Hand-Over Video Cache Policy for Mobile Users

Damien Charlet, Pascal Chatonnay, and François Spies

Laboratoire d'Informatique de l'université de Franche-Comté

Abstract. Nowadays, wireless links are shared between clients and the available bandwidth fluctuates, so the video stream must be adaptive in order to optimize the transmitted quality. In this article, we present a tool called MoVie to distribute multimedia streaming on large wireless networks. The two main aspects of MoVie deal with network access and managing systems. The management of video streams is realized using video caches linked to a physical area. This article describes the policy used to take into account the mobility of the clients and the hierarchical aspect of the streams by preparing the neighboring caches to receive roaming clients.

Keywords: Multimedia streaming, mobile devices, distributed video caches

1 Introduction

The delivery of digital multimedia contents over networks has dramatically increased over the last decade. Satellites can broadcast digital video over a wide population and, in the near future, terrestrial wireless digital transmissions will emerge. However powerful these technologies may be, they are still deprived of interactivity and mobility. Video and news on demand are currently well identified services of transmitting video content to one device. But, taking into account mobility, heterogeneity, adaptation and migration is hard to integrate on such services.

The development of the bandwidth of the wireless networks will change the way people consume multimedia contents [1]. Three technologies are emerging: GPRS [2], UMTS [3] and WiFi [4]. Various types of devices can access these networks which induce a large heterogeneity at the end of the connection. The need of adaptability is real in order to make the video transfer possible in such conditions. Only one video quality transmission is not possible to achieve and specific techniques for variable video quality must be used in order to tackle these problems. Few solutions allow variable multimedia quality such as stream switching, scalable streaming or transcoding. This induces new strategies of storing and managing data to avoid redundancy.

The number of clients connected to large wireless networks will be very important. The system delivery must share these accesses between various components in order to avoid bottlenecks. There are currently two ways to realize this partition: replicating a monolithic server many times or developing a distributed

J.-B. Stefani, I. Demeure, and D. Hagimont (Eds.): DAIS 2003, LNCS 2893, pp. 179–186, 2003.

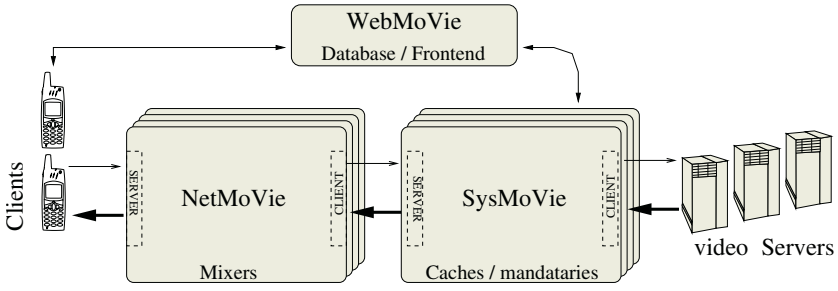

Fig. 1. The MoVie architecture.

system. The solution we retain is a distributed set of caches streaming multimedia sequences using specific management policies adapted to mobile clients.

The first section outlines our solution called MoVie and describes one of its components called SysMoVie, the distributed system layer. The next section introduces and evaluates by simulation one of the admission policies for distributed caches.

2 The MoVie Project

The architecture of the MoVie platform aims at delivering short-time multimedia streams on a large scale wireless network. MoVie must integrate the following characteristics in order to be efficient, fast and functional: being compatible with standard streaming clients (RTP/RTCP and RTSP); taking the mobility of clients and the stream migration into account; integrating a web-like query interface; being distributed and interoperable (ORB); managing the accessible video sequences from a hierarchical point of view.

We have structured this platform into modules in order to cover all the aspects of this new type of service: WebMoVie, NetMoVie and SysMoVie. WebMoVie represents the query interface of our platform. It is the entry point of clients where they are identified and decide which sequence they want to watch. NetMoVie [5] is the network level part. It integrates the RTP/RTCP protocol, receives different video sequence quality and selects the most adapted one depending on the current situation. SysMoVie [6] is the orchestration layer. It gathers ORB components and manages the cooperatives video caches. The strategy of video cache management is specific to the particular temporal data and the various quality levels of the same content. This is the part studied in this article and will therefore be described in more depth thereafter.

