Scalable Content Delivery Technology

Junya Akiba† *and Hirofumi Abe*

Abstract

Our scalable content delivery system can deliver content to hundreds or thousands of sites. It uses "leaf-to-leaf delivery", in which content is transferred between terminals, and "on-the-fly transfer", in which content is forwarded while it is being received, to achieve high-speed and reliable content delivery to a large number of sites, which is difficult with conventional content delivery technology.

1. Introduction

NTT Information Sharing Platform Laboratories has developed MDS-Dome [1] and MDS-Pack [2] as solutions for a content delivery network (CDN) (MDS: Mass Delivery System). MDS-Dome is a CDN for Internet use. To improve the scalability of Web servers, it shares the load among servers and networks by copying Web content to multiple synchronized mirror servers. MDS-Pack is a CDN for corporate use. It achieves mission critical content delivery for corporations and governments by providing content encryption, restart from the point of an interruption to ensure efficient delivery of large-volume content over an unstable network, and a tracking function that presents delivery status details to the user. Both MDS-Dome and MDS-Pack have been employed extensively by NTT operating companies as platforms for solutions and application service provider (ASP) businesses [3], [4]. However, now that content delivery services have become widely used by corporations, there are growing demands for larger-scale and more diverse forms of delivery. These demands cannot be met by MDS-Dome or MDS-Pack.

Customers want to be able to promote new products by displaying content on terminals installed in shops or on the street and to train franchise employees using digital content delivered to each site. However, as shown in Fig. 1, it has been practically impossible in terms of cost and performance to deliver a large volume of content to hundreds or thousands of sites.

This article explains how NTT Laboratories have solved the problems associated with content delivery to a large number of sites.

† NTT Information Sharing Platform Laboratories
Musashino-shi, 180-8585 Japan
E-mail: akiba.junya@lab.ntt.co.jp

2. Requirements for content delivery to a large number of sites

To be able to deliver content to a large number of sites, the content delivery system must satisfy the following requirements.

(1) Scalability

The delivery system should be able to handle, flexibly and efficiently, cases where a service starts on a small scale but grows rapidly to cover a large number of sites or cases where the number of sites changes frequently. Potential users of multipoint content delivery include local governments, schools, franchised distributors, convenience stores, sales kiosks, and restaurants. In these applications, the number of sites can reach hundreds or thousands. Thus the delivery system should be able to handle that number of sites.

(2) Real-time and reliable delivery

Even though the number of sites is large, the other requirements for content delivery remain the same. In applications such as advertising using content delivery for the promotion of new products and school education relying on the delivery of teaching materials over the network, the desired content must be delivered reliably within a specified time. In particular, the delivery time must remain more or less con-

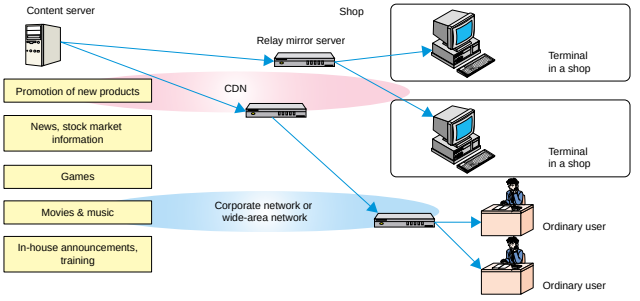

Fig. 1. Content delivery service.

stant even if the number of sites increases.

(3) Low delivery cost

Instead of using a CDN, a corporation could record the content on DVD disks or videocassettes and distribute them physically by mail or express courier. So the price of using a CDN should be competitive with these conventional means of distribution. In particular, minimizing the incremental cost for adding a new site can make a CDN an economic solution compared with physical delivery, whose cost rises in proportion to the number of sites.

(4) Low-cost and robust terminals

Conventionally, CDNs have been used to deliver Web content to mirror servers, or used by corporations to deliver business content. In contrast, for large-scale multipoint delivery, we assume direct delivery of content to customers' terminals installed in shops or on the street. In such applications, the terminal should be inexpensive (unlike a PC server), easy to operate, and robust.

3. Scalable content delivery system

For large-scale multipoint content delivery, we have developed a scalable content delivery system by combining leaf-to-leaf delivery and on-the-fly transfer technologies. This section describes the problems encountered in applying a conventional CDN to large-scale multipoint delivery, their solutions, and our scalable content delivery system that implements these solutions. The main features of our system are summarized in Table 1.

Table 1. Features of the scalable content delivery system.

