allow users or software to discover and select among available services. This removes the requirement that the user or client software know a server's location in advance in order for the client to communicate with the server.

Terms:

service: a particular logical function that may be invoked via some network protocol, such as printing, storing a file on a remote disk, or even perhaps requesting delivery of a pizza.

service characteristics: Characteristics provide a finer granularity of description to differentiate services beyond just the service type. For example if the service type is printer, the characteristics may be color, pages printed per second, location, etc.

service discovery protocol: A service discovery protocol enables clients to discover servers (or peers to find other peers) of a

particular service. A service discovery protocol is an application layer protocol that relies on network and transport protocol layers.

service protocol: A service protocol is used between the client and the server after service discovery is complete.

The scenarios are the discovery of a simple printer service.

3.4.1 Printer Service

Network-enabled printers allow various network clients to submit print jobs. A service discovery protocol MUST allow a printer service to be discovered by devices needing to print. This requires a service type as well as a service identifier to distinguish instances of a single service type. Service discovery MUST be independent from any particular printing protocol such as lpd, raw-tcp, ipp.

Printers vary in their characteristics such as location, status, dots per inch, etc. Discovering a service based on these characteristics SHOULD be part of the service discovery protocol.

Service discovery MUST complete in a timely (10s of seconds) manner.

Requirements:

- MUST allow a service to be discovered.
- MUST discover via service identifier and/or service type.
- MUST discover services without use of a service-specific protocol.
- SHOULD discover via service characteristics.
- MUST complete in a timely (10s of seconds) manner.

3.4.2 IPv6 Considerations

Service discovery protocols have no zeroconf related differences for IPv4 and IPv6.

4. Security Considerations

Zeroconf protocols are intended to operate in a local scope, in networks containing one or more IP subnets, and potentially in parallel with standard configured network protocols.

Application protocols running on networks employing zeroconf protocols will be subject to the same sets of security issues identified for standard configured networks. Examples are: denial of service due to the unauthenticated nature of IPv4 ARP and lack of confidentiality unless IPSec-ESP, TLS, or similar is used. However, networks employing zeroconf protocols do have different security characteristics and the subsequent sections attempt to draw out some of the implications.

Security schemes usually rely on some sort of configuration. Security mechanisms for zeroconf network protocols should be designed in keeping with the spirit of zeroconf, thus making it easy for the user to exchange keys, set policy, etc. It is preferable that a single security mechanism be employed that will allow simple configuration of all the various security parameters that may be required.

Generally speaking, security mechanisms in IETF protocols are mandatory to implement. A particular implementation might permit a network administrator to turn off a particular security mechanism operationally. However, implementations should be "secure out of the box" and have a safe default configuration.

Zeroconf protocols MUST NOT be any less secure than related current IETF-Standard protocols. This consideration overrides the goal of allowing systems to obtain configuration automatically. Security threats to be considered include both active attacks (e.g. denial of service) and passive attacks (e.g. eavesdropping). Protocols that require confidentiality and/or integrity should include integrated confidentiality and/or integrity mechanisms or should specify the use of existing standards-track security mechanisms (e.g. TLS [RFC2246], ESP [RFC1827], AH [RFC2402]) appropriate to the threat.

4.1 The Basic in/out Security Policy

The claim/collide approach to resource allocation is attractive in the zeroconf environment. To operate securely, hosts allocating resources need to ensure that messages indicating that a network resource is in use are authenticated. Hosts soliciting for a name or service must also be able to authenticate the responses they receive. A message is considered "authenticated" if it can be proved to have been sent by a member of a group of devices running zeroconf protocols. Note that in general, devices running zeroconf protocols must trust the other devices in the group because any device may claim to be using an address or name, or advertising a service. Zeroconf security mechanisms must at a minimum be able to distinguish between messages originating from a device "inside" the group or a device "outside" the group.

Requirement:

- Security schemes for zeroconf protocols MUST be able to implement a basic "inside/outside" security policy.

Access control mechanisms within a zeroconf network are beyond the scope of this document.