2.1 SysMoVie

The first task of the SysMoVie module is to enhance the quality of the transmission by bringing the sequences closer to the clients like web caches do. Thus, we have to deal with completely different type of content, large and continuous. The

phone network covering a very wide area and serving many users, a centralized server could neither cope with the huge number of clients nor have decent responsiveness. The application must be distributed. Moreover, the mobile phone network being already divided into cells, the territorial partitioning is simplified.

As SysMoVie is composed of a set of distributed caches, we have to deal with one of the main concerns of this type of architecture: referencing and localizing the contents in the system. Using a centralized database makes the system lose its attractiveness because of bottlenecks, while a distributed database brings greater latencies. Moreover, in a very large system where there may be lots of replicas, we need a fast decision process to get effective targets. In our system, this is achieved by taking advantage of the trading service framework of CORBA [7]. A trader has two complementary roles: it collects the sequences the caches attached to it declare and it answers queries about its content database. The sequences are described by the mean of static or dynamic properties. Traders also have the ability to be linked and therefore cooperate. This mechanism is described more in depth in [8].

Because of the architecture of the 3G frameworks and our choice to only use standards players, the only path between the user and the system during the transmission is the multimedia stream between the video player and the mixer. However we may need to collect more informations to put in place smarter policies, for example to prefetch sequences according to the user habits. We also need real time informations and the history of his localization to cope with mobility. To answer these needs, we have decided to use mobile agents. Each agent is a representative of one client inside the system and has its profile at its disposal. It updates in real time according to the actions of the user and can follow him along its way. It is placed nearest to the client, near the mixers, and can migrate from mixer to mixer. During the diffusion the mandatary performs three main actions: it manages the optimization of the quality streamed to its client, it cooperates with the NetMoVie components to prepare and supervise mixer swapping when the client moves and it moves each part of the streamed sequence closer. The mandatary is an adaptive component which copes with the context to make the system more suited to the current usage. It takes into account geographical information, clients' habits, streaming quality, and the responses of other components to evaluate the adequation between caches and mixers. The simultaneous work of all the mandataries allow the system to compose a matrix which is used to manage most of the operations done by caches.

2.2 The Caches

In continuous media caching, the documents are not as dynamic as web pages are. The contents of a video sequence neither change over time nor may be dynamically modified by the user's interaction. We can safely make the assumption that a document cached in the system will be valid whenever requested. Therefore we do not need to use a strict coherency algorithm for the sequences. However, we will see thereafter that we use a weak coherency algorithm for the pieces of data assigned to the management of prefetching.

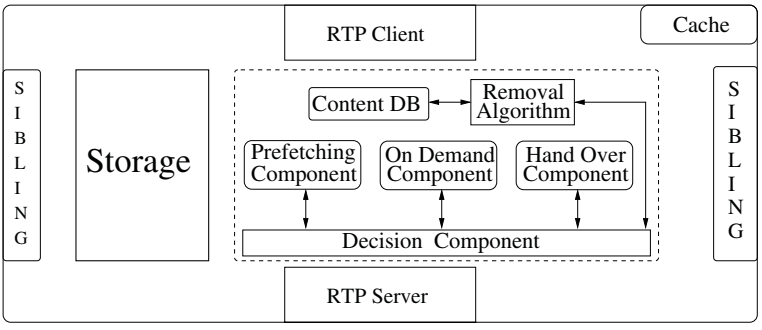

Fig. 2. Cache Architecture.

The video caches we introduce in SysMoVie implement three distinct mechanisms: video streaming, removal and insertion.

Video streaming to the mixer is performed by an RTP module integrated into the cache. Our architecture is a hierarchical client / server scheme. Each level is a server for its lower level, and a client for its superior level. The same server module can be found in the mixer on the user side. Like this, the cache is seen as a server for the mixer, and so is the mixer for the client. In the same way the RTP module has a client side used by the caches and the mixers. The caches can communicate with the exterior servers and they look like usual clients. A complete description of this module may be found in [5].

