An Efficient Bandwidth Sharing Scheme for Fast Multimedia Download Using Proxy Servers

Vincenzo MANCUSO and Giuseppe BIANCHI
D.I.E., University of Palermo
90128 Palermo, Italy
e-mail: {vincenzo.mancuso, giuseppe.bianchi}@tti.unipa.it

ABSTRACT

This work moves from the consideration that wireless technologies and multimedia world tend to be integrated in a worldwide mobile telecommunication environment, in which mobile customers want to experience multimedia service even while moving or travelling. On the other side, it is possible to recognize that a greater and greater number of entertainment systems are based on mutimediality and network content retrieval. Thus it can be easy to foresee the interest of entities like railway operators, or air companies, in the development and deployment of effective mobile multimedia servers, able to entertain passengers while reaching their destinations. A suitable architectural solution is proposed in this paper, consisting in a setellite content server that relays files to a local proxy, which in turn is responsible of storing data and forwarding the requested files to the right customer. The key of our approach consists in a suitable algorithm, namely A2M, for the allocation and sharing of the available bandwidth in the link between server and proxy. A first analysis of the proposed system is given in the paper, also considering pre-fetching techniques advantages. Simulation results are finaly reported, showing the benefits of the A2M approach especially in terms of reduced download outage probability, i.e. the eventuality that a download could stop due to a temporary disconnection between server and proxy.

Keywords: QoS, Proxy, Elastic Buffering, Pre-fetching, Bandwidth Allocation.

1. INTRODUCTION

In the last years a great effort has been produced in order to allow wireless users to access multimedia and interactive streaming with a suitable level of Quality of Service, in terms of various kinds of delay performance experienced by customers, like playout delay, or in terms of resolution of the received content, like image details and sound fidelity, or even in terms of service reliability. Two main fenomena are characterizing the evolution of users' behavior: i) the wide diffusion of wireless communication systems, including the effort in the definition of a new generation for mobile communication systems, and also the diffusion of low cost wireless local area network access; ii) the very high speed available for network connections, expecially in the network backbones.

In this paper we investigate a content distribution scenario in which users request their favourite files from the network through a proxy, which in turns is connected to a content server by means of a wideband wireless connection (i.e. a satellite server). In particular we refear to a group of customers moving together (e.g. while travelling on a train or an airplane); each customer can be connected to the local proxy either with a wired or wireless physical connection. This is an example of integration between wireless technologies and multimedia services environments, in which mobile customers want to experience multimedia services, even while moving or travelling. It results in a main businnes interest the deployment of an effective mobile multimedia platform for advanced services, able to offer high speed internet access, immediate web surfing and others entertainment resources like music and video-on-demand.

In this work, in particular, we consider a train running from its departing rail-station to destination. The train is provided with a proxy network node, which is an internet access point for travelling customers. Tre proxy is in charge of collecting users' requests and it operates as a content-retrieval server connected to the overall network through a satellite. The proposed architecture is shown in fig 1, where a content access network is represented, based on a proxy node linked to a satellite node. Customers can access network resources by setting a connection with proxy, which is meant to provide a content retrieval service with a suitable level of quality.

In what follows we consider the traffic generated by video-on-demand requests. For sake of simplicity, no other traffic sources are taken into account. This assumpion also follows a well recognized network trend: in a QoS-aware network, homogeneous flows are grouped and handled separately from others in order to obtain homogeneous and appropriate service (i.e. the DiffServ paradigm[1], [2]).

We propose a suitable architectural solution for QoS-aware multimedia file retrieval, consisting in a content server that relays files to a local proxy, which in turn is responsible of storing data and forwarding the requested files to the right customer. Each active client has a dedicated buffer on board of proxy, in which client data are stored. The goal of our approach is to make the system robust to the uncertain behavior of the link between satellite and proxy. In fact the satellite-proxy link is caracterized by a wideband, high-performing channel, but it requires a line-of-sight connection. When no line-of-sight is available, the proxy stop receiving data from server, while clients connected to the proxy continue to receive data untill their buffers, located on the proxy, go empty too.

The key of our approach consists in a suitable algorithm, namely A2M, for the allocation and sharing of the available bandwidth in the link between server and proxy.

In this scenario the file transfer is split into two sub-connection:

- from the server to the proxy, where files can be relayed with an high rate, but the actual connection rate is an on/off funtion of setellite server visibility;
- from the proxy to the client, where the file transfer occurs with tangible rate reduction, but the link (either radio or wired) shows very high performance.

