IPDM—A Novel Technology for High-speed Downloading of Large-scale Contents

Hiroshi Yamamoto, Yoshinori Kawamura, and Isami Nakajima

Abstract

There are growing demands for the delivery of large-scale content over the network, such as video and music. To expand the delivery business, it is essential to shorten the time required for delivery. It has become clear that using broadband networks alone is not a solution because the content to be downloaded is increasing enormously in size. This article introduces a new technology to solve this problem.

1. Expansion of content delivery services

As the use of always-on Internet connections and broadband access becomes widespread, the business of delivering music, video, and other types of content is beginning to flourish. Since most content providers assume customers are using ADSL (asymmetric digital subscriber line) access (at a few megabits per second), the main items of content delivered are compressed music files or short video clips. However, as optical access spreads and the performance of the delivery servers improves, demands are expected to rise for the delivery of large-scale content, such as high-quality PCM-encoded music, DVD videos with a duration of 1 to 2 hours, and games software (PCM: pulse code modulation). The content business market is forecast to grow to 2.8 trillion yen by 2007 [1].

2. Problems

One key to expanding the content delivery business and increasing customer satisfaction is a shorter content delivery time. If it takes only a few seconds or minutes to deliver content, then depending on the content type, the on-demand delivery business will grow rapidly [2].

To shorten the delivery time, we must increase the network access bandwidth and raise the performance of the delivery server. However, these measures alone cannot achieve the speed needed for the delivery of large-scale content because, as content grows in size, delivery speed becomes constrained by two factors: the throughput limit imposed by the network transfer protocol and the limit on the transfer speed of the delivery servers. These constraints are explained below.

2.1 Limit on TCP throughput

The network transfer protocol commonly used for content delivery is TCP (transmission control protocol), which includes retransmission control for packets that are lost. The maximum throughput that can be achieved by TCP depends on the packet loss rate and the round-trip delay between end terminals. **Figure 1** shows how the theoretical maximum throughput of one TCP session depends on the round-trip delay and packet loss rate. These days, the round-trip delay is increasing because of longer propagation distances and the introduction of firewalls and load sharing devices. It is difficult to reduce the delay caused by these constraints, even if a highly broadband network is introduced. A round-trip delay of 50 ms or more is not uncommon, even for communication between points within the same prefecture in Japan. (The maximum theoretical throughput is 16 Mbit/s with the widely used TCP New Reno [3].) Therefore, even if high-speed access (e.g., optical access) is used, TCP still imposes a limit on the data throughput.

† NTT Service Integration Laboratories
Musashino-shi, 180-8585 Japan
E-mail: ipdm@lab.ntt.co.jp

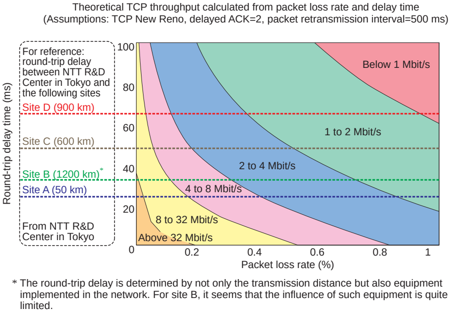

Fig. 1. Theoretical throughput of TCP New Reno.

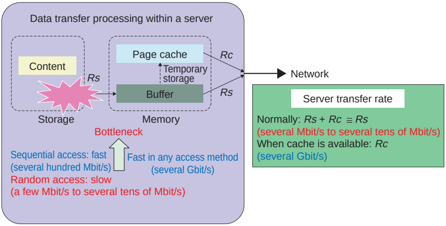

Fig. 2. Model for data transfer processing within a server.

2.2 Limit on the transfer rate of servers

The data transfer mechanism inside a delivery server can be modeled as shown in **Fig. 2**. Since network interface cards (NICs) have become faster recently, they are no longer a major limiting factor on the server transfer speed. Server transfer speed can be expressed as the sum of the storage reading rate *Rs* and the cache reading rate *Rc*.

Rs is high for continuous readout, but tends to be extremely low for random access. Access to a large item of content by many users becomes random, so a high data transfer cannot be expected. *Rc* depends on the cache hit rate. For a large item of content, the cache hit rate is necessarily low (**Fig. 3**).

Since neither *Rs* nor *Rc* can be increased, the server transfer speed often becomes the bottleneck for high-speed delivery.

3. IPDM: technology to overcome these limits

3.1 Basic idea

We have developed a high-speed data delivery technology called IPDM (intelligent parallel download method), which overcomes the two limits that appear as the content size increases. It can, for example, deliver the contents of a DVD within a few minutes over a 100-Mbit/s access line.

