Missouri University of Science and Technology
Scholars' Mine

Electrical and Computer Engineering Faculty Research & Creative Works

Electrical and Computer Engineering

01 Sep 2001

Proxy Servers for Scalable Interactive Video Support

Husni Fahmi

Mudassir Latif

Sahra Sedigh
Missouri University of Science and Technology, sedighs@mst.edu

Arif Ghafoor

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/1225

Follow this and additional works at: https://scholarsmine.mst.edu/ele_comeng_facwork

 Part of the Electrical and Computer Engineering Commons

Recommended Citation

H. Fahmi et al., "Proxy Servers for Scalable Interactive Video Support," *Computer*, vol. 34, no. 9, pp. 54-60, Institute of Electrical and Electronics Engineers (IEEE), Sep 2001.
The definitive version is available at https://doi.org/10.1109/2.947092

This Article - Journal is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Proxy Servers for Scalable Interactive Video Support

Targeting scalability, load balance, and resource efficiency in streaming multimedia applications, proxy servers facilitate faster real-time access to cached objects and reduce response time to document requests.

Husni Fahmi
Mudassir Latif
Sahra Sedigh-Ali
Arif Ghafoor
Purdue University

Peiya Liu
Liang H. Hsu
Siemens Corporate Research

Streaming audio and video-on-demand technologies support a broad spectrum of multimedia information applications, including digital libraries, telemedicine, distance education, and military communications. These information systems allow real-time access and sharing of documents stored on distributed servers. Deploying proxy servers as part of the network infrastructure helps to meet the increasing demand for access to multimedia documents.

Proxy servers manage quality-of-service guarantees and facilitate faster access to cached objects. Because of their physical proximity to clients, proxy servers usually cache recently requested documents, reducing server load and the time required to respond to document requests. Proxy servers can temporarily store frequently requested documents so that users can access these files without repeatedly connecting to servers that house them. Proxy servers also reduce traffic and relieve bandwidth congestion at network bottlenecks,[1-3] thereby increasing an organization's security and controlling access to internal and external information sources. Using multiple proxy servers for distributed object caching achieves scalability and load balancing, which are significant in the heavily loaded multiuser environments typical of many video-on-demand applications. As Figure 1 shows, in a groupware environment, proxies placed at the branching nodes of the multicast tree can perform multicasting data transmission.

Although proxy servers can cache textual and image documents, using them for streaming multimedia documents has not been effective because their large size and unique access patterns hinder effective caching.[4,5] The rapidly increasing demand for streaming applications has led to a critical need for streaming media caching techniques. Storing the entire volume of multimedia streams is prohibitive because it can deplete an ordinary proxy server's buffer capacity.

In the past two years, several researchers have addressed the issue of streaming multimedia caching. The central paradigm falls into one of three categories:

- *Treating multimedia streams as sequences of nonoverlapping segments.*[5] The proxy server divides a continuous media file into blocks of equal size and groups these blocks into variable-sized segments.
- *Dividing the streams into multiple layers of quality information.*[4] This approach uses hierarchical encoding to divide multimedia streams into multiple layers for quality adaptation. If the bandwidth between the proxy and a client can support high-quality streams, a proxy can gradually prefetch higher information layers from the servers. The base layer is cached in the proxy.
- *Prefetching initial portions of the streams.*[6,7] Prefetching avoids a bottleneck between the server and the proxy by providing connectivity to clients with high bandwidth. This technique can replace cached streams at a finer granularity than deciding for or against retrieving an entire file.