Thus, a very important role is played by the strategy adopted in order to transfer the requested files from the content server to the proxy, and in the value added services offered by the proxy, which can be able to retain in its cache memory a certain amount of highly requested data.

In fact the connection between server and proxy is the critical point in the overall scenario, since it is possible that the server-proxy link goes down due to the temporary lack of radio connection. For istance travellers using a video-on-demand service, have to setup a connection to a proxy linked to a satellite server: the proxy moves with the train and receives data from the satellite. This is not still true when the train meets, for example, a tunnel. In which follows we refear to such an event as a “channel outage”, since the proxy is made no more able to “see” the server and cannot receive data. When a channel outage occurs, and the proxy buffer goes empty, a client suffers a download blocking; we will refear to this event as “connection outage”.
The paper is organized as follows: section **2** describes the framework in which our work is running, presenting also some related works. Section **3** focuses on a proposed bandwidth allocation scheme for the setellite channel. In section **4** an insight of adopted models and relative performance is given. The subsequent session **5** contains the simulated performance results obtained using an event-driven C++ simulator. Finally section **6** summarizes conclusions and offers an insight in oncoming work.

2. Reference framework

A number of works dealing with file download issues can be found in the specialized literature. Basically the problem can be separated into two parts: the retrieval of a file and its physical download. The former problem is related to content network architectures deployment in terms of content discovery algorithms, and it is out of the scope of this paper, while the latter part is related to capacity allocation issues, local storage and replicas management and hardware performance as memory access speed. Since the aim of the present work is the minimisation of connection outage probability while downloading a multimedia stream, without introducing any addictional playout delay, this paper deals with local storage issues and optimized server-proxy connections.
Proxy is a well know approach, meant to mitigate problems as high packet loss rates and long delays, as shown in [3], where different caching and pre-fetching algorithms are tested. These tecniques consist in caching on the proxy data that are frequently required, even if there is not any user actually requesting those data. The pre-fetching approach allows a higher bandwidth utilization efficience, in fact pre-fetched data can be cached only once in corrispondence with a lot of requests.
Moreover different caching and pre-fetching strategies has been proposed in order to optimize others performance parameters as: network bandwidth utilization [4] and delay playout, as in [5] where only a file prefix is pre-fetched on the proxy memory; in [6] and [7] layered-encoded multimedia are also considered in pre-fetching algorithms, where incremental layers can be timely stored on the proxy; the hit ratio, i.e. the probability that a requested file matches an object stored on the local proxy, is the aim of the work presented by Reissline et al. in [8]. The role of a proxy-assisted delivery schemes is focused by Wang et al. in [9], where the authors show how a relative small amount of data is needed to be pre-fetched in order to experience very low startup delay with relative low network costs. Finally an insight in cost-based pre-fetching replacement schemes can be found in [10] and [11]
We face the proxy and pre-fetching problem from a different point of view, i.e. the QoS in terms of connection outage

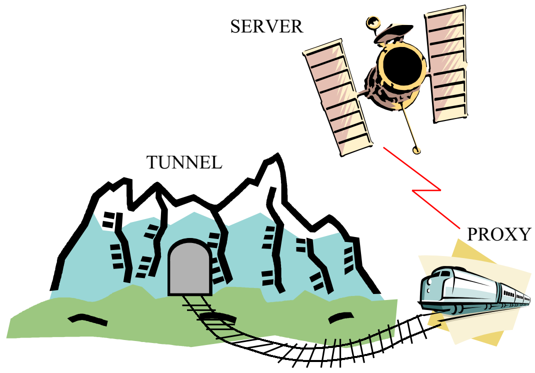

Fig. 1. Content access scenario

probability. The main contribution of the present paper consists in the definition of a new bandwidth allocation algorithm for the satellite link. The proxy operates a dinamical allocation of the satellite bandwidth in order to maintain a certain minimum buffer level for each active connection. In fact we argue that the proxy is able to minimize the connection outage probability by simply control the buffer size associated to each connection.
Moreover, a well known solution for speed-up client download consists in splitting the connection between the content server and the client, by interposing a proxy node able to quickly download files from server and also acting as a speed-buffer for the client. In fact client applications tipically need a reduced amount of bandwidth with respect to the server capacity. Splitting the download connection, using elastic buffering strategies on a proxy, allows network provider to better use available resources, decoupling the transmission speed [12] and it also prevents the client applications to be stopped due to a temporary server outage. The presence of the proxy does not imply additional playout delay experienced by the customer. In fact, the proxy relays data belonging to a stream as fast as a very small block size of data (tipically 4096 bytes) is received from the server.

