Application-Assisted Adaptation of Real-Time Streams over Wireless Links

Margaritis Margaritidis and George C. Polyzos

Center for Wireless Communications
and
Computer Systems Laboratory
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0114

Abstract – We introduce MobiWeb, a new proxy-based network architecture that enhances the performance of adaptive real-time streams over wireless Internet links. MobiWeb includes a priority scheme that preserves media smoothness despite short-term fluctuations of the link, as well as a filter-based scheme to adapt to long-term link changes. The two schemes are shielded from each other with the use of two timers which permit the continuous exploration of the link resources, increasing the utilization of resources when they are abundant, while forcing enhanced streams to back-off to more moderate resource usage when resources are scarce. MobiWeb uses admission control for real-time traffic in order to provide a base level of quality, while remaining fair and transparent to unaware, best-effort traffic.

I. Introduction

The design of the first and second generation mobile communication systems aimed at the transmission of voice, which made their integration with data networks a difficult task. Since then, new technologies and devices emerged, providing users the opportunity to have decent access to the Internet.

However, support for interactive, real-time continuous media is lacking. The performance of real-time applications appears to be significantly impacted by wireless channel impairments, much more so than that of other Internet traffic. Real-time continuous media demand significant amounts of bandwidth and pose rather tight delay and delay jitter requirements. These demands are difficult to satisfy because of the frequent and unpredictable fluctuations in the quality of wireless links. Even though continuous media can cope with occasional infrequent transmission errors, their performance can be devastated by bursts of them.

In addition, wireless mobile communication is often subject to handoffs as users move among cells. The different characteristics of the channel after the handoff might cause contention between the newly arrived streams brought to this cell (because of the mobile's movement) and the preexisting traffic. The traditional Internet protocols lack appropriate protection mechanisms and will probably lead to all applications suffering performance degradation or backing off (the ones using TCP), which will mostly disrupt the real-time ones.

MobiWeb introduces a new solution to the above problems by means of an inter-stream priority scheme, to address the short term fluctuations of the wireless link, and by supporting adaptive applications, which can gradually adjust their traffic and performance to the long-term changes of the channel quality. The priority scheme classifies packets according to the importance of the stream they belong to in the current environment. Priorities can change dynamically to reflect the relative importance of this stream, as new streams are initiated and old ones are terminated or as the focus of the users and their actions change. The adaptation interface allows applications to indicate their preferred adaptation method in the form of filters to be applied onto a stream to change its traffic (and presentation) characteristics. *MobiWeb* utilizes quality-of-service information provided by lower network layers (e.g., as in [1]) in order to adjust the traffic and the application expectations according to the resources that the network can provide at a certain moment.

The remainder of this paper presents the *MobiWeb* architecture in more detail and it is organized as follows. Section II discusses the features that such a scheme must have to effectively accommodate real-time traffic over wireless channels. Section III describes the *MobiWeb* architecture and justifies some decisions we had to make during the design of specific features. Section IV presents related work from other important projects in this area. Finally, Section V provides a summary and conclusions.

II. The Need For Priorities And Adaptation

Traditional Internet applications, such as ftp, telnet, http etc., can be operated fairly seamlessly from a mobile host. On the other hand real-time traffic has a hard time to perform well over wireless channels with the current Internet protocols and infrastructure. This disparity is due to the different characteristics of each traffic type. Traditional traffic is best effort and usually requires reliable delivery, but does not impose stringent timing requirements. On the other hand real-time traffic can tolerate some loss (depending on the specifics of the application and the coding scheme), but has stringent delay and jitter constraints and typically consumes a large portion of the link bandwidth. The current wireless data infrastructure, based on the TCP/IP protocol stack, can usually meet the requirements for non real-time traffic, but fails to accommodate effectively the real-time streams.

There is a need for a different approach when it comes to transport real-time traffic over wireless channels. The solution must cater for the special needs of real-time traffic, while it must still work correctly with the traditional, non real-time applications. The most important features of such a solution are:

a. *Bandwidth reservations and admission control*: These protect real-time streams from other real-time traffic and from being starved by best effort traffic.

b. *Scheduling*: In order to satisfy the delay bounds and provide a steady flow of packets, the scheme must provide intelligent scheduling and buffering for packets.

c. *A priority scheme*: Applying a priority scheme on packets of individual streams of different importance gives the opportunity to the scheduler to select the least important packets to drop during periods of decreased resource availability.

d. *An intelligent traffic management layer*: Based on the proxy model, this layer can provide enhanced features to advanced applications, while still serving transparently the unaware ones.

e. *An adaptation interface*: This allows adaptive applications to provide their adaptation specifications to the management layer, which in turn provides the mechanisms for the actual adaptation.

