Request-Routing Trends and Techniques in Content Distribution Network

Md. Humayun Kabir, Eric G. Manning, Gholamali C. Shoja
Parallel, Networking, Distributed Applications (PANDA) Laboratory
Department of Computer Science, University of Victoria
PO Box 3055, Victoria BC, V8W 3P6, Canada
e-mails: hkabir@csc.uvic.ca, Eric.Manning@engr.uvic.ca, gshoja@csc.uvic.ca

Abstract: *Content Distribution Networks (CDNs) have been developed and evolved to meet the growing commercial need for efficient and secure delivery of content over the Internet. Several proprietary CDN systems are now operating successfully, fulfilling the commercial demands to a certain extent. The protocols and algorithms used in these systems are mostly of proprietary nature and not standardized. Furthermore, the effectiveness of these protocols and algorithms has not yet been subject to analysis and exhaustive performance testing. However, the IETF Content Distribution Internetworking Group (CDN-WG) has begun the process of specifying the requirements and standardizing the protocols for CDNs. This paper presents an overview of Content Distribution Networks and then describes and analyzes different request routing techniques that have been developed by researchers so far to date.*

Keywords: *Content, Content Provider, Content Distribution, Content Distribution Network, Request-routing, Surrogate Server, Origin Server, Domain Name Server, Internet, Network Address Translation.*

1 Introduction

A CDN [11][12] is a value-added network, built on top of existing Internet infrastructure to offer new capability to process traffic flows based on any digital content including various forms of streaming media, route requests to the best server, and deploy content dynamically in one or more servers. A CDN has multiple replicas, or point of presence of content, in different locations that are geographically far apart from the origin server and from each other, but closer to the clients. A CDN directs the client's request to a *good replica*, which in turn serves the items on behalf of the origin server. A good replica means that the item is served to the client quickly, compared to the time it would take to serve the same item from the origin server, with essential quality, integrity and consistency. A network service provider can build and operate a CDN to offer content distribution service to a number of content providers. This helps content providers to outsource the network infrastructure and to focus their resources on developing high-value content, not on managing the network. For example, Akami, Sandpipper/Digital Island, and Adero are content-distribution service providers that provide content publishers such as CNN, Disney, AOL, Viacom and content aggregators such as Broadcast.com and Spinner , with the means to deliver content distribution and delivery. Though there are many proprietary CDNs operating, little work has been done to standardize their protocols and algorithms. In this paper, we are going to consolidate individual and isolated research works in the CDN request-routing problem domain. The rest of this paper is organized as follows: section 2 outlines the CDN and its peering systems, section 3 describes the research done so far to solve CDN request-routing problems and section 4 concludes the paper.

2 Content Distribution Network

2.1 CDN Overview

A CDN maintains a large number of replicas or surrogate servers in proximity to the end users, to act on behalf of the origin severs owned by different content providers. The CDN removes the delivery of content from a centralized site to multiple and highly distributed sites and overcomes the issues of network size, congestion, and failures. The CDN establishes business relationships with the content providers, to act on behalf of them. A typical CDN consists of several surrogate servers, a distribution system, a request-routing system, and an accounting system. Surrogate servers are the delivery servers other than the origin server. Figure 1 shows the schematic diagram of a CDN.

A surrogate server receives a mapped request and delivers the corresponding content to the client. A *Distribution system* consists of a collection of network elements called content-distributor. It supports the activity of moving a publisher's content from the origin server to one or more surrogate servers. Distribution can happen when a surrogate server either anticipates or receives a client request , or in response to a surrogate server receiving a client request. The former is called *push* and the later is called *pull*. The Distribution system also propagates *content signals*. Content signals specify information such as validation and expiration about the content. The CDN uses these content signals to maintain the integrity and consistency of the content in its surrogate servers. The Distribution system interacts with the request-routing system to inform the content availability in different surrogate servers. It also interacts with the accounting system to inform the content distribution activity so that the later can measure the volume of content distribution.

