Streaming Support for Vehicular Networks Using Elastic Proxy Buffers*

Vincenzo Mancuso[1], Giuseppe Bianchi[2] and Nicola Blefari Melazzi[2]

[1]Università di Palermo, Dipartimento di Ingegneria Elettrica
Viale delle Scienze 9, 90128 Palermo, Italy
email: vincenzo.mancuso@tti.unipa.it

[2]Università di Roma Tor Vergata, Dipartimento di Ingegneria Elettronica
Via del Politecnico 1, 00133 Roma, Italy
email: bianchi@elet.polimi.it, blefari@uniroma2.it

ABSTRACT

In this work we focus on vehicular area networks. A group of customers, located into a same public vehicle, is connected to the terrestrial network through a satellite backbone connectivity. We propose to deploy on-board proxies, which rely on elastic buffering mechanisms as a mean to decouple the multimedia information retrieval rate on the network backbone from the play-out streaming rate at the user terminal. We show that elastic buffering is an extremely effective mean to reduce, or even eliminate, streaming service outage due to intermittent backbone connectivity, such that occurring when a vehicle crosses tunnels. Moreover, we show that elastic buffering is not only a technique suitable for multimedia information retrieval services, but it can be effectively applied to delayed real-time services.

I. INTRODUCTION

In traditional wireless networks, each user sitting on a moving vehicle connects independently to the terrestrial network. Conversely, in a Vehicular Area Network (VAN) scenario, a network architecture is deployed inside the moving vehicle either through wireless or wired local area networking technologies, and end-user connectivity to the terrestrial network (i.e., the Internet) is managed through a specialized on-board gateway, connected to the terrestrial network through a wireless (e.g., satellite) link.

The application scenario tackled in the European Union funded IST project FIFTH (Fast Internet for Fast Train Hosts) falls in this general area, and specifically considers a moving train connected to the terrestrial network via a satellite link, hereafter also referred to as wireless backbone. Other VAN application scenarios are nowadays being considered in research projects (e.g., ships and airplanes are tackled in the IST projects MOBILITY and WIRELESS CABIN, respectively). But the case of trains is the one that presents major challenges at the datalink and network layer, due to the occurrence of tunnel crossing - which is not applicable in the cases of ships and airplanes.

In addition to internetworking functions, the gateway may act as proxy server. As such, it may provide local storage capabilities to support caching and/or pre-fetching algorithms, meant to support video and interactive stream [1,2], as well as multimedia with quality of service [3,4]. These mechanisms are devised to maximize the probability that a customer requesting information to download, may find it stored into a repository associated to the proxy, thus improving the retrieval performance, and reducing the traffic load on the wireless backbone [5,6,7].

When dealing with streaming sessions, the presence of a proxy server allows a further level of flexibility. In fact, the multimedia information may be retrieved on the wireless backbone at a variable speed, eventually higher than the play-out rate: the resulting excess information retrieved is accumulated into the proxy buffer for future play-out. Hence, the per-stream proxy buffer can be seen as an "elastic" buffer that empties at constant rate and fills at variable rate. This approach has been shown in [8] to provide a significant performance improvement in terrestrial networks.

The goal of this paper is to provide insights regarding the applicability of elastic buffering mechanisms as a mean to compensate outage periods occurring on the wireless backbone. Such an outage may occur while the vehicle crosses areas characterized by severe fading conditions, e.g., tunnels. This is a very critical issue when dealing with streaming services, which experience possibly long (order of several seconds) interruptions, thus causing highly negative performance impairments (service disruption) in terms of the end customer point of view.

The rest of this paper is organized as follows. Section II deals with multimedia information retrieval (e.g., video-on-demand) services, and proposes a resource management mechanism specifically designed to compensate channel outage periods occurring in a VAN scenario. Section III shows that the same concepts can be adapted to provide a more effective support for broadcast delayed services, i.e. delayed access to broadcast multimedia transmission (e.g. ,television channels, etc). Performance investigation is carried out in section IV for both multimedia on-demand services and delayed real-time services. Finally, concluding remarks are given in section V.

