Internet Draft B. Cain
Cereva Networks
O. Spatscheck
AT&T Labs
M. May
Activia Networks
A. Barbir
Nortel Networks
November 2001

Request-Routing Requirements for Content Internetworking
draft-cain-request-routing-req-03.txt

Status of this Memo

Copyright Notice

Abstract

Request-routing systems (RRS) are components of Content Distribution Networks (CDNs) that direct client requests to an available copy of content based on one or more metrics. To enable the interconnection of CDNs [MODEL][ARCH], it is necessary for their request-routing systems to interconnect and exchange information such that client requests can be routed between CDNs. This document specifies the requirements for request-routing internetworking.

1. Introduction

Request-routing systems (RRS) are components of Content Distribution Networks (CDNs) that direct client requests to an available copy of content based on one or more metrics. To enable the interconnection of CDNs [MODEL][ARCH], it is necessary for their request-routing systems to interconnect and exchange information such that client requests can be routed between CDNs. This is called request-routing internetworking. This document specifies the requirements for request-routing internetworking.

1.1 Document Organization

This document is organized as follows. Section 1 presents an introduction to request-routing systems. Section 2 presents the details of request-routing system components and protocols. Section 3 presents detailed requirements for each component, sub-component or protocol from sections 1 and 2.

1.2 Overview of Request-Routing Systems

Request-routing systems (RRS) are components of content networks (CN) that direct client requests to surrogates that can "best" service the request [KNOWN_MECH]. Request-routing decisions are based on a set of metrics that may include for example network proximity and server load. The basic functionality of a request-routing system can be summarized by the following:

1. It directs clients to surrogates that are able to service their requests.

2. It directs clients to surrogates that (per a set of metrics) are able to provide the "best" service.

A given client request may not necessarily cause a full redirection but may use cached information to fulfull the request (e.g. DNS-based request- routing systems). Nonetheless, we use the term "client request" within this document to refer (mostly) to a request not fulfilled from an intermediate cache.

For the sake of clarity, we now reiterate several important assumptions from [ARCH] [MODEL]:

1. Each content network is a "black box" to other networks to which it is interconnected. We use the term "neighbor CN" to refer to a directly interconnected content network.

2. Content is served by surrogates that act on behalf of an origin server that holds the "master" or "authoritative" copy of content. Surrogates are part of a distribution system.

3. A request-routing system is responsible for directing/servicing requests for one or more distribution systems.

4. Each distribution system may have its own internal (or intra-CN) request-routing system that is not exposed to other interconnected networks.

5. Request-routing systems interconnect through content internetworking gateways (CIG) that implement standards based interconnection protocols. A CN's CIG is the only "visible" element to other interconnected CNs.

1.3 Generic Request-Routing System Architecture

This section presents a generic architecture of a request-routing system to assist in understanding request-routing systems as well as the requirements for their interconnection. In Figure 1, a conceptual view of a request-routing system is presented; it consists of the following components: Content Topology Exchange, Content Topology Database and Route Computation. A brief summary of these components is provided below:

```
      ____________________________________________
     |  ___________    ________     ________     |
     | |Routing    |  |Content |   |Advert- |    |   Request-Routing
     | |Computation|<-|Topology|<->| ment    |    |<->Information
     | |___________|  |Database|   |Exchange|    |   Exchange
     |                |________|   |________|    |   Protocol
     |___________________________________________|
```

Figure 1.

1. Routing Computation: The computation of the best surrogate for a given set of clients based on information stored in the Content Topology DataBase, route computing algorithm, and configured policies.

2. Content Topology Database: The topology database includes detailed advertisement information received from CN neighbors and the associated metrics that are included.

3. Advertisement Exchange: This functional block is responsible for implementing the Request-Routing Information Exchange protocol.

4. Request-Routing Information Exchange Protocol: The actual protocol used to exchange sets of content advertisements and area advertisements [MODEL].

1.4 Interconnecting Request-Routing Systems

Within a single CN, a request-routing system is used to direct client requests to surrogates that are part of its own distribution system.