Cache removal management is a well-known, widely-studied domain in traditional caches. Lots of algorithms try to optimize the renewal of the cache contents, from traditional LRU, LFU or Size to more elaborate ones like GDSP, GDSF or Hybrid [9]. However, continuous media cannot be managed using these traditional techniques because of the huge size of each document and its transmission scheme. The size gives a grain much too large to be efficient, and the multimedia sequences cannot be completely downloaded at once and must be segmented. Tewari et al. [10] proposed the RBC algorithm to take these facts into account. None of these techniques take into account layered video, as used in this project. Even if the grain was refined, none of these techniques would be efficient. Podlipnig et al. [11] proposed a policy which treats the case of layered sequences. Our technique is a refinement, based on matrices of values for temporal and quality data. We also link the different streams representing the same sequence but encoded with a different quality [8].

Cache insertion is processed by three distinct mechanisms. Each of them manages a different level of prediction and differently takes the mobility of clients into account.

On-demand insertion is a traditional technique which treats none of these problems. It is equivalent to the answer given for a cache miss in web caches. In MoVie, users do not directly contact the caches. However a cache miss corresponds to the state where WebMoVie cannot find the document requested inside the system, or if it is present on a cache not usable for the mixer. The document

is then fetched from an outside server or a distant cache and is stocked on the cache to ensure an almost on-the-fly streaming.

Prefetching means inserting sequences into the cache which are supposed to be requested later. To accomplish this, our system uses local and distributed usage statistics. If both preceding techniques are time constraints, the latter is totally asynchronous and can be calculated while the system is low on load, and the movies fetched while the network is free.We try to anticipate future requests by processing a set of probable sequences that we insert in the cache. This method is based on a classification made by parsing the requests logs. As we are in a distributed system, we should also take advantage of the sorting made by the peer caches. Taking into account foreign request may lead to inserting sequences which are not locally known yet.

The technique named "hand over" consists in inserting the next part of a sequence being watched by a user in a bordering cell prior to its venue.

The on-demand algorithm looks much like traditional algorithms and therefore will not be described in more depth herein. A detailed description of our prefetching method may be found in [6]. The hand-over method brings interesting points and is described in further detail in the next section.

3 The Hand-Over Policy

The hand-over policy consists in inserting the remaining part of a sequence currently watched by a user in a cache of a bordering zone prior to its request. If the client moves out of its zone, one of the caches of the new zone already possesses the part being streamed. Therefore the system only has to initiate the connection and the user doesn't suffer from too much lag for a hand over, giving a good reactivity.

The mechanism used to perform this insertion is based on two types of information: the probability for a zone to receive a client in the near future and the cache and mixer adequation for the probable streaming zones. From this information we calculate an indicator evaluating the interest of loading the sequence. We use thresholds determining for each cache if we should insert the whole remaining part of the sequence, and which quality to use (i.e. the number of enhancement layers cached). In each case the reactivity is enhanced, in the worst case with a degraded quality for a small amount of time.

The most interesting part of the territory for this task is the set of zones bordering the current one. The probability of presence is processed by another component, the mandatary, which is described in section 2. The set of probabilities of presence $P_{0\ldots z}$ is presented in a vector $\overrightarrow{V_p}$ whose dimension is the number of bordering zones (z) plus one for the current. P_0 is the probability of presence for the current zone. With the knowledge of the list of caches able to accept the client in each new zone and the set of mixers, we rebuild a smaller cache-mixer adequation matrix $\mathcal{M}_{cm}$. We then multiply $\mathcal{M}_{cm}$ and $\overrightarrow{V_p}$ to obtain the evaluation vector V_e giving each cache a score.

$$\mathcal{M}_{cm} \times \overrightarrow{V_p} = \overrightarrow{V_e}$$

$$\begin{pmatrix} C_1M_1 & \dots & C_1M_z \\ \vdots & \ddots & \vdots \\ C_cM_1 & \dots & C_cM_z \end{pmatrix} \times \begin{vmatrix} P_0 \\ \vdots \\ P_z \end{vmatrix} = \begin{vmatrix} e_1 \\ \vdots \\ e_m \end{vmatrix} \quad (1)$$

