Automatic Router Configuration Protocol

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [1].

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/1id-abstracts.txt

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

Abstract

Automatic Router Configuration Protocol is a method by which Routers may be automatically configured according to a predefined or default scheme of behavior and address allocation. It is routing protocol independent and is based on a Client-Server model, working in conjunction with DHCP [2] and using a system of IP messages for communication between participating Router Clients and Servers. It is easy to use as a labor-saving device in less secure environments and will work even in conjunction with unmodified DHCP Server implementations. However, it also has inherent security features and is capable of highly secure implementation.

Glossary

ARCP - Automatic Router Configuration Protocol: the protocol described in this document.

ARCP-Active Interface - An interface on an ARCP Router Client which has been given an IP configuration by the ARCP Server.

ARCP-Disabled Interface - An interface which is detected by the router to be unconnected or which is administratively disabled.

ARCP-Passive Interface - An interface on an ARCP Router Client which has no IP configuration but which runs BOOTP/DHCP proxy and polls for ARCP peers, in order to conserve IP allocations until it detects another machine on the connected network.

ARCP Seed (‘Seed’) - An opaque-field text value which uniquely identifies an administrative domain for ARCP.
ARCP Server (‘Server’) - An IP node which processes requests from ARCP Clients for registration and configuration, and which can serve as a centralized point for automatic configuration control in an ARCP-enabled network. Distributed ARCP Server implementation and inter-server communication is possible but is beyond the scope of this document.

Client ID - A six-octet unique and opaque identifier of an individual router. By convention this is the MAC address of the router’s first Ethernet or similarly-addressed interface.

Connected Interface - An interface on an ARCP-enabled router which the router detects to be connected, or possibly connected, to another router or routers. The meaning of ‘Connected’ varies by Media-type, as different media provide varying amounts of information on their connection state.

Host Client - An ARCP-enabled router which has acquired a host IP configuration and can communicate via IP, but has not yet been sent any further configuration information by an ARCP Server.

Interface Type - Interface Type is the generic description of groups of Media-type, including Broadcast/NBMA/Point-to-Point, single/multiple-subnet, ARCP-Capable/Incapable and DHCP-Capable/Incapable.

Interface ID - An opaque identification string which uniquely identifies an interface or subinterface on a router.

Media-type - This is the specific type of medium for a router interface, such as Fast Ethernet or Frame Relay. This is an optional field in all ARCP messages, for informational purposes. It is beyond the scope of this document to define type formats for this field.

Router Client (‘Client Router’) - A fully-configured ARCP Client which has received routing configuration from an ARCP Server.

Secure Seeded Client - A Seeded Client with the Secure option set on its seed. Such clients will not include their seed in any messages except through authenticated and/or encrypted sessions.

Seeded Client - Any ARCP Host Client or Router Client which has an ARCP Seed. A Client SHOULD only have one Seed at a time. This document will not consider clients with more than one Seed.

Subinterfacing - For our purposes, this term is used to describe a physical router interface which is divided into one or more logical interfaces. For broadcast media such as ethernet this means separate virtual LANs/broadcast domains, not multiple IP addresses on a single logical segment. For point-to-point and non-broadcast multiple access media such as Frame Relay or ATM this means virtual circuits.

Unconfigured Client - An ARCP-enabled router which has not yet received any configuration data via ARCP. It may or may not also be an "Unseeded Client"

Unconnected Interface - An interface on an ARCP-enabled router which the router detects to be unconnected or disabled by configuration.

Unseeded Client - An ARCP-capable router which has not yet been assigned an ARCP Seed or any other configuration data via ARCP.
Requirements

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 RFC-2119 [3].

Table of Contents

1. Behavior Examples 5
   1.1 Basic Configuration Behavior 5
   1.2 High-security Configuration Behavior 6
2. Special Considerations 7
3. ARCP Extensions to BOOTP/DHCP 8
   3.1 Point-to-Point Extensions for Relay Agent 8
   3.2 Unnumbered Interface Relay Agent 8
   3.3 Relay Agent extensions for Peer Discovery 9
4. Peer Discovery Process 9
5. ARCP Server Requirements 10
   5.1 Client Registration Tracking Table 10
   5.2 IP Prefix Allocation Input Table 11
   5.3 IP Prefix Allocation Tracking Table 12
6. Examples of Operation 12
   6.1 ISP POP Expansion Example 13
   6.2 Secure Battlefield Networking Example 13
   6.3 Automated Enterprise Networking Example 14
   6.4 ARCP AutoServer - Zero Configuration Internetworking 15
7. Message Formats and Options 16
   7.1 DHCP ARCP-Seed Option (DHCP Option XXX) 16
   7.2 DHCP ARCP-Server Option (DHCP Option XXX) 16
   7.3 ARCPREG Message (Client to Server) 17
   7.4 ARCPCFG Message (Server to Client) 18
   7.5 ARCPREQ Message (Client to Server) 19
   7.6 ARCPSTATUS Message (Client to Server) 20
   7.7 ARCPQUERY Message (Server to Client) 22
8. ARCP Options 24
   8.1 All ARCP Options 24
   8.2 Router Options 25
   8.3 Interface Options 26
   8.4 Protocol Options 28
   8.5 Vendor-Specific Options 32
9. Security Considerations 35
10. IANA Considerations 35
11. Open Issues 35
12. References 35
13. Acknowledgments 36
14. Author’s Addresses 36

1. Behavior Examples

The following sequences describe typical ARCP Client-Server dialogues for unsecured and secured configurations.

1.1 Basic Configuration Behavior

i. An unconfigured ARCP-capable router is turned on. On any interfaces that the router determines are or may be connected to other devices, the unconfigured router begins to request an IP address using DHCP (or equivalent means for non-broadcast media). The
DHCPDISCOVER (or equivalent) message specifies the ARCP-Seed and ARCP-Server options.

ii. The DHCPDISCOVER is relayed to a DHCP server, which returns a DHCPOFFER with basic IP configuration data along with the ARCP Seed and the IP address of the ARCP Server. The requesting router now has an IP attachment to the network and can act as a regular IP host, but it cannot route.

iii. The newly host-configured router sends an ARCPREG message to the ARCP Server. This message contains the ARCP Seed and basic information about the requesting router.

iv. The ARCP Server registers the new router and returns basic router configuration data to the client using one or more ARCPcfg messages. This includes configuration lease time, IGP routing protocol data, and access/authentication data. The newly registered and configured router is now an ARCP Router Client.

v. The Router Client can now participate in the IGP on the interface from which it received its configuration, and its other interfaces are configured as "ARCP-Passive". It can also request further configuration from the ARCP Server using ARCPreq messages.

