Content Delivery Networks: Status and Trends

Athena Vakali and George Pallis • *Aristotle University of Thessaloniki*

CDNs improve network performance and offer fast and reliable applications and services by distributing content to cache servers located close to users.

The Web's growth has transformed communications and business services such that speed, accuracy, and availability of network-delivered content has become absolutely critical — both on their own terms and in terms of measuring Web performance. Proxy servers partially address the need for rapid content delivery by providing multiple clients with a shared cache location. In this context, if a requested object exists in a cache (and the cached version has not expired), clients get a cached copy, which typically reduces delivery time.

Web caching has three primary benefits. First, it reduces bandwidth consumption, network congestion, and network traffic because it stores the frequently requested content closer to clients. Second, it reduces external latency (the time required to transfer objects from the origin server to proxy servers) because it delivers cached objects from the proxy servers. Finally, caching improves reliability because clients can obtain a cached copy even if the remote server is unavailable. Using a shared proxy cache, however, has two significant drawbacks: If the proxy is not properly updated, a client might receive stale data, and, as the number of clients grows, origin servers typically become bottlenecks. When numerous users access a Web site simultaneously — such as when "flash crowds"[1] flooded popular news sites with requests in the wake of the September 2001 terrorist attack in the US — serious caching problems result and sites typically become unavailable.

Researchers have widely considered content delivery networks to be an effective solution to reducing these disadvantages.[2,3] CDNs act as trusted overlay networks that offer high-performance delivery of common Web objects, static data, and rich multimedia content by distributing content load among servers that are close to the clients.[4,5] CDN benefits include reduced origin server load, reduced latency for end users, and increased throughput. CDNs can also improve Web scalability and disperse flash-crowd events. Here we offer an overview of the CDN architecture and popular CDN service providers.

CDN Overview

CDNs first emerged in 1998[6] to address the fact that the Web was not designed to handle large content transmissions over long distances.[7,8] According to Network World Fusion (www.nwfusion.com), about 2,500 companies now use CDNs. In the US alone, for example, Storigen Systems (www.storigen.com) estimates that the CDN market generated US$905 million in 2000 and will reach US$12 billion by 2007.

Proxies and CDNs essentially address two different issues. ISPs use proxies to store the most frequently or most recently requested content; they're designed to work on a local basis. Web servers use CDNs to store content specified by the network administrator. CDNs can improve access to content that is typically uncacheable by caching proxies, including secured content, streaming content, and dynamic content.[8] (The "Dynamic Content"

 Published by the IEEE Computer Society 1089-7801/03/$17.00

Dynamic Content

Today's Web sites provide sophisticated e-commerce and personalized services that depend heavily on dynamic content generation.[1] Dynamic content includes HTML or XML pages that are built on the fly and unique to specific users. Because a proxy cannot cache dynamic content, network performance is seriously compromised.

To effectively manage dynamic Web content, CDNs support a new markup language, Edge Side Includes (www.esi.org). ESI enables dynamic content delivery directly from surrogate servers by facilitating the breakdown of Web pages into independently cacheable fragments. In ESI, each fragment is treated as a separate entity and maintained as a separate object in the surrogate servers. This significantly reduces bandwidth requirements for dynamic content and delivers important savings to Web servers.[2]

References

1. A. Datta et al., "Accelerating Dynamic Web Content Generation," *IEEE Internet Computing*, vol. 6, no. 5, 2002, pp. 27–36.
2. J. Dilley et al., "Globally Distributed Content Delivery," *IEEE Internet Computing*, vol. 6, no. 5, 2002, pp. 50–58.

sidebar discusses the latter in more detail.) CDNs can thus facilitate content delivery for the Semantic Web, which primarily aims to extend current Web content to computer-understandable content.

CDNs are complex, with many distributed components collaborating to deliver content across different network nodes. There are four basic steps in this collaboration:

- nonorigin, or *surrogate*, servers cache the origin servers' content,
- routers deliver the client's content request to a suitable surrogate server,
- various network elements distribute the requested content from the origin to the surrogate servers, and
- an accounting mechanism provides logs and accounting information to the origin servers.

Collaborations among CDN components can occur over nodes in both homogeneous and heterogeneous environments.

Surrogate Servers: The CDN Infrastructure

In a CDN, surrogate servers attempt to offload an origin server's work by delivering content on its behalf. A CDN is characterized according to its structure, the number of surrogate servers, where those servers are located, and which server will service a client's request.