These features are further explored and expanded next.

A. Bandwidth reservation and admission control

Real-time continuous media streams can be highly bandwidth intensive. This, in conjunction with the fact that wireless channel bandwidth is usually scarce, makes a strong case for a reservation scheme. Real-time streams like video can cope well with occasional packet loss, but they suffer when the available resources drop significantly. For example, a new ftp connection might push a large number of packets onto the wireless channel, consuming most of the available bandwidth for a few seconds. This can be devastating for a video application's performance sharing the same channel since video traffic will be delayed considerably or lost, causing the user to experience a brief freezing of the visual information or even total failure, depending on the robustness of the video application.

With a reservation scheme instead, a certain percentage of the bandwidth will be protected from use from the ftp traffic, allowing the video stream to provide the minimum acceptable quality to its user. The ftp transfer on the other hand will successfully complete with a certain delay, which, however, is usually not as important as the preservation of the smoothness of the real-time stream.

B. Scheduling

Real-time applications usually impose stringent delay and jitter bounds for their traffic. Late packets are usually useless and early ones consume valuable buffer space. Satisfying these requirements can be achieved through buffering and scheduling mechanisms at the two ends of the channel. Buffering can smooth jitter and decrease burstiness, while scheduling provides isolation among streams and a mechanism to meet the delay bounds.

C. Priority scheme

When sudden degradation of the quality of the link occurs, something quite typical for a wireless link, and no special measures are taken, packets are dropped arbitrarily at the transmitting end. Existing dropping policies are usually random-based or globally fair, which penalizes mostly real-time streams.

In environments where resources fluctuate often, it is important to prioritize the traffic in a way that will preserve the important characteristics, such as the smoothness of real-time streams. This necessitates traffic to be assigned priorities that can change dynamically based on available resources and user preferences. These priorities can be used by the scheduler to schedule traffic appropriately, in order to maximize the user's utility or the link's efficiency.

D. Intelligent traffic management layer

The traffic management layer is the intermediary between the applications and the traditional Internet transport protocols. Its task is twofold. First, it must provide an enhanced interface to advanced adaptive applications. Second, it must still preserve the functionality of the traditional interface for the unaware applications in order for them to still operate correctly.

The operation of such a layer is limited between the two ends of the wireless channel. This reduces the necessary system modifications to the wireless part of the network, thus easing deployment.

E. Adaptation interface

An important dimension of the prioritization scheme that must be considered is the point of view of the user. Even if we devise a reasonable static priority scheme covering all possible types of traffic, it might not necessarily reflect the preferences of a user at all times. For example, a user of an audiovisual application