vi. ARCP-Passive interfaces have no IP configuration but are configured with default Physical- and Link-layer parameters. The Router Client listens for Bootstrap DHCPDISCOVER messages on these interfaces. When it receives one of these messages, it obtains from the ARCP Server an IP configuration for the interface on which the request was received, then forwards the request on to the DHCP server, acting as a Bootstrap Relay Agent [4] on that interface from then on. Interfaces given an IP configuration and enabled as Bootstrap Relay Agents are designated ARCP-Active.

vii. Router Clients continue to request interface configuration data from the ARCP Server whenever a DHCP-style request is received on an interface.

viii. In order to allow ARCP-configured routers to discover and connect to each other, whenever a new ARCP-Passive or ARCP-Passive interface is activated, and then at regular intervals, the interface broadcasts a DHCPDISCOVER message containing the loopback interface address of the requesting router in its ciaddr field [ii].

ix. Upon receiving a DHCPDISCOVER message with the ciaddr field set, the relay agent on a neighboring router forwards the message to the DHCP and ARCP Servers. The ARCP Server detects a request from a registered Client Router though the relay agent of another registered client router and issues ARCPcfg messages to the two adjacent routers to configure their connected interfaces and to run the IGP. (This exchange of messages may also be coordinated between multiple linked ARCP Servers in a large network.)

1.2 High-security Configuration Behavior

This example of secure operation is included for illustrative purposes. This document describes a framework in which Secure ARCP may be implemented, but the details of such implementation are beyond the scope of this document.

i. The DHCP Server is connected directly to unconfigured routers in
an isolated, physically secure environment before the routers are deployed (for example, using an ARCP Server on a laptop with a cross-over ethernet cable). DHCP messages containing an ARCP Secure Seed should not be relayed across a network, as the Seed is in cleartext at this stage.

ii. When an unconfigured router is connected to the DHCP Seed Server, it receives a Seed in the standard manner, as per steps 1-2 of the Basic Behavior description.

iii. The DHCP Seed Server also gives the client any keys and configuration information needed to use DHCP authentication on the network (if applicable), and to establish IPSec sessions with the Secure ARCP Server.

iv. The DHCP Seed Server MAY give out the ARCP Server address if it is known, though this is not required. A router which has received a Secure Seed in this manner is a Secure Seeded Client.

v. The Secure Seeded Client is fielded and connected to another router during the lifetime of its Seed’s lease, and uses a BOOTREQUEST with the ARCP Server Option set to receive an IP address on the new network, and the IP address of the ARCP Server. This transaction SHOULD make use of the mechanism for authentication as described in rfc3118 [5].

vi. Having securely obtained an IP host configuration and the address of a Secure ARCP Server, the client sets up an IPSec-based secure TCP session with the ARCP Server [6], and proceeds to register and to obtain and update its configuration as described in the Basic Behavior section above, but exchanging all messages securely.

2. Special Considerations

Any router which receives an ARCP Seed via DHCP but not an ARCP Server Address SHOULD continue to use the standard DHCP process to try to obtain the address of an ARCP Server. Optionally, such routers MAY also use other service location protocols to locate an ARCP Server.

If a client has an ARCP Seed which is still within its valid lease time, any DHCP or ARCP messages received by that client which contain a Seed different from its own Seed MUST be ignored.

ARCP configuration data for interfaces SHOULD expire if the interface is left unconnected for a certain amount of time. This enables routers to be deployed and re-deployed securely in the field until the ARCP Seed lease expires.

ARCP Seeds and ARCP configuration data SHOULD have independent leases. In this way, ARCP configuration data for interfaces SHOULD expire if the interface is left unconnected for a certain amount of time, enabling routers to be deployed and re-deployed securely in the field until the ARCP Seed lease expires.

Unconfigured Clients receiving an ARCP-Server DHCP Option with an addressing set that the client does not support (e.g. IPv6 ARCP Server address for a client that only supports IPv4) SHOULD keep any other valid data in the message (such as an ARCP Seed) while ignoring the unsupported ARCP-Server Option and triggering an appropriate
error response.

Unconfigured Clients receiving more than one ARCP-Server DHCP Option MAY attempt to register with all of them successively or simultaneously, but MAY discard all addresses other than the first.

3. ARCP Extensions to BOOTP/DHCP

In order for the ARCP protocol to be fully automated and to work not only with broadcast media (e.g. Ethernet, Token Ring) but also with Point-to-Point and Non-Broadcast Multiple Access (NBMA) media (e.g. Serial, ATM, Frame Relay), some extensions are needed to the currently defined operation of DHCP.

These extensions, however, do not require any change to the functioning of a standard DHCP Server. Rather, there are two new DHCP Options defined for ARCP and some new BOOTP/DHCP Relay Agent functionality. The two new DHCP Options are defined in the Message Formats and Options section below.

It is worth noting that other methods are possible for the acquisition of IP and ARCP configuration data over non-broadcast media (e.g. RADIUS). The specification of a new DHCP Relay Agent function was felt to be the simplest in the context of a unified overall solution supported by a single unmodified server.

3.1 Point-to-Point Extensions for Relay Agent

In order for an ARCP-enabled router to connect automatically to a network via any non-broadcast medium, it will need a way to acquire an IP Address, an ARCP Server Address, and an ARCP Seed from the DHCP Server. In general, this will be done by means of a special Relay Agent function.

The specific method of this operation will vary depending on the type of medium involved. For example, simple Point-to-Point links without sub-interfaces would require a method different from that of a NBMA link such as Frame Relay or ATM. Each type will require a standard encapsulation or method of determining a common encapsulation. The determination of these standards will be addressed in a separate document.

3.2 Unnumbered Interface Relay Agent

In order to avoid unnecessarily assigning IP prefixes to interfaces not in use, ARCP-enabled routers will need to be able to listen for DHCP BOOTREQUEST messages on unnumbered interfaces. These are the ARCP-Passive interfaces as described above.

When a DHCP message is received by such an interface, the message much be cached while the receiving router acquires an IP configuration for that interface from the ARCP Server (by means of an ARCPREQ and an ARCP-CFG message). Only once the interface has been thus configured does the router forward the DHCP message to the DHCP server as a standard Relay Agent would.

3.3 Relay Agent extensions for Peer Discovery

The third special extension to the Relay Agent function is needed in order to allow configured ARCP Router Clients to discover each other
when a new connection is made between them. This requires that a router running the Relay Agent function be able to:

1. At regular intervals to send out a special DHCP BOOTREQUEST while configured as a Relay Agent.
2. To forward any incoming BOOTREQUEST messages to *both* the DHCP and ARCP Servers if the "ciaddr" field is populated.

