IEICE TRANS. COMMUN., VOL.E86–B, NO.6 JUNE 2003

PAPER *Special Issue on Content Delivery Networks*

Content Delivery Network Architecture for Mobile Streaming Service Enabled by SMIL Modification

Takeshi YOSHIMURA[†a)], ***Regular Member***, **Yoshifumi YONEMOTO**[†], ***Nonmember***, **Tomoyuki OHYA**[†], **Minoru ETOH**[†], ***Regular Members***, ***and*** **Susie WEE**[††], ***Nonmember***

SUMMARY In this paper, we present a CDN (Content Delivery Network) architecture for mobile streaming service in which content segmentation, request routing, pre-fetch scheduling, and session handoff are controlled by SMIL (Synchronized Multimedia Integration Language) modification. In this architecture, mobile clients simply follow modified SMIL files downloaded from a portal server; these modifications enable multimedia content to be delivered to the mobile clients from the *best* surrogates in the CDN. The key components of this architecture are 1) content segmentation with SMIL modification, 2) on-demand rewriting of URLs in SMIL, 3) pre-fetch scheduling based on timing information derived from SMIL, and 4) SMIL updates by SOAP (Simple Object Access Protocol) messaging for session handoffs due to client mobility. This architecture enhances streaming media quality for mobile clients while utilizing network resources efficiently and supporting client mobility in an integrated and practical way. The current status of our prototype on a mobile QoS testbed "MOBIQ" is also reported in this paper.
key words: *CDN, mobile network, streaming media, SMIL*

1. Introduction

With the birth and growth of the Internet and high-speed access links, Internet users can enjoy large amounts of web content on the Internet. However, network congestion and server overload have become major concerns for content delivery. To mitigate these effects, CDNs (Content Delivery Networks) are attracting a great deal of attention. In CDNs, web content is distributed to cache servers located at the edge of the Internet close to clients, and the cache servers deliver content to clients if they cached the requested content beforehand. Since the cached content does not have to be delivered from the origin servers to clients, this leads to lower latency for users, better network resource utilization for network operators, and scalable service provisioning for content providers. At the moment, several commercial companies provide CDN service on worldwide scales [1].

In mobile environments, Internet access has become very popular as represented by i-mode [2] and WAP (wireless access protocol) [3]. In addition, IMT-2000 (International Mobile Telecommunications-2000) has begun in Japan [4], and users can now access the Internet at up to 384 kbit/s through mobile devices. In the near future, mobile CDNs will be needed with the increasing number of IMT-2000 users.

At the same time, multimedia streaming service is becoming popular over the Internet. Mobile networks are also expected to provide IP-based multimedia streaming service in addition to web access service. From this requirement, 3GPP (3rd Generation Partnership Project), a standardization body of IMT-2000, has defined "packet streaming service (PSS) specifications [5]" as a standard for streaming service over 3G mobile networks.

Streaming service, however, presents a lot of challenges for network engineers. Unlike TCP applications, streaming service requires a certain amount of bandwidth to ensure the bit-rate needed by each media stream and the strict delay variation (i.e., jitter) needed to avoid buffer underflow at streaming clients. For such streaming service, CDNs are a very effective solution to relieve network congestion and to keep jitter at tolerable levels. Several papers have already considered how to cache streaming content effectively. In [6]–[8], streaming content is encoded with layered coding techniques and a proxy cache manages the number of layers to be stored for each stream. [9] proposed a prefix caching scheme that stores only the beginning of the content to minimize storage while decreasing waiting time for playback. [10] also utilizes the idea of prefix caching and describes resource management issues on cache systems. Selective caching was proposed in [11] as a generalization of prefix caching by storing various segments of the stream, specifically those that are subject to causing buffer underflow at clients. In [12], streaming content is divided into variable-sized segments in order to improve caching efficiency. Content-aware segmentation and a cache system using video summarization were proposed in [13]. In this system, streaming content is segmented at key frames and users can start streaming from any segment while watching the key frames.

Although these segment caching technologies have potential for enhancing network resource utilization and caching efficiency, they do not describe how to manage the segmented content and how to specify the

Manuscript received September 9, 2002.
Manuscript revised January 18, 2003.
[†]The authors are with Multimedia Laboratories, NTT DoCoMo, Inc., Yokosuka-shi, 239-8536 Japan.
[††]The author is with Hewlett-Packard Laboratories, Palo Alto, CA, 94304-1126 USA.
a) E-mail: yoshi@spg.yrp.nttdocomo.co.jp

Amazon v. Audio Pod
US Patent 9,729,907
Amazon EX-1006

locations of the proxy caches. Especially in mobile networks, since network and radio link conditions are dynamically changed depending on the time and place, it is required to specify the best cache locations flexibly. In addition, the best cache locations could be changed during a streaming session when a mobile client moves from one location to another location. The mechanism to update the cache locations is necessary for a mobile streaming media CDN. In [14], a streaming CDN architecture that has content naming, content management, and redirection mechanisms is described, but it does not consider network dynamics and client mobility expected in mobile networks.

The main goal of our work is to design a mobile streaming media CDN (MSM-CDN) that enhances streaming media quality in environments including mobile, while utilizing network resources efficiently and supporting client mobility in an integrated and practical way. To achieve this, we propose a dynamic SMIL framework in which all of the CDN technologies including content segmentation, request routing, pre-fetch control, and session handoff, are controlled by SMIL (Synchronized Multimedia Integration Language) [15] modification. The reasons for focusing on SMIL are itemized below.

- Since SMIL includes content location information, client requests can flexibly be redirected to the best cache servers by replacing the original content locations by the cache locations selected based on the network and caching conditions.
- To improve caching efficiency, large streaming content is likely to be divided into several segments. In this case, SMIL can easily manage the timing and spatial relations among these segments.
- SMIL makes user access patterns more predictable because it provides timing information about when a client is likely to request each segment. This enables efficient pre-fetch control to cache servers.
- Since SMIL is based on XML (eXtensible Markup Language) [16], XML-related technologies can be applied to this system.
- 3GPP PSS adopts SMIL as the scene description language. Our dynamic SMIL framework follows the 3GPP-compliant specifications. This is important for deploying this CDN system over 3G mobile networks.
- Content providers can easily create SMIL files for describing streaming content because SMIL is a standard and popular scene description language.

Figure 1 shows the concept of the dynamic SMIL framework. In this system, content providers register their streaming content with SMIL files describing the layout and media synchronization about the content, and do not need to worry about the existence of cache servers. Mobile clients, on the other hand, download the modified SMIL files from a portal server in the CDN. By simply following the downloaded SMIL files, the clients can receive multimedia content from the best cache servers. Even when mobile clients move to another location, the clients can continue multimedia streaming from the best cache servers by updating the SMIL files. The key elements of this system are as follows.

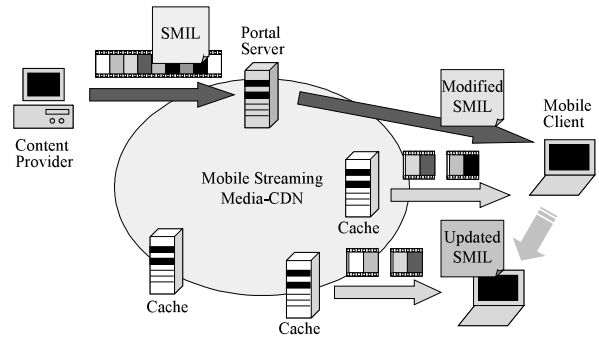

Fig. 1 Concept of dynamic SMIL framework.

1. The CDN provides content segmentation functionality to enable segment caching and to improve cache efficiency without complicating management of the segments. It accepts content registered from content providers and divides the content into several segments if it is too large. At the same time, it modifies the SMIL file corresponding to the segmented content to describe the timing relations among the segments.
2. On-demand URL rewriting described in [17] is applied to SMIL. That is, when clients request a SMIL file, the original content locations in the SMIL file are replaced by the cache server locations, and the modified SMIL file is sent to the clients. The clients, then, request session setup to the cache servers according to the modified SMIL file.
3. To minimize the waste of the network resources and cache memory, streaming segments are scheduled to be pre-fetched just before clients request the segments while the clients are receiving streaming segments. The request times are derived from the timing information described in SMIL files.
4. When mobile clients move from one location to another location, the clients request an update of the SMIL file. The portal server, then, updates the SMIL file to indicate the best cache server locations and creates the SOAP (Simple Object Access Protocol) [18] message that includes the update information from the previous SMIL file. After the clients reproduce the updated SMIL file from the SOAP message, they continue the streaming session according to the updated SMIL file.

The first three items enumerated above are effective in both wired and mobile networks though the last one is related only to mobile networks. Our proposed framework covers all of these features in an integrated

way by SMIL modification.

The rest of this paper is organized as follows. Section 2 explains our architecture overview. Then, each key element to support our CDN architecture, i.e., content segmentation, request routing, pre-fetch scheduling, and session handoff, is described respectively in Sect. 3 through Sect. 6. We report the current status of our prototyping work in Sect. 7, and finally, Sect. 8 concludes this paper.

2. Basic Architecture

The MSM-CDN architecture we propose is shown in Fig. 2. This architecture consists of the wired core network and the radio access network. The CDN overlays the core network, and the CDN control plane is in the middle of the CDN.

The CDN is composed of portal servers, content servers, surrogates, and a content location manager (CLM). The portal servers accept content registrations from content providers and provide content segmentation and SMIL modification functionality. The content servers are simple HTTP (Hypertext Transfer Protocol) [19] and RTSP (Real-Time Streaming Protocol) [20] servers and store a large number of streaming segments derived from the original media content. They do not have caching functionality, and work as HTTP servers for the surrogates and RTSP servers for mobile clients. The surrogates are the RTSP servers that temporarily store streaming segments and serve mobile clients as streaming servers via RTSP session control. The CLM is a server located in the CDN control plane, and it takes charge of managing content locations and scheduling pre-fetches.

The mobile clients can connect to the CDN via the radio access network. They are the RTSP clients that request multimedia content delivery according to the SMIL files sent from the portal server.

From the next section, we describe each key element of our architecture in more detail.

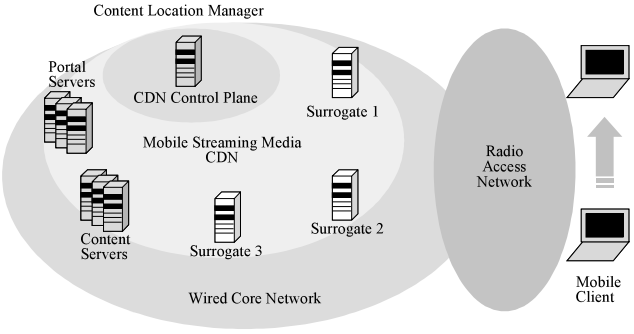

Fig. 2 Basic architecture of mobile streaming media CDN.

3. Content Segmentation

Content segmentation is effective especially for large streaming content. By dividing content into several segments, surrogates can store and remove the content at the granularity of the segments, which leads to efficient cache memory and network resource utilization. Prefix caching [9], selective caching [11], variable-sized segmentation [12], and content-aware segmentation [13] have been proposed as the intelligent variants of segment caching.

We also introduce content segmentation in our CDN architecture to improve cache efficiency. The problem with content segmentation, however, is to complicate the management of these segments. In other words, content segmentation increases the number of the files to be managed. And to realize such intelligent caching schemes proposed in [9]–[13], the servers in the CDN have to know the relations among the segments. Thus, how to manage the segments is an important issue.

For this purpose, we bind a SMIL file to the segments divided from a streaming content. Since SMIL provides the timing information for playing or displaying, it can easily manage the timing relations among the segments. By parsing the SMIL file, mobile clients and other servers can determine which segments are the next to the segments currently served and how long the segments will last.

The procedure of content segmentation is shown in Fig. 3. First, content providers register their own streaming content with the portal server. Here, the content providers also attach the SMIL files describing the layout and media synchronization about their streaming content. By registering content with SMIL files, content providers can specify how the content is displayed at mobile clients. If the size of the registered content is larger than a certain threshold, the portal server divides the content into smaller segments. At the same time, the portal server modifies the SMIL file attached to the content to describe the timing relations among these segments. Then, it sends the segments to a content server and requires it to store them.

Then, let us see an example of SMIL modification. Here, there is a streaming content whose length is sixty minutes, and its SMIL file is shown in Fig. 4. The SMIL

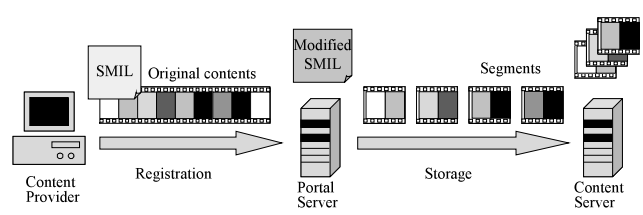

Fig. 3 Content segmentation and SMIL modification at portal server.

shows that content names are "content-A.mp4" for audio content and "content-V.mp4" for video content, and their durations are sixty minutes.

We assume the content is divided into three segments, and each segment is ten, twenty and thirty minutes long, respectively. In this case, the SMIL file is modified at the portal server as shown in Fig. 5. According to the modified SMIL file, these segments are renamed as "content-A-{1,2,3}.mp4" for audio segments and "content-V-{1,2,3}.mp4" for video segments, and the durations of the segments are ten, twenty, and thirty minutes long from the top segments. Since the part located between <seq> and </seq> indicates that these segments are sequentially played and displayed from the top segments, the timing relation among them is evident.

In this way, SMIL files are modified along with content segmentation, and the modified SMIL files indicate the names, durations, and relations of the segments.

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN"
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <region id="v" top="5" left="5" width="180" height="180"/>
    </layout>
  </head>
  <body>
    <par dur="60min">
      <audio src="rtsp://content-server/content-A.mp4" />
      <video src="rtsp://content-server/content-V.mp4" region="v" />
    </par>
  </body>
</smil>
```

Fig. 4 Example of original SMIL file.

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN"
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <region id="v" top="5" left="5" width="180" height="180"/>
    </layout>
  </head>
  <body>
    <seq>
      <par dur="10min">
        <audio src="rtsp://content-server/content-A-1.mp4" />
        <video src="rtsp://content-server/content-V-1.mp4" region="v" />
      </par>
      <par dur="20min">
        <audio src="rtsp://content-server/content-A-2.mp4" />
        <video src="rtsp://content-server/content-V-2.mp4" region="v" />
      </par>
      <par dur="30min">
        <audio src="rtsp://content-server/content-A-3.mp4" />
        <video src="rtsp://content-server/content-V-3.mp4" region="v" />
      </par>
    </seq>
  </body>
</smil>
```

Fig. 5 Modified SMIL file after content segmentation.

4. Request Routing

Request routing is the most important technology for CDN. Several request routing methods have already been proposed [17]. One popular method is DNS-based request routing. However, its resolution is domain level and, therefore, fine-grain request routing at object level is difficult. Another method is using RTSP REDIRECT. Although it's simple, the REDIRECT message has to be issued from original content servers to mobile clients each time the clients request a streaming segment, and this leads to session setup latency.

In our CDN architecture, we apply on-demand URL rewriting [17] to SMIL. Request routing is done by modifying SMIL files at the portal server when mobile clients request the SMIL files. In other words, the portal server replaces the content locations in SMIL files by the surrogate locations, and returns the modified SMIL files to mobile clients. The advantages of this request routing are:

- Request routing is done before clients request a streaming session by RTSP. This enables faster prefetching described in the next section.
- There is no extra procedure such as DNS-based request routing or RTSP REDIRECT method to set up a streaming session between clients and surrogates. This leads to low latency of session setup.

Figure 6 shows the request routing procedure by SMIL modification.

When a mobile client requests a SMIL file to start multimedia content streaming (1), the portal server reads the SMIL file corresponding to the streaming content stored in its memory. Then, the portal server queries CLM about the locations of the segments described in the SMIL file (2). At this time, the portal server notifies the estimated time when each segment will be requested, by parsing the timing information in the SMIL file. CLM utilizes this information for prefetching control described later. The client location (IP address) is also notified to CLM.

CLM selects the best surrogates for each segment

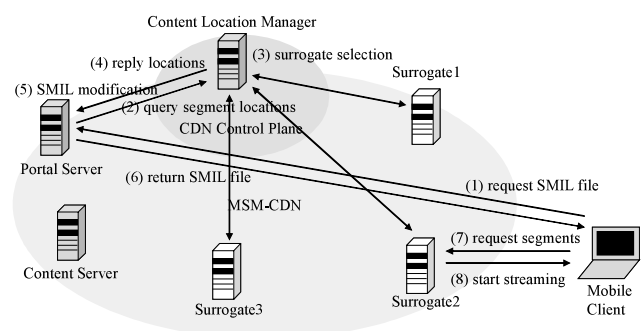

Fig. 6 Request routing procedure by SMIL modification.

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN"
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <region id="v" top="5" left="5" width="180" height="180"/>
    </layout>
  </head>
  <body>
    <seq>
      <par dur="10min">
        <audio src="rtsp://surrogate2/content-server/content-A-1.mp4" />
        <video src="rtsp://surrogate2/content-server/content-V-1.mp4 region="v" />
      </par>
      <par dur="20min">
        <audio src="rtsp://surrogate2/content-server/content-A-2.mp4" />
        <video src="rtsp://surrogate2/content-server/content-V-2.mp4" region="v" />
      </par>
      <par dur="30min">
        <audio src="rtsp://surrogate3/content-server/content-A-3.mp4" />
        <video src="rtsp://surrogate3/content-server/content-V-3.mp4" region="v" />
      </par>
    </seq>
  </body>
</smil>
```

Fig. 7 Modified SMIL file to indicate surrogate locations.

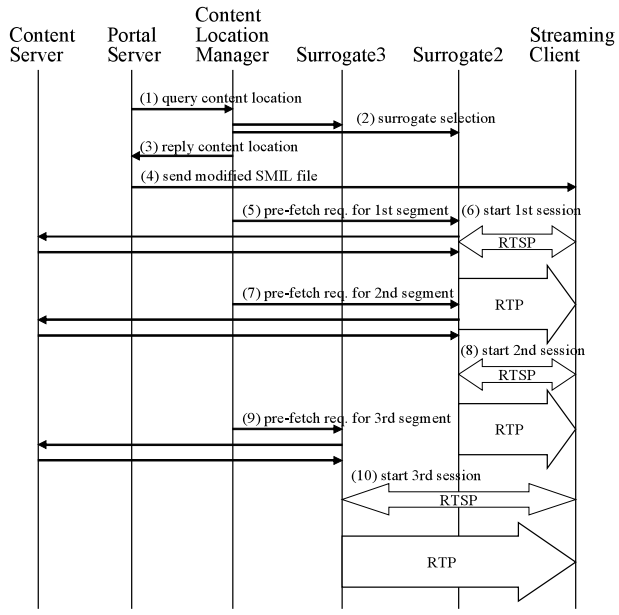

Fig. 8 Sequence of pre-fetching scheduled by content location manager.

based on the client location, network condition, caching status, surrogate load, and so on (3). In this paper, we do not specify how to select the surrogate locations. After the selection, it responds the locations of the selected surrogates to the portal server (4).

The portal server, then, replaces the original content locations in the SMIL file by the surrogate locations (5). Figure 7 shows an example of the SMIL file whose content locations are modified to indicate the surrogate locations. The original URLs of the segments shown in Fig. 5 are replaced by the surrogate URLs. The locations of the segments could be different. In the SMIL file shown in Fig. 7, surrogate2 is selected for the top four segments, while surrogate3 is selected for the last two segments.

After the modification of the SMIL file, the portal server returns the modified SMIL file to the mobile client (6). The mobile client receives the SMIL file and requests streaming sessions by RTSP according to the SMIL file (7). The surrogate requested to start the streaming session delivers RTP (Real-time Transport Protocol) [21] media packets that include audio and video payloads to the mobile client (8).

As long as the mobile client follows the SMIL file downloaded from the portal server, it always requests session setup to the surrogates selected by CLM.

5. Pre-Fetch Scheduling

To smoothly start streaming sessions, the surrogates have to pre-fetch the segments before the clients request the segments. As described in Sect. 3, the modified SMIL files provide the timing relations among the segments. By utilizing this information, the time when the clients will request each segment can be estimated. Based on the estimated times, pre-fetching is scheduled to be finished before the clients request the segments.

Figure 8 shows the sequence of segment pre-fetching. In our CDN architecture, CLM takes charge of pre-fetch scheduling.

Session ID			123456789		
Client IP address			10.20.30.40		
Start time			12:34		
Pre-fetch status			ACTIVE		
#	Time	Dur.	Seg. Name	Orig. Loc.	Surrogate
1	0 m	10 m	content-A-1.mp4	content-server	surrogate2
2	0 m	10 m	content-V.1.mp4	content-server	surrogate2
3	10 m	20 m	content-A-2.mp4	content-server	surrogate2
4	10 m	20 m	content-V-2.mp4	content-server	surrogate2
5	30 m	30 m	content-A-3.mp4	content-server	surrogate3
6	30 m	30 m	content-V-3.mp4	content-server	surrogate3

Fig. 9 Pre-fetch scheduling table.

When the portal server queries CLM about segment locations, it also notifies the timing information for the segments derived from the SMIL file (1). After the surrogate selection is done based on the client location, network condition, caching status, and surrogate load (2), it replies the locations of the selected surrogates (3).

At the same time, CLM creates pre-fetch scheduling table as shown in Fig. 9. The table is composed of session ID, client IP address, start time, pre-fetch status, and pre-fetch information for each segment. Pre-fetch information includes estimated request times, segment durations, segment names, original locations, and surrogates selected to serve the segment.

Based on this table, CLM starts pre-fetching. As shown in Fig. 8, the first audio and video segments are pre-fetched (5) while the client is requesting the segments (6). Before the client finishes receiving the whole

first segments, the second segments are pre-fetched (7). The third segments are pre-fetched to surrogate3 while the client is receiving the second segments from surrogate2 (9).

If the client pauses the streaming session, the surrogate serving the client notifies CLM of the pause. CLM, then, changes the pre-fetch status in the pre-fetch scheduling table to PAUSE and stops pre-fetch requests to surrogates. When the client restarts the streaming, the surrogate notifies CLM of the restart. CLM recalculates the start time and the request times for each segment, and changes the pre-fetch status to ACTIVE. The same procedure is taken when the client skips or goes back to the segments.

6. Streaming Session Handoff

When a mobile client moves from one location to another location, the best surrogate serving it could be changed. One way to redirect the client's request to the new surrogate is sending RTSP REDIRECT from the surrogate currently serving the client. In this case, however, the client will request the next segments from the obsolete surrogate if it is still guided by the SMIL file downloaded when starting the multimedia content delivery. Another way is client-driven session handoff. That is, the client disconnects the current surrogate and requests the new surrogate for a session setup. The problem with this is how the client gets to know the best surrogate locations for itself.

To enable session handoff for all of the streaming segments, the mechanism to update SMIL files is necessary when the best surrogates are changed due to client mobility. In our CDN architecture, a SOAP message is used for this purpose. When a mobile client moves to another location and the best surrogates are changed, the portal server sends the SOAP message that includes the locations of the new surrogates serving the client. The client, then, updates its SMIL file and continues to receive the streaming segments from the new surrogates by following the updated SMIL file.

The advantages of sending SOAP messages to update SMIL files are:

- The SOAP messages can explicitly notify the update of the current SMIL file.
- If the SMIL file to be updated is elaborate for describing the layout and if its size is very large, sending the updated SMIL file itself consumes network resources. In that case, the SOAP message that tells only the update information of the surrogate locations is more cost-effective.
- Both SMIL and SOAP are based on XML. The same modules to parse XML documents can be used at mobile clients. In addition, most of the mobile clients are expected to be SOAP-capable in the near future. This requires only small modification on the clients.

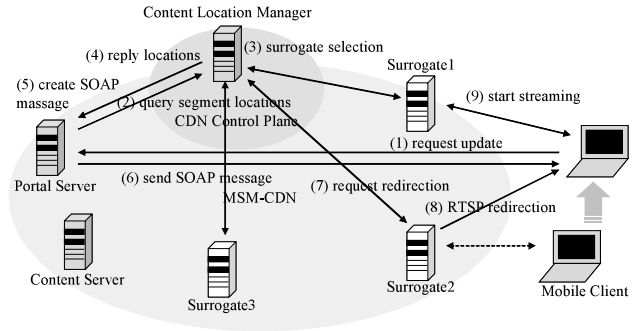

Fig. 10 SMIL update procedure when client moves.

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP-ENV:encodingStyle=
   "http://schemas.xmlsoap.org/soap/encoding/">
 <SOAP-ENV:Body>
   <m:update  xmlns:m="http://docomo.ne.jp/dynamicSMIL">
    <oldSrc>/smil/body/seq/par[2]/audio/@src</oldSrc>
    <newSrc>rtsp://surrogate1/content-server/content-A-2.mp4</newSrc>
    <oldSrc>/smil/body/seq/par[2]/video/@src</oldSrc>
    <newSrc>rtsp://surrogate1/content-server/content-V-2.mp4</newSrc>
    <oldSrc>/smil/body/seq/par[3]/audio/@src</oldSrc>
    <newSrc>rtsp://surrogate1/content-server/content-A-3.mp4</newSrc>
    <oldSrc>/smil/body/seq/par[3]/video/@src</oldSrc>
    <newSrc>rtsp://surrogate1/content-server/content-V-3.mp4</newSrc>
   </m:update>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 11 Example of SOAP message to update SMIL file.

Figure 10 shows the procedure of SMIL update when a mobile client moves. In this figure, we assume the best surrogates are changed from surrogate2 to surrogate1.

When a mobile client moves from one location to another location and detects its movement, the mobile client requests the portal server to update the SMIL file by the SOAP request message (1). After receiving the update request, the portal server requests CLM to recalculate the locations of the best surrogates (2). As similar to the case described in Sect. 4, CLM selects the best surrogates for each segment based on the client location, network condition, caching status, and surrogate load (3). After the selection, CLM responds the locations of the selected surrogates to the portal server (4).

The portal server, then, creates the SOAP response message to inform the client of the new surrogate locations for each segment (5). An example of the SOAP response is shown in Fig. 11. The parts located between <oldSrc> and </oldSrc> indicate the locations to be replaced in XPath (XML Path Language) [22] description. The next parts located between <newSrc> and </newSrc> indicate the URLs of the new surrogates selected by CLM. In Fig. 11, the surrogates serving the second and the third audio/video

segments are changed to surrogate1.

The SOAP message created in this way is transmitted to the client (6). From this SOAP message and the current SMIL file, the client creates the new SMIL file that includes the new surrogate locations. The client continues to request the subsequent segments from the surrogates described in the updated SMIL file.

While CLM responds the segment locations to the portal server (4), CLM also recalculates pre-fetch schedule and rewrites pre-fetch scheduling table. In addition, it may request the surrogate currently serving the client (surrogate2) to redirect the client's request to the newly selected surrogate (surrogate1) (7). Then, surrogate2 issues RTSP REDIRECT method to the client (8), and the client requests surrogate1 to send the segments currently being transmitted from surrogate2 (9). Note that surrogate2 does not need to issue RTSP REDIRECT methods for the subsequent segments because the client requests surrogate1 to deliver them directly according to the updated SMIL file.

This SMIL update mechanism may generate vast signaling traffic if a mobile client moves fast and frequently changes its location. Obviously there is tradeoff relation between accuracy of best surrogate locations and the cost of the signaling traffic. In this mechanism, the signaling cost can be flexibly controlled by a counter or timeout mechanism to prohibit the SMIL update for a while. And in an environment where multiple types of radio access networks (RANs) are connected, the change of the RAN type is a good indication for the SMIL update. That is, the best surrogate is expected to be changed with high probability when a mobile client changes its access network, for example, from 3G RAN to a wireless LAN. On the other hand, the best surrogate is expected to be the same as before when a mobile client merely changes its access point in wireless LANs. Some mechanisms to trigger the SMIL update from network to mobile clients may also be possible and more effective if the network knows the coverage area of each surrogate.

7. System Prototype

We have built the first version of MSM-CDN over our mobile QoS testbed called "MOBIQ [23]" as shown in Fig. 12. MOBIQ consists of Diffserv [24]-based IP core network and 3G RAN and IEEE802.11 wireless LAN. The core network is composed of FreeBSD routers with ALTQ [25] module, which provides functionality needed by Diffserv. The 3G RAN includes W-CDMA (Wideband Code Division Multiple Access) RAN emulator [26] and header compression nodes [27] to enable efficient IP packet delivery over 3G wireless links. We have also developed streaming servers, live steaming servers, and mobile clients. All of these streaming servers and clients are compliant with 3GPP-PSS standard.

The MSM-CDN overlay has been prototyped on this testbed. It includes the portal server, the content location manager(CLM), and the surrogates. At this moment, only the request routing procedure shown in Fig. 6 has been implemented. The CLM always selects the closest surrogate to mobile clients by requesting the surrogates to measure round trip times (RTTs) between mobile clients and themselves.

One of the features of this prototype is that all of the interfaces between the above CDN nodes are implemented by SOAP interfaces. Since SOAP is based on XML, these interfaces can be easily extended to include various security features including confidentiality, integrity, and access control. Another advantage of using SOAP interfaces is that content providers or other enterprises connecting to the Internet can easily utilize these interfaces and the CDN functions because SOAP is widely used in the Internet.

We have also developed live streaming splitters in our CDN prototype. In the live streaming case, there is no content to be cached on the surrogates. Instead of the surrogates, the splitters receive live streams from the live streaming servers and copy the streams to their clients. The same request routing procedure can be

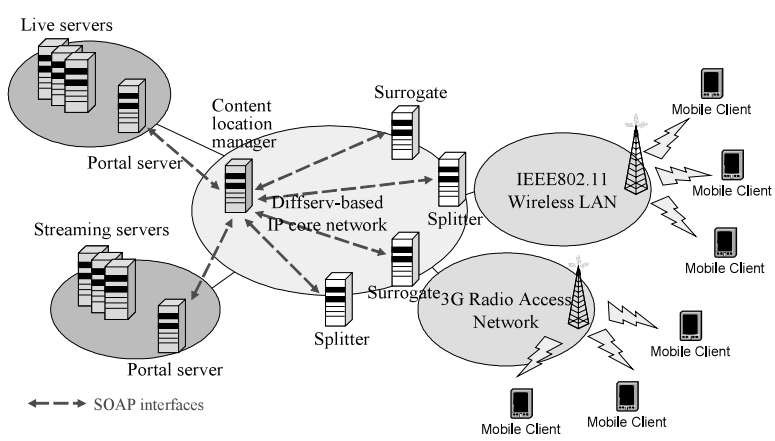

Fig. 12 Prototype of MSM-CDN on mobile QoS testbed.

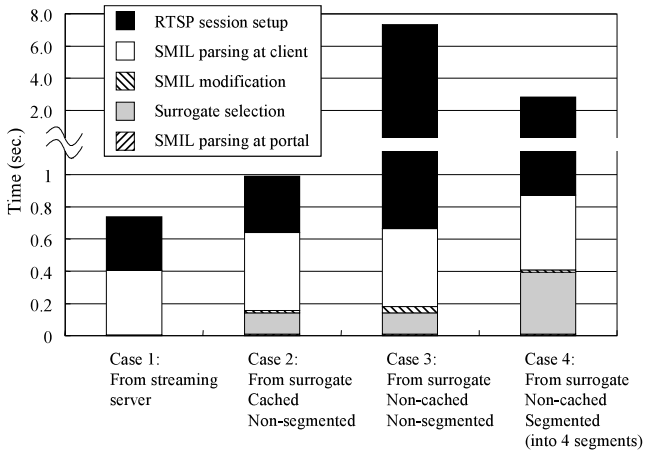

Fig. 13 Time needed for SMIL modification operation and RTSP session setup.

applied to the live streaming.

On this prototype, we can see better media quality with MSM-CDN than without it, when the bottle neck link in the core network is congested. However, mobile clients have to wait more for playback because of the SMIL modification operation including SMIL parsing/modification at the portal server and surrogate selection by CLM. Then, we measured the time needed for the SMIL modification operation as well as the time needed for SMIL parsing at clients and RTSP session setup in several test cases.

Figure 13 shows the results of the above measurement. The test case 1 is a usual streaming case without CDN. In this case, approximately 700 ms was necessary for parsing SMIL at clients and RTSP session setup. The case 2 is a streaming case from the closest surrogate that has already cached the requested content. Compared with the case 1, it took a little bit longer time because the SMIL modification operation was involved. The primary factor of this was the time for surrogate selection. However, we think the time is negligible because it was less than 200 ms. The case 3 is that the requested content whose size is 5.4 MB had not been cached. Here, the bandwidth of the bottle neck link in the core network was 10 Mbit/s. In this case, the time needed for RTSP session setup was nearly seven seconds because the surrogate could not respond RTSP DESCRIBE message until it finished downloading the requested content. On the other hand, the test case 4 where the content was divided into four segments shows that the time for session setup decreased by tolerable level due to shorter download time of the first segment. And our mobile clients could receive and play back the subsequent segments seamlessly. However, we can see slight increase in the time for surrogate selection. This is because the modified SMIL file became complicated and had more URLs in it by content segmentation. Therefore, segmentation of streaming content into unnecessarily many segments may cause bad effect on the waiting time for playback.

8. Conclusions

We presented a mobile streaming media (MSM)-CDN architecture in which all of the technologies related to CDN are enabled by SMIL modification. In this architecture, mobile clients simply follow the SMIL file downloaded from the portal server, and this leads to multimedia content delivery from the best surrogates in the CDN. The key components of this architecture are as follows.

1. To improve cache efficiency, content segmentation is provided along with SMIL file modification to describe the timing relations among the segments.
2. To redirect client requests to the best surrogates, the content locations in the SMIL file are modified to indicate the best surrogate locations.
3. Segment pre-fetching is scheduled just before clients request segments in order to minimize the waste of the network resources and cache memory. The request times are derived from SMIL files.
4. SOAP messages to update SMIL files are defined. This is necessary when mobile clients move and the best surrogates for them are changed.

The current status of our prototype was also reported. We have already built the MSM-CDN over our mobile QoS testbed called "MOBIQ." The MSM-CDN is completely compliant with 3GPP-PSS standard, and all of the CDN nodes have SOAP interfaces to control each other. We also showed that the time needed for the SMIL modification operation was negligible.

We are now investigating segmentation and resource management algorithms. These algorithms require choosing a media segment size for the content segmentation operation. An appropriately chosen segment size will consider client mobility, since each segment boundary provides an opportunity for a session handoff. Segment size also influences content use statistics, since knowledge that users only tend to view the first 30 seconds of a media clip can help the system efficiently allocate its storage resources. Resource management and caching algorithms can also benefit from knowledge of content popularity, since content that is likely to be accessed by other users should be left in cache storage whenever possible.

Interaction with the underlying QoS network is another research topic. Notice that the pre-fetch scheduling table gives a deadline for when the segment must be pre-fetched to the surrogate, however it does not specify a precise time for the pre-fetch to occur. This flexibility allows the CLM and surrogates to schedule the actual pre-fetch time while taking the underlying QoS base network resource usage into account.

We are also interested in personalized mobile

streaming. Since users have limited devices and time especially in mobile environments, it is important to deliver multimedia content dynamically and automatically summarized depending on user preferences and/or network conditions. Towards this, personalization of SMIL files [28] and MPEG-7 based personalization [29] have already been proposed. We are considering the integration of these personalization technologies into our MSM-CDN architecture.

References

[1] Akamai Technologies, Inc., http://www.akamai.com/
[2] NTT DoCoMo, Inc., i-mode, http://www.nttdocomo.co.jp/english/i/index.html
[3] WAP Forum, Wireless Access Protocol, http://www.wapforum.org
[4] NTT DoCoMo, Inc., FOMA (Freedom Of Mobile multimedia Access), http://foma.nttdocomo.co.jp/english/
[5] 3GPP TS 26.234 v 4.1.0, Transparent end-to-end packet switched streaming service (PSS); Protocol and codecs, Sept. 2001.
[6] R. Rejaie and J. Kangasharju, "Mocha: A quality adaptive multimedia proxy cache for Internet streaming," Proc. NOSSDAV 2001, 2001.
[7] R. Rejaie, H. Yu, M. Handley, and D. Estrin, "Multimedia proxy caching mechanism for quality adaptive streaming applications in the Internet," Proc. INFOCOM 2000, 2000.
[8] J. Kangasharju, F. Hartanto, M. Reisslein, and K.W. Ross, "Distributing layered encoded video through caches," Proc. INFOCOM 2001, 2001.
[9] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," Proc. INFOCOM'99, 1999.
[10] E. Bommaiah, K. Guo, M. Hofmann, and S. Paul, "Design and implementation of a caching system for streaming media over the Internet," Proc. Real-time Technology and Applications Symposium (RTAS 2000), 2000.
[11] Z. Miao and A. Ortega, "Proxy caching for efficient video services over the Internet," Proc. Packet Video Workshop 1999, 1999.
[12] K.-L. Wu, P.S. Yu, and J.L. Wolf, "Segment-based proxy caching of multimedia streams," Proc. 10th International World Wide Web Conference, May 2000.
[13] S.-J. Lee, W.-Y. Ma, and B. Shen, "An interactive video delivery and caching system using video summarization," Proc. WCW 2001, 2001.
[14] C.D. Cranor, M. Green, C. Kalmanek, D. Shur, S. Sibal, J.E. Van der Merwe, and C.J. Sreenan, "Enhanced streaming services in a content distribution network," IEEE Internet Computing, vol.5, no.4, pp.66–75, July–Aug. 2001.
[15] Synchronized Multimedia Integration Language (SMIL 2.0), W3C Recommendation, Aug. 2001. http://www.w3.org/TR/smil20/
[16] Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, Oct. 2000. http://www.w3.org/TR/REC-xml
[17] A. Barbir, B. Cain, F. Douglis, M. Green, M. Hofmann, R. Nair, D. Potter, and O. Spatscheck, "Known CDN request-routing mechanisms," IETF Internet-Draft, draft-cain-cdnp-known-request-routing-03.txt (work in progress), Nov. 2001.
[18] Simple Object Access Protocol (SOAP) 1.1, W3C Note, May 2000. http://www.w3.org/TR/SOAP/
[19] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol—HTTP/1.1," IETF RFC 2616, June 1999.
[20] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol," IETF RFC 2326, April 1998.
[21] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF RFC 1889, Jan. 1996.
[22] XML Path Language (XPath) Version 1.0, W3C Recommendation, Nov. 1999. http://www.w3.org/TR/xpath
[23] T. Yoshimura, T. Ohya, H. Matsuoka, and M. Etoh, "Design and implementation of mobile QoS testbed MOBIQ for multimedia delivery services," Proc. Packet Video Workshop 2002, April 2002.
[24] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," IETF RFC 2475, Dec. 1998.
[25] K. Cho, "Managing traffic with ALTQ," Proc. USENIX 1999, June 1999.
[26] H. Inamura, T. Ishikawa, and O. Takahashi, "Evaluation of TCP traffic over W-CDMA network," IPSJ Technical Report, MBL18-33, Sept. 2001.
[27] T. Yoshimura, T. Kawahara, T. Ohya, and M. Etoh, "Multiple-reference compression of RTP/UDP/IP headers for mobile multimedia communications," IEICE Trans. Fundamentals, vol.E85-A, no.7, pp.1491–1500, July 2002.
[28] R. Hjelsvold, S. Vdaygiri, and Y. Leaute, "Web-based personalization and management of interactive video," The 10th International World Wide Web Conference, May 2001.
[29] S. Sekiguchi, M. Etoh, K. Emura, W. Fujikawa, K. Masumitsu, and T. Echigo, "Video digest delivery using MPEG-7 media structure description and its authoring system," IEICE Technical Report, PRMU2001-92, Sept. 2001.

Takeshi Yoshimura received the B.E. and M.E. degrees from Department of Information and Communication Engineering at the University of Tokyo in 1997 and 1999, respectively. He joined NTT Mobile Communication Network, Inc. in 1999. He is currently a research engineer of Multimedia Laboratories, NTT DoCoMo, Inc. His current research interests include mobile streaming media technology and content delivery network architecture. He is a member of IEEE and IPSJ.

Yoshifumi Yonemoto received the B.E. and M.E. degrees in electrical and computer engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1990 and 1992, respectively. In 1992 he joined the Matsushita Electric Industrial Co., Ltd., and in 1999 he transferred to Matsushita Communication Industrial Co., Ltd. Since 2001, he is with NTT DoCoMo Multimedia Laboratories. His current research includes mobile multimedia content description and delivery architecture.

Tomoyuki Ohya received the B.E. and M.E. degrees in electronic engineering from Kyoto University in 1986 and 1988 respectively, and the M.S. degree in management of technology from Massachusetts Institute of Technology in 2000. He joined NTT in 1989, and since 1992 he has been working at NTT DoCoMo, Inc. He has been engaged in research and development of digital speech coding technologies for PDC and IMT-2000. Since 1996, he has been participated in the MPEG-4, ETSI and 3GPP international standardization activities, and mainly focusing on the research and development of digital signal processing technologies for the 3rd generation mobile communications systems. Currently his main research interest is on the multimedia signal processing and QoS architecture for 4th generation mobile communications networks. He is a member of IEEE. He received the Young Engineer Award from IEICE in 1995.

Minoru Etoh received the B.E. and M.S.E.E. degrees from Hiroshima University, Hiroshima, Japan, in 1983 and 1985, respectively. He received the Ph.D. degree from Osaka University, Osaka, Japan, in 1993. In 1985 he joined the Matsushita Electric Industrial. Co., Ltd. From 1988 to 1990 he worked for the ATR Communication Systems Research Laboratories, The Advanced Telecommunication Research Institute International, Kyoto. From 1991 to 1993, he was a Visiting Researcher at Osaka University. From 1994 to 1998, he was at the Central Research Laboratories of Matsushita Electric, and in the meantime he participated in the MPEG-4 standardization. He was also Adjunct Professor of Nara Institute of Science and Technology from 1996 to 2000, and Adjunct Lecturer of Osaka University from 1999 to 2001. In 2000, he moved NTT DoCoMo Multimedia Laboratories. He is now Director of Multimedia Signal Processing Laboratory. His current research includes media coding/processing, distributed concurrent software, pattern recognition, and networking. He received the 1995 Best Paper Award of IEICE Japan, the 14th Telecom System Technology Prize of the Telecommunications Advancement Foundation (1998), the 7th Sakai Commemorative Prize of IPSJ (1998), respectively. He is a member of IEEE and IPSJ.

Susie Wee received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology in 1991, 1991, and 1996, respectively. She performed her master's thesis work at the Jet Propulsion Laboratory in deep-space optical communication; she received a NASA Group Achievement award for this work. At MIT, her doctoral research was in the area of scalable video coding, video communications over broadcast channels, and high-definition television system design. In 1996, she joined HP Labs as a research scientist. Since 1999, she has been the R&D project manager of the HP Labs Streaming Media Systems Group. Her current research interests are in media delivery to diverse clients over packet networks; efficient video coding, transcoding, and compressed-domain processing algorithms; media-aware networking for next-generation wired and wireless networks; and mobile streaming media system design. In addition to working at HP Labs, she has been a consulting assistant professor at Stanford University since 1999 where she co-teaches a graduate-level course on digital video processing. She is currently an associate editor for the IEEE Transactions on Image Processing. She received Technology Review's Top 100 Young Investigators award in 2002.