Unit of data for delivery	Per directory (only a differential file is compressed and delivered)
Delivery trigger	Periodically; at a specified time; or on demand
Delivery direction	Push from the center to edges and pull from edges to the center
Delivery method	HTTP (SSL) (star-shaped; on the fly), Megacast
Security	SSL communication, support of a proxy server (only for the pull from edges to the center)
Restricted bandwidth transmission	The bandwidth for use in delivery can be restricted to an arbitrary value.
Partial mirror	Only certain files are delivered (mirrored). The rest are stored in mirror servers (caches) only when there is a request for this.
Operations and management	Creation of delivery groups, configuration of delivery parameters, on-demand delivery, collection of delivery progress information, collection of delivery result reports, display of error information, instructions for recovery from an error, and collection of server status information
Scale of delivery	Delivery to about 800 sites has been verified in an experimental environment.
Volume of data delivered	No restriction (can handle file sizes up to 2 GB)

3.1 Leaf-to-leaf delivery

In most conventional CDNs, the original content held in the original content server is delivered using a simple star-shaped topology. In such cases, the delivery time increases in proportion to the number of mirror servers. In addition, a large difference in delivery completion times between the first and last mirror servers may be unacceptable in some applications depending on the type of content handled. If content is delivered to multiple mirror servers simultaneously to solve this problem, then an extremely large bandwidth will be required for the access line to the central server. (For example, if the bandwidth of the access line to each mirror server is 1.5 Mbit/s and there are a thousand sites, then the bandwidth required for the access line to the central server is 1.5 Gbit/s.)

Solutions to this include using IP multicasting and peer-to-peer (P2P) file transfer [5]. However, IP multicasting over satellite links has a problem in that the required reception equipment may not be available at every site. Moreover, the application of IP multicasting over terrestrial links usually has geographically limitations because of the operational constraints of Internet service providers. P2P file exchange also has a problem: it cannot be sufficiently managed. To solve these problems, we propose using "leaf-to-leaf delivery" as a content delivery mechanism that is both scalable and reliable and is applicable to IP (Internet protocol) networks that are already used by or can easily be introduced by customers [6]. In leaf-to-leaf delivery, the server holding the original content and installed in the center is called the root server (or simply the root). A terminal to which content is delivered is called a leaf server (or simply a leaf). Leaves deliver content to each other (Fig. 2).

Leaf-to-leaf delivery looks similar to P2P file exchange in that terminals exchange content with each other. However, in an ordinary P2P file exchange, each terminal autonomously finds a peer that has the desired file and directly asks it for the file. In some cases the request goes via the nearest server or even the original content server, so delivery routes may be established at random. In contrast, in our leaf-to-leaf delivery, the root server optimizes the delivery routes based on the performance and access line bandwidth of each leaf (Fig. 3). As a result, content can be delivered over a delivery tree that makes the most of the available access line bandwidths, making content delivery faster and less expensive than P2P file exchange. In addition, if the file size is known, the delivery time can be predicted.

Furthermore, while most P2P file exchange tools cannot manage the content delivery status centrally, in leaf-to-leaf delivery the content delivery status of every terminal can be monitored because the root server controls all deliveries. In conventional CDNs, if content cannot be delivered successfully due to a mirror server or network link failure, the content is simply re-transmitted from the original content server and re-synchronized. As the number of sites increases, so will the number of mirror servers that fail to deliver content. In our system, if there is a faulty server, the root server informs it of a nearby, normally operating leaf, which re-synchronizes the content delivery of the faulty leaf. When the faulty leaf is restarted, it sends an inquiry to the root server to check that the content it holds is correct. Then it

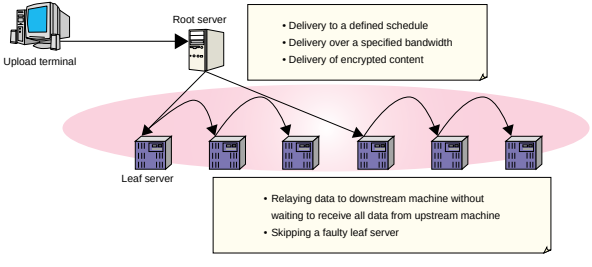

Fig. 2. Basic operations of the scalable content delivery system.

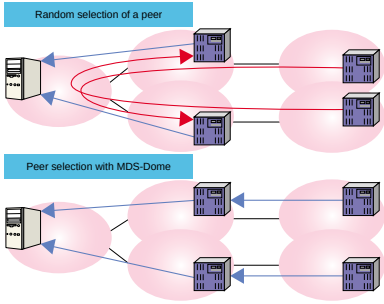

Fig. 3. Selection of a peer in leaf-to-leaf delivery.

performs all the processes needed for recovery with the help of the nearby normally operating leaf. This substantially reduces the processing load on the root server for recovery of a faulty leaf.