4. Peer Discovery Process

When two or more Router Clients have been attached to a network and configured by the ARCP Server and are later attached by a new physical link or bridged connection, they need to be able to discover each other and begin routing to each other when appropriate. This is accomplished by a novel use of DHCP messages and Relay Agent functionality. (See the preceding section ‘Relay Agent extensions for Peer Discovery’.)

The process operates in this way:

1. Two or more ARCP-Active or ARCP-Passive router interfaces on different routers are attached to the same physical or bridged network segment.

2. At regular intervals, each router sends a "dummy" BOOTREQUEST to each of its interfaces with the ciaddr field set to the router’s own loopback interface address as registered with the ARCP Server. (The sent message is called a "dummy" because the sending router will never accept any configuration offers it receives in reply to such messages - such replies MUST be ignored.)

3. When the neighboring router receives such a request and sees that the ciaddr field is populated, it forwards the BOOTREQUEST message not only to the DHCP Server but also to the ARCP Server, in the same format as if it were a DHCP Server. When a router forwards a BOOTREQUEST message, it SHOULD check the ciaddr field, forwarding a copy to the ARCP Server only if the field is populated. Note that if the DHCP and ARCP Servers have been integrated, only one forwarded message is needed.

4. The ARCP Server MUST listen for incoming BOOTREQUEST messages as if it were a DHCP Server. The ARCP Server examines each incoming BOOTREQUEST against its tables of registered client routers. If it determines that both the address of the relay agent and that of the requesting router (in the ciaddr field) are locally registered, it generates the ARCP_CFG message(s) to configure them to route data as appropriate through the new link.

5. ARCP Server Requirements

This section outlines the minimum set of tables and fields that must be managed by an ARCP Server. Other details of ARCP Server behavior is left to the implementor. It is not the purpose of this document to dictate the particular embodiment of tables and fields; they may take the form of flat text files, distributed databases, or any other embodiment deemed appropriate by the implementor.

Server implementations involving multiple coordinated ARCP Servers in a single administrative are possible but are beyond the scope of this document.
Field descriptions below are followed by a parenthetical notation of the type or minimum size of the field; these are RECOMMENDED but left to the discretion of the implementor.

5.1 Client Registration Tracking Table

The purpose of this output table is to track the current status of clients as they register and deregister with the server. The table itself SHOULD NOT be user-editable, but information from the table SHOULD be viewable by an administrator to determine the current state of the clients. Likewise, an administrator MAY be able to use this table as the basis for issuing manual changes to client configuration using ARCPFCFG and ARCPRESET messages.

The following information MUST be tracked in this table for each registered client:

- Client ID - See definition in glossary (6 Octets)
- Seed - This is only REQUIRED if the server is capable of managing clients with multiple seeds (128 Octets)
- Unicast IP address - The unicast address used to communicate with the client; this would usually be the DHCP-assigned address for a Host Client and a loopback address for a Router Client. (4 Octets for IPv4 or 16 Octets for or IPv6)
- Host/Router Flag - Records whether the client is a Host Client or a Router Client (binary flag)
- Make - An opaque field intended to be a human-readable record of the client router Hardware Manufacturer (32 Octets)
- Model - An opaque field intended to be a human-readable record of the client router’s model name or number (32 Octets)
- OS Version - An opaque field intended to be a human-readable record of the client router’s Operating System version (32 Octets)

RECOMMENDED fields for this table include:

- Authentication/Encryption data for the client, if any (Implementation-specific size)
- Multicast Group the client belongs to, if any (4 Octets)
- Hostname - An opaque field intended to be a human-readable record of the client router’s hostname, if any (32 Octets)
- Location - An opaque field intended to be a human-readable record of the client router’s physical location and situation (128 Octets)

In addition to the above fields it will be necessary for the ARCP Server to maintain information about each Client’s configuration details. This MAY be done by means of:

- Additional large fields in the Client Registration Table containing full individual configuration details,
- Pointers in this table to individual client configuration records,
- Pointers in this table to configuration profiles for many clients,
- Any combination of the above or similar methods, as deemed appropriate by the implementor.

5.2 IP Prefix Allocation Input Table

This table allows the administrator to input IP address blocks to be allocated to the clients. Each of the following attributes MUST be tracked for each prefix in the table:

- Block Prefix - The IP address prefix to be allocated (4 Octets for
IPv4 or 16 Octets for or IPv6)
- Block Mask - The number of bits in the mask of the whole block (1 Octet)
- Allocation Mask - The number of bits in the mask of prefixes to be allocated from within the block, which MUST be greater than or equal to the Block Mask (1 Octet)
- Network Type - Point-to-Point or Broadcast (binary flag)
- Status - Allocate/Hold/Reclaim (ternary flag)

The following attributes SHOULD be tracked for each prefix:

- Up/Down - A flag to determine whether prefixes from the block are allocated from the lowest to the highest or vice versa (binary flag)
- Low-Reserved - Number of allocation prefixes to reserve at the beginning of the block. (1 Octet)
- High-Reserved - Number of allocation prefixes to reserve at the end of the block. (1 Octet)
- Address Type - An opaque field, intended to record whether the prefix is from Public or Private address space. The field may contain one of the keywords "Public" and "Private", or a user-defined keyword. (64 Octets)

For implementations which integrate ARCP and DHCP functionality, the following attributes MAY be tracked:

- ARCP/DHCP - This attribute records whether a prefix is to be used for ARCP or DHCP allocation only, or either (exclusively), or both.
  - Any other DHCP-specific data.

IP address blocks within this table SHOULD NOT be allowed to overlap. The ARCP Server MUST either reject changes to the table which cause overlapping blocks, or include implementation-specific rules for the handling of overlapping blocks.

5.3 IP Prefix Allocation Tracking Table

This table records the current status of IP allocations. Like the Client Registration Tracking Table, it SHOULD NOT be directly user-editable, but information from the table SHOULD be viewable by an administrator to determine the current state of the IP allocations. Likewise, an administrator MAY be able to use this table as the basis for issuing manual changes to client configuration using ARCP CFG and ARCPRESET messages.