II. ELASTIC BUFFERING FOR CHANNEL OUTAGE COMPENSATION

Let us focus on multimedia information retrieval services (such as video-on-demand). Our goal is to provide a streaming service robust to events hereafter referred to as satellite "channel outage", which may last for several seconds, and during which all sessions are interrupted.

We propose to endow the on-board gateway with proxy functionalities, devised to hide channel outage periods to the final user, i.e., to reduce the impact of channel outage in terms of resulting "connection outage".

*This work has been partially funded by the European Union in the framework of the IST FIFTH Project

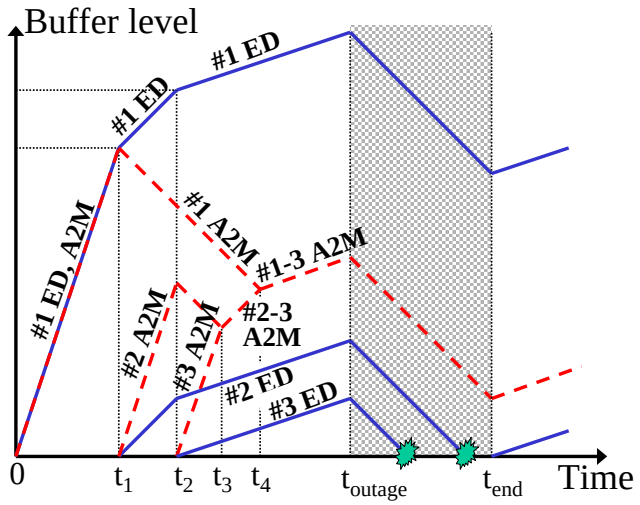

Figure 1 A2M vs. ED Operation

This can be accomplished by decoupling, from a service level point of view, the delivery service provided inside the moving network from that enforced on the wireless backbone. Specifically, the presence of an on-board proxy enables us to split the streaming service in two independent parts. Within the moving network, the stream is delivered to the end user at its natural play-out speed (for convenience of presentation, in what follows, we'll assume a constant play-out rate). Conversely, on the wireless backbone, the retrieval rate for multimedia information occurs at a rate higher than the play-out rate, and the excess stream information downloaded from the wireless backbone is temporarily buffered in the on-board proxy.

The described operation gives raise to an elastic buffer, which is filled during the periods in which the satellite link is active, and whose buffered data is consumed at a fixed rate (the play-out speed for each active streaming session) during channel outage periods. Hence, streaming sessions may remain active even during satellite outage periods, provided that a sufficient amount of information has been buffered in the proxy local storage. The streaming session outages (connection outage) only when both the satellite link is in outage and the buffered information exhausts.

A. The Equally Distributed resource sharing algorithm

For the sake of simplicity, assume that the multimedia information repository is directly placed at the terrestrial satellite gateway. Let us consider the eventuality that an on-board customer requests a file. Once the streaming request is received at the proxy, it firstly checks if the requested stream is locally cached. If this is not the case, the request is forwarded to the remote server, provided that satellite resources are available (i.e., after a positive response of an admission control decision based on the number of already active streaming sessions on the satellite link). All the satellite link capacity is shared between all active sessions so that the multimedia information is retrieved on the satellite link at the highest possible rate.

The simplest approach is to fairly share the satellite resources among all the concurrent multimedia information retrievals. This policy is referred to as "Equally-Distributed" (ED). Figure 1 shows the ED operation (solid lines) by reporting the per-session proxy buffer occupancy level versus time. It is assumed that at time 0 a new connection, session #1, starts. Let C bits/s be the satellite channel capacity, and let R bits/s the streaming play-out rate (supposed equal for all the sessions). Since a single session is offered to the satellite link, all the channel capacity C is reserved to download information. This information is in turns played-out at rate R. We conclude that the buffer-level grows linearly with time, being $(C-R)t_1$ bits the buffer-level at time t_1. Assume now that at time t_1 session #2 starts: the two sessions will equally share satellite link resources, with the result that the per-stream buffer level will linearly grow at rate $(C/2-R)$. In general, when n session share the link, each is granted a retrieval rate equal to C/n.