However, when request-routing systems are interconnected, a request-router has the ability to redirect client requests to neighbor CNs. That is, when neighbor CN can "better" serve a set of clients, it may be desirable to direct requests to that neighbor CN. In order to determine which CN may best serve a client request, one or more protocols may be required to exchange various types of information and associated metrics.

This document describes the components of request-routing systems and requirements for interconnecting them.

2. Overview of Request-Routing System Components and Protocols

This section provides a detailed description of the basic components of a request-routing system. Section 3 provides a description of the specific requirements for each component.

2.1 Request-Routing System Types

The methods in which a client request is directed may be different depending on the architecture of the request-routing system. Currently, there are two well-known types of request-routing systems [KNOWN_MECH]. These two types are described below:

1. DNS-based Request-Routing Systems: The Domain Name System (DNS) is used for the direction of client requests. In this approach, one or more domain names are assigned to the request-routing system; these names are then used as part of a URI reference to direct client requests [DNSMAP]. The limitations of DNS-based systems are described [KNOWN_MECH] and in section 2.1.1.

2. "In-Line" Request-Routing Systems: These request-routing systems are "in-line" to client requests. Examples of in-line request-routing systems are those that may be implemented within a proxy or a layer-7 router. In-line request-routing systems have full visibility into content requests (e.g. full URL) as well as visibility of the client's IP address [note: this isn't always true if transparent proxies are in place].

The distinction between these request-routing system types is important because of the differences in:

- The view of the content identifier (partial vs. whole).

- The view of the client (e.g. client's IP vs. client's local DNS).

- The implementation requirements of the two types (e.g. DNS caching).

2.1.1 DNS-Based Request-Routing Systems

In DNS-based request-routing systems [ARCH] [DNSMAP], only aggregate sets of content may be "directed" because a domain name (e.g. images.blah.com) can only (reasonably) represent a larger set of content. A DNS-based request-routing system works well in scenarios where many surrogates share large sets of content.

DNS-based request-routing systems suffer from the following limitations:

- The request-routing system knows only the domain name of the requested content. This precludes the RRS from knowing the full content path (e.g. URI) and the content type (e.g. HTTP, RTSP).

- The request-routing system knows the client's local DNS server, not the client itself.

- The request-routing system responses may be cached in DNS servers. The result is that a client request may not be individually directed by the request-routing system.

2.1.1.1 DNS Example

Content network CN-A is authoritative for http://images.blah.com (or CNAMEs are used to ultimately force a resolution of this name to CN A). Assume that DNS-based request-router R is part of CN-A and is also a CIG for CN-A. When R receives a client DNS request for images.blah.com, it makes a request-routing decision. This decision may be to direct the request to its own surrogates or to direct the request to another CN. This decision is based on the routing computation by CIG-A that in turn is based on "area" and/or "content" advertisements [MODEL] received from neighbors. For example, CIG-A can make a request-routing decision based on the following:

1. Information contained in area advertisements that have been received from interconnected CNs. An example may be an IP prefix advertised with an associated metric.

2. The ability of interconnected CNs to support the (content) type of the request.

3. Information contained in content advertisements that may include: content metrics, availability of content, etc. With DNS-based request-routing systems, content specific information is only relevant to the DNS name (e.g. in a URI).

4. Local request-routing policy.

If the choice is made to direct the request to another CN, the appropriate CNAME is used to direct the client's DNS to the chosen neighbor CN. The process then continues.

2.1.2 "In-Line" Request-Routing Systems

A Layer-7 router or Proxy situated close to a client may be used as an "in-line" request-routing system. Such a RRS is capable of directing client requests based on individual full content requests. This is possible because layer-7 information (e.g. HTTP headers) is exposed to the layer-7 router or proxy. In this type of RRS, a surrogate can be chosen based on, for example, a full URL. Another example of in-line request-routing is when an origin server (or reverse proxy) performs a layer-7 redirection by "URL-rewriting".

There are three major differences between an "in-line" request-routing system and a DNS-based request-routing system. The first is that the full content request is exposed (e.g. a full URL). The second is that the content type of the request is exposed (again from the full URL). The third is that all client requests can be received by the request-routing system; this is in contrast to DNS-based systems where caching may prevent this.