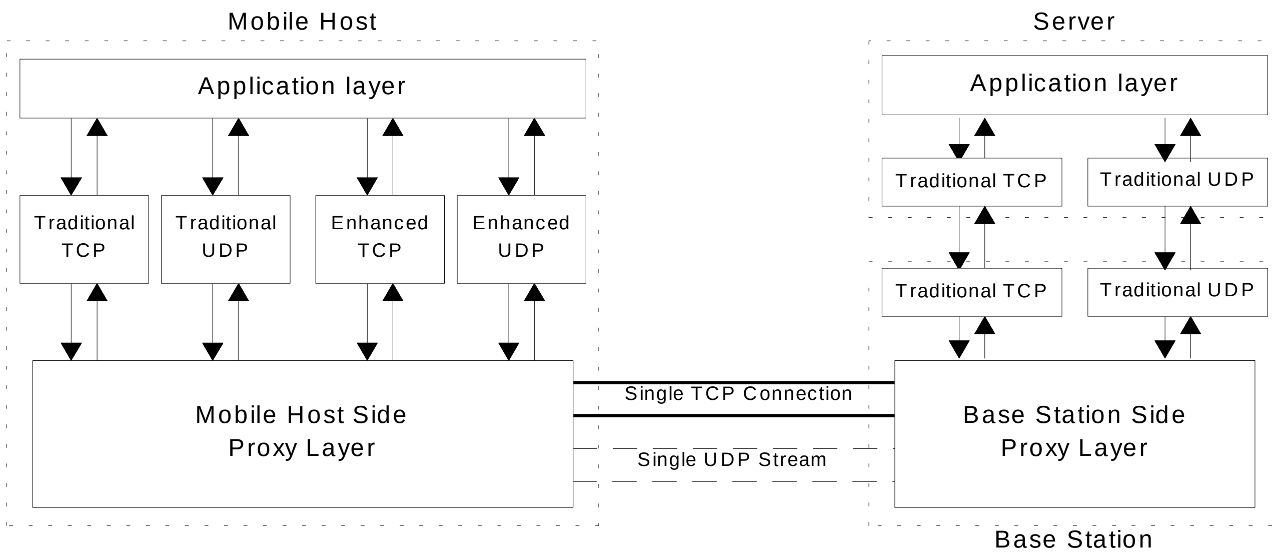

Figure 1. The Proxy-Based Architecture of MobiWeb.

might prefer to mostly hear clear sound and have an inferior visual quality, but might need to observe something important in detail, without sound at all, some of the time. These two conflicting objectives cannot be captured by a single static priority scheme.

It is obvious that the relevant priority of each stream can vary depending on the content and the preference of the user, which cannot be always accurately predicted. A more efficient solution to this problem would be to give the user the opportunity to declare the relevant priorities of the streams according to his preferences, when he does not want to use the default ones. This can be done by an adaptation interface presented to the applications.

The adaptation interface permits applications to manage the adaptation of their streams both in the short and in the long term. In the short term they can provide the relative priority of the streams based on the current preferences of the user. In the long term they can supply the management layer with filters that will adjust the streams according to the prevailing link conditions. The traffic management layer can then use this information to dynamically adapt the operation of each stream according to the desired performance of all other streams and the available resources.

III. Architecture

We are implementing and experimenting with the *MobiWeb* architecture, in order to overcome the difficulties of transporting real-time continuous media streams over wireless links, following the guidelines presented above. This section describes our design decisions and their rationale.

A. Proxy architecture

MobiWeb is based on the proxy model. One of its most important features is that it remains transparent to unaware applications. Since the architecture is intended to be deployed locally, at the two ends of the wireless channel, the proxy model provides all the necessary features that *MobiWeb*'s functionality needs.

Figure 1 illustrates the *MobiWeb* architecture. Two symmetric proxy layers are added in both the Mobile Host (MH) and the Base Station (BS) at the two ends of the wireless channel. Vertically, the proxy layer is inserted between the application and the transport layer of the current Internet protocol stack. At both ends the proxy layer intercepts each TCP or UDP stream from an application. If the application is *MobiWeb* aware, the proxy layer applies first the appropriate filter on each stream initiated from this application, which will adapt its quality according to the currently selected Level of Quality. Otherwise, each packet of the stream is only checked for time expiration and is being queued. The scheduler is then responsible for arranging the actual transmission of these packets. In order for the scheduler to be able to apply the prioritization scheme, a single TCP and a single UDP streams are permanently open between the MH and the peer BS. Each packet is been encapsulated in a new packet that carries the stream's ID before transmission through the TCP or UDP stream [3,4].

The reason for the use of a single TCP and a single UDP stream is twofold. First, it eliminates setup overhead between the two ends for each new TCP connection. Second and more important, it enables the scheduler to enforce the priority scheme. Since *MobiWeb* is designed to be dynamic and transparent to unaware traffic, compatibility with the current TCP and UDP protocols is imperative. Since there are no appropriate provisions for priorities for TCP/UDP traffic, the transmission of multiple streams through a single TCP connection or UDP stream is the best solution for realizing the prioritization scheme.