To deliver content to end users with quality-of-service (QoS) guarantees, CDN administrators must ensure that surrogate servers are strategically placed across the Web. Generally, the problem is to place M surrogate servers among N different sites ($N > M$) in a way that yields the lowest cost (widely known as the *minimum K-median* problem). Many researchers have focused on surrogate server placement and proposed several placement algorithms, including tree-based,[9] Greedy, and Hot Spot.[10] These algorithms specify the surrogate server locations that improve performance with low infrastructure cost. Another example is Scan,[11] a scalable replica management framework that generates replicas on demand and organizes them into an application-level multicast tree. Appropriately placed surrogate servers benefit ISPs by reducing bandwidth consumption, and benefit Web servers by reducing latency for their clients.

CDN administrators must also determine the optimal number of surrogate servers because this directly affects a given traffic pattern's cost. There are two typical approaches for determining this optimal number:

- The *single-ISP approach* deploys many surrogate servers — at least 40 around its network's edge[6] — so that each supports numerous objects. An ISP with a global network can thus achieve adequate geographical coverage without relying on other ISPs. The most common policy is to deploy surrogate servers at one or two locations in each major country. The ISPs associate the surrogate servers with large caches (the larger the cache, the higher the hit ratio). A disadvantage of this approach is that it might locate the surrogate servers at a distance from the CDN's clients.
- The *multi-ISP approach* locates numerous surrogate servers in as many global ISP points of presence (POPs) as possible. The goal is to get surrogate servers close to users and thus deliver content quickly and reliably. One advantage of this approach is that CDNs can deliver content from the requesting client's ISP. This minimizes the number of network access points required to deliver content. Some large distributed systems, for example, have more than 8,000 servers.[6] A disadvantage of this approach is that each surrogate server receives fewer hits,