Because a multimedia stream has an unusually large volume, client-side playback is prone to high latency, loss rates, and increased delay. To correct

0018-9162/01/$10.00 © 2001 IEEE

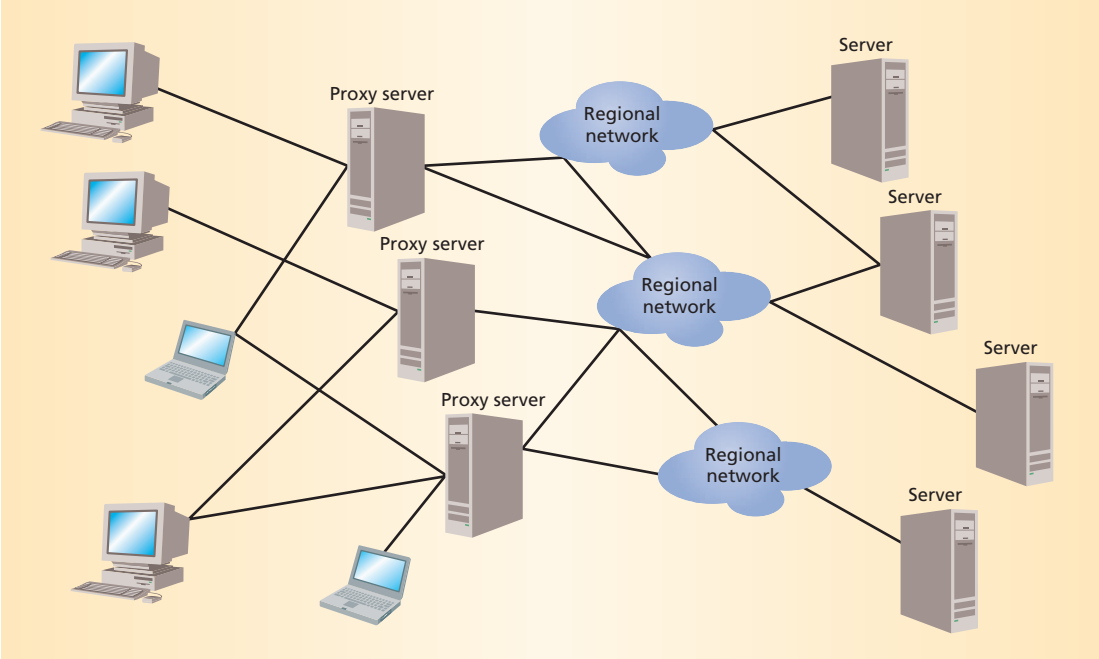

Figure 1. Distributed streaming multimedia environment with proxy servers. Proxy servers reduce network traffic, relieve bandwidth congestion, and help to achieve scalability and load balancing.

these problems, proxy servers can use *prefix caching*[6] to store the initial frames of popular clips. The server transmits subsequent frames to the proxy while the proxy transmits data to the client. Zhi-Li Zhang and associates[7] developed *video staging*, a related video-delivery technique that retrieves only a portion of a video stream from the central video server across the backbone network; local proxy servers deliver the remaining video. Video staging significantly reduces bandwidth requirements, particularly for a multiuser environment in which many users access the same local network.

SUPPORTING INTERACTIVE FUNCTIONS

Allowing the real-time interactive presentation of multimedia documents to users is an essential requirement of streaming multimedia applications. Users can manipulate VCR-like playout controls to perform random seek, fast forward, and rewind operations for audio or video segments. Before playing out an entire video, users may want to preview a segment by browsing through *hotspots*. A hotspot in a video stream represents a salient object that is semantically related to other media, such as image, audio, or text that provides details about the video object. Embedded hyperlinks semantically relate these highlighted portions of the video to other documents. Viewers seek these hotspots rather than using conventional VCR forward or reverse functions to find video data.

Seek latency, the delay in retrieving a portion of the video, is the principal impediment to providing interactive functions for hotspots in a video-on-demand system. In interactive applications, in addition to continuous playout of the video data, video servers respond to many requests for small portions of the video data. By directing these requests to proxy servers, video servers can devote more connections to continuous playouts, which improves resource use, increases scalability, and facilitates responding to requests for lengthy videos. Caching hotspots at proxy servers reduces the load on video servers and decreases the response time for random seek because proxy servers are located close to their clients. Caching hotspots lets users preview a media file before playing the entire file. To provide a meaningful preview, a proxy should cache a hotspot segment that constitutes at least one complete video scene—a sequence of video shots that conveys a meaningful message. In MPEG video files, one hotspot segment consists of an integral multiple of a group of pictures (GOP). This basic MPEG playback unit is a collection of frames that the client can decode in isolation from the rest of the file.