3. The A2M algorithm

Usually a proxy caches data while serving a client or when there are no requests to satisfy (idle periods). The way the proxy operates is:

- a client requests a file;
- the proxy checks if the requested file is locally stored;
  - if NO, proxy asks for the file to the server;
  - proxy starts receiving file from server at a fixed rate;
- proxy sends the requested file to the client;
- if the request rate of the served file is above a threshold: the file is stored (eventually only for a fraction)

The wireless channel bandwidth is shared between active connections, and there are no connection privileged. Basically proxy downloads a certain number of files using the same rate for each one. If a new connection is accepted, the entire bandwidth is fairly reallocated and redistributed to all connections. We call this proxy operational-mode “bandwidth Equi-Distributed”, namely ED.

Our approach consists in dinamically adapt the bandwidth allocated between server and proxy, carrying data belonging to a stream. In particular we monitor the buffer level at the proxy side and make all the avaible bandwidth assigned to the stream, or the group of streams, suffering the lower buffer level. This is done with the resolution of a memory block size, for istance 4 KBytes.

We call this operational-mode "bandwidth reserved All to Minimum cache stream", namely A2M. Thus the A2M operation can be summarized as follows:

- a client requests a file;
- the proxy checks if the requested file is locally stored;
  - if NO, proxy asks for the file to the server;
  - proxy starts receiving file from server at a full rate;
- proxy starts sending the requested file to the client;
- untill the file is not fully downloaded by the proxy, the proxy checks for streams with lower buffer level and make the total server bandwidth shared among these streams
- if the request rate of the served file is above a threshold: the file is stored (eventually only for a fraction)

The temporal behavior of proxy buffer level is plotted in figure 2. In that figure, the proxy starts from an idle state and than progressively accept three incoming connections requesting non-pre-fetched files. Only the beginning phase of the connections is shown, being the remaining phases analogous. The figure plots the buffer level for each connection vs. the simulation time. Firstly all the available bandwidth C is reserved to the first incoming connection, so that the buffer level grows with a rate $C - R$, where R is the proxy-to-client rate. When a second connection is accepted, all the bandwidth C is switched to this new stream, so that the old stream buffer decreases with a rate R. The same occurs when the third connections is accepted. After the buffer level of the third stream ($buffer3$) reachs the above one ($buffer2$), A2M algorithm operates a re-distribution of the bandwidth. Streams #2 and #3 are now using half the full server rate and their buffer levels grow together untill $buffer1$ level is reached. From that instant on, the three buffers are served at the same rate, that is $C/3$.

A similar dynamic bandwidth allocation mechanism for circuit-switched satellite network has been proposed in [13], where authors show the improvements due to the adoption of such a scheme in terms of bandwidth utilization and QoS guaranties.

The described system is stable if the total amount of bandwidth allocated to clients assumes a value less than or equal to the total satellite capacity. This is, a suitable connection admission control mechanism has to be implemented. Nonetheless, there exists a probability that a stream is blocked while running, and this is due to the probability that satellite and mobile proxy are not in line-of-sight (P_{ch-out}) for a long time. For what it concerns the effects of A2M on the connection outage probability $P_{conn-out}$, it is shown in section **5** that a benefical impact is obtained. In fact, using A2M each new connection is temporarily privileged and made able to accumulate a certain buffer level on the proxy. Since the higher is the buffer level, the lower is $P_{conn-out}$, A2M tries to fill proxy buffers as fast as possible, preventing outage occurrences.