The following attributes MUST be tracked for each prefix that has been allocated:

- Allocated Prefix - The IP address prefix allocated (4 Octets for IPv4 or 16 Octets for or IPv6)
- Mask - The mask of the allocated prefix (1 Octet)
- Requesting Client ID - The Client ID of the Router Client to which this prefix was allocated (6 Octets)
- Connected Client IDs - The Client IDs of any other Router or Host Clients which have been determined to be attached to the network segment for which this prefix was allocated. Multiple entries are possible for this field. (6 Octets per entry)

6. Examples of Operation

This section describes examples of operational scenarios using ARCP,
and is intended partially in the sense of an Applicability Statement.

6.1 ISP POP Expansion Example

An Internet Service Provider wants to implement several new Points of Presence (POPs) at locations remote from their engineering staff. They have contracted a Collocation provider which provides "Remote Hands and Eyes" services to place their equipment and connect cables as instructed by phone, without configuring the equipment.

Rather than having routers for the new POPs shipped to the ISP’s headquarters for configuration and then reshipped to the new POPs, or flying engineers to the new POPs to configure the equipment manually, the ISP purchases routers with "ARCP-Ready" Operating Systems. The routers are shipped directly to the new POPs and plugged in by the Collocation site technicians.

When turned on, the new routers find some interfaces connected via contracted leased service lines (e.g. DS-3, OC-3, or long-haul Ethernet services) to an ARCP-enabled Router at one of the ISP’s preconfigured POPs. Polling these interfaces, their requests are relayed to a DHCP Server and an ARCP server, which register them and return to them the ISP’s ARCP Seed, followed by configuration parameters. The routers are configured as members of the same OSPF Area as their upstream routers, with passwords and access lists permitting login only from a predetermined range of addresses. The routers immediately become visible to the ISP, with secure configurations, before the Collocation technician is off the phone.

From this point, additional configuration is done remotely. Additional equipment for the new POP may be ordered and activated in similar fashion. Additionally, any equipment ordered and shipped first to the ISP before shipping to remote sites may be minimally preconfigured for security, by giving each piece of equipment the ISP’s ARCP Seed only, in order to ensure that it will only take configuration from the ISP’s other equipment.

This procedure can save several days and potentially thousands of dollars by reducing shipping time and expense, and reducing the need for travel by engineers.

6.2 Secure Battlefield Networking Example

The Army Signal Corps needs to support rapid deployment and redeployment in the field while reducing the time taken by technicians using networking equipment configuration checklists. In order to facilitate this, they preconfigure all networking equipment for secure ARCP.

All routers are preloaded from the manufacturer with ARCP-enabled software with strong encryption and authentication capability for DHCP [2] and IPSec [6]. Before fielding, Signal Corps technicians attach each unconfigured router to a laptop computer configured as a DHCP Seed Server. However, instead of giving the routers a full configuration, the Server gives each an appropriate ARCP Seed and an individual private key which is registered in a central key repository.

Once fielded and connected, the routers use strongly-authenticated DHCP messages to request a basic IP configuration and an ARCP Server address. They then initiate a strongly-authenticated and encrypted
IPSec connection which identifies the new router to the ARCP Server and the ARCP Server to the new router, before the Router may receive configuration.

All configuration data on the routers, including the ARCP Seed, secure keys, and configuration data, is encoded with lease times after which they will be wiped from memory and overwritten. In cases where a communications node is suspected to have been overrun or otherwise compromised, technicians at a rearward post can have ARCP Servers send ARCPRESET messages to remove all configuration data from compromised equipment and lock out their Router IDs from the Servers.

Lease times may be tailored to facilitate rapid redeployment of routers in this scenario. For example, ARCP Servers may use periodic unicast or multicast ARCPQUERY messages to verify presence of routers in the field, and to de-allocate their assigned resources after five minutes of missed responses. The configuration parameters on the routers themselves (not including Seeds and Keys) may be configured with a five-minute expiration time after disconnection or lack of authenticated messages from the Server. In this way, when a command post or communications node disconnects and redeploy, it is immediately ready to obtain a new configuration upon reconnection.

### 6.3 Automated Enterprise Networking Example

An enterprise is expanding rapidly and does not employ a full-time Network Engineer, and has only a small block of public IP addresses. The company purchases an integrated Firewall/DHCP/ARCP Server appliance with a simple Web-based GUI for configuration. The appliance connects on one side to the company’s business-class DSL service, and provides NAT service and [RFC 1918](https://www.rfc-editor.org/rfc/rfc1918) Private IP Addresses via DHCP to internal networks.

In addition, the appliance performs basic routing functions including RIPv2 or Single-area OSPF, and is capable of routing Public IP traffic to the Internet in addition to its normal NAT operation. The ARCP Server automatically assigns 30- or 32-bit IP prefixes to Point-to-Point links, and 24-bit prefixes to broadcast media, all from [RFC 1918](https://www.rfc-editor.org/rfc/rfc1918) private address space.

The company’s Server Administrator can use the Web-based GUI to allocate Public IP addresses to the particular LANs and particular MAC addresses where machines are present that require external visibility with public IP space. Alternately, externally-visible servers can participate in the network’s internal routing protocol using 32-bit host routes, for the greatest flexibility and most conservative us of limited public IP space.

### 6.4 ARCP AutoServer - Zero Configuration Internetworking

A router manufacturer decides to build both ARCP Client and DHCP/ARCP Server capabilities into its router’s operating system. When such a router boots it acts only as a client for a preset time, after which it will continue to act as a Client but will also activate its preconfigured Server capability in response to any DHCP/ARCP requests it receives from connected machines.

Users determine which Router should act as Server for the network by turning it on first, and then attaching other boxes. If the routers do not already have a common Seed, the first router generates a Seed at random (or uses a null-valued Seed) and automatically allocates
rfc1918 private prefixes and basic standardized configurations in response to requests from DHCP and ARCP. For example, the following trivial AutoServer configuration could be used:

- Default encapsulations for all interface types
- Allocate a 24-bit prefix from 172.16.0.0/16 to each broadcast-capable interface that comes up in either DHCP or ARCP.
- Allocate a 30-bit prefix from 172.17.0.0/16 to each point-to-point interface that comes up in ARCP.
- Server router also acts as a Multicast RP, all routers run PIM- SM/SSM on all interfaces and listen to a standard multicast group for configuration changes from the Server.
- All routers run OSPF, with all interfaces initially in Area 0
- All interface configurations are released when disconnected for one minute, and the routers revert to unconfigured state if disconnected for 10 minutes.
- Routers have no initial password but accept login initially only via telnet from the Server.

In this way a network of virtually any size can be spontaneously constructed and self-organized. When a system operator is ready to perform manual configuration, he/she logs into the Server’s console port. From the Server he can use simple configuration commands to distribute security and other configuration parameters to the rest of the network via unicast or multicast.