A closer look to Figure 1 allows us to underline a major limit of the ED policy. In fact, whenever a channel outage occurs (in the figure, from time t_{outage} to time t_{end}), the most recently admitted sessions are the first ones for which connection outage occurs. In turns, the buffer-level reached by the first admitted flow is unnecessarily high. In other words, a fair resource sharing on the channel yields an unfair share of the proxy buffer space, and increases the probability that connection outage occurs during a channel outage period.

B. The All-to-min resource sharing algorithm

In the case of channel outage, the buffer-level represents the margin before an outage occurs. For this reason we will use also the term "outage margin". It naturally comes out that an effective resource sharing approach is to devise a policy targeted to converge as fast as possible to a situation in which all sessions have the same outage margin.

We refer to such a new policy with the name "All to Minimum" (A2M). The A2M operation is designed to dynamically reserve all the channel resources to the session(s) with lower buffer-level. This operation is implemented at the proxy, which can access the information regarding the per-flow buffered data. In turns, the proxy dynamically signals (through layer 2 - satellite-specific – signalling mechanisms) to the terrestrial gateway the identity of the sessions which suffer of minimum buffer-level, so that this latter gateway is able to properly schedule the information download.

The A2M operation is exemplified in Figure 1 (dashed lines). At time 0, the A2M algorithm operates as the ED one, since a single session is admitted to the channel. The differences start from time t_1 in which session #2 starts. In fact, from time t_1, all the channel capacity is reserved to session #2. Hence, its buffer-level grows at rate $C-R$. Conversely, since no channel resources are assigned to session #1, its buffer-level decreases at rate R. This would last until the two buffer-levels would become the same. If, in the mean time, a new session #3 starts (time t_2 in the figure), it will exploit all the channel resources until it reaches the same buffer-level of session #2 (this occurs at time t_3 in the figure). Then, the channel capacity C will be equally shared between sessions #2 and #3, so that their buffer-level grows with rate $C/2-R$ while the buffer-level of session #1 decreases with rate R. Once all the n (3 in the example) per-session buffered data reach the same level (time t_4), a uniform share C/n of the channel capacity will be again enforced.

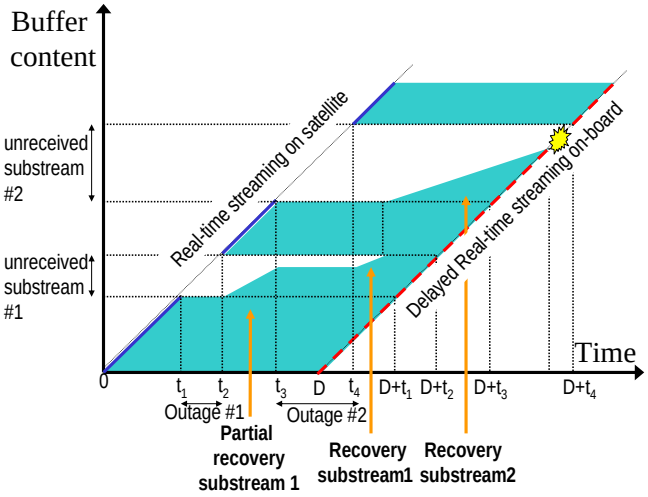

Figure 2 Real-Time Connection outage after multiple channel outages

As apparent from Figure 1, the A2M mechanism minimizes the probability that a connection outage occurs during a channel outage period. The price to pay is that, in the case a connection outage occurs, this is likely to simultaneously involve all the admitted sessions.