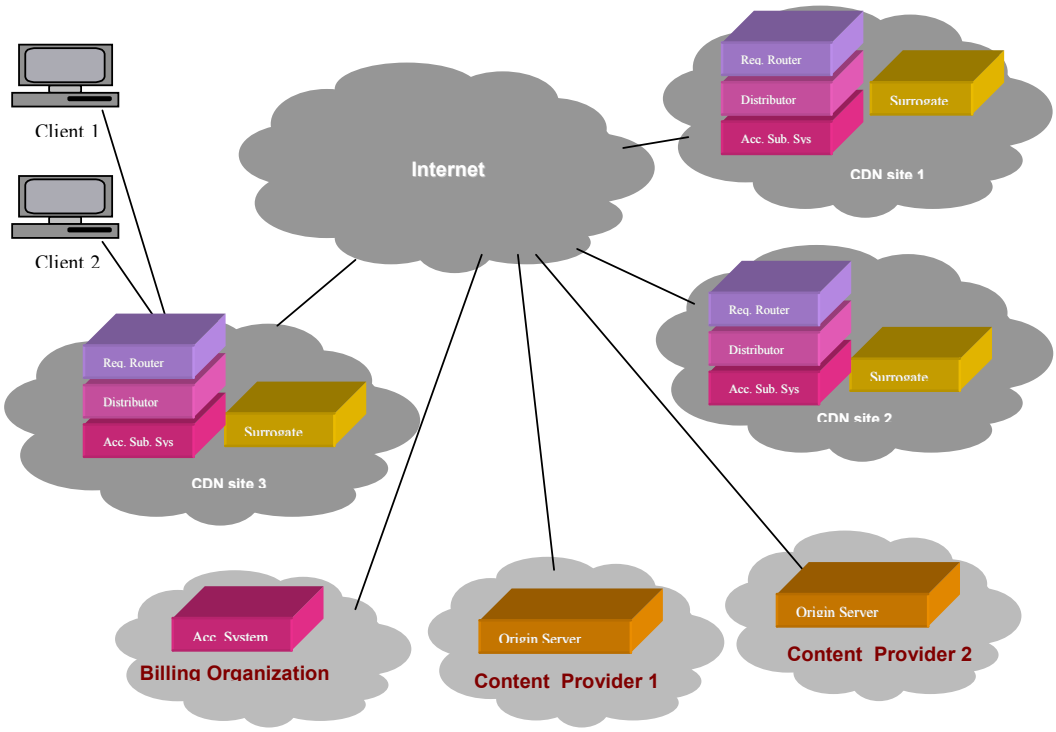

Figure 1: A typical Content Distribution Network (CDN)

A request-routing system [11][4] enables the activity of directing a client request to a suitable surrogate server. It consists of a set of network elements called *request-router*s. Request-routers work in cooperation to direct a request. They also use dynamic information about network conditions and load on the surrogate servers to balance the load while directing requests. The Request-routing system interacts with the accounting system to inform the content delivery to the clients. It also interacts with the distribution system to inform the demand of content. Distribution system uses this information to place the content into the surrogate servers.

The Accounting system supports the measurement and recording of content distribution and delivery activities. Information recorded by the accounting system is used as a basis for the transfer of money, goods, and obligation among the network service providers and the content providers. The Accounting system consists of several sub-accounting systems distributed over the network.

2.2 CDN Peering

The value of a CDN to the content providers is the combination of its scale and reach. There are limits to how large any one CDN's scale and reach can be. The scale and the reach are unfortunately limited by the cost of equipment, the space available for deploying equipment, and by the demand for that scale/reach of infrastructure. Operating a single CDN to cover the whole world is not feasible. Rather, it is natural that a large number of manageable-size CDNs will operate worldwide, each covering a portion of the world's scale and reach. They will be peered to cooperate with each other to gain the worldwide scale and reach, while building and administrating only a part of that scale and reach [11][12][10].