Note that the proxy layer uses cross-layer information in order to operate properly. As figure 2 shows, the interface to the application layer is similar to a typical transport layer one, with the only difference being that it is enhanced with optional fields for the adaptive applications, in order for them to be able to assign priority values to each of their streams. These applications can also provide an association between each Level

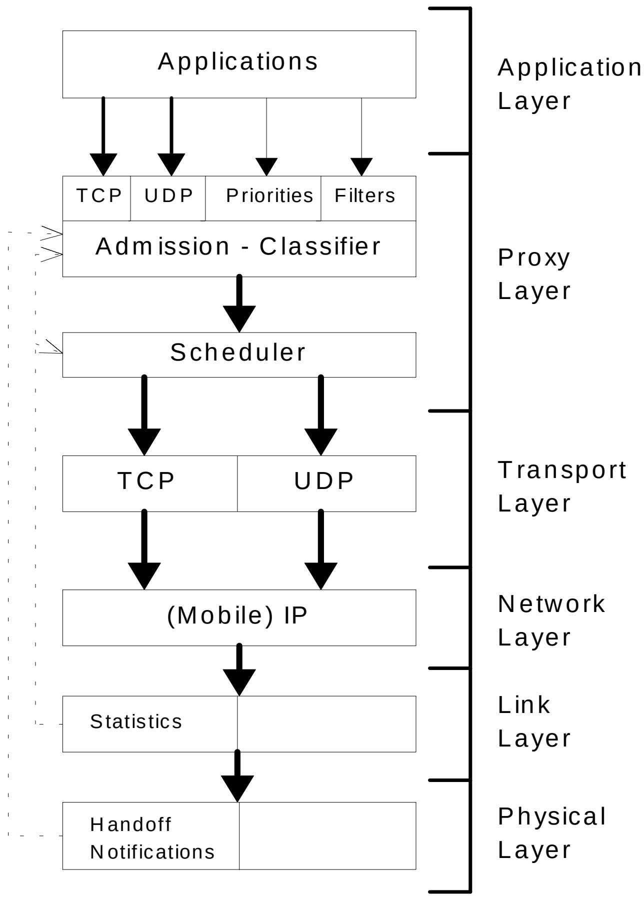

Figure 2. The Proxy organization and interface.

of Quality (LoQ) of their operation and each filter that the proxy will use in order to transform the stream to meet this LoQ. More details about the LoQ are given below.

At the lower end, the scheduler needs to have information about the current quality of the wireless link, in order to decide which packets to transmit and which to drop. The channel statistics can be gathered by the link layer and recorded continuously in files directly accessible by the scheduler, e.g., as suggested in [1]. These statistics not only affect the decisions of the scheduler as part of the short term drop packet policy, but also play an important role in the admission policy for new streams and in the long term adaptation of the existing *MobiWeb* aware applications.

B. Initiating a new stream

Each time a new stream is initiated, the proxy layer intercepting it scans for additional information indicating an adaptive, *MobiWeb* aware application. The new socket interface includes new fields for an application to indicate its Levels of Quality (LoQ) and their accompanying priorities. Each LoQ corresponds to a certain operation level of the application. These levels form a list in ascending order of quality. It is obvious that for a higher level there is higher demand for resources, the exact amount of which depends on the filter that is associated with each LoQ.

The filters are usually simple, type-specific, lossy algorithms that transform the stream by adjusting its bandwidth and timing requirements. Since they are type specific, it is necessary that the adaptive application designates or even provides them, especially in the case of a proprietary data type. In order to increase the adoption of *MobiWeb* though, we believe that the provision of several pre-installed filters for well-known data types, that the applications can use directly, will be necessary.

C. Admitting a new stream

Since *MobiWeb* is designed to provide performance guarantees to streams that require them, the incorporation of admission control is imperative. Research on this topic revealed that it is fairly difficult to provide hard guarantees [2], especially in extremely variable environments such as wireless links. Thus, our decision was to provide only base level statistical guarantees and only for *MobiWeb* aware real-time traffic.

The base requirements for a new stream to be admitted are extracted from its lowest LoQ. The proxy layer records the base requirements of all the current adaptive streams and compares their totals with the current quality of the wireless link. If the remaining resources are enough to accommodate the new stream, the admission is successful and the new stream is initiated with its lowest LoQ. Otherwise, a rejection message is sent to the application and the stream is terminated. Finally if the stream is originated from an unaware application it is always admitted, since it is expected to be a best-effort stream.