The A2M implementation only requires the knowledge of a global proxy parameter, that is the number of clients that are downloading a file not entirely cached/pre-fetched on the proxy, while the associated admission control function simply needs to know the number of clients already connected. As a

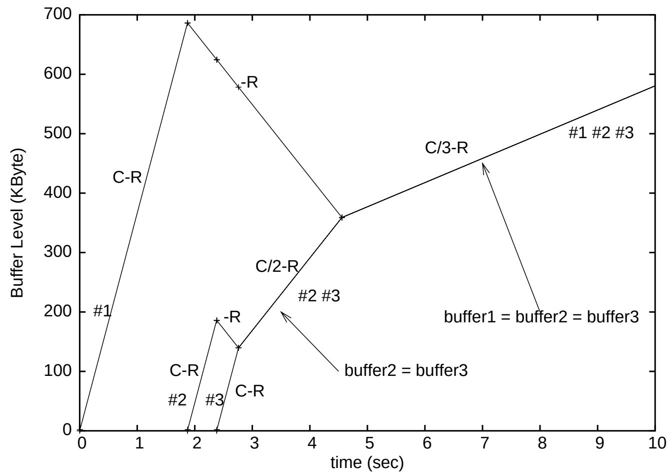

Fig. 2. A2M operation: all the available bandwidth is shared between connection with lower buffer values

consequence A2M can be considered a well suited bandwidth sharing algorithm, able to improve the robustiness of the proxy system; it is also a scalable mechanism, since no complex state information has to be managed, but the number of downloads.

4. Models and performance overview

In this section we describe most important parameters and models considered in the evaluation process. Details about the simulations can be found in the next section **5**. Main parameters are:

- requests statistics;
- file size distribution;
- admission control scheme;
- download protocol adopted;
- channel characterisation;
- proxy buffering;
- proxy pre-fetching strategy.

Firstly it is relevant the statistical characterisation of the way an object is requested by a client. By analizing proxy server traces and content distribution network statistics, it is possible to see that web objects are requested according to a Zipf-like distribution [14]. The Zipf law was originally formulated in terms of popularity of words in a text, and it states that ranking by decreasing popularity each object, the frequency of the i-th object is

$$p_i = \frac{k}{i}$$

$$i = 1, 2, ..N$$

where

$$k = p_1 = \frac{1}{\sum_{i=1}^{N} \frac{1}{i}}$$

Zipf law has been generalized as follows:

$$p_i = \frac{k}{i^{\alpha}}$$

where α assumes the value 0.986 for web object popularity [1] .

[1]In this paper we have mainly adopted a Zipf-like model for file requests; moreover we tested also different distribution laws, in particular the uniform and geometric distributions. It is clear that the distribution law adopted determines the behavior of the system expecially for what regards the optimal amount of data to be pre-fetched, but this is out of the scope of this paper and no comparisons are reported here.

Moreover we assume a file request process is a Poisson arrival process, i.e. the inter-request time is exponentially distributed.
A second aspect is related to the file size distribution. It has been shown that the distribution of file size for web downloads fits a heavy-tailed distribution function (see [15], in which log-normal and Pareto distributions are used). Since we consider essentially a video retrieval scenario, a mosty uniform file size distribution can be a resonable adoption.
Admission control schemes and transport protocol selection affect QoS performance in terms of reliability and download times. In fact, according to the selected admission control scheme it is possible to prevent congestions and otpimize the channel utilization, while the transport protocol determines the effective file download rate. For istance a standard TCP protocol could severely affect the transmission over a satellite link, while an UDP protocol should maximize the data rate at the expence of reliability, since no delivery checks are performed. In a video-on-demand scenario the client download rate is almost-fixed, while the proxy can store bytes at a variable rate. This implies that different protocol could be used on the server-proxy connection and on the proxy-client one; this is a further advantage of using a split download connection.
As regards the channel characterisation, the satellite transmits data over a reliable channel, using a suitable transmission power and adopting an appropriate coding scheme for error correction. Anyway the satellite transmission requires a line-of-sight between sender and receiver. When server and receiver (i.e. the proxy) are out-of-sight a channel outage occurs. The statistic of such events is related to the mobile behavior of the proxy. In particular a channel outage occurs when the train, with proxy on board, enters a tunnel. Thus the channel outage probability is route-dependent. For the sake of simplicity we consider the line-of-sight/out-of-sight alternation as an on/off process with both on and off periods exponentially distributed.
Finally the proxy caching and buffering strategy is responsible for channel usage optimization and for clients' connections outage prevention, as previously shown in section **3**.