Note that this scenario would operate most effectively if certain parameters are standardized; this effort is beyond the scope of this document. However, many variations of this scenario are possible once routers exist with both ARCP Client and ARCP/DHCP Server capabilities.

7. Message Formats and Options

This section describes the message and field formats and the extensions to related protocols necessary to ARCP operation. All ARCP messages are conveyed by TCP. Both clients and servers listen on TCP port XX, with message types distinguished by the Message Type field in the first eight bits of the message, as described below.

7.1 DHCP ARCP-Seed Option (DHCP Option XXX)

This DHCP Option is used to give ARCP clients an ARCP Seed, so that they will know which unique ARCP administrative domain to accept configuration from.

<table>
<thead>
<tr>
<th>Code</th>
<th>Len</th>
<th>Option</th>
<th>Lease</th>
<th>Seed</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXX</td>
<td>n</td>
<td>op</td>
<td>l1</td>
<td>s1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>l2</td>
<td>s2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>l3</td>
<td>s3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

The Option field is 0x00 for Insecure. Other values for this field are beyond the scope of this document.

The Lease field (4 octets) is the lease time in seconds, using the same format as a standard DHCP lease; 0xffffffff = infinity.

The Seed field is the ARCP Seed, up to 128 octets.

7.2 DHCP ARCP-Server Option (DHCP Option XXX)
This DHCP Option is used to tell an ARCP-enabled router which ARCP Server to register with in order to receive configuration. In domains with more than one ARCP Server, an IP Anycast address SHOULD be used in order to simplify client operation. Similarly, the DHCP Server SHOULD give out more than one ARCP-Server Option only when both an IPv4 and an IPv6 ARCP Server address are known.

<table>
<thead>
<tr>
<th>Code</th>
<th>Len</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXX</td>
<td>n</td>
<td>a1 a2 a3 a4 ...</td>
</tr>
</tbody>
</table>

The Len field is 4 for IPv4 and 16 for IPv6

The Address field is the IP address of the ARCP Server

7.3 ARCPREG Message (Client to Server)

The ARCPREG message is used by a Client to register with the ARCP Server. This message is normally only sent once, when the client initially loads and acquires an IP address and the address of the ARCP Server.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message Type | Flags | Client ID (6 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Client IP Address (4 or 16 octets) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Seed Length | ARCP Seed |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Auth Option | Auth Data |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Message Type is 0x01 for ARCPREG messages.

Flags:
- Bit 0: Address Version - 0=IPv4, 1=IPv6
- Bit 1: Authentication - 0=Not Included, 1=Included
- Bits 2-7: Must be Zero (Reserved for future use)

(Full descriptions TBD)

Client ID is as defined in the Glossary, above

Seed Length is the length in Octets of the ARCP Seed

ARCP Seed is as defined in the Glossary, above

Auth Option and Auth Data are only included in authenticated and/or encrypted ARCPREG messages, which are beyond the scope of this document.
7.4 ARCPCFG Message (Server to Client)

The ARCPCFG message is used by a Router or Host Client to request configuration parameters from the ARCP Server. The Options in this message are construed as new configuration orders rather than requests or current-status information.

The ARCPCFG Message may be sent from an ARCP Server to and ARCP Client at any time to convey new configuration information. This message MAY be sent either to individual Client IP addresses or to the Client MUST check the Source IP Address of the IP header of the message to confirm that it is from the ARCP Server with which the client is registered, and discard the message if it is not.

The Client MUST check that either:

a. The Client ID field matches zero and the Destination address in the message’s IP Header is the Client’s Multicast address, or
b. The Client ID field matches the Client’s own ID.
and MUST discard the message if it is neither.

The contents of an ARCPCFG message to a client should be considered an "order": the Client MUST implement all valid configuration data received that it is capable of implementing on any ARCP-enabled part of the router.

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message Type  | Flags         | Client ID (6 octets)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Seed Length   | ARCP Seed                                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| NumOptions    | ARCP Options...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Message Type is 0x02 for ARCPCFG messages

Flags:

Bit 0: Seed
0=ARCP Seed is not included in this message
1=ARCP Seed is included in this message

Bit 1: Acknowledge
0=Reply with an ARCPSTATUS message only if there is an error
1=Reply with an ARCPSTATUS message irrespective of error

Bit 2: Discard
0=Use all good data in the message if there is an error
1=Discard entire message if there is an error

Bits 3-7: Must be Zero (Reserved for future use)

Client ID is as defined in the Glossary, above.
Seed Length is the length in Octets of the ARCP Seed

ARCP Seed is as defined in the Glossary, above.

Note that the Seed Length and ARCP Seed fields are only present if the 'Seed' Flag is set to 1 as described above.

NumOptions is the number of ARCP Options included in this message.

ARCP Options convey the specific configuration data to be implemented by the Client. Options in ARCPREQ messages use the same format as those used in ARCPCFG and ARCPSTATUS messages, as defined below in the ARCP Options section.

7.5 ARCPREQ Message (Client to Server)

The ARCPREQ message is used by a Router or Host Client to request configuration parameters from the ARCP Server. The Options in this message are construed as requests rather than new configuration orders or current-status information.

| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 |
|---------------------------------------------------------------+
| Message Type | Flags         | Client ID (6 octets)          |
|---------------------------------------------------------------+
| | | |
| | Seed Length   | ARCP Seed      |
| +-------------------------------+-------------------------------+ |
| +-------------------------------+-------------------------------+ |
| NumOptions    | ARCP Options...   |
| | | | | | | | | | |

Message Type is 0x03 for ARCPREQ messages

Flags:

Bit 0: Seed

0=ARCP Seed is not included in this message
1=ARCP Seed is included in this message

Bits 1-7: MUST be zero (Reserved for future use)

Client ID is as defined in the Glossary, above.

Seed Length is the length in Octets of the ARCP Seed

ARCP Seed is as defined in the Glossary, above.

Note that the Seed Length and ARCP Seed fields are only present if the 'Seed' Flag is set to 1 as described above.

NumOptions is the number of ARCP Options included in this message.

ARCP Options in an ARCPREQ message convey the particular configuration data being requested by the Client. Options in ARCPREQ messages use the same format as those used in ARCPCFG and ARCPSTATUS messages, as defined below in the ARCP Options section.
ARCPREQ messages may use options without content (for example, the DNS Server Router Option with zero length as described in the Router Options section below) to request any matching configuration parameter (in this case, any DNS Server).