The A2M algorithm can be easily extended to streaming sessions characterized by different data rates. The only difference is that the per-session buffer-level has to be measured in terms of play-out time, i.e., in terms of time to reproduce a content on a user's terminal.

III. ADAPTATION TO DELAYED REAL-TIME STREAMING SERVICES

The seamless support of broadcast streaming services, such as digital television channels, on moving networks is an extremely appealing challenge for a public vehicle operator, and perhaps is of even greater interest with respect to multimedia-on-demand service support. Of course, due to the presence of tunnels, it is possible to compensate channel outage for real time streams only if i) their delivery inside the train is delayed by a suitable amount of time D seconds, and ii) the proxy will buffer all the information related to such a delay.

Delayed play-out can be accomplished by simply using fixed buffers whose size is exactly equal to the one needed to store D seconds of broadcast video. Of course, when an outage lasting $\delta > D$ seconds occurs, the proxy will be able to play-out only the first D seconds of the considered stream, while connection outage will occur for the remaining δ-D seconds. Thus, D should be set to be larger than the maximum tunnel crossing time. However, this is not a sufficient condition to avoid outage. Figure 2 reports the real time streaming as received from the satellite (solid line) and the delayed real time streaming delivery occurring on-board (dashed line). The x-axis reports the elapsed time. As shown in the figure, the delayed play-out starts with an initial offset equal to D. Two channel outage periods are illustrated in the figure: one occurring in the range (t_1, t_2) and the other in the range (t_3, t_4). If no action is taken, it is evident that, during an outage period, a fraction of the original stream (referred to as sub-stream #1 and #2 in the figure) will not be stored in the proxy buffer. Hence, after a delay D, this outage will be experienced also within the train, and specifically in the time ranges $(D+ t_1, D+ t_2)$ and $(D+ t_3, D+t_4)$. In other words, during a channel outage, the un-received sub-streams cannot be temporarily stored in the buffer for subsequent play-out.

We propose to solve the above problem by treating each sub-stream as an independent special case of multimedia information retrieval session. As soon as an outage period ends, the proxy buffer uses extra bandwidth available on the satellite channel to recover the sub-stream, and thus refill the "buffer hole" caused by the channel outage. This implies that, to effectively support delayed real-time services, the channel capacity must be split in two parts: one used to transmit the real-time streams, and the other used for the described sub-stream recovery procedure.

It will be shown in the numerical results section that the extra bandwidth made available to delayed real-time services must be quite large. Since, in addition, this extra bandwidth is used only at the end of a channel outage, and generally for a short time (i.e., in a very bursty mode), it is convenient to let the sub-stream recovery procedures share the same bandwidth reserved to multimedia-on-demand streaming sessions. In doing this, it is worth noting that the A2M algorithm can be adopted to manage this extra bandwidth, i.e., no distinction is in principle needed between sub-stream recovery procedures and normal on-demand multimedia streaming support.

To better understand how connection outage may arise, consider again the example illustrated in figure 2. We have here reported a scenario in which recovery of multiple un-received sub-streams may occur. In fact, the recovery phase for sub-stream #1 starts at time t_2, right after the end of the relevant outage period. However, due to scarce extra bandwidth availability in the time interval (t_2, t_3) and/or too short time range (t_2, t_3) elapsing before outage #2 occurs, it is possible that only part of sub-stream #1 is recovered. As illustrated in the figure, at time t_4, the recovery procedure for the sub-stream #1 restarts (in fact, following the A2M rules defined in section II-A, the outage margin for the sub-stream #1 is lower than the margin for sub-stream #2). Hence, all the available bandwidth is assigned to sub-stream #1 until its complete recovery. However, this delays the beginning of the recovery procedure of sub-stream #2. In other words, even if the outage margin after channel outage #2 was large enough to allow full recovery of the sub-stream #2, the presence of an additional sub-stream to be recovered leads to connection outage.