5. Simulation

The proposed scenario has been tested by means of a C++ simulator. The simulator is an event-driven tool and implements a fluidic model of the download process. In other words no packets containing data have been simulated, but the data flow itself, computed taking into account parameter as:

- server to proxy download rate C, set to $32Mbps$;
- proxy buffer fill rate (as a fraction of the channel rate C);
- client download rate R, set to $2Mbps$;
- maximum number of admitted connection, ranging between 1 and 16 (=100%);
- proxy buffer emptying rate (the same as above: R);
- file length S, fixed to $100,000KB$;
- connection request arrival, modeled as a Poisson process with rate λ, ranging in [0.02;200] requests/sec;
- simple admission control rule: a maximum number of clients is set, and a new connection request is accepted only if the number of already admitted connection is lower than the maximum
- channel outage probability P_{ch-out}, ranging from 0 to 0.6, and the channel outage process is modeled as an $on-off$ process with exponential distribution for both on and off periods. Specifically the channel outage mean time is T_{off}=10s, while the mean on time is computed as

$$T_{on} = T_{off}\frac{1 - P_{ch-out}}{P_{ch-out}}$$

- file population, ranked from the most frequently requested file to the least frequent one, according to a Zipf-like disribution with $\alpha = 1.0$

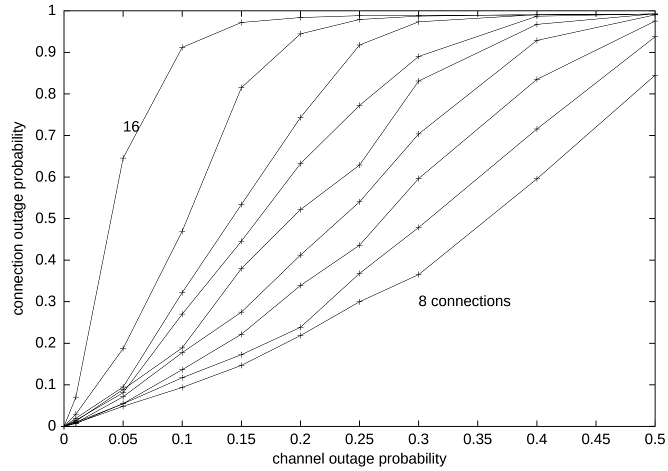

Fig. 3. connection outage probability vs. channel outage probability with uniform content requests distribution and without pre-fetching

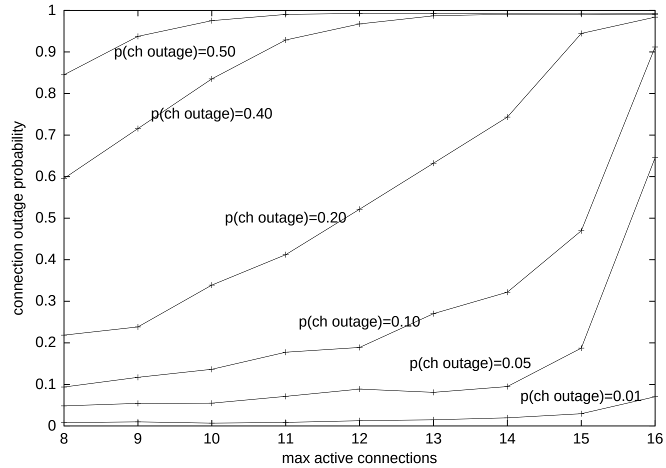

Fig. 4. connection outage probability vs. proxy utilization with uniform content requests distribution and without pre-fetching

For the sake of simplicity, simulations do not take into account transport protocols and downloading files behaves as a constant bit rate (CBR), eventually stopping and resuming after a failure.
A lot of the reported performance results has been obtained in very high load conditions, i.e. with a very high request rate, also referred as "continuous load" when $\lambda \rightarrow \infty$. This turns in a framework in which the accepted load is always to the maximum admissible by the admission control algorithm. Even if this is not the real case, the high load assumption is intended to stress the A2M mechanism in order to show its robustiness to traffic load. Moreover the continuous load assumption allows us to study the intrinsic benefical effects of A2M, since the P_{co-out} performance is a funcion of the buffering scheme instead of both proxy buffering and variable accepted load conditions.