CDN peering is the interconnection among two or more separately administered CDNs. CDNs are interconnected via *CDN Peering Gateways* (CPG), which provide Request-Routing Peering System, Distribution Peering System, and Accounting Peering System. These peering systems collectively control the selection of the delivery CDN, content distribution between peering CDNs, and usage accounting, including billing settlement among the peered CDNs. CDN peering makes a larger set of highly distributed surrogate servers available to the clients, which otherwise could not be achieved by an individual CDN. A peered CDN that is directly connected with the origin server of a content provider is the origin CDN for that content provider. Content that has been *pulled* or *pushed* into any one CDN may be distributed into any other peered CDN. Even a non-origin but peered CDN can issue commands to initiate content distribution across the CDN peers from the origin server. Accounting information regarding content delivery (to the clients) and distribution activity is made available to the origin CDN by the peered

CDN. The Origin CDN passes this information to the content provider. Client requests can be directed to any surrogate server of any CDN from any peered CDN. Different CDN peering scenarios are sedcribed in [10].

3 Request-Routing Problem in CDN

Major CDN problems are surrogate server implementation, request-routing from clients to either a surrogate server or the origin server, content distribution and synchronization from origin server to surrogate servers, client authentication, authorization and accounting (AAA), and the same level of security enforcement by the surrogate servers and the origin server. On the other hand, CDN peering has the following major requirements: request-routing peering, distribution peering, accounting peering, and security consideration. CDN peering requirements and present state of the peering solutions are discussed in [12]. In the remainder of this section we are going to discuss only the CDN request-routing problem along with the research works done so far that addrersses the solution for the problem.

3.1 Request Routing Problems

The Request routing system in a CDN routes the client requests to a suitable surrogate server to serve the request. The selected surrogate server should be closest to the client and least loaded. Request routing system needs to maintain proximity metrics and server load metrics for this purpose.

3.2 Request Routing Solutions

There are a number of research works on possible solutions for CDN request routing problems [5][6]. Each work has its own approach. Each of them or a combination of them can be used in a CDN. This section describes those solutions individually.

3.2.1 DNS based Request Routing

Domain Name Server (DNS) based request routing is widely used in the Internet at present. DNS based request routing is also used in many CDNs because of its ubiquity as a directory service. DNS servers handle the domain name of the desired web site or content. The Client initiates a name lookup in a local DNS server, which is supposed to return the address of a surrogate server near the client. If local DNS cache misses, it forwards the name lookup to the DNS root server. DNS root server returns the address of the authoritative DNS server for the web site. The Authoritative DNS server then returns the address of a surrogate server near the client based on specialized routing, load monitoring and Internet mapping mechanism. Finally the client retrieves the content from the designated surrogate server.
Authoritative DNS server can return addresses of multiple surrogate servers to the client site DNS server. Client site DNS server can use those multiple addresses one after another in a round robin fashion to route requests from different clients for the same content to different surrogate servers. This technique increases the reliability and load balancing of CDN. Multiple DNS servers in a single DNS resolution can be used to distribute more complex decisions from a single DNS server to multiple, and more specialized DNS servers. This can be done by redirecting the authority of the next level domain to another DNS server using Name Server (NS) records or Canonical Name (CNAME) records [4]. Multiple physical DNS servers that combine request routing and metric measurement can share an anycast IP address. The packet containing the DNS resolution request will reach one of these DNS servers, which is the closest to the client site DNS server. After receiving the packet, the DNS server knows that it is the closest and can use this information in making routing decision.

DNS based request routing has several limitations. It involves multiple levels of redirection and does not scale well, since on a DNS cache miss, lookup incurs the long round-trip time to centralized DNS servers (root and authoritative) irrespective of clients location. Furthermore, the short time-to-live used in this system to respond quickly to changes in network conditions, increases load on DNS servers. Use of network-level metrics does not respond to application-level failure, a client may continually be redirected to an unresponsive web server. As DNS requests go through intermediate DNS servers, client's actual location may be hidden and the chosen surrogate server may not be suitable from client's perspective. DNS based systems have difficulty scaling to support multiple content provider networks and large numbers of content providers.