Alternately, the Client MAY include current, past, or otherwise desired specific configuration information in the ARCPREQ in order to request a particular configuration. Use of this method is left as an OPTIONAL behavior to the Client implementor; it SHOULD be supported by all ARCP Servers.

7.6 ARCPSTATUS Message (Client to Server)

The ARCPREQ message is used by a Router or Host Client to request configuration parameters from the ARCP Server. The Options in this message are construed as current-status information rather than new configuration orders or requests.

ARCPSTATUS messages are sent in reply to:

1. ARCPCFG messages with the Acknowledge Flag set to 1
2. ARPQUERY messages
3. Status changes on the router, such as interface state changes and manual configuration changes

| Message Type | Flags         | Client ID (6 octets)         |
|---------------+---------------+-----------------------------|
|               |               |                             |
| Seed Length   | ARCP Seed     |                             |
|               |               |                             |
| NumOptions    | ARCP Options. |                             |
|               |               |                             |

Message Type is 0x04 for ARCPSTATUS messages

Flags:

Bit 0: Seed
0=ARCP Seed is not included in this message
1=ARCP Seed is included in this message

Bits 1-7: MUST be zero (Reserved for future use)

Client ID is as defined in the Glossary, above.

Seed Length is the length in Octets of the ARCP Seed

ARCP Seed is as defined in the Glossary, above.

Note that the Seed Length and ARCP Seed fields are only present if the ‘Seed’ Flag is set to 1 as described above.

NumOptions is the number of ARCP Options included in this message.

ARCP Options in an ARCPSTATUS message convey the current
configuration data in use by, and general current status of, the
Client. Options in ARCPSTATUS messages use the same format as those
used in ARCPREQ and ARCPCfg messages, as defined below in the ARCP
Options section.

Note that the correct ARCPSTATUS message requires new triggered
events in a client router. ARCPSTATUS messages should be triggered
not only when an interface comes up or down, but also any time a
manual configuration change is made which alters the ARCP automated
configuration. This includes in particular any manual resetting or
erasure of the client router’s configuration.

7.7 ARCPQUERY Message (Server to Client)

The ARCPREQ message is used by a Router or Host Client to request
configuration parameters from the ARCP Server. The Options in this
message are construed as current-status information rather than new
configuration orders or requests.

ARCPSTATUS messages are sent in reply to:

1. ARCPCfg messages with the Acknowledge Flag set to 1
2. ARCPQUERY messages
3. Status changes on the router, such as interface state changes and
   manual configuration changes

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message Type  | Flags         | Client ID (6 octets)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Seed Length   | ARCP Seed
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| NumOptions   | ARCP Options...|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Message Type is 0x04 for ARCPSTATUS messages

Flags:
   Bit 0: Seed
      0=ARCP Seed is not included in this message
      1=ARCP Seed is included in this message

   Bits 1-7: MUST be zero (Reserved for future use)

Client ID is as defined in the Glossary, above.

Seed Length is the length in Octets of the ARCP Seed

ARCP Seed is as defined in the Glossary, above.

Note that the Seed Length and ARCP Seed fields are only present if
the ‘Seed’ Flag is set to 1 as described above.

NumOptions is the number of ARCP Options included in this message.
ARCP Options in an ARCPSTATUS message convey the current configuration data in use by the Client. Options in ARCPSTATUS messages use the same format as those used in ARCPREQ and ARCPCFG messages, as defined below in the ARCP Options section.

8. ARCP Options

ARCP Options are the basic building block used by ARCPCFG, ARCPREQ, ARCPQUERY, and ARCPSTATUS messages to request and convey all sorts of information about router configuration. They are similar in function to but generally more complex than DHCP Options. They take the following forms:

8.1 All ARCP Options

All ARCP Options begin with the following fields:

```
Con   Pri  Type
+-----+-----+-----+---
|  c  |  p  |  t  | ...|
+-----+-----+-----+---
```

Con is the "Context" or "Virtual Router" to which this option applies, if applicable. Routers which are not subdivided in this way SHOULD ignore this field.

Pri is an octet containing Priority Flags, as described below:

Bit 0: Override
0=Make the change only if this item is not already configured
1=Make the change whether or not the item is already configured

Bit 1: Additive
0=New data overwrites previous configuration
1=New data is added to previous configuration if possible

Bit 2: Immediate
0=Change should take effect when router or interface is reset
1=Change should take effect immediately

Bits 3-7: Must be Zero (Reserved for future use)

Note that these flags are not applicable to ARCPREQ/ARCPSTATUS Messages and all MUST be set to zero for such messages.

Type is the Option Type. Option types, the value of this field for each, and the format of the following octets for each, are described below:

8.2 Router Options

Router Options are used to define parameters which affect the entire Router (or context), such as hostname, DNS Server, login security parameters, and PIM RP Configuration.

```
Type  CCode Length
=========+-----+-----+---
...|  t  |  c  |  l  | ...
=========+-----+-----+---
```

Type is 0x00 for any Router Option
CCode is the Content Code of this Router Option.

Length(1 Octet), is the length in Octets of the remaining content of this Option. 0xFF in the Length field means that the content is continued in next Option.

The octets following the Length parameter which form the remainder of the Router Option vary based on the Content Code. This document gives the following preliminary but not exhaustive examples of content for well-known Router Option Content Codes:

0x00: ARCP Seed - New ARCP Seed
0x01: ARCP Server - New ARCP Server
0x02: Router Reset - Content is seconds before reboot, 4 Octets
0x03: ARCP Reset

Content is seconds before all ARCP configuration data is erased (4 Octets). Note that this code does not reset ARCP Seed or ARCP Server, only configuration data received from an ARCP Server. This can be accomplished by using the 0x00 and 0x01 codes with all three Priority flags set to zero.

0x04: Config Reset

Content is seconds before erase full config, 4 Octets.

Note that options 0x02 through 0x04 MUST only be used in ARCP_CFG messages with Priority Flag "Immediate" set to 1, and in ARCP_STATUS messages (with the content seconds counter set to zero) sent to notify a server when a some part of a router has been manually reset. Other combinations are in error and MUST be discarded.

0x05: Hostname - Content is the hostname in ascii, up to 64 Octets
0x06: ReadPass - Content is read-only password, up to 64 Octets
0x07: WritePass - Content is write-enable password, up to 64 Octets
0x08: DNS Server IPv4 - 4-Octet content is address of a DNS Server

Zero is used in the Length field for Router Options which have no specific content. For example, the Router Option field in an ARCPREQ message requesting a DNS Server address, and the corresponding ARCP_CFG in the ARCP_CFG reply, would look like the following:

(Octet values are given in three-place dotted decimal for brevity)

ARCPREQ Router Option to request DNS Server:
|000.000.000.008.000|

ARCP_CFG Router Option in reply for DNS Server 192.168.0.1:
|000.224.000.008.004.192.168.000.001|

In the reply, the 224 decimal value in the second octet represents the first three bits; Override, Additive and Immediate; with values of 1.

