**Cache Array Routing Protocol**

With the Cache Array Routing Protocol (CARP), an [*array*](http://msdn.microsoft.com/en-us/library/ms812582.aspx) containing multiple Internet Security & Acceleration (ISA) Server computers can act as a single logical [*cache*](http://msdn.microsoft.com/en-us/library/ms812607.aspx). CARP allows an array to efficiently balance Web-based client load and distribute cached content among [*array members*](http://msdn.microsoft.com/en-us/library/ms812582.aspx). CARP provides client computers with the information and algorithms required to identify the best server in the array to serve their request, eliminating the need for array members to forward requests between them. CARP also supports array member selection by the servers and by chained [*proxy servers*](http://msdn.microsoft.com/en-us/library/aa503258.aspx).

CARP uses [*hash*](http://msdn.microsoft.com/en-us/library/ms826764.aspx)-based routing to determine the best path for resolving a request within an array. The request resolution path is based upon hashes of the array member identities and the host name in the [*Uniform Resource Locator (URL)*](http://msdn.microsoft.com/en-us/library/aa503419.aspx) specified in the request. For any given host name from a URL, a Web browser can determine exactly where in the array the information is stored if the information has already been cached from a previous request, and array members can determine where the information is to be cached for future requests.

CARP provides powerful benefits:

* Because CARP determines the best request resolution path, there is no query messaging between proxy servers in an array, as is found with conventional Internet Cache Protocol (ICP) networks. By doing this, CARP avoids the heavier query congestion that normally occurs with a greater number of servers.
* CARP eliminates the duplication of content that otherwise occurs on an array of proxy servers. With an ICP network, an array of five proxy servers can rapidly evolve into duplicate caches of the most frequently requested objects. The hash-based routing of CARP keeps this from happening by allowing all five ISA Server computers to exist as a single logical cache. The result is a faster response to queries and a far more efficient use of server resources.
* CARP has positive scalability. Due to its hash-based routing and its resultant independence from peer-to-peer pinging, CARP becomes faster and more efficient as more proxy servers are added. ICP arrays must conduct queries to determine the location of cached information. This is an inefficient process that generates extraneous network traffic. ICP arrays have *negative scalability*: the more servers added to the array, the more querying required between servers to determine location.
* CARP automatically adjusts to additions or deletions of servers in the array. The hash-based routing means that, when a server is either taken offline or added, only minimal reassignment of caches for specific URLs is required.
* CARP ensures that the cached objects are either distributed evenly between all servers in the array or by the load factor that you configure for each server.

### How CARP works

The CARP process provides efficient routing for requests.

1. All servers are tracked through an array membership list, which is maintained in Active Directory. Array members are notified when servers are added or removed from the array.
2. Periodically, the Web Proxy client or a downstream server polls and, if necessary, updates the array membership list.
	* Web Proxy clients send a **array.dll?get.routing.script** request to the member server.
	* Downstream servers send a **array.dll?get.info.v1** request to the member server.
3. When requesting an object, the client or downstream server uses the membership list, together with a hash function it computes for the name of each requested URL, to determine which server should service the request.
4. The hash value of the URL is combined with the hash value for each ISA Server. The URL+ISA Server hash that comes up with the highest value becomes "owner" of the information cache.
5. The server checks if it should handle the request. If not, then it sends the request to another member server, specifying its intra-array IP address.

The result is a location for all cached information, meaning that the Web browser or downstream server can know exactly where a requested URL is either already stored locally or will be located after caching. Because the hash functions used to assign values are so numerous, the load is statistically distributed and balanced across the array.

The request resolution path that CARP provides means that there is no need to maintain massive location tables for cached information. The browser simply runs the same math function across an object to determine where it is.

Because ISA Server computers in an array may have different hardware and may be more powerful than others, you may want to divide the cache load differently. For this reason, you can configure the CARP functions, specifying the load factor for any given server in the array.

Furthermore, you can configure CARP differently for incoming and outgoing Web requests. For example, CARP can be enabled for all outgoing Web requests, and disabled for all incoming Web requests.