Then, we define two values: $T, \alpha \in [0 \dots 1]$. From these two values we build two thresholds $T.P_0$ and $\alpha.T.P_0$ with $\alpha.T.P_0 < T.P_0$ such as if the evaluation of a cache is over $T.P_0$ we fetch the whole remaining part of the sequence, if the evaluation is less than $\alpha.T.P_0$ we do not fetch anything, else if the evaluation is between those two thresholds we only fetch part of the remaining sequence. This ensures that almost wherever the user moves the streaming will go on, in the worst case temporarily with lower quality, without jagging the caches. The caches are just hinted and their admission control and insertion policies could still refuse the new sequence or client.

The main improvement in the near future will consist in optimizing the bandwidth used for the prefetching. As the base layer and several enhancement layers are diffused on to different caches of the neighborhood at the same time, we plan using a hierarchical multicast algorithm [12] where the caches can subscribe to the group where the layers they request are sent.

Case Study

In this case study we show by simulation the impact of the two parameters "Threshold" (T) and the coefficient "α" on the fetching and hit rate for three characteristics distributions of probabilities. Here, we consider seven zones isolated (ie. caches do not interact outside their own zone), each one possessing one cache. We choose these characteristics to emphasize the role of α and T in the optimization of both reactivity and space saving.

In these simulations we consider that the current zone is already fulfilled with the full sequence and that the others do not have any parts of this sequence. We consider the probability of a user to be in each zone according to different situations: staying in place, going north, and a user in a train. These are depicted on the first column of figure 3. We then apply our hand-over policy with two different couples (T, α), extract the types of fetchs which apply and count the number of full hits, partial hits and miss. This is described in the other columns. The values of the couples used in the first figure of each distribution are chosen to optimize the reactivity (ie. full fetch). The values of the couples used in the seconds figures are chosen to save storage with good chance of avoiding misses (ie. partial).

As can be seen on the results of the figure 3, our policy gives at least 8 full hit out of 10 when we maximize the reactivity. This is done by loading only one cache in the first case, two in the second case and three in the third case instead of the whole six. In the second set of results, where we save storage while avoiding miss by fetching partial sequence, we obtain the same number of

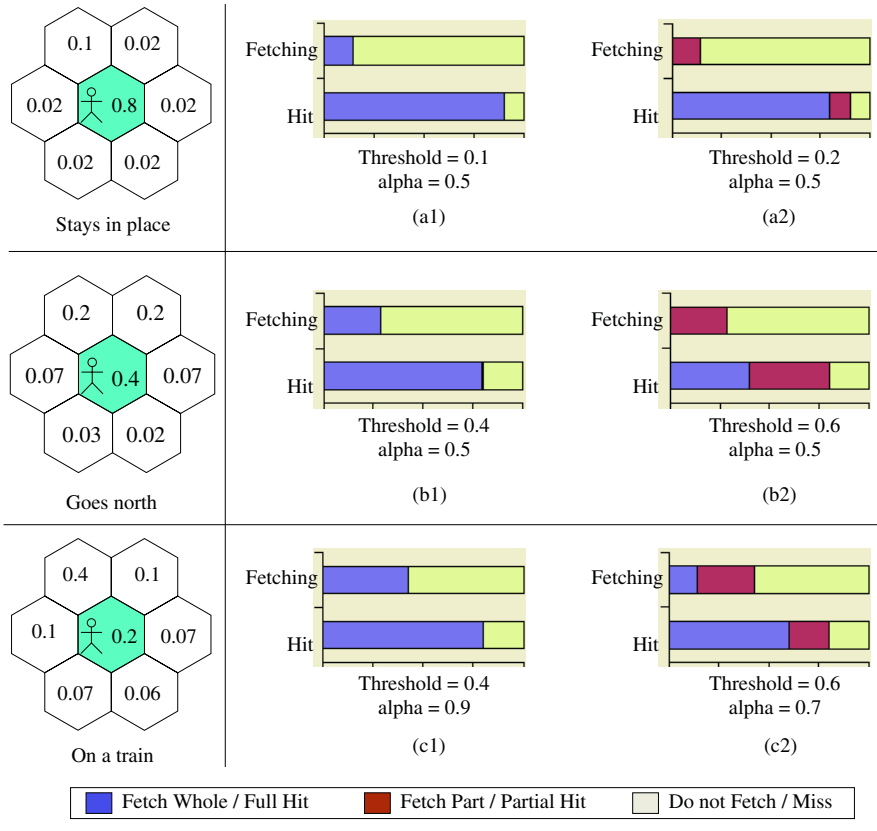

Fig. 3. Data from the Simulation.