Our simulations evaluating the response time to interactive functions and server load indicate that delegating interactive function support to the proxies significantly reduces the response time and allows more continuous server connections.

HOTSPOT CACHING AT PROXY SERVERS

Figure 2 shows the HyperStreaming document player—the Web-based streaming multimedia application we used in our experiments. This application facilitates the playout of multimedia documents such as technical manuals for machine-operation procedures. High-quality videos and multimedia data can

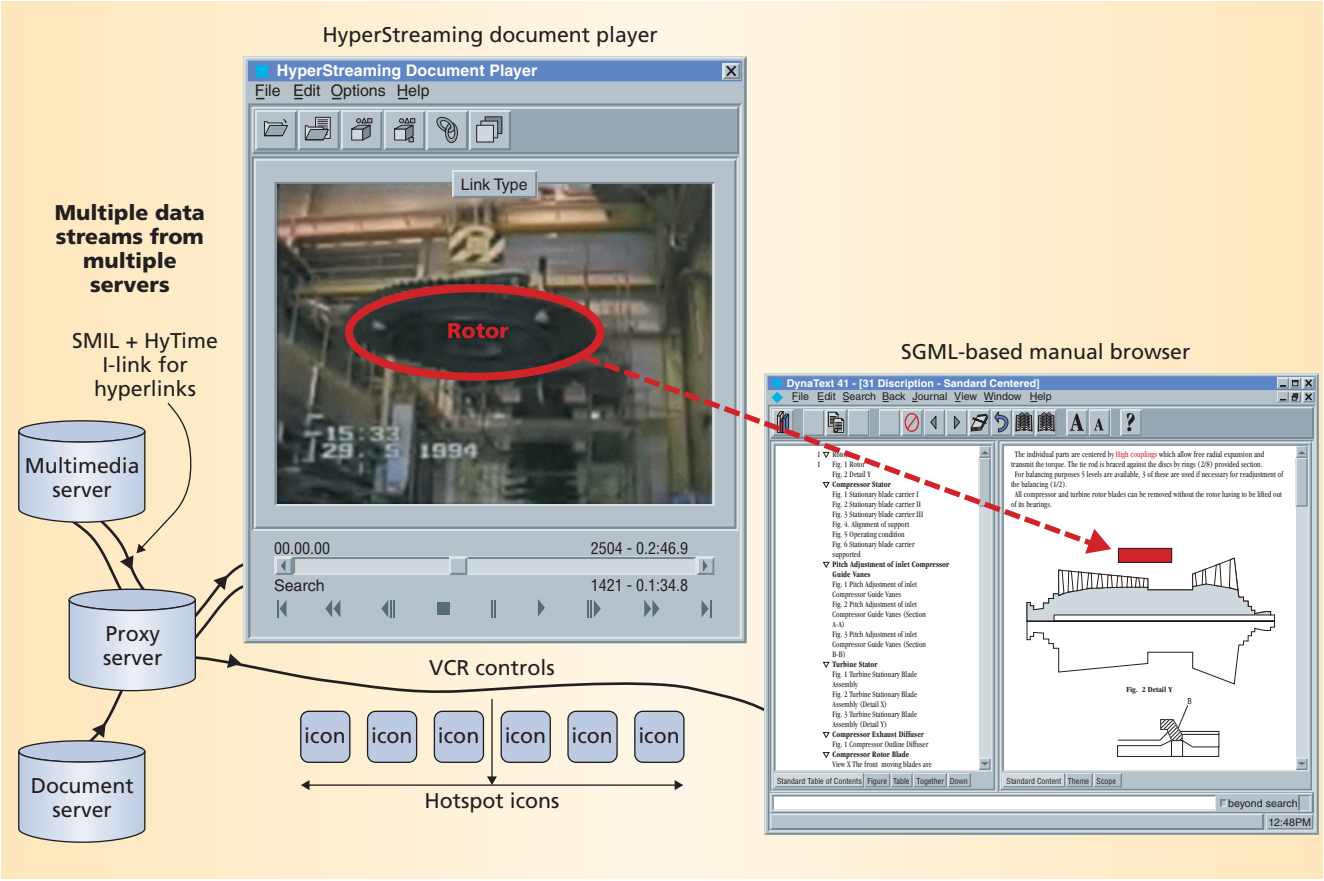

Figure 2. HyperStreaming document player. In a video for a plant management application, a hotspot indicates embedded hyperlinks from the turbine to text and image data in a technical manual containing detailed information about the turbine.

enhance the details of operational procedures consisting of physical locations and sequences, which textual information alone cannot describe succinctly. Users can preview portions of a media file before deciding to play the entire file, or they can make several playback requests to preview a specific portion of the media file.

Caching hotspots at multiple proxy servers facilitates document playback interaction by allowing users to seek a hotspot in both forward and reverse-time directions. Because the total size of the hotspot segments in a video file is significantly smaller than the complete video file, caching the hotspot segments instead of the complete file uses buffer space more effectively.

Figure 3 depicts the caching of hotspot segments of a typical manual at proxy servers. These segments may not be contiguous in the original video file, and each segment is a different size depending on the size of the video scene associated with the hotspot.

