Improved Quality of Service for High Demand Multimedia Applications in the Wireless ATM Environment

Mo Adda, Amanda Peart
School of Computing, University of Portsmouth, Lion Terrace, Portsmouth, UK
mo.adda@port.ac.uk, amanda.peart@port.ac.uk

Abstract

The Quality of Service (QoS) within the dynamically changing nature of wireless networks, experiences many issues, specifically in providing high and maintainable levels of QoS for high demand multimedia applications. This paper explores the solutions developed to overcome these issues and looks particularly extending Mobiware - a middleware solution currently in research. A proposed extension to Mobiware that could provide a solution to enhance QoS in dynamic wireless networks for high demand multimedia content is presented in this paper. This will illustrate how multiply user's can access a finite resource while still maintaining the required level of quality of service demanded for multimedia applications.

Keywords
Quailty of Service (QoS), middleware, wireless networks, multimedia applications.

1 Introduction

Quailty of Service (QoS) has been defined as *"The collective effect of service performance which determines the degree of satisfaction of a user of the service"* [1]. QoS is simply a mechanism for provisioning guaranteed bandwidth over a computer network for high demand content. For example, a simple home network may consist of an upstairs PC, and a downstairs living room TV, wirelessly connected to each other. The user may want to watch a digitally stored movie, located on the PC, on the downstairs TV without copying and storing the movie locally. First, the data must be streamed from the PC to the TV. Current streaming technologies (e.g. Windows Media 9 and RealVideo 9, introduced in 2002) permit near-DVD quality streaming at a rate of 500Kbps. This paper focuses on a lower resolution form of video, approx. 320x240 with a frame rate of 20fps (frames per second) (A Review of Video Streaming over the Internet n.d.). At this resolution and frame rate, video requires around 350Kbps to stream (Windows Media Encoded).

At these speeds, an IEEE 802.11b network (11Mbps) could sustain 31 simulations streams in optimal conditions, providing all streams were running a constant rate of 350Kbps. This does not include any network overhead, or any extrinsic factor affecting the quality of the network connection. The aim of QoS in this situation is to ensure each stream receives as close to 350Kbps as possible, and to insure a seamless change of bandwidth allocation to the user when necessary. The goal is to provide a dynamically changing wireless network (that is, a wireless network with devices leaving and entering the network on a ad hoc basis), with the ability to sustain every connected device running streamed multimedia applications with the required bandwidth and network conditions for, satisfactory use. The principle scenario would permit the complexity of multiple devices to simultaneously steam media.

2 The Technology

The evolution of ATM into wireless networks enables wireless technology to encompass the advantages of the QoS provided by ATM. QoS in a wireless environment, aims to guaranteed bandwidth and service to the end user, fundamentally the QoS is essential for high demand applications such as video and audio streaming. These applications require a minimal level of service to function, and seamless transition between levels of service.

This paper proposes a system that enhances the advantage of ATM's QoS and incorporates it within a wireless infrastructure. The proposal builds on the Mobiware system, which claims to provide QoS support that allows multimedia applications to operate transparently during handoff and through heavy QoS requirement fluctuations. Mobiware is a highly programmable middleware platform designed to run between the radio link layer and the application layer [2]. The proposed Intelligent Adaptive Buffer Control (iABC) will integrate low-level network protocols and end, high-level user applications. It is envisaged that this will maintain QoS within a saturated network, while permitting short-term use for additional applications. Incorporating Mobiware's 'flow adaptive policy' with application intelligence could achieve this. Mobiware guarantees the QoS though a secured bandwidth, the drawback being that this is a finite resource, iABC proposes to dynamically alter the size of the receiving applications buffer once additional demands arise. iABC provides an interesting proposition to provide all parties with their respective requests, while maintaining a high and maintainable QoS for all.

3 Adaptive Buffer Control (iABC)

Figure 1, illustrates n High priority users (H_1, H_2, …, H_n) sharing the available bandwidth, B_r, with the required QoS. For instance, each high priority user requires 350Kbytes/sec bandwidth for video streaming. During the window of opportunities, window of accepted requests (WAR), we assume that m low or medium priority users (L_1, L_2, …, L_m) have requested bandwidths for QoS of the same magnitude of the high priority request for a limited duration only. For instance, each low or medium priority user requires a video clip to get football news, and then releases the bandwidth. We denote by B_a the remaining bandwidth which B_T - nB_r, which the total bandwidth takes away the requested bandwidth by all the high priority users. In the figure below we denote by T_p, T_b, T_e and T_s the durations of the WAR, the buffering time, the buffering emptying time and the viewing or streaming time respectively. Finally we refer to the additional buffer size for each user as bf.