miss in each case so we can easily maintain the streaming for eight out of ten zones but eventually with a degraded quality. At the same time, at least 55% of storage space has been saved. This simulation has shown that the hand-over policy gives a good reactivity and that we could tune this reactivity by selecting a couple (T, α) according to the resources we accept to use. By putting in place a mechanism letting the system compute the values for T and α, the system would self adapt according to the amount of resources available and the reactivity it has been asked to maintain.

4 Conclusion

The aim of the MoVie project is to allow the diffusion of video sequences on large wireless networks. The problems consist in increasing the capacity of diffusion, ensuring the extensibility of the system, maintaining a good reactivity and managing the clients' heterogeneity (phone, pda, notebook computers, desktop computers).

We have presented the principles and the architecture of MoVie and particularly of the SysMoVie layer which is in charge of ensuring the extensibility and the reactivity of the video on-demand streaming system. The extensibility is ensured by cooperative and distributed caches. The reactivity is taken in charge by the pre-loading of caches with sequences in the more or less short term. The constraints linked to the heterogeneity of the clients and their mobility are assumed in part by the representatives of the clients inside the system: the mandataries.

We have approached the set of problems release of the storage space in the caches. We have shown how we insert sequences into the caches in real time (on demand), in advance in the short (hand-over) and medium term (prefetching). For the short-term insertion we use probabilities of migration and an adequation matrix from a set of caches to a set of streaming zones. This matrix is built in a distributed, asynchronous way while the system is running. It is built with the caches, the mixers and the mandataries cooperating. It's an auto-adaptive mechanism to regulate the sysMoVie layer.

Several points are still to be studied. Particularly, the optimization at the network level of data insertion into the cache has not been approached in this article. We are working on a mechanism using a peer-to-peer communication paradigm for the exchanges between the caches and the mixers, mixed when possible with layered multicast transmissions.

References

1. B. Girod and N. Färber. *Compressed Video over Networks*, chapter Wireless Video. Signal Processing Series, Marcel Dekker, 2000.
2. Regis J. Bates. *GPRS: General Packet Radio Service*. McGraw-Hill Prof., 2001.
3. A. Samukic. UMTS Universal Mobile Telecommunications System: Development of standards for the third generation. *IEEE VTC*, 47(4):1099–1104, 1998.
4. R. Flickenger. *Building Wireless Community Networks*. O'Reilly edition, 2001.
5. J. et al. Bourgeois. Optimization of multimedia contents delivery towards mobile devices. *Journal of System Architecture, Elsevier Science*, 2003.
6. D. Charlet et al. SysMoVie: Managing interoperability in a video distribution framework for mobile environment. Technical report, LIFC, University of Franche-Comté, 2003. Ref: TR/CDM03-02.
7. Trading object service specifications, omg/00-06-27, June 2000.
8. D. Charlet et al. SysMoVie: Managing interoperability in a video distribution framework for mobile environment. In *Proceedings of PDPTA'02*, 2002.
9. S. Williams et al. Removal policies in network caches for World-Wide Web documents. In *Proceedings of the ACM SIGCOMM '96*, Stanford, CA, 1996.
10. R. Tewari et al. Resource-based caching for Web servers. In *Proceedings of ACM/SPIE MMCN'98*, 1998.
11. S. Podlipnig and L. Boszormenyi. Replacement strategies for quality based video caching. In *IEEE International Conference on Multimedia and Expo*, 2002.
12. D. Rubenstein et al. The impact of multicast layering on network fairness. In *SIGCOMM*, pages 27–38, 1999.