8.3 Interface Options

Interface Options are used to convey, request, and set interface-specific parameters.
<table>
<thead>
<tr>
<th>Type</th>
<th>Interface</th>
<th>Subinterface</th>
<th>Flags</th>
<th>CCode</th>
<th>Length</th>
</tr>
</thead>
</table>
| ...  | t         | i1 | i2 | i3 | s1 | s2 | s3 | f | c | l | ...

Type is 0x01 for any Interface Option

Interface is the identifier of an individual interface within the Router Context (see Glossary above). This field is treated as opaque with the following exceptions: 0xFFFFFFFF indicates that the configuration data applies to all interfaces within the router context.

Subinterface is the identifier of an individual subinterface within the Interface (see Glossary above). This field is treated as opaque with the following exceptions: 0x000000 indicates that the configuration data applies to the parent interface, not to any subinterfaces. 0xFFFFFFFF indicates that the configuration data applies to all subinterfaces within an interface, not to the interface itself.

Flags is an octet with flags bits that convey basic interface configuration information. The following are their meanings in the context of an ARCPSTATUS message:

Bit 0: Bcast
0=Broadcast is supported on this interface
1=Broadcast is not supported on this interface

Bit 1: SubIntOpt
0=Subinterfacing is optional on this interface
1=Subinterfacing not optional (it always is or always isn’t)

Bit 2: SubIntAct
0=Subinterfacing is not active on this interface
1=Subinterfacing is active on this interface

Bit 3: Enable
0=Interface is currently disabled
1=Interface is currently enabled

Bit 4: Connect
0=Interface is currently disconnected
1=Interface is (or may be) connected

Bit 5: Passive
0=Interface is not ARCP-Passive
1=Interface is ARCP-Passive

Bit 6: Auto
0=Interface is in automatic configuration mode
1=Interface has been manually configured

Bit 7: Must be Zero (Reserved for Future Use)

For ARCPSTATUS and ARCPREQ messages these flags convey the actual status of an interface. ARCPCFG messages can use some of these same flags to convey the status to which the interface should change. For example, bit 3 set to 0 in an ARCPCFG message tells the router client to disable an interface. This usage only
applies to bits 2, 3, 5, and 6; other bits in ARCPCFG messages SHOULD be set to their true current status if known and MUST be ignored by the receiver of the ARCPCFG message.

Note also that bits 0 and 1 are only used in ARCPSTATUS, since they convey inherent and unchangeable information about an interface. In other messages these bits SHOULD be set to their true current status if known and MUST be ignored by the receiver of the message.

CCode is the Interface Option Content Code which identifies the specific type of interface configuration information being requested or conveyed.

Length(1 Octet), is the length in Octets of the remaining content of this Option. 0xFF in the Length field means that the content is continued in next Option.

The octets following the Length parameter which form the remainder of the Interface Option vary based on the Content Code. This document gives the following preliminary but not exhaustive examples of content for well-known Interface Option Content Codes:

0x01: IP Address with mask in bits, 5 Octets total

<table>
<thead>
<tr>
<th>IP Address</th>
<th>MaskBits</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>i1</td>
</tr>
</tbody>
</table>

0x02: Interface Type-Specific Encapsulation, 1 Octet

Content is 0x00 for Interface-Type default encapsulation, which varies by interface type. Specific values of this field for each interface type will be described in a separate document.

8.4 Protocol Options

<table>
<thead>
<tr>
<th>Type</th>
<th>Proto</th>
<th>Instance</th>
<th>CCode</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Type is 0x02 for any Protocol Option

Protocol is a code designating the protocol to be configured. Initial values include:

0x00: Any routing protocol
0x01: RIP version 2
0x02: OSPF
0x03: IS-IS

Instance MUST be 0x0000 (Reserved for future use)

CCode is the Protocol Option Content Code which identifies the specific type of protocol configuration information being requested or conveyed.

Length(1 Octet), is the length in Octets of the remaining content of
this Option. 0xFF in the Length field means that the content is continued in next Option.

The octets following the Length parameter which form the remainder of the Protocol Option vary based on the Content Code. This document gives the following preliminary but not exhaustive examples of content for well-known Protocol Option Content Codes:

0x00: No Code - Zero length, no content. Used to specify just the protocol itself, without parameters.

0x01: RIP Interface - Configuration of RIP on interfaces

Length is 7 Octets

<table>
<thead>
<tr>
<th>Length</th>
<th>Interface</th>
<th>Subinterface</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x07</td>
<td>i1</td>
<td>i2</td>
<td>s1</td>
</tr>
<tr>
<td></td>
<td>i3</td>
<td>s2</td>
<td>s3</td>
</tr>
<tr>
<td></td>
<td>f</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Interface and Subinterface define the interface(s) or subinterface(s), as defined in the Interface Options section above, to which the protocol data applies.

Flag bits for RIP Interface:

Bit 0: Activate
0=Deactivate RIP on this (sub)interface
1=Activate RIP on this (sub)interface

Bit 1: Overwrite RIP
0=Ignore this data if RIP is configured on (sub)interface
1=Overwrite any previous RIP configuration on (sub)interface

Bit 2: Overwrite all
0=Ignore any other protocol on (sub)interface
1=Remove any previous protocol from (sub)interface

Bits 3-7: MUST be zero (Reserved for future use)

0x02: OSPF Interface - Configuration of OSPF on interfaces

Length is 11 Octets

<table>
<thead>
<tr>
<th>Length</th>
<th>Interface</th>
<th>Subinterface</th>
<th>Area</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0B</td>
<td>i1</td>
<td>i2</td>
<td>a1</td>
<td>a2</td>
</tr>
<tr>
<td></td>
<td>i3</td>
<td>s1</td>
<td>a3</td>
<td>a4</td>
</tr>
<tr>
<td></td>
<td>s2</td>
<td>s3</td>
<td>f</td>
<td></td>
</tr>
</tbody>
</table>

Interface and Subinterface define the interface(s) or subinterface(s), as defined in the Interface Options section above, to which the protocol data applies.