D. Priorities

In addition to the filter, each LoQ is also associated with a priority value. Whenever a LoQ is selected, all packets of the stream receive its priority value. During resource degradation, the scheduler compares the relevant priorities of all the packets currently in its queue and drops those with the lower priority. Taking the current environment in consideration, it transforms the decision for priority values for each stream from a trivial automated task to a complicated multidimensional one.

Deciding on *MobiWeb*'s priority scheme, required analyzing all different dimensions of importance that each stream can have in a certain transmission environment. Firstly, it is not uncommon for a user or an application to initiate more than one real-time stream at the same time. In the case of multiple streams from a single application the relevant priorities between streams can be predefined. From the user's point of view however, the priority values for each stream can change unpredictably according to his perceptual objectives. In addition, in a shared environment where several MHs are using the same wireless channel, the transmitting BS might face priority schemes introduced by several different applications and users.

Evaluating the conflicting requirements of each scenario, there are two different approaches for a priority scheme selection. The first uses standard predefined priority values for each data type in any environment, while the second gives the application and the user the flexibility to rearrange them as fit (chosen from a well known predetermined set). The advantages of a standardized scheme are that neither the application has to be *MobiWeb*-aware, nor the user will be forced to decide the priority of each stream. Plus, standardizing a scheme allows for worldwide compatibility. The disadvantages of such a scheme however, are that it represents effectively only the general use of all real-time media and that it prohibits the user from fully customizing the media according to his objectives.

On the other hand, the advantages of a flexible scheme are that it allows the application and the user to dynamically modify priorities. It gives the freedom of fine tuning the application performance according to the user's preferences and to the current environment consisting of contenting streams over the limited resources of the wireless channel. This freedom though can turn into a disadvantage for this scheme if an application or a user abuses it in the case of a shared channel. It is impossible to enforce good behavior, without an external scheme. Such a scheme might be a pricing policy, forcing the user to pay according to the amount of resources he uses and eliminating a malicious behavior.

The implementation of *MobiWeb* is not favoring any of the extreme solutions. Instead it favors the use of a hybrid scheme, with a tendency towards the power of adaptation and customization. It provides a default prioritization scheme for all existing data types, which can be useful for traditional real-time streams, while it permits *MobiWeb*-aware applications to alter their priorities at will. *MobiWeb*'s current scheme cannot protect from malicious behavior, but the anticipated deployment of several pricing schemes will eventually provide *MobiWeb* with the appropriate protection mechanisms.

E. Short term adaptation

In a volatile environment, like a wireless link, the decision of assigning priorities to streams can be a troublesome and error prone procedure. Therefore, *MobiWeb* provides some guidelines for a default priority assignment scheme, which can be used as the initial selection of all *MobiWeb*-aware applications.

The first recommendation is that traditional non real-time traffic, i.e. TCP and UDP, will have the same priority assigned to it permanently, at the middle of the range of priority values. Since *MobiWeb*-aware applications require protection from such best-effort traffic, it is recommended that the priority for the base LoQ to be at least one level higher. The actual value of it will depend on the relevant priority that it has compared to the remaining other *MobiWeb*-aware traffic currently on the channel. For example in a scale of 0 to 10, with best-effort traffic having a priority of 5, an appropriate decision is to assign to a base video stream a priority of 6 and to a base audio stream a priority of 7. That is because audio is usually more important than video for a user.

Once a stream advances a LoQ, a new priority value must be carefully picked for it. Considering fairness over the wireless channel, the more resource expensive a stream is the lower the priority it should have. This rule permits real-time streams to utilize resources when they are available, while forcing them to back off in favor of new or poorly supported streams. Such a configuration can be realized by decreasing the level of priority of each stream when it is ascending to a higher LoQ. For example an audio stream that starts with base priority 10 will have lower priority after 2 advancements in LoQ than a video stream that had a base priority of 9 and stayed there. At this point, the quality of the sound is expected to be much better than the quality of the video, therefore making more appealing the utilization of any spare resources in the future by the video stream first.

Another important consequence of the suggested configuration is that it gives priority to *MobiWeb*-aware applications but it still remains fair to unaware ones. In the above example, if traditional best-effort traffic has a priority of 8, then the video stream will eventually advance to a higher LoQ with lower priority than 8. This restrains the video stream from abusing the channel and advancing arbitrarily many LoQ. The best-effort traffic will first be accommodated and the remaining resources will then be available for the video stream.