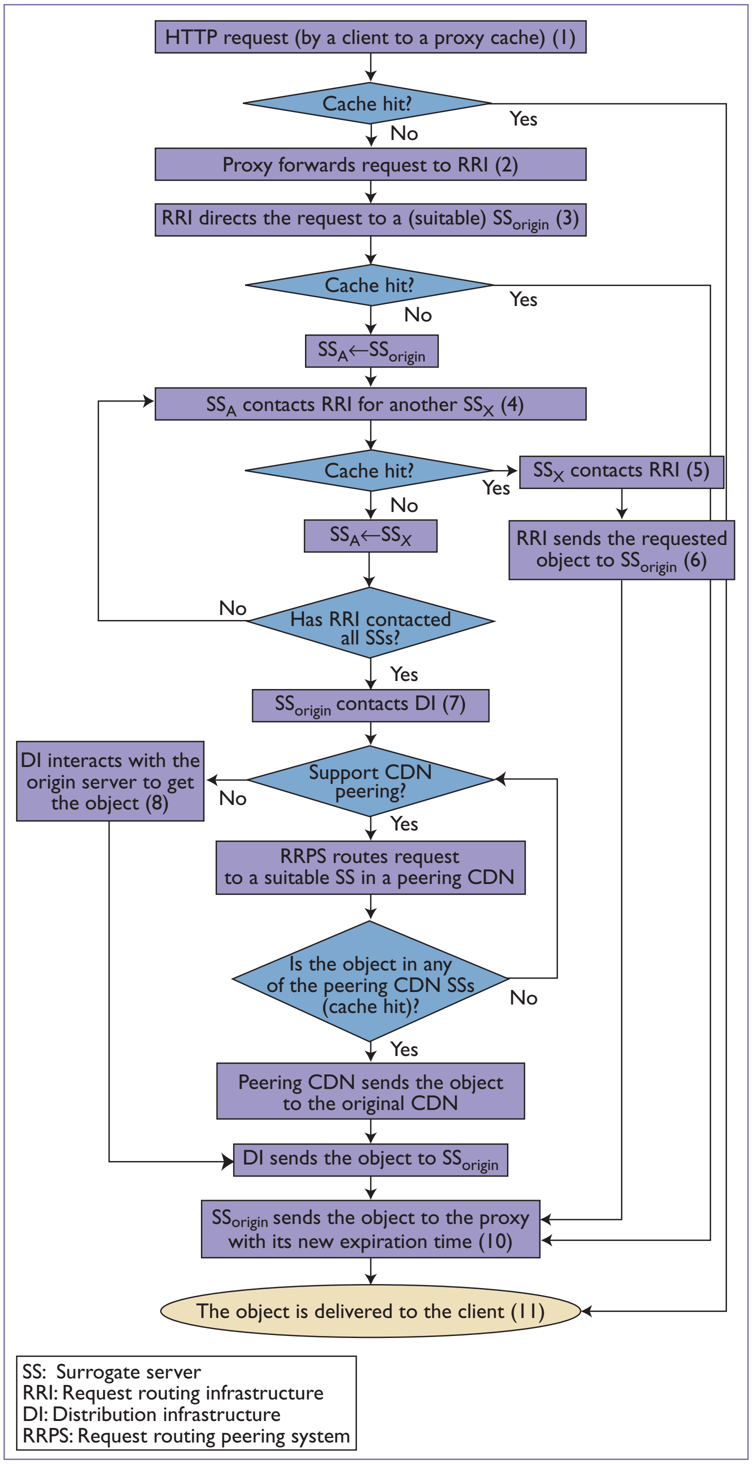

Figure 1. Delivering the requested object to the client. "Cache hit" indicates that the cache contains a current version of the object.

resulting in poor CDN performance. This network is also more complex and more difficult to maintain than with the single-ISP approach.

Estimates of performance variations under different traffic volumes seem to indicate that the single-ISP approach works better for sites with low-to-medium traffic volumes, while the multi-ISP approach is better for high-traffic sites.[6]

Estimating the most appropriate surrogate server is another critical issue. The "best" server does not always mean the one nearest to the client, the fastest, or the most reliable, but rather is a combination of these attributes (among others). In general, the most appropriate surrogate server is the one with the closest topological proximity to the visitor's browser. Topological proximity is an abstract metric that considers a variety of criteria, such as physical distance, speed, reliability, and data transmission costs. Some CDNs support surrogate server selection based on a balance of multiple metrics, such as proximity, server load (the load of servers or network paths to the servers), and an aggregate of the two. As a typical example, a proximity-load-threshold algorithm[7] uses a proximity metric to select the nearest surrogate server, and then uses a load metric to ensure that the server isn't overloaded.

Routing and Distribution

Figure 1 shows the content delivery process, from a client request submission to delivery of the requested object. A cache hit indicates that the object is in the cache and the cached version has not expired; if CDN providers support peering, the client request servicing procedure is more complex.

To select the most appropriate surrogate server for content routing, most CDN providers currently use Domain Name System (DNS) redirection; some also use URL rewriting.[6]

DNS redirection. In this approach, the DNS performs the mapping between a surrogate server's symbolic name and its numerical IP address. There are several steps to locating the appropriate surrogate server:

1. The user sends a DNS query to the local DNS server, which forwards the query to the CDN's request-routing infrastructure (RRI).
2. The RRI queries each surrogate server, asking them to examine their particular route to the local DNS server.
3. Each surrogate server sends measurement results to both the local DNS server and the RRI; it also sends other criteria to the latter that let the infrastructure compare each server's topological proximity to the local DNS server. Server measurements are based on network metrics — such as latency, packet loss, and router hops from

surrogate servers to requesting entities — and feedback from a pool of surrogate servers, from which the proximity-load-threshold algorithm selects the least-loaded surrogate server.

4. The RRI compares all previous measurements and selects the most appropriate surrogate server to deliver the requested content, sending a DNS response to the user's local DNS server.
5. The user's local DNS server sends that response to the user.

Both full and partial-site CDN service providers use DNS redirection. With full-site content delivery, the RRI redirects all origin server requests through the DNS to a CDN server. In contrast, partial-site content delivers only embedded objects — such as Web page images — from the corresponding CDN. With partial-site content CDNs, the surrogate server fetches only noncacheable (or expired) objects from the origin Web server. This reduces

- the load on the site's content generation infrastructure, and
- the data that the surrogate server must retrieve from the origin server.

Because embedded objects change infrequently, partial-site deployment typically achieves better performance.

DNS redirection does have drawbacks, however. The most significant is that DNS lookup times can increase latency. To solve this problem, CDN administrators typically split the CDN DNS into two levels for DNS load distribution.[7] Another drawback is that DNS redirection is not scalable because clients don't access the actual domain names that serve their requests. As a result, there is no alternative server to fill client requests if the target surrogate server fails.