An important parameter is the ratio K between the spare bandwidth made available for recovery and the delayed streaming service rate. This parameter can be intended as the amount of video seconds that will be recovered in a second. In practice, when $K>1$, connection outage can occur only when the train is in a tunnel. But $K>1$ means that more than 50% of total bandwidth is allotted to recovery operations; the resulting efficiency in link utilization is rather poor. When $K<1$, the time needed to recover a sub-stream is longer than the sub-stream itself; this means that it cannot be recovered while delivering it, and an outage occurs.

IV. PERFORMANCE EVALUATION

The proposed scenario and algorithms have been tested by means of a fluidic C++ event-driven simulator.

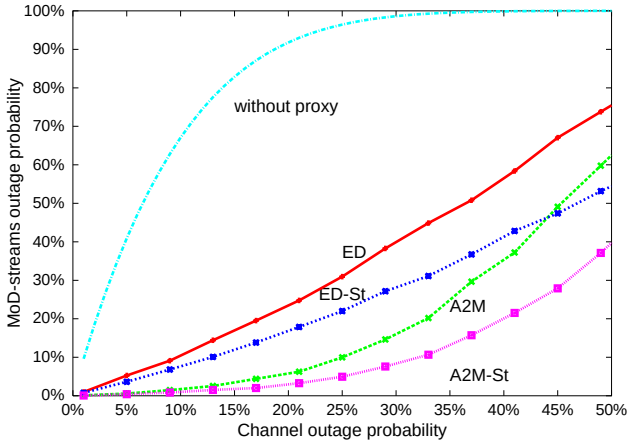

Figure 3 Multimedia-on-demand: connection outage probability vs channel outage probability

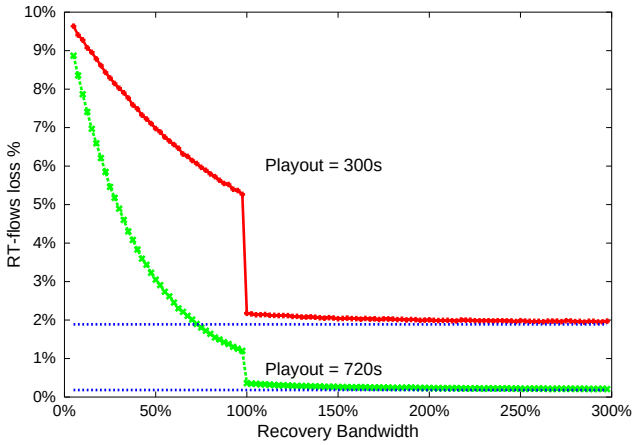

Figure 4 Delayed Real Time stream: fraction of unrecovered (lost) data

In section IV-A we deal with multimedia-on-demand (MoD) services, while the case of delayed real-time (RT) services is discussed in section IV-B. Simulation results taken from a train path scenario currently under deployment are presented in section IV-C.

A. Multimedia on demand services

We have considered a scenario characterized by homogeneous multimedia-on-demand sessions, each requiring a constant play-out rate equal to *R=1* Mbps. The wireless backbone capacity has been set to *C=16* Mbps. The play-out time for each streaming session has been set to a constant value equal to 1800s. To stress the wireless backbone, we assumed a worst-case infinite load regime referred to as "saturation load". Specifically, the number of simultaneous retrievals on the wireless backbone has been enforced not to overcome a maximum threshold (8 retrievals, in the simulation runs). As soon as a retrieval ends, we assume that a new streaming request is immediately available.

We remark that the number of users served, in average, in saturation load conditions is actually greater than the above mentioned threshold, as a new streaming session starts as long as the previous streaming session has completed the information retrieval (but this can happen well before the end of the stream play-out in the internal network).

Channel outage has been modelled with an ON-OFF pattern, meaning that, during an outage period, i.e., inside a tunnel, no connectivity is possible, while connectivity at full capacity *C* is provided in visibility. We assumed outage periods (tunnels) with an exponentially distributed duration, with mean value 180s. The duration of the ON periods is set as a function of the channel outage probability target $P_{ch}= T_{ON}/(T_{ON}+T_{OFF})$.