Unlike continuous playback requests for complete files, which exhibit temporal locality and predictability, short-duration random access puts a heavier load on the server, particularly in a multiuser environment. Excessive loading can decrease service performance and diminish the quality of service. Caching hotspots at the proxy servers reduces this overload. This caching scheme isolates the servers from random requests and improves the response time from the users' perspective. In addition, the server can process more requests for continuous playouts, increasing the multimedia information system's scalability.

Data transmission techniques

Two well-known data transmission techniques are used for streaming multimedia applications:[8]

- Specialized real-time methods focus on achieving optimal presentation quality and efficient data transmission.
- HTTP transfers multimedia files.

While real-time methods provide efficient data transmission, using them for multimedia streaming requires specialized servers. Another drawback of this technique is that users cannot get inside a firewall to access multimedia data from sources other than Web servers. On the other hand, HTTP streaming uses a Web server's existing infrastructure, and its traffic is generally allowed through firewalls. HTTP 1.1[9] lets clients issue *byte-range* requests to obtain a designated range of bytes in requested files. This function lets clients control the downloading of different portions of multimedia files. Byte-range requests offer flexibility because users can seek a particular position in the media.

In the streaming and caching protocol, the proxy sends and receives two distinct streams—the main video and the hotspot segments. The user progresses through the main video stream and sends an HTTP byte request corresponding to a hotspot ID parameter to the proxy. The proxy uses this parameter to synchronize the flow of hotspots with the main video stream. For each hotspot, the proxy maintains a video segment of suffi-

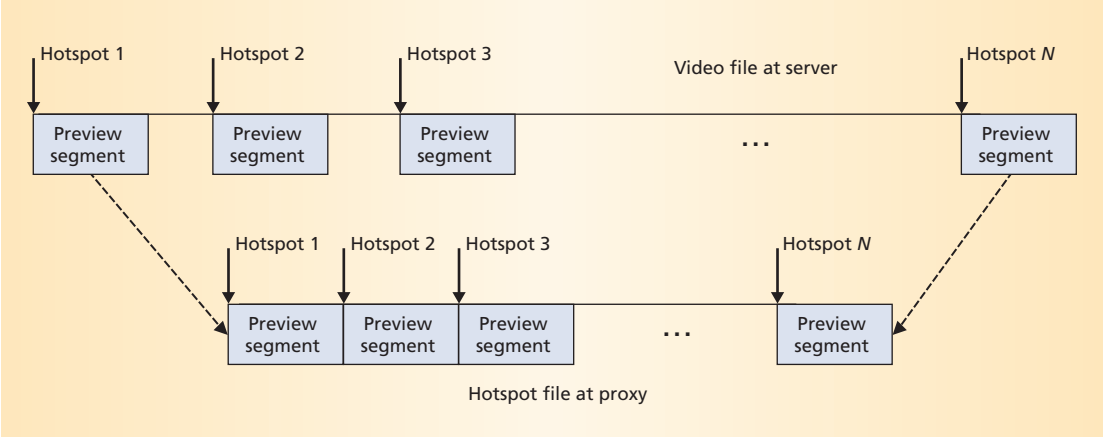

Figure 3. Hotspot caching of a video file at a proxy server. The hotspot segments may not be contiguous in the original video file, and each segment is a different size depending on the associated video scene.