IPDM maximizes the use of the data available in the

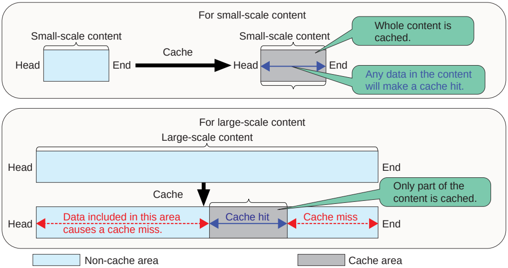

Fig. 3. Relationship between content size and cache hit rate.

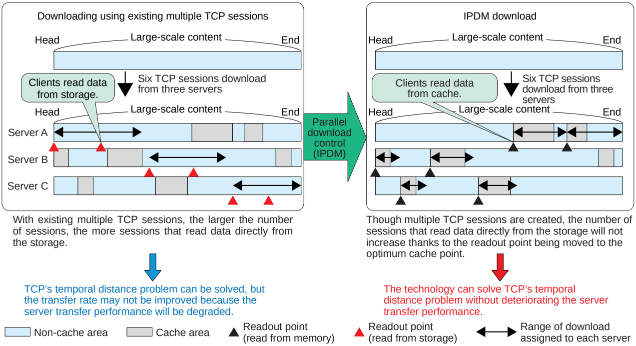

Fig. 4. Use of multiple TCP sessions with IPDM.

page cache within the server to overcome the limit on the server's data transfer rate. Specifically, it estimates which part of the content is in the cache and when a delivery request arrives from a client, IPDM moves the readout point to the cache. This increases the data transfer rate.

To overcome the limit of the TCP throughput, IPDM uses multiple TCP sessions simultaneously. **Figure 4** shows a case in which a large item of content is delivered in parallel from three servers (A, B, and C) to a single client using multiple sessions. Each triangle indicates the point where a TCP session is reading the content at a certain point in time. If the existing method of downloading data using multiple TCP sessions is used, the number of sessions reading data from the storage device increases, which reduces the server transfer rate (left side of Fig. 4). IPDM uses its data availability estimation to deliver data from several caches. This maximizes the use of cached data, so it does not aggravate the server transfer rate problem that normally arises when multiple sessions are used (right side of Fig. 4).

3.2 Example of cache control

Figure 5 shows how the readout point is controlled when delivery of a large-scale content to a new client is started. Since the LRU*1 algorithm is used in the operating system of most delivery servers, the data most recently read by the client remains in the cache at the point shown by s in Fig. 5. Consider the case where content delivery to the client starts at the head of the content (upper part in Fig. 5). During Times 1 to 3, data reading by Client C1 causes some of the downloaded content to be stored in the cache. At Time 4, when C2 begins to receive the content, the front part of the previous content is lost in the cache. With the conventional technology, which always downloads from the head of the content, even C2 receives content from the storage device. As time progresses, C3 starts to read data from the storage device. Thus, the three clients must share the readout capability of the storage device.

In contrast, in IPDM at Time 4 when C2 begins to read data, it does not read the data in the storage device but the data in the cache stored there for C1 (lower part of Fig. 5). Similarly, C3 begins to read data cached for C1 and C2. As a result, even though all three clients are receiving data, only C1 is reading data in the storage device. Since C1 monopolizes the storage device, it can read data at high speed. C2 and C3 read cached data, which increases the data readout speed.

3.3 IPDM functions

The functional arrangement of IPDM is shown in **Fig. 6**. To use IPDM, it is necessary to install cache management server software (Function 1) in the server and an IPDM plug-in (Function 2) in the user's browser. It is not necessary to modify existing delivery servers at all. Therefore, a provider with an existing delivery server can adopt IPDM at a very low cost.

3.4 Client interface

IPDM's client interface is shown in **Fig. 7**. An embedded Javascript program ensures that delivery is automatically based on IPDM, so the user need not be conscious of using IPDM.

*1 LRU (least recently used): a cache updating algorithm in which the least recently used data is deleted first when new data is to be cached.