2.1.2.1 "In-Line" Example

Assume client X is configured to forward its requests to layer-7 request-router R. Furthermore assume that request-router R is a CIG for content network CN-A. When a request from client X is received, request-router R makes a request-routing decision based on its content topology database constructed from information communicated from other neighbor CNs. If request-router R can service the request within its own distribution system then the request is sent to a surrogate that is part of CN-A. If request router R decides to direct the client to another neighbor CN, a redirect is sent to the client to direct the cilent to another layer-7 request-router in a neighboring CN. In summary, when "in-line" request routing is used, the redirection decision is based on the following:

1. Information contained in area advertisements that have been received from neighbor CNs. An example may be an IP prefix advertised with an associated metric.

2. The ability of neighbor CNs to support the content type of the request. An example may be a set of content types supported.

3. Information contained in content advertisements from neighbor CNs that may include: content metrics, availability of content, etc. For "in-line" request-routing systems this may include full URLs or URL sets.

4. Local request-routing policy.

2.2 Request-Routing Interconnection Model

Request-routing systems (RRS) present a "black-box" view of their associated distribution systems. Since in such an environment no CN possesses a global view of all other CNs, the request-routing system must also rely on a peer-to-peer model in which each request-routing system is only aware of its direct neighbor. [Note: A direct neighbor of the request-routing systems does not have to be a direct neighbor at Layer-3].

There are two methods for redirecting a request between two interconnected request-routing systems. The first method is an iterative method where a RRS directs the request to the next-best (neighbor) RRS. This continues until a surrogate is finally selected. The second method is recursive where a RRS directs a request to the next-best RRS but expects an answer to return to the client. These two methods are analogous to recursive vs. iterative DNS lookups.

An example of how requests can be directed between CNs is through the use of DNS CNAMEs. When DNS-based request-routing systems are interconnected and redirecting requests using CNAMEs, a clients DNS resolution is redirected using a DNS CNAME record to another DNS-based request-routing system until a surrogate is found that is appropriate (according to a set of metrics) to serve the content. The drawbacks of CNAME based request-routing are discussed in [KNOWN MECH].

2.3 CN Capabilities

Request-routing systems are associated with one or more distribution systems. When a request-routing system directs a client request it must ensure that:

1. The client request type can be serviced by the distribution system (e.g. HTTP vs. RTSP).

2. The distribution system to which a client is directed has the capacity to service the request.

In order to ensure that an interconnected (neighbor) CN can service a request, a request-routing system is required to have the following information about neighbor CNs:

1. Request-routing system types.

2. Content types that can be served by the CN.

3. Sets of metrics that are used for direction.

This information maybe obtained manually (off-line) or through the use of dynamic (on-line) information exchange protocols.

2.4 Request-Routing Information Exchange

Interconnected request-routing systems need to exchange information in order to make request-routing decisions. The two request-routing system types presented in section 1.2 have slightly different requirements with respect to the types of information exchanged. In summary, interconnected request-routing systems need to exchange two basic types of information:

1. Area Advertisements: Advertisements from a CN's request-routing system about aspects of topology, geography and performance of a CN.

2. Content Advertisements: Advertisements from a CN's request-routing system about the availability of one or more collections of content on CN. This may include for example: urls, content types, distribution model, authoritative request-routing system, etc.

Request-routing information exchange follows the model of layer-3 routing protocols. That is, advertisements are sent to neighbor CNs and each request-routing system makes its own decisions. The design of an information exchange protocol must take the following into consideration:

- Information exchange may occur over highly unreliable networks.

- Information exchange protocols may be required to exchange large sets of advertisement information.

- Information exchange may occur over insecure networks.

- Arbitrary meshed topologies may exist for information exchange protocols.

2.5 Request-Routing Decision

Request-routing systems make decisions based on one or more advertisement types and their associated metrics. Both content advertisements and area advertisements may be used to construct a request-routing content topology database. This table is used to determine how requests should be directed. The request-routing decision process is complex for the following reasons:

- Content delivery networks are overlay networks which inherently makes decision processes more complex.