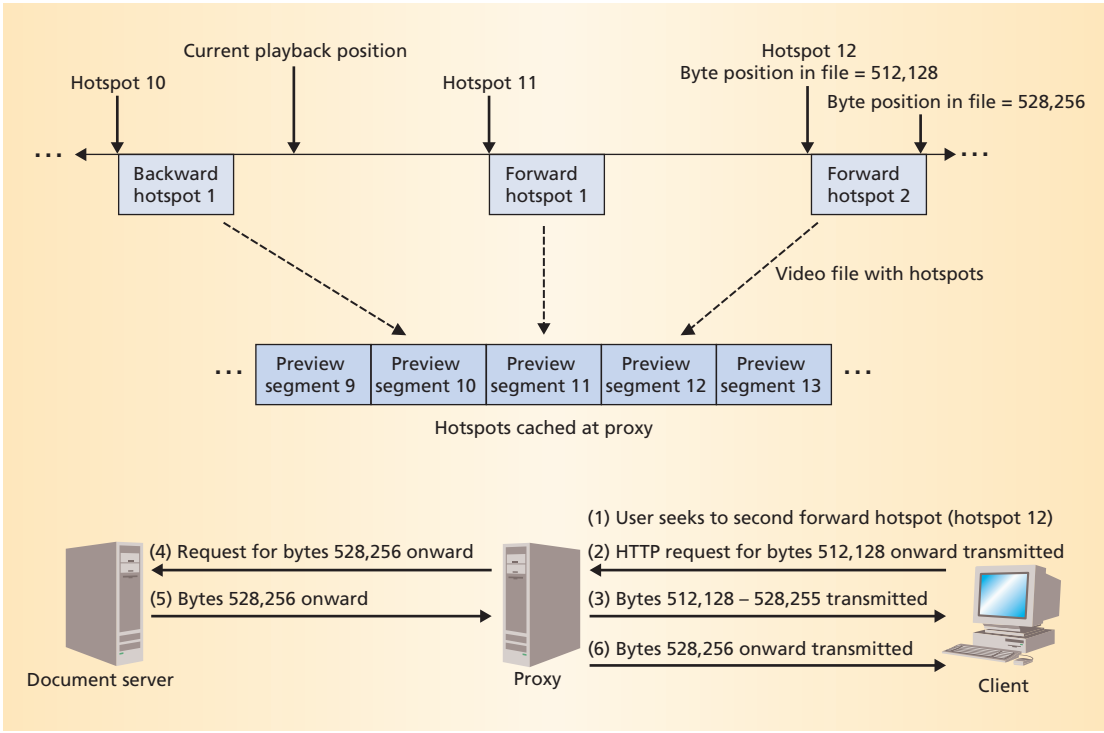

Figure 4. Search for a hotspot cached at the proxy. The user is currently playing a video segment between hotspots 10 and 11 and then seeking to hotspot 12. The user issues byte requests corresponding to hotspot 12. The proxy transmits the requested segment and fetches the subsequent data.

cient length to provide a preview of the hotspot's content. Hotspot segments do not need to be distributed at a uniform interval or be of equal size. The media authors define the distribution of hotspot segments. When a client uses a proxy to request a video, the proxy sends the starting clips of hotspots from the requested video. Icons that let the user preview random hotspots can represent these clips.