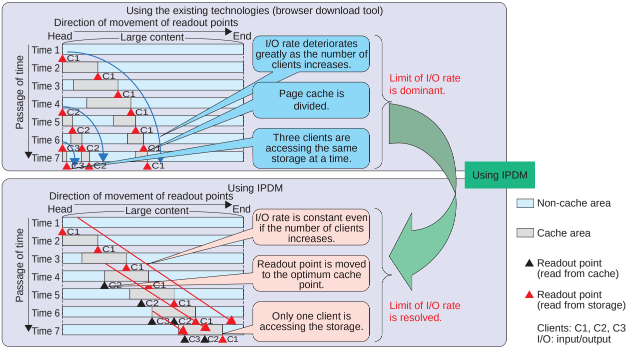

Fig. 5. Cache control in IPDM.

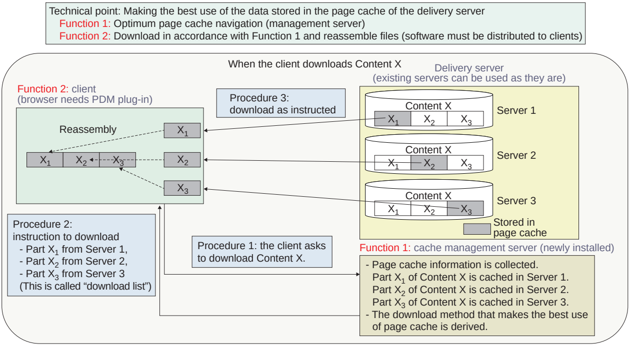

Fig. 6. Functional layout of IPDM.

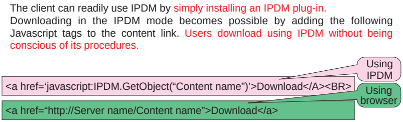

Fig. 7. Client interface of IPDM.

3.5 Throughput comparison with alternative technologies

We compared the throughput of IPDM and competing technologies: load balancer and Flashget (a method of downloading data using multiple existing TCP sessions) [4]. The results are shown in **Fig. 8**. The left side of Fig. 8 plots the total throughput of six servers for different volumes of data delivered. As the data volume increased, the throughputs of the load balancer and Flashget decreased dramatically while the throughput of the IPDM remained nearly constant at a high level.

The right side of Fig. 8 shows the throughput per client for different values of round-trip delay. With load balancer, as the round-trip delay increased, the problem of TCP (temporal distance) became pronounced and throughput suffered. Although Flashget used multiple TCP sessions, the server transfer problem was prominent and kept the throughput almost constant at a very low level. In contrast, when IPDM was used, multiple TCP sessions could be used without any server transfer problem. Thus, it was possible to download data at the maximum speed possible for a given bandwidth.

4. Future issues

We have developed technology for delivering data at almost the maximum throughput of the broadband network or delivery server used. We will study ways to increase the reliability of this technology to guarantee operations in the event of data loss and we will explore business opportunities.

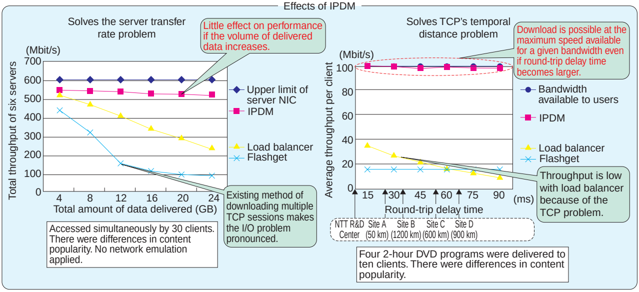

Fig. 8. Performance comparison between IPDM and competing technologies.

References

[1] White Paper on Digital Content, Digital Content Association of Japan, 2003 (in Japanese).

[2] Nikkei Communication, Oct. 2, 2000 (in Japanese).

[3] RFC2582: The NewReno Modification to TCP's Fast Recovery Algorithm.

[4] http://www.amazesoft.com/

Yoshinori Kawamura

Senior Research Engineer, NTT Service Integration Laboratories.

He received the M.S. degree in mathematics from Nagoya University, Nagoya, Aichi in 1989 and joined NTT the same year. He has been working in R&D of quality control for broadband network services.

Hiroshi Yamamoto

Research Engineer, Communication Traffic & Service Quality Project, NTT Service Integration Laboratories.

He received the M.S. degree in information and computer science from Waseda University, Tokyo in 2001 and joined NTT Service Integration Laboratories the same year. He has been working on the architecture and performance evaluation of content delivery networks. He has also been contributing to ITU-T SG13 activities.

Isami Nakajima

Group Leader, NTT Service Integration Laboratories.

He received the M.S. degree in mathematics from Tohoku University, Sendai, Miyagi in 1984. Since joining NTT Laboratories in 1984, he has been engaged in research on network design, traffic control, and QoS management. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan and the Operations Research Society of Japan.