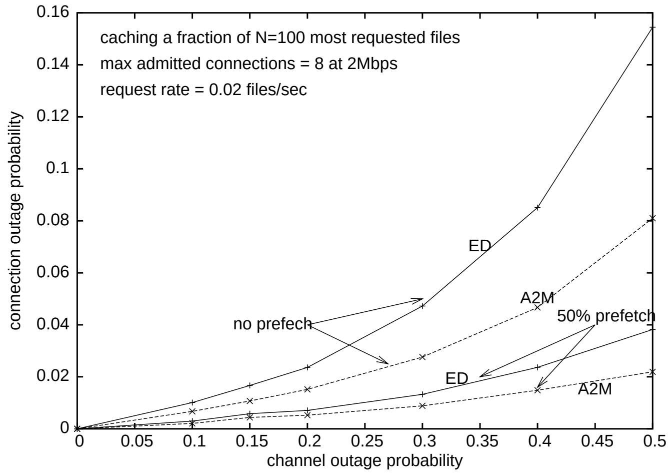

Fig. 5. connection vs. channel outage probability with max 8 connections in low load conditions

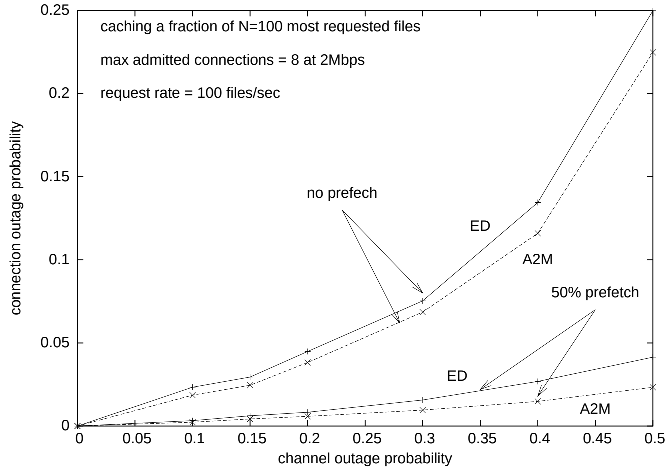

Fig. 6. connection vs. channel outage probability with max 8 connections in high load conditions

As a starting point, figure 3 shows the relation between the download failure (connection outage probability), the channel outage probability, and the max number of admitted connections in high overload conditions ($\lambda = 100$ requests/sec) when A2M algorithm is adopted. No pre-fetching is considered and a uniform distributed frequency request rate is adopted. In the same simulation conditions fig 4 shows how the connection outage probability reacts to the variations of the accepted load, in terms of maximum number of admitted connections. It is remarkable that with a small channel outage probability (i.e. 5%), the connection outage probability become intolerable as the number of admitted connection grows and approaches the maximum (i.e. 16 = 32 Mbps / 2 Mbps).

In the remaining figures a Zipf-like distribution law is adopted and pre-freching effects are also evaluated.

Figures 5 and 6 give a performance comparison between A2M and ED operation effects. In particular figure 5 is related to low request rate conditions ($\lambda = 0.02$ requests/sec), while figure 6 reports simulation results obtained in a almost-continuous load framework ($\lambda = 100$ requests/sec). Each figure plots four curves, two for A2M and two for ED simulations; for each bandwidth allocation scheme, the two curves represent simulations with and without pre-fetching. It is obvious that pre-fetching algorithms improve outage performance, due to the probability that a client request matches a (partially) pre-fetched file. Moreover simulation results show that using exactly the same pre-fetching algorithm, A2M performance looks better than ED. A2M performance is well below the ED one, expecially in low load conditions (figure 5), but this is still true also in high load conditions (figure 6). We want to stress that, using A2M, it is possible to obtain a connection outage that is less than a half of the probability experienced in traditional ED framework.

The pre-fetching gain is also clearly shown in figures 7 to 9. In those figuses we propose a partial pre-fetching scheme, in which only a fraction of the overall file is permanently stored in the proxy cache. Plots has been obtained by varying the pre-fetching fraction. Moreover, only files being requested with an high frequency are pre-fetched: as a reference we set the pre-fetching algorithm so that the hundred most requested files are selected for pre-fetching.

Firstly in figure 7 we consider the case that only 8 connections are symultaneously admitted, while the overall request rate is quite high. A small pre-fetching fraction as large as 10% of the hundred most frequrently requested files, allows a remarkable reduction of connection outage probability of one order of magnitude. On the other side, considering figure 8, using all the proxy-client connections (i.e. 16 connections) yields a smaller gain using pre-fetching, i.e. a greater pre-fetching fraction is needed in order to obtain the same performance as before, even if the offered load is well reduced with respect to the case shown in figure 7. The same can be noticed in figure 9, with very high connection request rate. In any case the connection outage probability can be reduced by 50% simply by using a 10% pre-fetching fraction of the hundred most frequently requested files [2].