The assigned proxy can wait until the client has viewed all but the last b_τ seconds of the hotspot buffer before prefetching the data. The value of b_τ depends on the proxy-server delay; prefetching should begin earlier for a slower proxy. Once the user has viewed all but b_τ seconds of the hotspot, the proxy server must start prefetching data beyond the end of the hotspot to avoid discontinuity in playout if the user continues to view the video after viewing the hotspot. In our simulation, we assumed that the preview never exceeds the length of the hotspot buffer, eliminating the need for prefetching.

Figure 4 describes the sequence of events required for a successful search for a hotspot cached at the proxy.

Storing hotspot segments

Proxy servers maintain two types of information: starting clips of the hotspot segments delivered to clients who request a multimedia connection, and the sequence of hotspot segments for an entire video file.

In one approach to storing these hotspot segments, the proxy holds them in a contiguous file. Because these segments may not be contiguous in the original video files, the proxy uses a translation table to map the segment's location in the cached file to its location in the original file. For example, hotspot segment h is located at byte 1,050 in the original file. The segment is located at byte 400 in the proxy cache, which does not store the gaps between hotspot segments. Because this approach performs cache replacement at the file level, not the segment level, when the proxy makes a replacement on the corresponding video file, it completely replaces the file

Table 1. Parameters of experimental proxy model.

Parameter	Definition
C	Cache size (Kbytes).
B	Block size (Kbytes). Each block is 0.5 second long.
H	Hotspot spacing—the distance between successive hotspots (number of blocks). The default value is 600 blocks, corresponding to a five-minute interval.
S	Hotspot size (number of blocks). The default value is 60 blocks, corresponding to 30 seconds.
α	Probability that the user, after previewing the hotspot, will jump to another hotspot.
β	Probability that the user will stop viewing the video after previewing the hotspot.
P	Probability that the user will decide to view the complete video after previewing the hotspot.

that contains the sequence of hotspot segments.

In the second approach to storing hotspots, the proxy stores each segment in an independent file, allowing finer-grained replacement. This independent-file approach increases the complexity of proxy servers because they must distinguish between multimedia-streaming request and ordinary requests and keep track of hotspot-segment information. We use the first approach for simplicity.

PERFORMANCE EVALUATION

We used a simulator that models a proxy cache to evaluate our hotspot-caching technique. Table 1 lists this experimental proxy model's parameters.

To generate a trace representing typical Internet video-file access, we viewed a one-month log of HTTP accesses to NASA's Kennedy Space Center Web server,[10] discarding all accesses to files other than MPEG videos. We used the log, which consisted of 28,619 accesses to 61 distinct video files over the one-month period, to generate a synthetic trace for 1,220 individual files. We assumed that the file durations were uniformly distributed over 3,600 (30 minutes) to 14,400 (two hours) blocks. This synthetic trace simulated both a conventional proxy server and a proxy server implementing the hotspot-caching technique. Both proxy servers implemented a least recently used caching scheme that treats each file, or its associated hotspots, as an atomic unit for caching purposes.

Performance metrics

We used the simulation to compute two performance metrics: the *hit rate*, the fraction of user requests the proxy services; and the *hit ratio*, a measure of server load that represents the ratio between the total number of user-requested bytes fetched from the proxy cache and the total number of user-requested bytes. These two metrics quantify the reasons for using a proxy server—reducing server load and decreasing the user's perceived response time.

Our simulation makes several assumptions about proxy caching:

- The user does not issue a single continuous playback request, but instead previews a video before viewing its entire content. During the preview, the user views a portion of the hotspot buffers, then either jumps to another hotspot for the same file, stops playing the current file, or continues and plays back the entire file. We used probabilities α, β, and p, respectively, for each of these decisions.
- On average, a user previews a segment that is half the size of the hotspot buffer.
- As a matter of policy, the proxy does not prefetch the video segment after a hotspot if the preview exceeds the hotspot's length. In other words, a user who previews the entire hotspot and then decides to continue watching must wait for the server to send the rest of the file.
- The MPEG media file frame rate is 30 frames per second, and the GOP is 15 frames, corresponding to a block of 0.5-second duration.

Our simulation compared complete video caching and hotspot caching for various configurations of user seek patterns, block size, and cache size.