F. Long term adaptation

So far we described the details of the short-term adaptation of the streams to temporary but frequent fluctuations of a wireless channel. We also mentioned that adaptation in the long-term, when resources change permanently and dramatically, is achieved by changing the LoQ of a stream and the corresponding filters. They only question left unanswered until now is *when* do the streams adapt to a different LoQ?

Whenever a new adaptive stream is initiated, the proxy layer associates two timers with it, the relaxation timer and the exploration timer. Their initial values are usually fixed and determined based on the inherent characteristics of the link (WLAN, CDPD, Cellular, etc) and its current quality (round trip delay, buffer availability, etc). The relaxation timer is activated when degradation occurs on the reception of a real-time stream. If until expiration the quality of the reception hasn't reached back the level indicated by the current LoQ for the stream, the proxy layer changes the LoQ to the next lower LoQ available for it and informs the responsible application for the change. If the stream was already at its lowest LoQ, a further degradation forces it to terminate.

The exploration timer on the other hand is responsible for the utilization of excessive resources that are available on the link. It triggers when the quality of the received stream is equal to or better than the expected one, determined by the current LoQ. As long as the quality stays at this level and the exploration timer expires, the proxy layer changes the LoQ to the next higher LoQ available and again informs the responsible application for the change. If the stream was already at its highest LoQ, additional available resources leave its performance unaffected.

The introduction of those timers intents to fine-tune the response of adaptive streams to link fluctuations. Since the performance of a real-time stream can be severely degraded when it is forced to change its LoQ, the design of *MobiWeb* is focused on performing such an adaptation only when it is absolutely necessary. This way the stream will preserve its flow smoothness by entering a stable-state, which is often more desirable to the user than a higher quality but variable stream. On the other hand, whenever spare resources become available on the link, the user would expect the stream to rapidly exploit them, which is one of *MobiWeb*'s goals.

The initialization and modification of the exploration-relaxation timer pair should preserve the effectiveness of the scheme in all situations. The conflicting requirements of each case prevent a permanently fixed value for them as a vital solution. Therefore the timers are designed to be dynamic, each of them in its way, to better reflect the reaction that a stream should have each time the channel changes.

During the initialization of the stream, the proxy layer initiates the transmission with the base LoQ. This reduces the impact that the introduction of the new stream has on the rest of the streams. After the initiation at the lowest LoQ, the stream is expected to promptly start exploring the available resources. Thus the initial value of the exploration timer should be the lowest possible as well. If resources are available, the exploration of the next level will be successful. This is determined after a period of time equal to the relaxation timer from the moment the new LoQ is selected. After this period the proxy layer is assured that the existing resources are enough to support the new LoQ. The current condition of the stream switches then from the exploration mode to the normal mode of operation. The exploration timer remains the same but the relaxation timer is reduced now by a certain predetermined amount. The reason for that will become clear later on.

As long as the stream is in its normal operation, it will continue to explore the channel until it reaches a level where there are not enough resources to accommodate a higher LoQ. Thus, an attempt to advance to this higher LoQ will trigger the relaxation timer and it will force, when it expires, the stream to back off a LoQ. This backing off has an effect on the values of both timers. Since the stream is in exploration mode, the exploration timer will be doubled so that the next trial for a higher level will occur in twice the time. Such a change gives the stream the chance to explore the resources in a later time when the conditions will be different, while it prevents the flooding of the wireless links from sources that try constantly but unsuccessfully to utilize more resources. This behavior is crucial for the performance of real-time streams, since they avoid frequent unnecessary adaptations and keep a steady smooth streaming.

Moreover, backing off during exploration results in a small increment of the relaxation timer. Since the stream's quality is reduced to a lower LoQ, it gains more robustness by letting other streams, with higher quality and smaller relaxation timers, to adapt first in the next severe degradation period. This contributes to the provision of fairness in utilization of the channel, because it forces streams with higher LoQ to retreat and surrender their resources first.