URL rewriting. Although most CDNs use DNS-based schemes, some use URL rewriting, in which the origin server redirects clients to different surrogate servers by rewriting the dynamically generated pages' URL links. With a Web page containing an HTML file and embedded objects, for example, the Web server would modify references to embedded objects so that the client could fetch them from the best surrogate server. To automate this process, CDNs provide special scripts that transparently parse Web page content and replace embedded URLs.[7]

URL rewriting is also called *content modification*. CDNs use it a priori in one of two ways:

- *Statically*: the Web server modifies content and rewrites embedded URLs before the content is stored on the origin server and made available to clients.
- *Dynamically:* the content is modified on-demand when the server receives a client request.[7]

For dynamic Web pages, the basic drawback of URL rewriting is that it imposes a significant performance overhead as scripts must be continually executed.[5,7]

CDN peering. Peered CDNs deliver content on each other's behalf. A CDN can thus expand its reach to a larger client population by using partnered CDN servers and their nearby forward proxies. When client requests are serviced, it's possible that the RRI won't find the requested object on any surrogate server. In such a case, the RRI can request the object from a peer CDN that has a contract with the origin CDN. This case is shown in Figure 1 where, after action 7, the RRI routes the requested object to a peer CDN through the request-routing peering system module.

A content provider usually contracts with only one CDN, and each CDN contacts other peer CDNs on the content provider's behalf. The authoritative CDN — the one with the provider contract — acts on the provider's behalf, working with other CDNs to deliver content and paying them accordingly. In general, peered CDNs provide better performance at a relatively low cost compared with solo CDNs.

Accounting Mechanism

CDNs and peering CDNs typically support an accounting mechanism that collects and tracks information related to request routing, distribution, and delivery.[12] Such mechanisms are composed of network elements, such as identifiers, measurers, and counters. CDN administrators implement the accounting mechanism using known protocols (FTP, simple network management protocol, and session initiation protocol, for example). The mechanism then gathers information in real time and collects it in log files for each CDN component. To provide customers (Web server administrators) with accounting and billing information, the mechanism uses a common data format to collect, partition, and analyze the component logs and determine billable usage based on a minimum set of attributes.

CDN administrators also use this information to maintain a network overview and, in conjunction

Table 1. Content delivery network provider characteristics.

CDN service provider	Service type	Content distribution	Fees	Customers
Akamai www.akamai.com	Multi-ISP, partial-site request servicing, peering	More than 12,000 surrogate servers spanning 1,000 networks in 62 countries	US$1,995 per month for each Mbps of delivered content	Covers 70 percent of the market, with more than 3,600 customers including Apple, CNN, MSNBC, Reuters, and Yahoo
Adero www.webvisions.com/adero/	Multi-ISP, full-site request servicing, peering	Surrogate servers in more than 30 countries	Depends on resellers (CDNs that buy Adero services)	Serves 30 customers, including resellers Exodus and UUNET
Digital Island www.sandpiper.net	Multi-ISP, partial-site request servicing, peering	2,500 surrogate servers spanning 327 networks in 35 countries	Starts at US$1,500 per month	More than 900 customers including AOL, Canon, Cisco Systems, Microsoft, and Hewlett Packard
Mirror Image www.mirror-image.com	Multi-ISP, partial-site request servicing, peering	22 surrogate servers in North America, Europe, and Asia	US$2,100 per month for each Mbps of delivered content	More than 200 customers including Creative, Open Systems, and SiteRock
Inktomi www.inktomi.com	Single-ISP, full-site request servicing, peering	10 surrogate servers across China	Starts at US$4,000 per month	13 CDNs including Adero and Digital Island and more than 200 Web sites

with existing billing and reporting systems, to analyze content-usage costs. Studying these reports can help administrators reduce a CDN's communication and traffic costs. Moreover, for peering CDNs, an accounting mechanism can help administrators monitor and establish accounting and authorization policies. Finally, the mechanism's user-friendly interface provides customers with effective and secure access to accounting and billing information.

Criteria for Selecting a CDN

There are currently only a few CDN service providers; Table 1 shows some of the most popular. To our knowledge, there is little published information about each CDN's design and infrastructure (although an article was recently published on Akamai[13]). In general, most providers have distributed topologies, all support peering and all content types, and several offer both DNS redirection and URL rewriting technologies.

CDNs serve two groups:

- CDN *customers* are the Web server administrators who contract with the CDN.
- CDN *clients* are the Web server's end users who download content through the CDN.

The HTRC Group (www.htrcgroup.com/pages/white.html) asked 100 CDN customers to rate their criteria for choosing a CDN on a scale of 1 to 7. These customers considered end-user performance most crucial, followed by service, support, and the CDN's reputation.