We analyse in this paper three schemes. During the window accept requests period low priority use the remaining bandwidth **Ba** to broadcast their requests and requirements for the streaming period. The requests could be denied, if not enough remaining bandwidth is available, or the requirements cannot be met. Each high priority user adjust dynamically its buffer with the number of requests and the remaining bandwidth $\mathbf{B_a/n}$, while maintaining a connection to the server at the $\mathbf{B_r}$ rate for QoS.

$$\frac{B_a}{n} * T_p = T_b * B_r \Rightarrow T_p = \frac{nB_r}{B_a} * T_b \qquad (1)$$

The additional buffer size for each High priority user would be

$$Bf_H = \frac{B_a}{n} * T_p = B_r * T_b \qquad (2)$$

and for each low priority user would be

$$Bf_L = \frac{B_T}{m} * T_b \qquad (3)$$

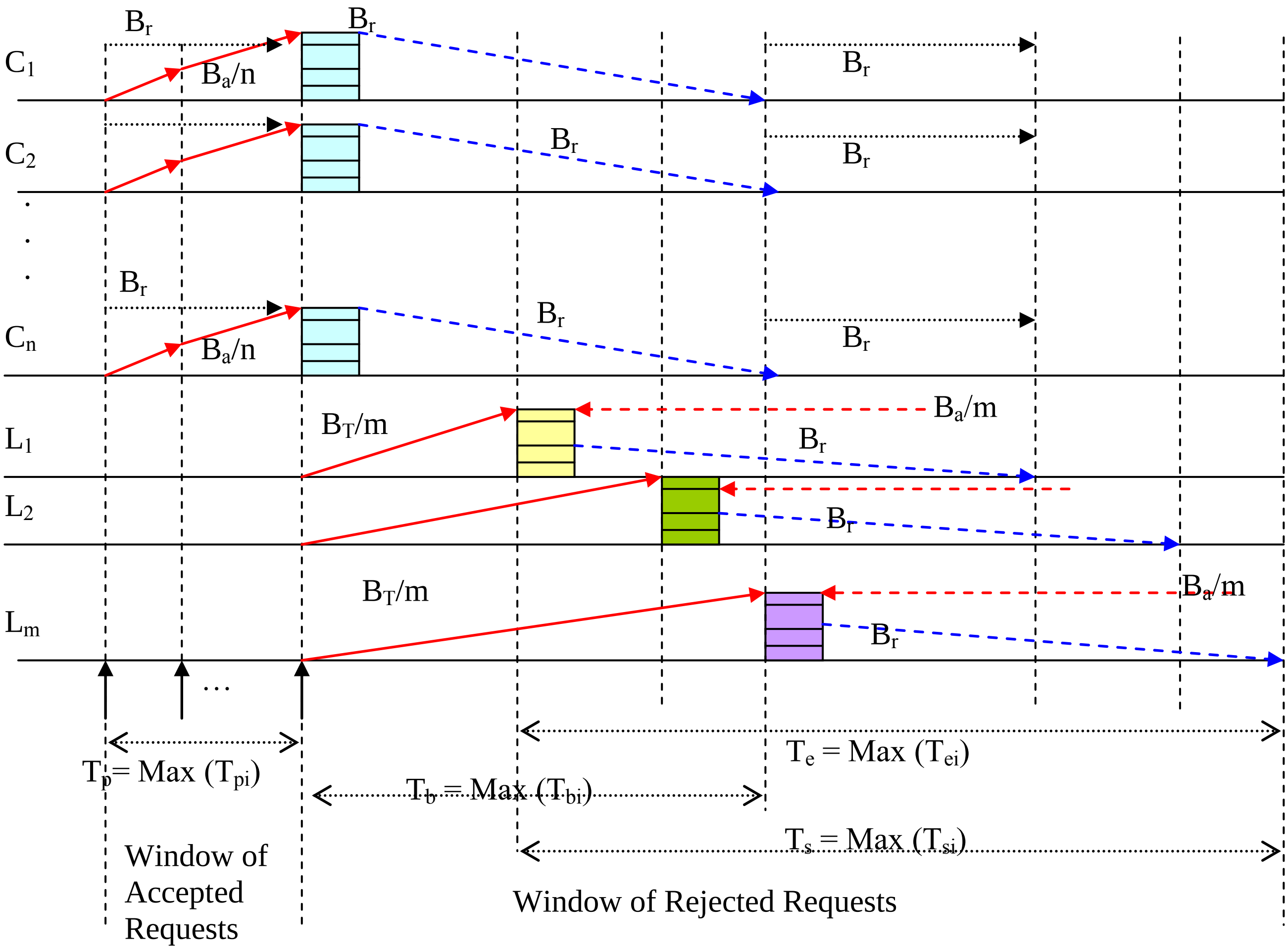

Figure 1, High priority users ($H_1, H_2, \ldots, H_n$) sharing the available bandwidth, B_r, with the required QoS.

In the three schemes if the total bandwidth is more than enough for all high and low priority users ($B_T > (m+n)\ B_r$), the viewing time starts immediately ($T_v = 0$). Otherwise, the bandwidth will be shared equally among them. However, if the total bandwidth is enough for high priority users only ($B_T <= nBr$) all requests from low priority users will be denied and the viewing time is infinity ($T_v = \infty$). In all the other cases, ($nBr < B_T < (m+n)\ B_r$), requests will be accepted with durations as computed in each case with the expense of extra buffering as follows. There are no preferences between low priority users. It is possible to implement preferences within low priority users as explained in section 4.

3.1 Fully Loaded Buffer before Streaming (FLB)

In this scenario the lower priority users start streaming, when their buffers are full. The streaming period is equal to the buffer emptying time ($T_s = T_e$). Each low priority user fills it buffer at the rate of the total shared bandwidth (B_T/m).

$$\frac{B_T}{m} * T_b = B_r * T_s \Rightarrow T_b = \frac{mB_r}{B_T} * T_s \qquad (4)$$

Combining equations 1 and 4 and after some manipulations, we deduce the duration of the viewing delay as