- There are many possible metrics; if multiple metrics are exchanged, loop prevention may be difficult.

- Request-routing systems may have specific policies with respect to direction.

- Request-routing decisions are independent; therefore request-routing loops must be prevented.

2.6 Request-Routing Protocol Design

In order to interconnect request-routing systems, one or more protocols are required to exchange request-routing information. These protocols are designed to operate in an inter-domain context and therefore have the following considerations:

- Protocol sessions will need to be debugged across CN boundaries.

- Large sets of information may be exchanged between CNs.

- Policy based request-routing is needed in many scenarios.

- Protocol designs should be "Internet" scalable.

3. Request-Routing System and Protocol Requirements

3.1 General Requirements

In the following section we describe the general requirements for protocols to be used in the interconnection of request-routing systems.

- Request-routing protocols MUST use an administrative identity to identify themselves in protocol exchanges.

- Request-routing protocols SHOULD support arbitrary direction topologies; this means "peer-to-peer" design.

- Request-routing protocols MUST treat other content networks as "black boxes"; that is, a given CN A does not normally posses direct visibility into another neighbor CN B.

- Request-routing protocols MUST support methods to determine the authoritative request-routing system for content.

- Request-routing protocols SHOULD be compatible with existing applications and protocols.

3.2 Request-Routing System Type Requirements

The following section describes the information exchange protocol requirements that apply to both DNS-based and in-line request-routing systems.

- Request-routing protocols SHOULD support DNS-based and in-line request-routing system types.

- Request-routing protocols MUST be extensible to support other request routing system types.

- Request-routing protocols MUST communicate their request-routing system type to neighbors (e.g. DNS-based).

- Request-routing protocols MAY allow for utilization of more than one request-routing system type for content.

- Request-routing protocols MUST be able to identify content types for content.

3.2.1 DNS-Based Request-Routing Requirements

The following section describes the information exchange protocol requirements that apply to DNS-based request-routing system types.

- Request-routing protocols MUST support CNAME based DNS redirection.

- Request-routing protocols MUST be able to map content types to CNAMEs in order to make proper direction decisions.

3.2.2 In-Line Based Request-Routing Requirements

The following section describes the information exchange protocol requirements that apply to in-line based request-routing system types.

- Request-routing protocols MUST support application layer redirection (e.g. HTTP redirection).

- Request-routing protocols SHOULD support explicitly configured application gateways and proxies.

3.3 Request-Routing Interconnection Model Requirements

The following section describes the information exchange protocol requirements for the model of CN interconnection.

- Request-routing protocols MUST allow for delegation of requests to another request-routing system.

- Request-routing protocols SHOULD support both iterative and recursive redirection models.

- Request-routing protocols SHOULD require that content have only one authoritative request-routing system.

- Request-routing protocols MUST verify that neighbor CNs have the ability to deliver content before directing requests to that neighbor.

3.4 Request-Routing System Capabilities Requirements

The following section describes the information exchange protocol requirements for request-routing system capability information.

- Request-routing protocols MUST support the advertisement of content type information between neighbors.

- Request-routing protocols SHOULD have primitive methods for capability advertisement.

3.5 Request-Routing Information Exchange Requirements

3.5.1 General Information Exchange Requirements

The following section describes the information exchange protocol requirements with respect to general types of information exchanged.

- Request-routing protocols MUST define standardized methods for identifying an atomic unit of content.

- Request-routing protocols MUST define standardized methods for identifying distribution system capabilities (e.g. content types, layer-3 coverage, etc).

- Request-routing protocol MUST not preclude request-routing systems from implementing policy based routing decisions.

- Request-routing protocols MUST support the exchange of multiple basic information types (e.g. area and content advertisements).

- Request-routing protocols MUST be able to associate multiple (and optional) metrics with each basic information types.

- Request-routing protocols MUST exchange information sufficient to avoid looping of information advertisements.

- Request-routing protocols MAY exchange information sufficient to prevent request-routing loops.