Finally figure 10 gives an insight of the dependence between offered load, in terms of λ, and the connection outage probability P_{co-out}, given a prefecting setting and a channel outage probability. When the channel outage probability is resonable, figure 10 shows that the A2M approach, with a light form of pre-fetching, appears to be robust to λ fluctuations: with a full loaded proxy (16 connections), and a 10% pre-fetch of the hundred most frequently requested files, a satellite channel outage probability equal to 10% yields a connection outage probability of about 0.3% whatever the value of λ.

6. CONCLUSIONS AND OPEN ISSUES

In this paper we consider travelling customers attempting to download multimedia contents from the network. A proxy server has been introduced in order to split the download into two different connections: a secure and reliable proxy-client connection, and a server-proxy connection showing a time variant behavior, according to satellite visibility. A new bandwidth sharing algorithm, A2M, has been proposed in the server-side in order to prevent a download blocking at client-side, resulting in a connection outage event. The effectiveness of such an approach has been tested via simulation. The results show that A2M, jointly to a light pre-fetching scheme, offers high reliable download performance, minimizing the probability that a client's download is stopped due to proxy buffer emptying, when satellite server is out-of-sight.

[2]using a Zipf distribution with a population of 1000 elements, the 100s most popular fi les cumulate about 69% of the overall request probability.

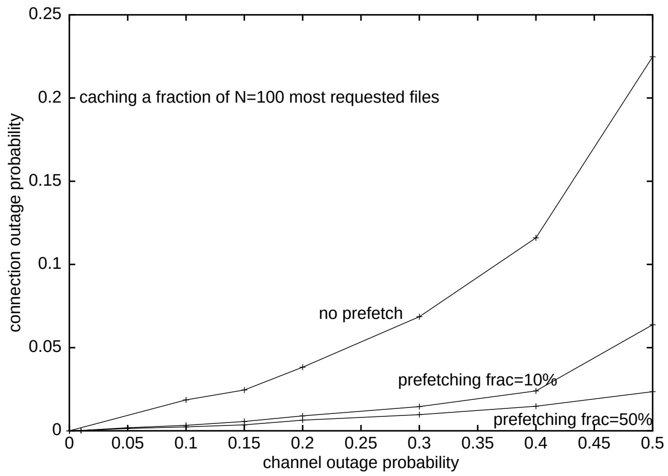

Fig. 7. connection outage vs. channel outage probability with max 8 connections admitted and high offered load

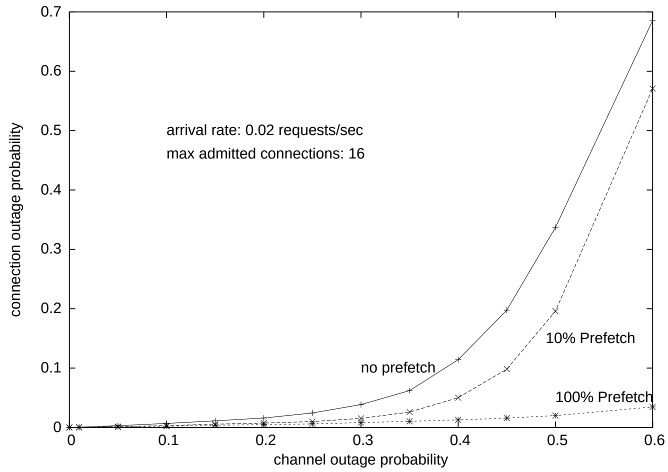

Fig. 8. pre-fetching gain in low request rate conditions

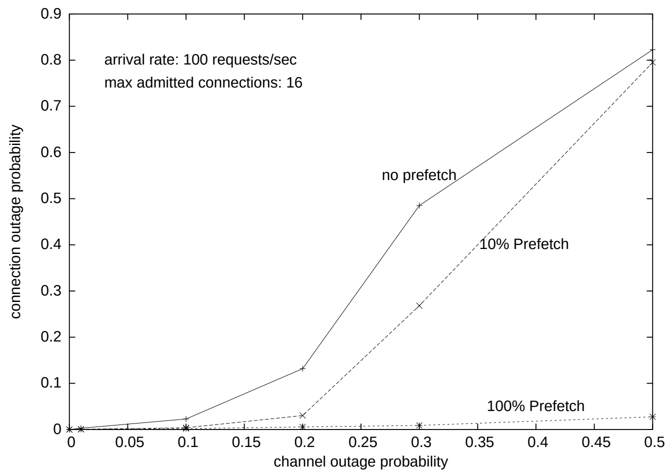

Fig. 9. pre-fetching gain in very high request rate conditions

Next steps in recearch mainly relate to the download transport protocol influence, and the implementation of an advanced admission control scheme, taking advantage from the amount of pre-fetched data in the local proxy server. Our work will be extended to a more general scenario in which proxy is connected