Krishnamurthy and colleagues offer a detailed study of how the most popular CDNs perform.[5] Typically, CDN customers evaluate CDN performance using five key metrics.[6,7,14]

- *Cache hit ratio*: the ratio of cached documents versus total documents requested. A high hit rate reflects an effective cache policy. Gadde and colleagues defined the CDN cache ratio for a fixed object (x), as

$$\frac{C_{N_I} - C_{N_L}}{1 - C_{N_L}},$$

  where C_{N_I} is the cache-hit ratio (of the surrogate servers) for a client population of fixed size N_I and C_{N_L} is the cache-hit ratio at leaf node (a proxy, for example) serving client population of a fixed size N_L.[14]
- *Saved bandwidth*: the decrease in bytes retrieved from the origin servers.
- *Latency*: the set of surrogate servers that least delay the requested document. Reducing latency generally decreases the saved bandwidth.
- *Surrogate server utilization*: the fraction of time that the surrogate servers are busy. Adminis-

trators use this metric to calculate the CPU load, the number of connections being served, and each surrogate server's storage I/O.
- *Reliability*: packet-loss ratio. Administrators use packet-loss measurements to calculate a CDN's reliability. High reliability indicates that the content is always available.

From a client's perspective, the most important CDN criteria are cost, performance, and availability. In terms of *cost*, using a CDN is initially more expensive than not using one.[6] As traffic volume increases, however, the overall economic cost declines. Most commercial CDN service providers base their charges on bandwidth, using a per-gigabyte fee (Table 1 shows recent fees charged by the most popular CDN providers).

With increasing CDN deployment, the underlying *performance* must be quantitatively evaluated in terms of storage, communication, and bandwidth costs; throughput performance; latency; and scalability. Although researchers recently compared how two CDN providers, Akamai and Digital Island, performed in terms of selecting servers with minimum latency to the client,[4] there have been few other experiments in this area.

Given this, we strongly recommend that CDN customers either perform their own tests or use an independent performance-measuring company such as Keynote Systems (www.keynote.com) or Web Lens (www.appliant.com). A CDN provider's own performance testing can be misleading because the service might be ideal for one Web site, for example, but perform poorly for another. The reason is that performance depends both on the site's content (file types and streaming formats), and where that content is stored. Some CDN providers support only certain streaming formats in certain countries and geographical regions. Independent performance-measuring companies can efficiently measure CDN performance because they support benchmarking networks of strategically located measurement computers connected through major Internet backbones in several cities. These computers measure how a particular Web site performs from the end user's perspective, providing meaningful metrics on Web application performance in critical areas. CDN customers can thus select a CDN based on their particular site configurations.

Performance and availability are closely related; if a Web site is busy, end users might have trouble reaching it. Most independent performance-measuring companies have found that using a CDN gives end users a lower error rate on such things as DNS lookup failures and connection and page time-outs.

Write for Spotlight

Spotlight focuses on emerging technologies, or new aspects of existing technologies, that will provide the software platforms for Internet applications. Spotlight articles describe technologies from the perspective of a developer of advanced Web-based applications. Articles should be 2,000 to 3,000 words. Guidelines are at www.computer.org/internet/dept.htm.

Contact department editor Siobhán Clarke at siobhan.clarke@cs.tcd.ie.

CDNs: Looking Ahead

When combined with replication technologies, such as mirroring, CDNs can offer efficient multicast delivery of especially rich content. Streaming media proliferation will drive the CDN market; the Internet Research Group predicted that CDN traffic volumes will dramatically increase in the next few years (see www.netsedgeonline.com/press_releases/cdn_market_share.html). As a result, the cost of CDN products and services will decrease over time, driving up adoption rates. In anticipation, vendors are introducing new value-added services for content distribution, adaptation, and negotiation. In particular, vendors have implemented content service networks.[15] CSNs act as another network infrastructure layer built around CDNs and provide next-generation CDN services through interactions with Web servers, surrogate servers, and ISPs' proxies.

Several companies also offer CDN software. Cisco, for example, has introduced several network management software products (www.cisco.com/en/US/products/sw/conntsw/ps845/products_user_guide_book09186a0080080e68.html). Novell has released another set of "velocity" CDN products under the Volera name (www.novell.com/products/volera), and Unitech Networks offers software for the IntelliDNS CDN service provider. Other companies offering CDN software include CacheWare, Radware, SinoCDN, and WARP Solutions.