3.5.2 Specific Information Exchange Requirements

The following section describes the information exchange protocol requirements with respect to specific types of information exchanged.

- Request-routing protocols MUST support the exchange of area advertisements (e.g. IP prefixes) between request-routing systems.

- Request-routing protocol area advertisements MUST support the inclusion of multiple capabilities and metrics (e.g. X Mbps, Y CIDR blocks, Z static http).

- Request-routing protocols SHOULD define a minimum set of metrics for area advertisements.

- Request-routing protocols MUST support the exchange of content advertisements (e.g. URIs) between request-routing systems.

- Request-routing protocol content advertisements MUST support the inclusion of multiple metrics.

- Request-routing protocol content advertisements MUST support the ability to advertise the availability of content.

- Request-routing protocol content advertisements SHOULD identify the authoritative request-routing system.

- Request-routing protocols SHOULD define a minimum set of metrics for content advertisements.

- Request-routing protocols MUST accommodate hierarchy and aggregation in content and area advertisements.

3.6 Request-Routing Decision and Policy Requirements

The following section describes the information exchange protocol requirements with respect to request-routing decision making.

- Request-routing protocols MUST be "policy friendly" (e.g. support additional neighbor-to-neighbor extensible attributes).

- Request-routing protocols SHOULD support exchange of information sufficient to prevent routing loops.

- Request-routing protocols MAY support multiple metrics for direction decisions as long as routing decisions can be guaranteed loop free.

3.7 Request-Routing Information Exchange Protocol Attribute Requirements

The following section describes the information exchange protocol requirements with respect to the specific attributes of the protocol design itself. Note that some of these requirements are redundant with other sections; we repeat them here for organization.

- Request-routing protocols MUST use a reliable transport protocol.

- Request-routing protocols MUST make use of existing IETF developed security mechanisms for encryption and authentication.

- Request-routing protocols MUST include protocol notifications for protocol error conditions.

- Request-routing protocols SHOULD be connection oriented.

- Request-routing protocols MUST provide mechanisms to prevent looping of advertisement information.

- Request-routing protocols MUST have extensible packet formats.

- Request-routing protocols MUST properly identify neighbors.

- Request-routing protocols MUST properly authenticate neighbors.

- Request-routing protocols MUST scale to accommodate the exchange of large sets of content and area advertisements.

- Request-routing protocols MUST support (at a minimum) a simple capability exchange/advertisement.

- Request-routing protocols MUST NOT exchange policy information.

- Request-routing protocols MUST accommodate policy based request-routing systems.

4. Security Considerations

TBD

5. References

[MODEL] Day, M., Cain, B., Tomlinson, G., P. Rzewski "A Model for Content Internetworking (CIG)", draft-day-cdnp-model-08.txt (work in progress), October 2001.

[KNOWN MECH] Barbir, A., Cain, B., Douglis, F., Green, M., Hofmann, M., Nair, R., Potter, D. and O. Spatscheck, "Known CDN Request-Routing Mechanisms", draft-cain-cdnp-known-req-route-02.txt (work in progress), June 2001.

[ARCH] Green, M., Cain, B., Tomlinson, G., Thomas, S. and P. Rzewski, "Content Internetworking Architectural Overview", draft-green-cdnp-gen-arch-03.txt (work in progress), March 2001.

[DNSMAP] Deleuze, C., Gautier, L., and M. Hallgren, "A DNS Based Mapping Peering System for Peering CDNs", draft-deleuze-cdnp-dnsmap-peer-00.txt (work in progress), November 2000.

6. **Author's** Address:

Brad Cain

Cereva Networks
bcain@cereva.com

Oliver Spatscheck

AT&T Labs
spatsch@research.att.com

Martin May
Activia Networks
Martin.May@activia.net

Abbie Barbir
Nortel Networks
abbieb@nortelnetworks.com

7. Acknowledgements

Thanks to the following people for their contributions: John Martin, Nalin Mistry, Mark Day, Stephen Thomas, Hillary Orman, Phil Rzewski, and Fred Douglis.

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

Full Copyright Statement

Acknowledgement

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