After a few unsuccessful explorations, the value of the exploration timer becomes pretty large allowing essentially the stream to enter a steady state. The stream remains there until degradation or upgrade occurs on the link, while small fluctuations are absorbed by the relaxation timer. Once a severe degradation occurs though (because of a new stream or a natural object interfering with the wireless channel), and unless another stream adapts first, the relaxation timer will eventually expire. This sets the stream to the degradation mode where it should reduce its LoQ by one level and readjust its performance and its timers. The relaxation timer is being increased as in the exploration mode, but the exploration timer is now halved. This permits the same stream that lost its resources to be the first to explore the next available when they will be released. Thus, by reducing its exploration timer to half, the stream has a better probability to be the first to attempt advancing a LoQ, considering that the exploration timer of the other streams remains large, since they remained in their steady state.

G. Handoffs

Roaming with a Mobile Host connected in a wireless environment causes a series of horizontal (e.g. cell to cell) and vertical (e.g. WLAN to Cellular to Satellite) handoffs between the different wireless interfaces. The major issue that concerns the users and *MobiWeb*'s design is the shielding of the performance of real-time applications from the transition effects of a handoff.

In *MobiWeb*, handoffs are handled as a special case of quality fluctuation of the wireless link. The traffic management layer uses most of the mechanisms described before to seamlessly adapt *MobiWeb*-aware traffic to the new environment and retain performance and smoothness intact. However, some unique features associated with handoffs cannot be handled by the general mechanisms, so they require special consideration. First, a vertical handoff leads to a new interface with different inherent characteristics. The result is demand for a different approach on manipulating the values of the relaxation and exploration timers. Thus, *MobiWeb* should be able to distinguish between simple fluctuations and vertical handoffs, since the parameters of the new environment affect the calculation of the initial and current values of the timers.

The second feature associated with handoffs is the special consideration for admission control. After a handoff the MH might end up in a cell that doesn't have enough resources to accommodate all its streams like the previous one. This can happen during a horizontal handoff in a shared medium, like a WLAN, or during vertical handoffs between two interfaces, like a WLAN and a Cellular. The general mechanisms would lead *MobiWeb* to perform admission control for all the newly arrived streams whenever a handoff occurs, as it does when a new stream is initiated. In a new environment with limited resources, admission control would result in dropping all the streams of the recently arrived MH. This is arguably not a viable solution in shielding the performance of the already established real-time applications from the handoff's transition effects.

MobiWeb's solution takes advantage of handoff notifications and does not perform admission control during handoff. The introduction of new streams make all of them compete for the channel's resources and a shortage will force the higher quality and lower priority streams to back off first, allowing the rest to perform seamlessly. The streams keep surrendering resources until a steady state is reached. In case that the resources are not enough to support the base quality of the lowest prioritized streams, some of them will be terminated for the more important among them to survive.

To handle these special issues introduced by handoffs effectively, *MobiWeb* requires handoff notifications from lower layers. A mechanism that keeps track of the current environment can be implemented in either the physical or the link layer and it can pass the notifications up to the proxy layer along with the network quality statistics [1]. In an implementation that lacks such support, *MobiWeb* still works correctly, although less effectively. The readjustment of the timers can be inaccurate, until the new environment parameters are precisely measured, and the new streams introduced by a handoff have to go through admission control with a possibility of rejection.

IV. Related Work

The notion of application adaptation over wireless channels became a very hot topic for research recently. Several major projects addressing the unique features of wireless links are in progress. WebExpress [3, 4] uses caching, document differencing and protocol overhead reduction to optimize text-based web transfers over wireless links. It uses a proxy-based scheme like *MobiWeb*, but it does not provide any enhancements for real-time traffic.

The BARWAN project [5] is an ongoing effort at Berkeley that covers aspects of wireless communications spanning the whole Internet protocol stack. The application adaptation part of BARWAN [6, 7] is focused on on-demand dynamic distillation of real-time streams and non real-time objects. It uses the notion of LoQ, but this statically reflects the wireless interface only and not the dynamic fluctuations in the quality of the channel.

The Odyssey project at CMU [8] also uses a feature similar to the LoQ. It performs dynamic adaptation of real-time streams that captures the link fluctuations, but it lacks two important features: the priority scheme and the adaptation timers that provide the robustness and preserve the smoothness of the real-time stream.

Mobiware at Columbia [9] bundles all streams from a single Mobile Host (MH) in order to manage them better while the MH roams inside a switched ATM infrastructure. It provides adaptation methods within the real-time stream bundle but without the notion of priorities, and it seems to lack an arbitration method during handoffs.