$$T_v = T_p + T_b = \frac{mB_r}{B_T - nB_r} * T_s \qquad (5)$$

The buffer sizes at for high and low priority users can simply be derived from using equation 4 into 2 and 3 respectively. The results are shown in the figure 2,3 and 4.

3.2 Partially Loaded Buffer before Streaming (PLB)

In this scenario the lower priority users start streaming, when their buffers is filled up to a threshold value. During the streaming, each lower priority user relies on the remaining bandwidth to keep filling it buffer at a lower rate. The streaming period is equal to the buffer emptying time ($T_s = T_e$).

$$T_b * \frac{B_T}{m} + T_s * \frac{b_a}{m} = T_s * B_r \quad \Rightarrow T_b = \frac{mB_r - B_a}{B_T} * Ts \qquad (6)$$

Using equations 1 and 6 we can deduce the viewing time as

$$T_v = T_p + T_b = (\frac{mB_r}{B_T - nB_r} - 1) * T_s \qquad (7)$$

3.3 Continuously Loaded Buffer while Streaming (CLB)

In this scenario the lower priority users start streaming, while buffering. During the streaming, each lower priority user relies on the remaining bandwidth to keep filling it buffer at a lower rate. The streaming period is equal to the filling and emptying times of the buffer ($T_s = T_b + T_e$).

$$T_b * \frac{B_T}{m} - B_r * T_b + T_e * \frac{b_a}{m} - B_r T_e = 0$$

$$T_b * \frac{B_T}{m} - B_r * T_b + (T_s - T_b) * \frac{b_a}{m} - B_r (T_s - T_b) = 0$$

$$\Rightarrow T_b = (\frac{(m+n)b_r - B_T}{nB_r}) * T_s$$

(8)

Using equations 1 and 8 we can deduce that the viewing time ($T_v = T_p$) is similar to equation 7.

3.4 Preferences between low priority users using PLB scheme

In this case we assume that low priority users have preference, and when enough bandwidth remain, some of them will use it to fulfil their QoS requirements while the remaining share the left over bandwidth.

$X = (B_T - nB_r)/B_r$

If X is greater than 1 then

$n = n + \lfloor X \rfloor$ // Low priority user is promoted to High priority
$B_a = B_T - n\,B_r$ //Recalculate the new remaining bandwidth
$m = m - \lfloor X \rfloor$ // the remaining share the new available bandwidth

else

All requests are denied.