3.2.2 Transport-layer based Request Routing

Transport-layer based request routing is used to achieve finer and the next level of request routing after the first level is done by DNS based request routing. It uses information such as client's IP address and port number, available in the first packet from the client, in the request routing process and hands off the session to a more appropriate surrogate [4].

3.2.3 Application-layer based Request Routing

Several application-layer based request routing mechanisms are described in [4]. Like transport-layer based request routing application-layer based request routing works with DNS based request routing to provide fine-grained request routing down to the level of individual content item. Two types of application-layer based request routings are header inspection and content modification. Header inspection can be Uniform Resource Locator (URL) based, Mime Header based, or Site Specific. URL based request routing uses URL or URL prefix of the requested content to make the routing

decisions. Two types of URL based request routings are 302-Redirection and In-path Element. Mime Header based request uses mime headers such as Cookie, Language, User-Agent available in the content request to make routing decisions. In Site Specific request routing, site specific identifiers such as Secured Socket Layer (SSL) Session Identifier are used to direct a content request. Content modification technique can be used by the content providers not by the CDNs. Typically, a content item is made up of a basic structure that includes references to additional embedded content items and the client fetched embedded items from the origin server. Using content modification technique, a content provider can modify references to the embedded items so that the client can fetch an embedded item from the best surrogate server. This type of modification is also called URL Rewriting and the URL Rewriting can be either priori or dynamic [4].

An application-layer based request routing system using application-layer *anycasting is described in* [17]. . A CDN request routing system can also use this technique. It uses Anycast Domain Names (ADNs) and that uniquely identifies a collection of IP addresses of an anycast group. An application-layer anycast service maps an ADN into one or more IP addresses. It does not require any change the network layer operations. A client generates an anycast query to an anycast resolver and the resolver processes the query and selects a server based on the performance data about the servers. The resolver replies with an anycast response. An ADN is of the form:

<Service> %<Domain Name>.

Where <Domain Name> indicates the authoritative anycast resolver for this ADN. <Service> identifies the service within the authoritative resolver. Each network location is pre-configured with the address of its local anycast resolver, which resolves queries and/or consults the higher authoritative resolver, as in DNS. The resolver maintains the information necessary to perform the mapping from ADN to IP address. The resolver also collects and maintains the performance information associated with each member of the anycast group. Resolver collects the response time, measured from just prior to sending a query to just after receiving the complete response. This response time and the information about the size of the response data could be used to compute the throughput. Response time is collected because it is directly correlated with a user perception of the quality of service. Response time depends on server capabilities (speed, processors), current server load, network path characteristics and current load on the path. The technique used to collect metric data is scalable to a large number of servers, anycast groups and clients. Though the technique used is relatively accurate, the performance penalty for slightly inaccurate metric data will not be severe; rather than selecting the best server it may select a nearly best server. The server can monitor its own performance and push this information to the resolvers when significant change is observed. The update information can be network layer multicast to all resolvers that maintain information about the server. The server controls the network traffic generated by this mechanism by adjusting the monitoring and push schedule. This push technique is a scalable technique and it collects accurate server performance, but it needs to modify the server and it does not measure the network path. A probing agent on behalf of large number of clients can make periodic queries to the server to determine the performance that a client would experience. The probing agents co-locate with the resolvers. This probe technique measures the network path performance and does not require modification to the server. . Both push and probe techniques have limitations but they are complementary to each other. To get the best result, a hybrid push-probe technique is generally used. Another type of application-layer based request routing using URL Forwarding and Compression for adaptive Web caching is described in [3]. This technique can also be used in CDN. Due to space limitation we are not discussing it in this paper.