In the simulation, resource management on the wireless backbone is assumed to be instantaneous. We remark that, even for geostationary satellites, such signalling delays are negligible, when compared to the time needed to cross a tunnel. The proxy buffer space is assumed to be infinite (though, in practice, its occupation level always remains bounded).

Figure 3 shows the ED and A2M performance in terms of connection outage probability versus channel outage probability. In particular, connection outage probability is defined as the probability that the connection outage occurs for at least one time during the stream duration. As shown in the figure, if no proxy were employed, this probability would rapidly converge to 100%, as the channel outage probability grows. We see that the improvement provided by elastic buffering is remarkable.

Moreover, A2M significantly outperforms ED in all cases. Regarding buffer space consumption, it is very interesting, and perhaps not intuitive, to note that the buffer space needed on the proxy is very limited. Numerical results obtained during the 50% channel outage probability simulation run show an average buffer occupancy level of about 2500s (i.e., less than 1 and a half stream size), while this value decreases to as low as 500s in the 75% channel outage probability case (clearly, the more severe the outage, the less loaded the buffers will be).

To demonstrate that the proposed proxy management scheme can be efficiently run in conjunction with a caching or pre-fetching mechanism, figure 3 also reports the performance of the ED and A2M schemes (labelled as ED-St and A2M-St) in the assumption that 30% of the incoming requests find files pre-loaded in the proxy memory for half of their size (see [1,7] for insights on partial pre-fetching schemes). As expected, pre-fetching leads to a performance improvement, but what is interesting from our point of view is that A2M shows a further relative advantage over ED.

B. Delayed real-time services

Performance results concerning the support of delayed Real-Time services are reported in figure 4. The wireless backbone was loaded with a single never-lasting broadcast transmission. In order to present cleaner results, we have not included multimedia-on-demand sessions in this scenario. The two major design parameters to be considered in such simulations are i) the play-out delay to be adopted before delivering a stream to the end user, and ii) the amount of extra bandwidth resources that are reserved to recovery procedures.

Figure 4 reports the percentage of real-time stream lost as a consequence of connection outage (un-recovered data), versus the recovery bandwidth, i.e., the extra-bandwidth with respect to the natural real-time broadcast rate. Two initial play-out delays are considered: 300s and 720s. The

channel outage has been modelled by assuming exponentially distributed outage periods lasting, in average, 180s, and exponentially distributed visibility periods with mean value 1620s (i.e. a 10% channel outage probability).

The figure shows that as long as the recovery bandwidth increases, the performance improves. The amount of lost fraction of the stream converges to an asymptotic value which simply represents the probability that a single tunnel (i.e., an outage period) lasts more than the initial play-out delay: in such a case, a fraction of the stream will be lost regardless of an eventually unlimited extra bandwidth available. A sharp improvement in the performance is suddenly encountered when the extra bandwidth available is equal to the stream rate. This operational point corresponds to $K=1$, as defined in section III. In fact, for $K \geq 1$, the recovery bandwidth is greater or equal than the stream rate, and thus a connection outage can occur only inside a tunnel. Conversely, when $K<1$, the recovery rate (the rate at which the proxy buffer is filled) is lower than the stream rate (the rate at which the proxy buffer is emptied), and thus connection outage may occur outside a tunnel, as long as the buffered information exhausts (see also Figure 2).

C. Train scenario: Italian railways

Figure 5 presents performance results taken from the reference deployment scenario tackled in the European Community IST project FIFTH. It considers a deterministic tunnel pattern taken on the railways path between the Italian cities Rome and Florence (covered in about 1.5 hours by high speed trains). About 23% of this path is covered by 44 tunnels. The average crossing time for a tunnel is 24 seconds, with the longest one lasting for about 180s. Moreover, in many cases, tunnels are very close each other and leave very little time for stream recovery.