Finally, the TIMELY project at UIUC [10] uses the HPF transport protocol [11, 12]. It is the most recent effort and the only other known architecture that supports priorities between streams for intelligent management over a wireless link. HPF transmits all the real-time streams of a single application over a single connection, thus having the opportunity to prioritize packets and drop the less important of them during periods of link degradation. Its drawback is that it does not have a provision for inter-stream priorities between different applications. This severely limits its effectiveness, plus its end-to-end nature will be a major burden during deployment.

V. Conclusions

The unique characteristics of wireless links, in conjunction with the current Internet protocols, can severely impact the performance of real-time streams for interactive multimedia applications. Such applications typically impose stringent delay and jitter constraints in order to preserve the smoothness of continuous media. We introduced *MobiWeb*, an architecture designed to enhance the performance of real-time Internet streams over wireless links.

MobiWeb provides a solution capable of shielding the fluctuations in quality of the link with the aid of adaptive applications. It introduces inter-stream priorities to protect important streams during short-term fluctuations and uses a filter-based Level-of-Quality specification to facilitate stream adaptation to long-term changes in the wireless environment. The selection of the appropriate action in each instance is based on an essential feature of *MobiWeb*, the use of both relaxation and exploration timers.

MobiWeb supports enhanced services to adaptive applications, while remaining transparent and fair to unaware applications. Finally, it effectively utilizes handoff notifications from lower layers to better shield the end user from the effects of the transition to a new environment.

References

[1] G.C. Polyzos and G. Xylomenos, “Enhancing Wireless Internet Links for Multimedia Services,” *Proceedings of the 5th International Workshop on Mobile Multimedia Communications (MoMuC’98)*, Berlin, Germany, October 1998, p. 379-384.

[2] S. Shenker, “Fundamental Design Issues for the Future Internet,” *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, September 1995, p. 1176-88.

[3] B.C. Housel and D.B. Lindquist, “WebExpress: A System for Optimizing Web Browsing in a Wireless Environment,” *MobiCom’96 Demonstration Session*, November 1996.

[4] R. Floyd and B. Housel, “Mobile Web Access Using eNetwork WebExpress,” *IEEE Personal Communications*, Oct. 1998, p. 47-52.

[5] E.A. Brewer, *et al*., “A Network Architecture for Heterogeneous Mobile Computing,” *IEEE Personal Communications*, Oct. 1998, p. 8-24.

[6] A. Fox, S.D. Gribble, E. A. Brewer, and E. Amir, “Adapting to Network and Client Variability via On-Demand Dynamic Distillation,” *Proceedings of the Seventh International ACM Conference on ASPLOS, Cambridge MA*, October 1996.

[7] A. Fox, S.D. Gribble, Y. Chawathe, and E.A. Brewer, “Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perspectives,” *IEEE Personal Communications*, vol. 5, no. 4, August 1998, p. 10-19.

[8] M. Satyanarayanan, B. Noble, D. Narayanan, J.E. Tilton, J. Flinn, and K.R. Walker, “Agile Application-Aware Adaptation for Mobility,” *Proceedings of the 16th ACM Symposium On Operating Systems Principles*, vol. 31, no. 5, December 1997, p. 276-87.

[9] O. Angin, A.T. Campbell, M.E. Kounavis, and R.R.-F. Liao, “The mobiware toolkit: programmable support for adaptive mobile networking,” *IEEE Personal Communications*, vol. 5, no. 4, August 1998, p. 32-43.

[10] V. Bharghavan, L. Kang-Won, L. Songwu, H. Sungwon, L. Jin-Ru, and D. Dwyer, “The TIMELY Adaptive Resource Management Architecture,” *IEEE Personal Communications*, vol. 5, no. 4, August 1998, p. 20-31.

[11] D. Dwyer, H. Sungwon, L. Jia-Ru, and V. Bharghavan, “An Adaptive Transport Protocol for Multimedia Communication,” *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1998, p. 23-32.

[12] J.R. Li, D. Dwyer, and V. Bharghavan, “A Transport Protocol For Heterogeneous Packet Flows,” to appear in the *Proceedings of IEEE INFOCOM’99*.