4 Results

Results at 5000Bkps a normal IEEE 208.11a with access point for different users request a short period of the bandwidth at the MPEG rate of 350Kbps, with 10 high priority users. The size of the buffer for the PLB is lower in all cases. For instance for 6 requests during the WAR period, the buffers of the high priority and low priority users will increase to 42KBytes*60 = 2520Kbytes and 6000Kbytes for a streaming time of 60sec.

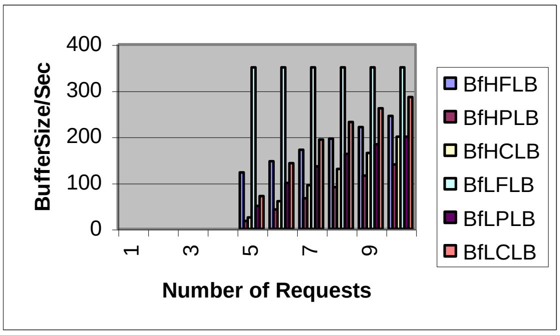

Figure 2, A normal IEEE 208.11a connection with access point for different users requesting a short period of the bandwidth at the MPEG rate of 350Kbps, with 10 high priority users.

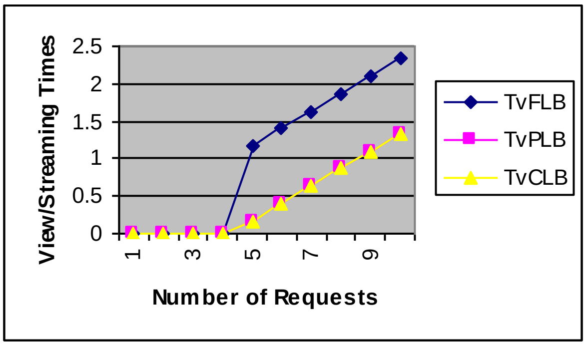

Figure 3, viewing time defined as the time taken to start viewing the video clip

The viewing time defined as the time taken to start viewing the video clip is lower for both PLB and CLB (figure 3). For instance for 6 requests, on average for a 1 minute of video streaming, the viewing time will start about .4*60sec = 24Sec.

In figure 4, we vary the number of High priority users. For a total bandwidth of 2500Kbits/sec with 5 low priority requests, we notice that as the number of high priority increases, the is insufficient bandwidth left to assign to low priority, in which case the viewing time tends to infinity. For 3 users, and 5 requests, the viewing time for PLB and CLB is about 20% if the streaming time giving 12sec for a video clips of 1-minute duration.

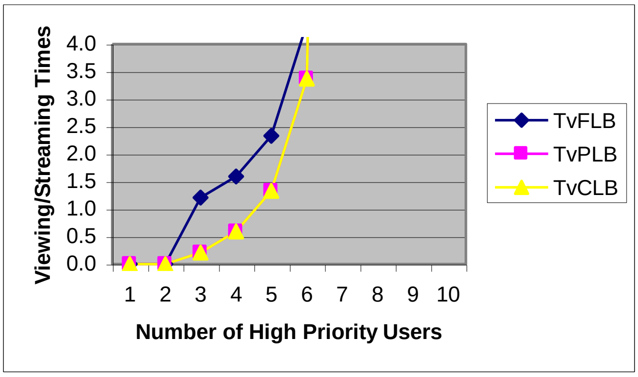

Figure 4, Viewing time with a variable number of High priority users

4 Conclusion

The Proposed solution (iABC) demonstrated a viable solution. Despite all solutions, QoS methods, additional bandwidth etc, we are trying to make a resource that is finite appear infinite. Ultimately, bandwidth can only support a certain number of devices at a certain level of quality. When that limit is reached, additional requests will simply have to wait for freed resources. Networks, like every other aspects of IT are continually and rapidly evolving. The Internet is starting to outgrow its roots, and unless new technologies like those covered in this paper can be refined and implemented on a mass and commercially viable basis, the Internet will eventually cease to function in any usable state, and will certainly be unable to cope with the future of high demand real-time content.

References

[1] Sitaraman, V. R., (n.d), Retrieved December 15th, 2004, from http://www.cse.ohio-state.edu/~jain/cis788-99/ip_qos_atm/
[2] Campell, A. T., (n.d), QoS-aware Middleware for Mobile Multimedia Communications, Retrieved December 28th, 2004, from http://comet.ctr.columbia.edu/~campbell/papers/multimedia98.pdf