Although CDNs' future is potentially prosperous, several important issues remain open — including security. The ability to intelligently link and monitor content, for example, is critical to CDN deployment. Without scalable and reliable distributed storage and surrogate servers, CDNs are vulnerable to attackers (such as worms or other various hackers efforts; see www.telin.nl/Enindex.shtml). The secure sockets layer (SSL) protocol has become the CDN standard for establishing and maintaining a secure session among clients and CDN servers.[7] In particular, the SSL protocol is widely accepted to secure data transmitted between CDN infrastructures over the Internet (sockets between client and server are coupled with RSA Security's public key encryption).

In any case, we expect CDNs to expand in terms of number of clients, infrastructure distribution, technology integration, and peering contracts. The challenge for such CDNs will be to offer secure content delivery with high QoS guarantees.

References

1. Y. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites," *Proc. 11th Int'l World Wide Web Conf.* (WWW 02), ACM Press, 2002, pp. 293–304.
2. M. Baentsch et al., "Enhancing the Web's Infrastructure: From Caching to Replication," *IEEE Internet Computing*, vol. 1, no. 2, 1997, pp. 18–27.
3. K.L. Johnson et al., "The Measured Performance of Content Distribution Networks," *Computer Comm.*, vol. 24, nos. 1-2, 2001, p. 202; www.cs.bu.edu/pub/wcw01/206.
4. D. Kaye, *Strategies for Web Hosting and Managed Services*, John Wiley & Sons, 2002.
5. M. Rabinovich and O. Spatsheck, *Web Caching and Replication*, Addison Wesley, 2002.
6. F. Douglis and M.F. Kaashoek, "Scalable Internet Services," *IEEE Internet Computing*, vol. 5, no. 4, 2001, pp. 36–37.
7. B. Krishnamurthy, C. Wills, and Y. Zhang, "On the Use and Performance of Content Distribution Networks," *Proc. 1st Int'l Internet Measurement Workshop*, ACM Press, 2001, pp. 169–182.
8. I. Lazar and W. Terill, "Exploring Content Delivery Network," *IEEE IT Professional*, vol. 3, no. 4, 2001, pp. 47–49.
9. B. Li et al., "On the Optimal Placement of Web Proxies in the Internet," *Proc. 18th Joint Conf. IEEE Computer and Comm. Soc.* (IEEE INFOCOM), IEEE CS Press, 1999, pp. 1282–1290.
10. L. Qiu, V. Padmanabham, and G. Voelker, "On the Placement of Web Server Replicas," *Proc. 20th Joint Conf. IEEE Computer and Comm. Soc.* (IEEE INFOCOM), IEEE CS Press, 2001, pp. 1587–1596.
11. Y. Chen, R.H. Katz, and J.D. Kubiatowicz, "Dynamic Replica Placement for Scalable Content Delivery," *Proc. Int'l Workshop on Peer-to-Peer Systems* (IPTPS 02), LNCS 2429, Springer-Verlag, 2002, pp. 306–318.
12. M. Day et al., "A Model for Content Internetworking (CDI)," Internet draft, IETF, Feb. 2002; work-in-progress.
13. J. Dilley et al., "Globally Distributed Content Delivery," *IEEE Internet Computing*, vol. 6, no. 5, 2002, pp. 50–58.
14. S. Gadde, J. Chase, and M. Rabinovich, "Web Caching and Content Distribution: A View From the Interior," *Computer Comm.*, vol. 24, nos. 1-2, 2001, pp. 222–231.
15. W.Y. Ma, B. Shen, and J.T. Brassil, "Content Services Network: Architecture and Protocols," *Proc. 6th Int'l Workshop on Web Caching and Content Distribution* (IWCW6), 2001; www.cs.bu.edu/pub/wcw01.

Athena Vakali is an assistant professor at the Aristotle University of Thessaloniki in Greece. Her research interests include Web data management, data caching, content delivery, and information clustering. She received a PhD in disk subsystems performance from the Aristotle University, Greece, and an MS in computer science from Purdue University. She is a member of the IEEE Computer Society and the ACM. Contact her at avakali@csd.auth.gr.

George Pallis is a PhD student at the Aristotle University of Thessaloniki. His research interests include Web data clustering, usage characterization, and content delivery. He received a BS in computer science from the Aristotle University. Contact him at gpallis@ccf.auth.gr.