4.2 Security Scenarios

In the next few sections, several scenarios are examined to highlight the security implications of employing zeroconf protocols in various environments.

4.2.1 Use on an isolated, secure network link

In this scenario all the devices in the network are connected to a secure link (e.g. a control network in a car, or a private network between two devices used for configuration). The link might be a separate physical wire or shared media with layer-2 security enabled (e.g. wireless or power-line networks). Any host that has access to the link is trusted and any packet received from the secure link is considered to be authenticated. In this case, the inside/outside policy is implemented by controlling access to the link itself.

4.2.2 Use on a shared (insecure) link

In this scenario, a group of devices use zeroconf protocols on link(s) they share with other devices who are NOT part of the group. Various pieces of network configuration MUST be consistent across all hosts sharing the link (e.g. addresses, netmask, routing information) in order for communication to occur at all. Consequently, hosts inside the group are potentially at risk from hosts outside their group when they try to configure such information (e.g. via ARP insecurity). Host names and service identifiers MUST be unique within a group, but with authentication may not need to be unique across the link. Assuming that requests and responses can be associated with a group and authenticated, solicitations and responses for host names and services that are not located inside the group can be ignored.

4.2.3 Use in conjunction with configured protocols

Zeroconf protocols are likely to be used in conjunction with configured network protocols. In general, zeroconf protocols must not allocate resources from the same address or name spaces used by configured network protocols. Locally allocated zeroconf network resources must not mask global assigned resources.

5. IANA Considerations

No known IANA considerations arise from this document.

6. Acknowledgments

Thanks to Peter Ford and Stuart Cheshire for hosting the NITS (Networking In The Small) BOF that was the catalyst to forming the Zeroconf Working Group.

Thanks to Erik Guttman and Stuart Cheshire for forming and chairing the Zeroconf Working Group, which is responsible for this document.

Thanks to Erik Guttman for providing key input to the service discovery and the security sections.

Thanks to Dave Thaler for providing key input to the IP multicast address allocation sections.

Thanks to Stuart Cheshire for providing key input to the introduction and IP interface configuration sections.

Thanks to Myron Hattig for acting as editor for previous versions of this document.

Additional recognition goes the following people for their influential contributions to this document and its predecessors: Brent Miller, Thomas Narten, Marcia Peters, Bill Woodcock, Bob Quinn, John Tavs, Matt Squire, Daniel Senie, Cuneyt Akinlar, Karl Auerbach, Kanchei Loa, Dongyan Wang, James Kempf, Yaron Goland, and Bernard Aboba, Ran Atkinson.

Normative References

Informative References

[I-D.ietf-ssm-arch] Cain, B. and H. Holbrook, "Source-Specific Multicast for IP", draft-ietf-ssm-arch-00 (work in progress), February 2002.

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC1827] Atkinson, R., "IP Encapsulating Security Payload (ESP)", RFC 1827, August 1995.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2246] Dierks, T., Allen, C., Treese, W., Karlton, P., Freier, A. and P. Kocher, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC2251] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[RFC2365] Meyer, D., "Administratively Scoped IP Multicast", BCP 23, RFC 2365, July 1998.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[RFC2461] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[RFC2535] Eastlake, D., "Domain Name System Security

Extensions", RFC 2535, March 1999.

[RFC2541] Eastlake, D., "DNS Security Operational Considerations", RFC 2541, March 1999.

[RFC2608] Guttman, E., Perkins, C., Veizades, J. and M. Day, "Service Location Protocol, Version 2", RFC 2608, June 1999.

[RFC2730] Hanna, S., Patel, B. and M. Shah, "Multicast Address Dynamic Client Allocation Protocol (MADCAP)", RFC 2730, December 1999.

[RFC2931] Eastlake, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 3306, August 2002.

Author's Address

Aidan Williams
Motorola Australian Research Centre
Locked Bag 5028
Botany, NSW 1455
Australia

Phone: +61 2 9666 0500
EMail: Aidan.Williams@motorola.com
URI: http://www.motorola.com.au/marc/

Full Copyright Statement

Acknowledgement

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