Area is the OSPF Area number (see Flag Bit 3 below for format)

Flag bits for OSPF Interface:
Bit 0: Activate
0=Deactivate OSPF on this (sub)interface
1=Activate OSPF on this (sub)interface

Bit 1: Overwrite OSPF
0=Ignore this data if OSPF is configured on (sub)interface
1=Overwrite any previous OSPF config. on (sub)interface

Bit 2: Overwrite all
0=Ignore any other protocol on (sub)interface
1=Remove any previous protocol from (sub)interface

Bit 3: Area Format
0=Area is undotted 32-bit integer (Big-endian order)
1=Area is four-place dotted integer

Bit 4: Stub
0=Not a Stub Area
1=Stub Area

Bit 5: NSSA
0=Not an NSSA (Not-so-stubby Area)
1=NSSA

Bits 6-7: MUST be zero (Reserved for future use)

0x03: IS-IS Interface

Length is 7 Octets

<table>
<thead>
<tr>
<th>Length</th>
<th>Interface</th>
<th>Subinterface</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>i1 i2 i3</td>
<td>s1 s2 s3</td>
<td>f</td>
</tr>
</tbody>
</table>

Interface and Subinterface define the interface(s) or subinterface(s), as defined in the Interface Options section above, to which the protocol data applies.

Flag bits for IS-IS Interface:

Bit 0: Activate
0=Deactivate IS-IS on this (sub)interface
1=Activate IS-IS on this (sub)interface

Bit 1: Overwrite IS-IS
0=Ignore this data if IS-IS is configured on (sub)interface
1=Overwrite any previous IS-IS config. on (sub)interface

Bit 2: Overwrite all
0=Ignore any other protocol on (sub)interface
1=Remove any previous protocol from (sub)interface

Bit 3: Level-1
0=Level-1 adjacency not enabled on (sub)interface
1=Level-1 adjacency enabled on (sub)interface

Bit 4: Level-2
0=Level-2 adjacency not enabled on (sub)interface
1=Level-2 adjacency enabled on (sub)interface
Bits 5-7: MUST be zero (Reserved for future use)

0x04: IS-IS Net

Length is 8 to 20 Octets

Length

<table>
<thead>
<tr>
<th>l</th>
<th>a1</th>
<th>a2</th>
<th>a3</th>
<th>a4</th>
<th>...</th>
<th>an</th>
</tr>
</thead>
</table>

IS-IS System ID               N-Sel

| s1 | s2 | s3 | s4 | s5 | s6 | 0x00 |

0x05: Passive Interface

Length is 6 Octets. This Content Code indicates that the Protocol indicated by the Proto field for this Protocol Option should be set to Passive for the indicated interface(s) or subinterface(s). Note that a Proto field value of 0x00 means all protocols should be passive.

Length  Interface         Subinterface

| 0x06 | i1 | i2 | i3 | s1 | s2 | s3 |

Interface and Subinterface define the interface(s) or subinterface(s), as defined in the Interface Options section above, to which the protocol data applies.

Passive in this context means that routing protocol updates MUST NOT be sent out from this interface, though the interface MAY listen for incoming protocol data depending on the router’s capability to do so. Routers which do not have this sort of "passive" interface capability SHOULD deactivate the indicated protocol on the indicated interface.

TBD: Metrics, Designated Router Priorities, and Filtering. Possibly also IS-IS level control, authentication.

Note that some combinations of Protocol Option fields are nonsensical and MUST not be used. For example, a Protocol Option with the Proto field set to 0x01 for RIP version 2 cannot be used with a Content Code of 0x02 for configuring an OSPF Interface. Such mismatched messages MUST be ignored by any receiving station. It is permissible, however, to use protocol-specific Content Codes with the 0x00 value Proto field, indicating ‘Any protocol’.

8.5 Vendor-Specific Options

<table>
<thead>
<tr>
<th>Type</th>
<th>Vendor</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>v1</td>
<td>v2</td>
</tr>
</tbody>
</table>

Type is 0xFE (decimal 254) for all Vendor-Specific Options

Vendor is a two-octet field designating a vendor. A list of these codes is outside the scope of this document and will need to be separately managed.

Length (1 Octet), is the length in Octets of the remaining content of this Option. 0xFF in the Length field means that the content is continued in the following Option in this message.

The remaining content of this option is an opaque field to be specified by the individual implementers. Message Sequence Example

<table>
<thead>
<tr>
<th>DHCP Server</th>
<th>Client</th>
<th>ARCP Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>Begins initialization</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard DHCP Procedure.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\ DHCPDISCOVER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\ DHCPoffer \ Client</td>
<td>acquires IP host config, ARCP Seed and Server Address</td>
</tr>
<tr>
<td></td>
<td>\ DHCPREQUEST</td>
<td>\ DHCPACK</td>
</tr>
<tr>
<td></td>
<td>\ DHCPRELEASE</td>
<td>\ ARCPCFG Send top-level config parameters</td>
</tr>
<tr>
<td>Request Initial -&gt; Config</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARCPREQ detailed router config</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\ ARCPCFG</td>
<td></td>
</tr>
<tr>
<td>Interface Event -&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARCPREQ Interface-specific config</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\ ARCPCFG</td>
<td>Reset or update Needed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARCPQUERY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARCPSTATUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
</tbody>
</table>

Timeline diagram of ARCP client / server messages
9. Security Considerations

ARCP relies on pre-existing mechanisms to enable operation at the appropriate level of security as determined by the system operator. These mechanisms include Secure DHCP for initial Host-mode configuration, followed by IPSec for client-server messaging. Basic operation also provides a limited form of inherent security through the Seed-checking mechanism. Specific details of highly-secure ARCP operation will be addressed in a separate document.

10. IANA Considerations

Consistent and correct ARCP operation necessitates several lists to be maintained by IANA. These include:

- DHCP Options (Addition of two to the current list)
- ARCP Options (New list to maintain, four items initially)
- ARCP Content Codes (One list per Option Type other than Vendor-Specific)
- ARCP Vendor Codes

11. Open Issues

Editorial: Several sections need cleaning up. In general more textual explanation needed. Also need to formalize message formatting.

Accompanying Documents: Need to formulate drafts on Secure ARCP, ARCP Options and Content Codes, and Relay Agent Extensions for Non-Broadcast Media (at least)

Peer Discovery: Need to clarify/validate the current mechanism and determine a message format.

DHCP Interaction: Clarify/validate mechanism for an ARCP router requesting and receiving a host address for a Point-to-Point interface. Consider using the RFC 3046 or RFC 3011 [7] options for this.

12. References

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


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


13. Acknowledgments

TBD

14. Author’s Addresses

Jeb R. Linton
8235 Depot Place, Manassas, VA 20112
Email: jrlinton@ieee.org