3.2.4 Content-layer based Request Routing

Reference [14] presents a content-layer based request routing technique useful for CDN request routing. It provides content routing support in the core of the Internet performed by content routers, which are extended to support naming. Content routers act as conventional IP routers and name servers, and participate in both IP routing and named based routing. Not every router needs to be a content router, only firewalls, gateways, and Border Gateway Protocol (BGP) level routers need to be content router. Internet Name Resolution Protocol (INRP) is used to perform lookup, and Name-Based Routing Protocol (NBRP) is used to perform routing. Aggregation mechanism is used to enhance NBRP. INRP uses Redirection mechanism for finding isolated names not advertised in the NBRP.

INRP is reverse-compatible with DNS; uses same record types and packet format, but with different underlying semantics. Clients initiate content requests by contacting a local content router. Each content router maintains a set of name-to-next-hop mappings in a routing table. When an INRP request arrives in a router, the desired name is looked up in the name based routing table and the next hop is chosen from the table. The content router forwards the request to the next hop content router. In this way, the request proceeds toward the best content server. When an INRP request reaches the content router adjacent to the best content server, that router sends back a response message containing the address of the preferred server. This response is sent back along the same path of content routers. If no response appears, intermediate routers can select alternate routes and retry the name lookup. It can recover from a failing server or an out-of-date routing information by providing an “anycast” capability at the content level, with network and client control to reselect alternatives based on direct experience with the chosen

server. Routing is done at the granularity of server names rather than full URLs. Relaying the name lookup across the same path as the packets ensures that naming is as available as endpoint connectivity and that the replica selected is actually reachable.

NBRP performs routing by name with a structure similar to BGP. Just as BGP distributes address prefix reachability information among autonomous systems, NBRP distributes name suffix reachability to content routers. NBRP is also a distance vector routing algorithm with path information. NBRP routing advertisement contains the path of content routers toward a content server. Routing advertisement from content servers also includes a measure of the load at that server, specified in terms of the expected response latency. Load attribute indicates that content which takes longer to access appears "further away" from a routing perspective, and treated internally by a content router as extra hops in the routing path. This load information is propagated to a limited distance to keep the number of routing updates manageable. NBRP updates are authenticated by cryptographic signature, in a manner similar to Secure BGP. A content server's authenticity is verified by the signature on its initial routing update; content routers receive explicit permission from this content server to advertise routes with their name added to the path list. If a content peer becomes unreachable, then all the contents available through that peer are unreachable as well. IP routing information is used to select among routes that appear identical at the content routing level. Content routing policies are kept consistent with the IP routing policies so that the decisions made at the content level are faithfully carried out by the IP forwarding level.
To handle large numbers of names, which appear globally in name-based routing tables, NBRP supports combining collections of name suffixes that map to the same routing information, into routing aggregates [14]. Isolated names not advertised in the name-based routing systems have records indicating their actual topological location in the Internet in terms of a well-known name. Content routers return this redirection record in response to name lookup requests. After receiving the redirection records, a client or a first hop content router sends a lookup query using the name in it.

3.2.5 NAT based Request Routing

Reference [8] proposes a Network Address Translation (NAT) based request routing technique named TRIAD (Translating Relaying Internet Architecture integrating Active Directories). NAT was introduced to allow Internet address reuse. It is used for address allocation autonomy, which allows an enterprise to assign addresses independent of its ISP. It supports multi-homing, switching ISPs and decoupling the number of hosts and the number of addresses provided by the ISP. But with NAT an IP address is only meaningful within one address realm. Application-specific proxies are required in NAT routers to make some Internet applications function correctly. It needs to modify DNS responses that transit through NAT router and to update the transport checksum of a packet, compromising end-to-end reliability and conflicting with end-to-end security. In NAT it is hard to communicate freely between separate private realms without renumbering. TRIAD uses NAT but overcomes above problems. TRIAD techniques are applicable in CDN request routing using NAT.

4 Conclusions

In this paper, we have presented an overview of Content Distribution Networks and CDN request-routing problem. We have also described and analyzed different request routing mechanisms that can be deployed in CDNs. Comparative performance studies of the described request routing mechanisms constitute highly valuable future work in this area.