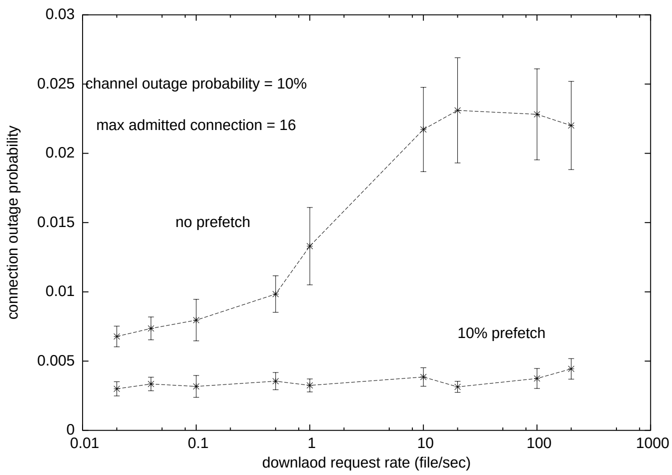

Fig. 10. connection outage probability as a funcion of request rate and pre-fetching gain for a 10% channel outage probability

to wireless servers, not necesarily a satellite one.

REFERENCES

[1] S.Blake, D.Black, M.Carlson, E.Davies, Z.Wang, W.Waiss, "An architecture for differentiated services", RFC2475, December 1998

[2] J.Heinanen, F.Backer, W.Waiss, J.Wroclawsky, "Assured forwarding PHB groups" RFC2597, June 1999

[3] S.Jin, A.Bestavros, A.Iyengar, "Accelerating Internet Streaming Media Delivery using Network-Aware Partial Caching" proceeding of the 22th International Conference on Compuyting Systems - IEEE ICDCS'02, 2002

[4] Y.Wang, Z.L.Wang, D.H.Du, D.Su, "A network-conscious approach of end-to-end video delivery over wide area networks using proxy servers", Proceeding of Infocom 98, 1998

[5] S.Sent, J.Rexford, D.Towsley, "Proxy prefi x caching for multimedia streams", Proceeding of Infocom 99, April 1999

[6] R.Rejaie, H.Yu, M.Handley, D.Estrin "Multimedia proxy caching mechanisms for quality adaptive streaming applications in the internet", proceeding of Infocom 2000, March 2000

[7] Z.Miao, A.Ortega, "Scalable Proxy Caching of Video Under Storage Constraints ", IEEE journal on selected areas in communications, vol 20, no.7, p.1315-1327, September 2002

[8] M.Reissline, F.Hartanto, K.W.Ross, "Interactive video streaming with proxy servers", proceeding of IMMCN, February 2000

[9] B.Wang, S.Sen, M.Adler, D.Towsley "Optimal Proxy Cache Allocation for Effi cient Streaming Media Distribution", proceedings of Infocom 2002, 2002

[10] Z.Xiang, Q.Zhang, W.Zhu, Y.Zhong, "Cost-Based Replacement Policy for Multimedia Proxy across Wireless Internet", IEEE Globecom'01, 2001

[11] A.Venkataramani, P.Yalagandula, R.Kokku, S.Sharif, M.Dahlin, "The potential costs and benefi ts of long-term pre-etching for content distribution", Computer communications journal, 25(4): 367-375, 2002

[12] G.Bianchi, R.Melen, "The role of the local storage in supporting video retrieval services on ATM networks", IEEE/ACM Transaction in networking, vol.5, no.6, December 1997

[13] N.Blefari-Melazzi, G.Reali, "Improoving the effi ciency of circuit-switched satellite networks by means of dynamic bandwidth allocation capabilities", IEEE journal on selected areas in communications, vol.18, no.11, November 2000

[14] G.K.Zipf, "Houman behavior and the principle of least effort",Reading, MA: Addison-Wesley, 1949

[15] M.Crovella, P.Barford,"The network effects of prefetching", Proceedings of IEEE Infocom'98, 1998