Caching performance

For the range of parameters we examined, hotspot caching improves the user's perceived response time because the proxy serves more requests than the server. However, hotspot caching is not as effective as conventional caching techniques in reducing the server load—the total volume of video the server plays. Even for popular video files, the proxy caches only a fraction of the file while the server must supply the remaining portion.

In our simulation, the proxy either caches all of the hotspots or none of them. Thus, the average number of seeks within the same file has no bearing on the percentage of requests that result in a hit. Our studies confirmed that hotspot caching decreases the average response time and the number of short-duration server hits, but it can increase the server load because

caching smaller portions of the video files increases the number of lengthy playouts. This is in comparison with the complete caching of media at proxies. Small requests are deferred to proxies, but servers manage long playouts. In complete caching, proxies serve long playouts, but this requires significantly more buffer space than hotspot caching. This finding was apparent in our synthetic trace, in which 45 percent of the accesses were for 10 percent of the files.

Figure 5 illustrates the effect of changing the proportion of cached video data. The hit rate is extremely sensitive to the hotspot buffer size. Increasing the hotspot size reduces the number of files the proxy can cache, thus increasing the percentage of requests fetched from the server rather than the proxy. Decreasing the hotspot buffer size dramatically decreases the average latency of the system response, as more requests begin playback from the proxy rather than the server. However, reducing the hotspot size increases the chance that the buffer will fail to provide a satisfactory preview of the file, and is likely to produce a server hit when the user decides to continue playback. The hotspot buffer size should be proportional to the network delay. In wide area networks, delays are long, so the proxy servers should have larger buffers to avoid decreasing the hit ratio.

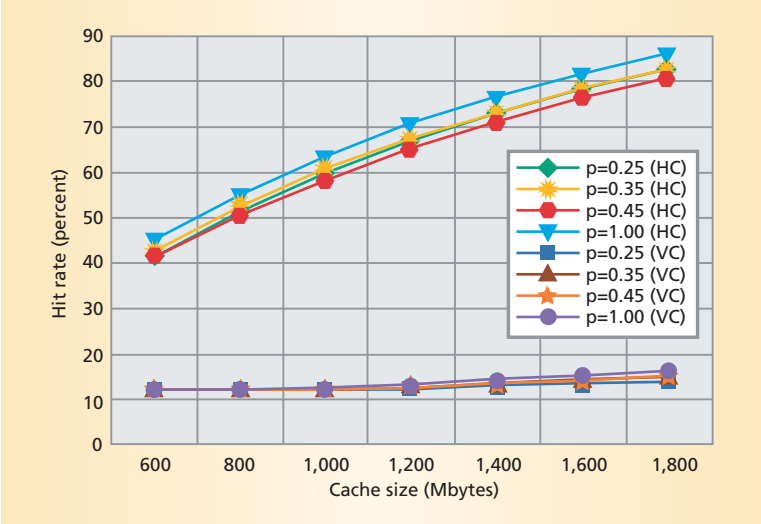

Figure 5. Hit rate of complete video caching versus hotspot caching. Increasing the hotspot size reduces the number of cached files and increases the percentage of requests fetched from the proxy rather than the server. HC, hotspot caching; VC, video caching.

Proxy servers support interaction in multimedia streaming applications by allowing users to preview hotspots before playing out a video's entire content. Caching only hotspot segments improves the proxy server function because these segments require significantly less buffer space than a complete video file. Storing each hotspot segment independently as a single file, rather than collecting all hotspot segments into one file, allows finer granularity for the cache-replacement policy. Because hotspot caching reduces the number of short requests sent to multimedia servers, the servers can devote more connections to continuous playouts. This approach improves storage retrieval efficiency and scalability because a smaller number of requests for lengthier objects is less demanding than a large number of short requests. Further, the proxy servers' physical proximity to clients reduces the response time to a random seek for previewing a video. Based on this study's results, we anticipate the use of proxy servers for quality-of-service management of multimedia document delivery, both in wired and wireless networks. Their role in wireless networks is increasingly prominent because they can provide quality adaptation for multimedia data transmission in the presence of rapidly varying, and often limited, network resources.