5 References

[1] Andrzej Duda, and Mark A. Sheldon, "*Content Routing in a Network of WAIS Servers*", Broadcast Technical Report, Esprit Basic Research Project 6360, Laboratory for Computer Science, MIT, Cambridge, MA 02139, USA.

[2] Antony Rowstron, and Peter Druschel, "*Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems*", in Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms, Germany, 2001.

[3] B Scott Michel, Konstantinos Nikoludakis, Peter Reiher, and Lixia Zhang, *"URL Forwarding and Compression in Adaptive Web Caching*", in Proceedings of the IEEE Infocom, 2000.

[4] B. Cain, F. Douglis, M. Green, M. Hofmann, R. Nair, D. Potter, and O. Spatscheck, "*Known CDN Request-Routing Mechanisms*", http://www.contentalliance.org/docs/draft-cain-cdnp-known-req-route-00.html (work in progress) , November 2000.

[5] Bakker, E. Amade, G. Ballintijn, I. Kuz, P. Verkaik, I. Van der Wijk, M. van Steen, and A. S. Tanenbaum, "*The Globe Distribution Network*", http://www.cs.vu.nl/globe

[6] D. A. Johnson, "*A Protocol for the Interoperability of Content Distribution Networks"*, Masters Thesis, Department of Computer Science, University of Victoria, Victoria, British Columbia, Canada, 2002.

[7] D. Gilletti, R. Nair, and J. Scharber, "*CDN Peering Authentication, Authorization, and Accounting Requirements*", http://www.contentalliance.org/docs/draft-gilletti-cdnp-aaa-reqs-00.html (work in progress), November 2000.

[8] David R. Cheriton, and Mark Gritter, "*TRIAD: A Scalable Deployable NAT-based Internet Architecture*", Computer Science Department, Stanford University.

[9] F. Douglis, I. Chaudri, and P. Rzewski, "*Known Mechanism for Content Internetworking*", http://www.contentalliance.org/docs/draft-douglis-cdi-known-mech-00.txt November 2001.

[10] M. Day, and D. Gilletti, "*Content Distribution Network Peering Scenarios*", http://www.contentalliance.org/docs/draft-day-cdnp-scenarios-02.html (work in progress), November 2000.

[11] M. Day, B. Cain, and G. Tomlinson, "*A Model for CDN Peering*", http://www.contentalliance.org/docs/draft-day-cdnp-model-03.html (work in progress), November 2000.

[12] M. Green, B. Cain, and G. Tomlinson, "*CDN Peering Architectural Overview*", http://www.contentalliance.org/docs/draft-green-cdnp-framework-00.html , (work in progress), September 2000.

[13] Mark A. Sheldon, Andrzej Duda, Ron Weiss, James W. O'Toole, Jr., and David K. Gifford, "*A Content Routing System for Distributed Information Servers*", Technical Report MIT/LCS/TR-578, Laboratory for Computer Science, MIT, Cambridge, MA 02139, USA, 1993.

[14] Mark Gritter, David R. Cheriton, "*An Architecture for Content Routing Support in the Internet*", http://www.dsg.stanford.edu/papers/contentrouting/ , 2001.

[15] Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, "*Harvest: A Scalable, Customizable Discovery and Access System*", Technical Report CU-CS-732-94, Department of Computer Science, University of Colorado-Boulder, USA, 1995.

[16] Y. Charlie Hu, Daniel A. Rodney and Peter Druschel, "*Design and Scalability of NLS, a Scalable Naming and Location Service*", in Proceedings of the IEEE Infocom, 2002.

[17] Zongming Fei, Samrat Bhattacharjee, Ellen W. Zegura, and Mostafa H. Ammar, "*A Novel Sever Selecttion Technique for Improving the Response Time of a Replicated Service*", Networking and Telecommunication Group, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332.