The leftmost curve in Figure 5 shows results for multimedia-on-demand services. For the case of eight streams in saturation load, it plots the extra-time needed to complete the vision of the streams (percentage, left y-axis) versus the extra-bandwidth available on the satellite channel[1]. For example, a 20% extra sizing of the satellite bandwidth leads to about an extra 13% of the stream delivery time. The figure shows that a small extra-bandwidth can drastically reduce the download time. However, if the goal is to achieve a marginal completion time overhead, a considerable satellite bandwidth over-provisioning (up to 70% and more) is required.

The rightmost curve in Figure 5 reports results for a delayed broadcast (real-time) stream. Here, the obvious service requirement for deployment purposes is to provide a seamless support of the broadcast channel, i.e., design the system so that no outage periods occur. The plot shows the initial play-out delay (y-axis, right scale) and the extra bandwidth sizing (x-axis) necessary to achieve an uninterrupted service (zero-loss).

[1] Since, during a connection outage, advertising might be delivered to the customers, this plot might be used to estimate how much advertising overhead a customer could suffer versus a given satellite bandwidth sizing.

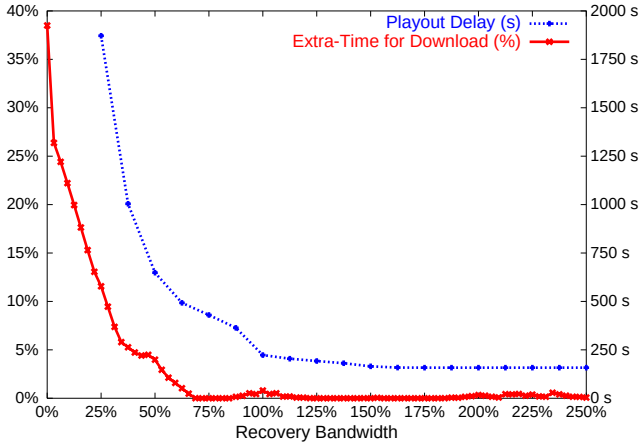

Figure 5 Rome-Florence railways path: Delayed Real-Time zero-loss trade-offs (right y-axis) and Multimedia-on-demand performance (left axis)

Clearly the initial play-out delay can be traded off with extra bandwidth. It is also clear that a minimum initial play-out delay, equal to the longest involved tunnel (about 180s), is necessary to provide an uninterrupted service.

V. CONCLUSIONS

In this paper we have shown that the usage of proxy servers and elastic buffering mechanisms allows to effectively reduce, or even eliminate, connection outage events. Moreover, we have proposed a resource management mechanism for the satellite link capacity, called A2M, and we have shown its superiority with respect to resource management policies based on an equal share of the satellite link capacity among the active streams. The basic ideas proposed in this paper are currently developed within the frame of the IST FIFTH project, which employs a bidirectional communication between a GEO satellite and a travelling train equipped with an auto-seeking on-board parabolic antenna.

REFERENCES

[1] S.Sent, J.Rexford, D.Towsley, *Proxy prefix caching for multimedia streams*, Proceeding of Infocom 99, April 1999

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

[3] R.Rejaie, H.Yu, M.Handley, D.Estrin, *Multimedia proxy caching mechanisms for quality adaptive streaming applications in the internet*, Proc. IEEE Infocom 2000

[4] B.Wang, S.Sen, M.Adler, D.Towsley, *Optimal Proxy Cache Allocation for Efficient Streaming Media Distribution*, Proc. IEEE Infocom 2002

[5] 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*, Proc. IEEE Infocom 1998

[6] M.Crovella, P.Barford, *The network effects of prefetching*, Proc. IEEE Infocom 1998

[7] S.Jin, A.Bestavros, A.Iyengar, *Accelerating Internet Streaming Media Delivery using Network-Aware Partial Caching*, Proc. of the 22th Int. Conf. on Computing Systems - IEEE ICDCS'02, 2002

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