References

1. M. Abrams et al., "Caching Proxies: Limitations and Potentials," *Proc. 4th Int'l World Wide Web Conf.*, Elsevier Science, Amsterdam, Dec. 1995, pp. 119-133.
2. J. Shim, P. Scheuermann, and R. Vingralek, "Proxy Cache Algorithms: Design, Implementation, and Performance," *IEEE Trans. Knowledge and Data Eng.*, July/Aug. 1999, pp. 549-562.
3. S. Acharya, "Techniques for Improving Multimedia Communication over Wide Area Networks," doctoral dissertation, Dept. Electrical Eng., Cornell Univ., Ithaca, N.Y., Jan. 1999.
4. R. Rejaie et al., "Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet," *Proc. IEEE Infocom 2000*, IEEE Press, Piscataway, N.J., 2000, pp. 980-989.
5. K.L. Wu, P.S. Yu, and J.L. Wolf, "Segment-Based Proxy Caching of Multimedia Streams," *Proc. 10th Int'l World Wide Web Conf.*, Elsevier Science, Amsterdam, 2001, pp. 36-44.
6. S. Sen, J. Rexford, and D. Towsley, "Proxy Prefix Caching for Multimedia Streams," *Proc. IEEE Infocom 99*, IEEE Press, Piscataway, N.J., 1999, pp. 1310-1319.
7. Z. Zhang et al., "Video Staging: A Proxy-Server-Based Approach to End-to-End Video Delivery over Wide-Area Networks," *IEEE/ACM Trans. Networking*, Aug. 2000, pp. 429-442.
8. Microsoft, Streaming Methods: Web Server vs. Streaming Media Server; http://www.microsoft.com/windows/windowsmedia/en/compare/webservvstreamserv.asp (current Aug. 2001).

9. Hypertext Transfer Protocol—HTTP/1.1, RFC 2616, June 1999.
10. NASA-HTTP; http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html (current Aug. 2001).

Husni Fahmi *is a PhD candidate at the School of Electrical and Computer Engineering, Purdue University. His research interests include multimedia networking, quality-of-service management, traffic scheduling, proxy servers, and streaming multimedia. Fahmi received an MS in electrical engineering from Purdue University. Contact him at fahmi@ecn.purdue.edu.*

Mudassir Latif *is a graduate student at the School of Electrical and Computer Engineering, Purdue University. His research interests include proxy servers for streaming media. Latif received a BS in electronic engineering from the Ghulam Ishaq Khan Institute, Pakistan. Contact him at latifm@ecn.purdue.edu.*

Sahra Sedigh-Ali *is a PhD candidate at the School of Electrical and Computer Engineering, Purdue University. Her research interests include software testing, quality management, and component-based software development. Sedigh-Ali received an MS in electrical engineering from Purdue University. Contact her at sedigh@ecn.purdue.edu.*

Arif Ghafoor *is a professor at the School of Electrical and Computer Engineering, Purdue University. His research interests include multimedia information systems, database security, and distributed computing. Ghafoor received a PhD in electrical engineering from Columbia University. He is an IEEE Fellow. Contact him at ghafoor@ecn.purdue.edu.*

Peiya Liu *is a project manager and senior member of the technical staff of the Multimedia Documentation Program, Siemens Corporate Research. His research interests include multimedia standards, training, and education, SGML/XML-based markup technologies, and multimedia applications in industrial environments. Liu received a PhD in computer science from the University of Texas, Austin. Contact him at pliu@scr.siemens.com.*

Liang H. Hsu *is a Distinguished Member of the technical staff and head of the Multimedia Documentation Program, Siemens Corporate Research. His research interests include conversion of legacy documents into SGML/XML, SGML/XML-based document composition and hyperlinking for complex products, multimedia training, document browsing, and multimedia navigation support for service-related applications. Hsu received an MS in computer science from the University of Pittsburgh. Contact him at lhsu@scr.siemens.com.*