Adaptive Multimedia System Architecture for Improving QoS in Wireless Networks

Amit Mahajan, Padmavathi Mundur, and Anupam Joshi

Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County, Baltimore, MD 21250, USA
{amitm1, pmundur, joshi}@csee.umbc.edu

Abstract. In this paper, we present an adaptive end-system based architecture for improving QoS in wireless networks. The proposed system adapts to fluctuating network resources by transmitting lower fidelity streams, chosen based on user preferences. Adaptation based on user preference leads to selection of data that satisfies both the network (avoids congestion) and user (better perceptual value). The system does not have any dependency on the underlying network, making its implementation possible in any wireless network.

1 Introduction

With the growth of bandwidth available in wireless networks, it is feasible to stream multimedia rich audio/video content to mobile clients. The available bandwidth has increased from 9.6 Kbps-14.4 Kbps (2G - GSM and TDMA wireless networks of 1990s) to 64 Kbps (3G networks). Increasing bandwidth is a necessary first step for accommodating real-time streaming applications, however it is not sufficient due to unpredictable and large bandwidth fluctuations experienced in wireless networks. Some minimum *quality of service (QoS)* must be provided to support smooth audio/video playback. Fluctuations in network resource availability due to channel fading, variable error rate, mobility, and handoff, makes QoS provisioning more complex in wireless networks.

In this paper, we present an adaptive end-system based architecture for improving QoS in wireless networks. We use layered-encoding feature provided by ISO (MPEG) and ITU (H.26x) video standards to achieve graceful adaptation in case of bandwidth variation. The adaptation is based on user preference in order to increase the perceptual value of the multimedia stream by making better use of available bandwidth. The end-system based architecture consists of modules at the two ends of the networks, namely the mobile client and the multimedia server. Thus, the system does not have any dependency on the underlying network, making its implementation possible in any wireless network. After the connection is established with a multimedia server, the client periodically sends feedback about bandwidth availability to the server. The server stores multiple copies of streaming data encoded at different fidelity levels. Based on the feedback and user preference, the scheduler at the server dynamically selects the appropriate copy of audio/video stream. The adaptation to the available bandwidth also provides means of avoiding the network congestion. User preferences are specified in terms of user-level QoS parameters such as resolution and frame-rate, to keep the interface simple for the user. We propose *perceptual-value based analysis* to obtain the value of data received at mobile client.

The paper is organized as follows. Related work in adaptive mobile architectures is presented in Section 2. In Section 3 we describe the architecture of the proposed adaptive system. We present the simulation environment and results in Section 4. The conclusion of the paper is given in Section 5.

2 Related Work

Several projects address the issue of bandwidth variation in wireless networks by providing an adaptive architecture. The MobiWeb project [1], is based on the proxy model in which the proxy layer at the base station (BS) intercepts TCP or UDP streams and applies the appropriate filter. Bandwidth reservation and priority scheme are used to provide continuous smooth audio/video stream. The Odyssey system [2] uses similar proxy based approach to provide smooth audio/video streaming. The system includes client components to request lower fidelity of data. Proxy based system in [3] utilizes MPEG standard features to achieve smoother video delivery. The system uses Resource Reservation Protocol (RSVP) to reserve bandwidth for high priority real-time packets. The PRAYER [4] framework is based on QoS-unaware servers and QoS-aware clients. A concept similar to home network in Mobile IP is used to achieve QoS by dynamic adaptation. Most of the proposed solutions follow proxy based approach, and also rely on the underlying network to provide services like bandwidth reservation and priority routing. Though the approach is transparent to the applications, lack of support from any intermediate network or node can render the architecture useless. For example, in case priority routing is not supported by a router on the transmission path the whole scheme will fail. Moreover, proxy based solutions have scalability problems [5], especially in case of computation intensive proxy functionality like transcoding. Most systems do not use video standard (MPEG and H.26x) features and user preference to maximize the perceptual quality of video. We propose an end-system based architecture which does not depend on either the proxies or the underlying network for additional services.

3 Adaptive Multimedia System Architecture

The block diagram of our end-system based adaptive system is shown in Figure 1. We first list the factors dictating our design and then describe each of the system component.

End-System Based Design The end-system based architecture consists of modules only at the two ends of the networks, namely the mobile client and the multimedia server. