STREAMOBILE: Pay-per-View Video Streaming to Mobile Devices Over the Internet *

Josep Domingo-Ferrer and Antoni Martínez-Ballesté
Universitat Rovira i Virgili,
Dept. of Computer Engineering and Maths,
Av. Països Catalans 26,
E-43007 Tarragona, Catalonia, Spain
e-mail {jdomingo,anmartin}@etse.urv.es

Abstract

As new mobile communication technologies are becoming broadly available, there is urgent pressure to populate them with services that provide returns for the huge investements made by telecom operators. Services around video transmission are expected to play a key role: news broadcasting, videoconferencing, movie channels, on-line gambling, etc. The goal of the STREAMOBILE project is to demonstrate an Internet pay-per-view distribution system toward GPRS and UMTS mobile devices. Copyright-protected contents are streamed to the customer and a micropayment scheme is used so that contents are not paid for in advance, but as they are being received by the customer. This paper presents an evolving prototype, whose client software has been designed in such a way that it can be easily migrated to 3G mobile devices.

Keywords: *Pay-per-view systems, Trust and privacy issues in mobile environments.*

1 Introduction

New generation mobile communications have attracted huge investments by telecom operators in the recent past. As a result, there is heavy pressure to populate those new technologies with services yielding financial returns proportional to the investments made. Unless such services are made available in the short term, telecom operators will be facing a hard time under the weight of debt, which may even cause some of them to disappear.

On the optimistic side, there is a growing market for multimedia content creation and delivery, which features news broadcasting, videoconferencing, movie channels, on-line gambling, etc. Thus, it seems only natural to use mobile communications as a new privileged outlet for digital content delivery in pay-per-view mode.

Current pay-per-view services are to be found basically in digital TV platforms. In most cases, pay-TV content is paid for by the customer before viewing it. The customer pays a minimum amount per month, which grants her access to a basic offer consisting of several TV channels. She can also watch an occasional extra movie or football match by making a credit card payment, always before viewing the content. The problem is that the customer pays for the whole piece of content: if she wants to stop watching anytime, she is losing part of her money. Due to the characteristics of mobile communications (relatively narrow bandwidth and reliability), advance payment of contents before delivery does not seem a good business model. Pay-per-view as contents are being streamed from the server to the customer seems an option better adapted to the technology. In Europe, two mobile technologies are likely to allow pay-per-view video services: GPRS (General Packet Radio Service, also known as 2.5 mobile generation) and UMTS (known as 3rd generation).

The system described in this paper allows video contents to be copyright-protected and paid for as they are being delivered and watched.

Section 2 addresses video transmission over the Internet. In Section 3, the use of micropayments in pay-per-view systems is discussed. Section 4 describes the copyright protection strategy used in STREAMOBILE. Section 5 describes the system components and their interaction. Section 6 contains some conclusions

*This work has been partly supported by the Spanish Ministry of Science and Technology and the European FEDER fund under project TIC2001-0633-C03-01 "STREAMOBILE".

Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02)
1529-4188/02 $17.00 © 2002 IEEE

and sketches future work.

2 Video streaming over the Internet

Digital video is greedy in terms of storage requirements. For example, a few minutes of high-quality video may need a few gigabytes to be stored. The MPEG formats[11] are one possibility to mitigate this problem through compression; MPEG-1 is very used for low-resolution videos, and MPEG-2 is common for DVDs and digital TV.

However, downloading entire movies from the Internet usually takes a long time even in the presence of compression. Besides, video downloading precludes the delivery of live broadcasting. Streaming is better than downloading in order to avoid long waiting times and enable transmission of live events. With streaming, the client can view the contents as soon as they are received. In fact, streaming has become the dominant way to deliver continuous media over the Internet.

In spite of technological advances and market acceptance, streaming over the Internet remains a challenging problem. The basic issue is that the streamed data must be presented to the viewer at a constant bit rate, whereas the network delivering the content has a different and fluctuating bandwidth. A common solution is to use an internal buffer which carries out an isochronous transmission. The buffer will not deliver any data to the video decoder until it has accumulated enough data to guarantee a constant bit rate. For things to run smoothly, the network bandwidth should be slightly greater than the video bit rate.

The streaming buffer will always keep some data until the delivery ends. The buffer size can be constant or can be dynamically increased or decreased, depending on the client player implementation. Thus, streaming occurs on the client side, whereas the server can deliver data using a protocol as simple as HTTP.

2.1 Bandwidth and quality

PAL quality television requires a bandwidth of about 25Mbps. With MPEG compression, this bit rate can be substantially decreased at the cost of some quality loss (less resolution, less frame details, less frames per second, etc.). Table 1 shows a comparison of different technologies in terms of bandwidth and qualities of video and audio[14]. It can be observed that the quality of video transmission reachable with current Internet or GPRS bandwidths is still poor. In spite of offering a narrower bandwidth, GPRS allows higher quality than standard fixed telephone lines. Recently reported experiments on video transmission over a GPRS network[2] yielded 15 frames per second at a quality of 176x144 pixels per frame.

Table 1. Bit rate and audio/video quality. BTN stands for Basic Telephone Network.

Bandwidth	Technology	Audio	Video
40Kbps	GPRS	ok	poor
56Kbps	BTN	poor	very poor
64Kbps or 128Kbps	IDSN	good	ok
256Kbps to 2Mbps	ADSL	very good	good

3 Pay per view

In the system proposed in this paper, the user pays as she is receiving the streamed content. The user only pays for the part of the contents she actually watches and can quit at any time. Thus, the amount of individual payments is likely to be very small. Also, payments will take place very frequently: a payment will be made every minute or for every certain number of received frames.

For such small and frequent payments, credit card transactions or electronic payment systems like SET are too expensive, too complicated or both. The operating costs of those systems are unaffordable for small amounts and can be split into communication and computation costs, the latter being caused by the use of complex cryptographic techniques such as digital signatures. Micropayments are electronic payment methods specifically designed to keep operating costs very low; in most micropayment systems in the literature, computational costs are dramatically reduced by replacing digital signatures with hash functions[12].

3.1 PayWord micropayments

Several micropayment schemes have been proposed in the literature, such as NetCard [1], μ-*i*KP [7], PayWord [13], PayTree [8] and spending programs [4]. Commercial implementations include IBM's MiniPay[10] and Digital's MilliCent [6]. Our system uses PayWord coins[13] for micropayments.

In PayWord, the customer (payer) establishes an account with a broker who gives her a certificate that contains the customer's identity, the broker's identity, the customer's public key, an expiration date and some other information. A hash chain is produced by the customer using a random root. When the customer wants to start making micropayments (to start receiving contents), she sends to the service provider a commitment to a chain. The commitment includes the ser-

Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02)
1529-4188/02 $17.00 © 2002 IEEE

vide provider's identity, the broker certificate, the last hash value (coin) of the chain, the current date, the length of the chain and some other information. In this scheme, the broker certificate certifies that the broker will redeem any payment that the customer makes before the expiration date, and the customer commitment authorizes the broker to pay the service provider. After that, micropayments in return for a certain amount of streamed content are made by the customer by revealing successive hash values (coins) of the chain to the service provider.

Note that a scenario is conceivable where a service provider operates not just one but a set of content servers. In this case, a customer can obtain contents from any content server in the set of those operated by the same content provider. It is also worth pointing out that coin generation is distinct from coin spending: a customer can generate a hash chain representing a set of coins, but these are only spent when they are sent to the service provider.

Technical details of our PayWord implementation are that hash chains are produced by a wallet software on the customer's side. This wallet software has been programmed in a JavaCard that the customer can carry and plug into the terminal she wants to receive the contents in. The one-way hash function being used is SHA-1[17], which yields 160-bit coins. Coin generation by the wallet software is exceedingly fast: thousands of hash values can be calculated in just one second. Verifying a coin is also easy and fast. The service provider uses little storage for coin verification, as he only needs to keep the last coin spent by each customer.

4 Copyright protection

Securing payment for the streamed content is not enough to ensure a sound business model. The intellectual property of the content should be protected in order to discourage unauthorized redistribution by the customer.

Two copy protection strategies are ordinarily considered:

Copy prevention Preventing unlawful copying requires some kind of hardware enforcement and has failed even in scenarios where players had special-purpose specifications, as is the case of DVD players[18]. In a setting with general-purpose mobile players like PDAs or GPRS phones equipped with color displays, it is completely unrealistic to rely on the player hardware to enforce copy prevention.

Copy detection Detection techniques appear as the main solution for protecting the copyright of content in electronic format. The idea here is not to prevent copying, but to track whether copying has taken place. Watermarking is the main technique for hiding copyright information in the content. The information being hidden or embedded is called *watermark*; the amount of watermark bits that can be embedded in a piece of content with a given watermarking scheme is called the capacity of the scheme. A special case of watermarking is fingerprinting whereby the copyright information embedded in the content, called *fingerprint*, can be viewed as a serial number identifying the buyer. Watermarking techniques must be robust and imperceptible.

Fingerprinting inherits the robustness and imperceptibility of the watermarking scheme used to embed fingeprints. However, in fingerprinting each buyers gets a slightly different copy of the content. Thus, collusion of a set of buyers to locate and destroy their fingerprints by comparing their copies is a feasible attack. *Collusion-secure* fingerprinting guarantees that, if up to c buyers collude, they will *not* be able to come up with a copy of the content that identifies none of them (c-secure fingerprinting).

In STREAMOBILE, the intellectual property of the streamed content is protected using our own watermarking and fingerprinting technology. Specifically, we use the following tools:

- *Oblivious robust watermarking.* Oblivious watermarking allows the embedded watermark to be recovered from the protected content without requiring the original (unprotected) content. This is especially convenient for storage-consuming and mass-distributed content such as video, since the original content can be "forgotten" by the verifier. This has a twofold advantage: 1) the verifier does not need to store the original content; 2) the content owner does not need to entrust the verifier with the original content, so that distributed verification by third-parties is feasible. In [15], we presented the first public-domain oblivious watermarking scheme for images which survived a broad range of manipulation attacks (including scaling and geometric distortion).
- *Collusion-secure fingerprinting.* The oblivious watermarking method is used to embed collusion-secure fingerprints. Rather than using the general c-secure fingerprinting code described in [3], we use the 3-secure fingerprints proposed in [16]. The latter are an extension of the 2-secure codes we proposed in [5] and they are dramatically shorter than

Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02)
1529-4188/02 $17.00 © 2002 IEEE

general codes in [3]; thus, much less embedding capacity is required. The fact that only collusions of up to 3 buyers can be survived is less problematic than it might seem, because collusions tend to be small due to their clandestine nature.

In this way, the algorithm for copyright protection in STREAMOBILE is as follows:

Algorithm 1 (Copyright protection) *1. If N is the size of the buyer community, generate a 3-secure collusion-secure fingerprinting code for N buyers, as defined in [16].*

2. *If a copy of the content is to be streamed to buyer i, embed the i-th codeword of the fingerprinting code into the content using the oblivious watermarking scheme [15].*

5 System design

The system presented here consists of a client part and a server part (see Figure 1). The client part implements the customer functionality. The client part has been implemented on a GPRS-integrated PDA. A client on a Java-enabled GPRS mobile phone is now being implemented. The two main components of the client are:

- The web browser. Windows Pocket PC includes Internet Explorer as a web browser. The 65K color TFT display and the integrated speakers allow a quality MPEG-1 video viewing.
- A wallet application, to generate, keep and deliver PayWord coins. The 240x320 PDA display allows video and wallet to be displayed at the same time. This wallet is stored in a 32MB Flash ROM memory.

The server part encapsulates the service provider and the content server functionalities. It consists of the following components:

- A web server (Apache web server). The web server hosts the main page of the Internet pay-per-view video services.
- A content server that delivers content only if the client pays for it. This server is implemented in Java. It can be configured to request a payment before delivering n Kbytes, where n is a parameter.
- A process called shop or service provider which manages payments between the customers and content servers. It is implemented in C.

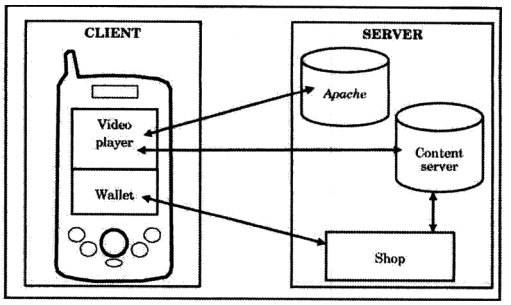

Figure 1. Mobile devices pay-per-view video system block diagram.

5.1 Description of a session

When the wallet program starts running, it sends a log-in message to the shop. This message includes the credit card number (so that the shop can pay off the customer's account whenever necessary).

The first time the customer has an empty wallet. In order to fill it with coins, the wallet sends a request to the shop telling it that the customer wants to create some coins. Once coin minting is allowed, the wallet program generates the coins. Then the wallet sends the last generated coin to the shop.

When the customer clicks on the link corresponding to a piece of content, say a movie, content streaming begins. The website hosted in the Apache server points to the content hosted in the content server. Then the protocol below is followed:

Protocol 1 (Pay per stream)

1. *The content server opens a socket connection to answer the HTTP request. Thanks to this socket, the server knows the client's IP address.*
2. *The content server asks the shop for a client coin.*
3. *The shop sends a payment request to the client located at the IP specified by the content server.*
4. *The wallet receives a payment request. If there are coins left in the wallet, the wallet sends the next coin to the shop.*
5. *The shop checks that the received coin is an authentic one. If payment succeeds, a pair of acknowledgment messages are sent: one to the wallet (which deletes the last used coin) and another to the content server (which sends the first n Kbytes of the movie to the client).*

Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02)
1529-4188/02 $17.00 © 2002 IEEE

6. *The payment request process is repeated until the movie has been entirely sent or until the client drops the connection.*

As mentioned in Section 2, there is a client buffer that holds data to keep the perceived bit rate constant. Some initial payments will probably be needed in order to fill the buffer with enough data to start viewing some movie frames; at this moment, the customer has paid but she has seen nothing of the movie. However, the difference between the amount of content paid for and the amount viewed is small as compared to the size of the entire content. For example, if the user quits watching a movie at minute 35, it might happen that she actually has paid for 37 movie minutes.

6 Conclusions and future work

The above described prototype has been tested and the server supports concurrent interaction with several clients. Preliminary laboratory tests confirm that this system is a good testbed to advance mobile video streaming sale, the ultimate goal of project STREAMOBILE of which this prototype is a deliverable. The next steps are:

- Conduct a large-scale test with real users and with the assistance of a telecom operator.
- Migrate the client implementation to UMTS technology as soon as it becomes available.

References

[1] R. Anderson, C. Manifavas and C. Sutherland, "NetCard - A practical electronic cash system", 1995. Available from author: Ross.Anderson@cl.cam.ac.uk

[2] R. M. Bernárdez, J. M. López, A. Farreras, J. L. García, J. Rufino and J. Lorente, "Video services on mobile networks of newer generation" (in Spanish), *Comunicaciones de Telefónica I+D*, no. 21, Jun. 2001.

[3] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data", *IEEE Trans. on Information Theory*, vol **IT-44**, no. 5, pp. 1897-1905, 1998.

[4] J. Domingo-Ferrer and J. Herrera-Joancomartí, "Spending programs: A tool for flexible micropayments", in *Information Security-ISW'99*, eds. M. Mambo and Y. Zheng, LNCS 1729, Springer-Verlag, pp. 1-13, 1999.

[5] J. Domingo-Ferrer and J. Herrera-Joancomartí, "Short collusion-secure fingerprints based on dual binary Hamming codes", *Electronics Letters*, vol. 36, no. 20, pp. 1697-1699, 2000.

[6] S. Glassman, M. Manasse, M. Abadi, P. Gauthier and P. Sobalvarro, "The Millicent protocol for inexpensive electronic commerce", in *World Wide Web Journal, 4th Intl. World Wide Web Conference Proceedings*, O'Reilly, pp. 603-618, 1995.

[7] R. Hauser, M. Steiner and M. Waidner, "Micropayments based on *i*KP", IBM Research Report 2791, presented also at SECURICOM'96. http://www.zurich.ibm.com/Technology/Security/publications/1996/HSW96.ps.gz

[8] C. Jutla and M. Yung, "PayTree: " Amortized-signature" for flexible micropayments", in *Second USENIX Workshop on Electronic Commerce*, Oakland CA, Nov. 1996.

[9] J. Lu, "Reactive and proactive approaches to media streaming: from scalable coding to content delivery networks", in *Proceedings of IEEE ITCC 2001*, Los Alamitos CA: IEEE Computer Society Press, pp. 5-9, 2001.

[10] MiniPay, http://www.minipay.com

[11] Moving Pictures Experts Group, http://www.mpeg.org

[12] R. Oppliger, *Security Technologies for the World Wide Web*, Norwood MA: Artech House, 1999.

[13] R. L. Rivest and Adi Shamir, "PayWord and MicroMint: two simple micropayment schemes", Technical Report, MIT LCS, Nov. 1995. http://theory.lcs.mit.edu/ rivest

[14] M. Scott, "High bit rate and low bit rate video streaming", http://www.indiana.edu/ ccumc/

[15] F. Sebé and J. Domingo-Ferrer, "Oblivious image watermarking robust against scaling and geometric distortions", in *Information Security*, eds. G. Davida and Y. Frankel, LNCS 2200, Berlin: Springer-Verlag, pp. 420-432, 2001.

[16] F. Sebé and J. Domingo-Ferrer, "Short 3-secure fingerprinting codes for copyright protection", manuscript, 2002.

[17] U. S. National Institute of Standards and FIPS PUB 180-1 Technology, *Secure Hash Standard*, 1995. http://csrc.ncsl.nist.gov/fips/fip180-1.txt

[18] `http://www.lemuria.org/DeCSS`

Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02)
1529-4188/02 $17.00 © 2002 **IEEE**