3.2 On-the-fly transfer

Leaf-to-leaf delivery makes it possible to increase the number of sites without needing to upgrade the central equipment. However, since communication between leaf servers is repeated sequentially, there can be a substantial difference between the delivery completion times of the first and last leaf servers. "On-the-fly transfer" reduces this time difference: while a leaf is receiving and storing data in its memory, it simultaneously relays the data to the next leaf. As shown in Fig. 4, this on-the-fly transfer can prop-

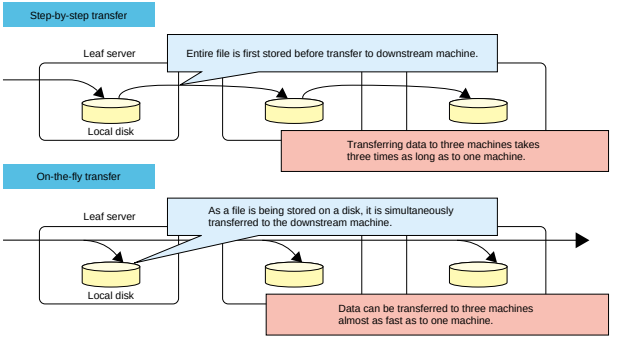

Fig. 4. On-the-fly transfer.

agate a file faster than step-by-step transfer, in which the leaf server in the middle stores the entire file before relaying it to the next leaf. In the ideal case where all leaf servers have the same access line bandwidth and there is no overhead time for relaying data in each leaf server, it theoretically takes approximately as long to deliver content to many sites as to a single site. Employing leaf-to-leaf delivery and on-the-fly transfer, the proposed system reduces the difference between the delivery completion time of the first and last leaf servers without decreasing the delivery throughput to each leaf server.

4. Built-in OS

Recently, with built-in operating systems (OSs), such as EmbededLinux and WindowsCE, becoming more powerful, products that meet the requirements for terminals of a large-scale multipoint delivery system have become available. Most of these products have the advantages of being easy to use with only a limited set of operations required. The OS can be started or shut down by simply turning the power on or off. In addition, they are highly robust because they do not have many moving parts. In order to fully benefit from all the above-mentioned features of the scalable content delivery system using built-in OS terminals, we have reduced the required memory size and simplified procedures. As a result a fully-fledged PC is not required as a terminal at each site: an appliance such as a set-top box (STB) with a built-in OS is sufficient. We have also added many functions to make the system easy to operate and reliable. For example, power can be turned on or off without requiring special care. Pressing the power button starts content delivery automatically and causes the terminal to join the delivery tree automatically. Consequently, it is now possible to use low-cost STBs or home servers as terminals to deliver content, such as a promotional video for a new product to shops or an entertainment video to homes, easily and inexpensively.

5. Conclusions

This article has introduced our work on achieving content delivery to hundreds or thousands of sites. Table 1 summarizes the main features of the Scalable Content Delivery System we have developed for that purpose.

The system is based on the two conventional CDNs: MDS-Dome and MDS-Pack, and uses leaf-to-leaf delivery and on-the-fly transfer technologies to realize high-speed content delivery to a large number of sites, a requirement which has been difficult to achieve with conventional CDNs.

References

[1] K. Yamada, T. Shiroshita, and S. Ushijima, "Large-capacity Content Delivery System for B-to-E and B-to-C: MDS-Dome/Megacast, I.SS," NTT Technical Journal, Vol. 14, No. 4, pp. 46-49, 2002 (in Japanese).

[2] T. Arai and K. Ishikawa, "Highly Efficient Content Delivery System for B-to-B Applications: MDS-Pack," NTT Technical Journal, Vol. 14, No. 4, pp. 39-43, 2002 (in Japanese).

[3] http://www.gtrax.ne.jp/

[4] http://www.ntt.com/e-con/

[5] http://www.gnutella.com/

[6] H. Abe, S. Ushijima, and K. Kamiya, "Proposal for Edge-to-Edge Multi-point Content Delivery," IEICE General Conference, B-06-258, 2003 (in Japanese).

Junya Akiba
Senior Research Engineer, Software Architecture Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in applied physics from Tohoku University, Sendai, Miyagi in 1990 and 1992, respectively. In 1992, he joined NTT Communication Switching Laboratories, Tokyo, Japan. Currently, he is engaged in the development of CDN systems. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).

Hirofumi Abe
Engineer, NTTPC Communications, Inc.

He received the B.E. and M.E. degrees in electronic engineering from Keio University, in 1994 and 1996, respectively. In 1996, he joined NTT Network Service Systems Laboratories, Tokyo, Japan, where he engaged in research on web applications and multimedia contents delivery. In 2003, he moved to NTTPC Communications, Inc. Currently, he is developing web applications and business databases. He is a member of IEICE and the Information Processing Society of Japan.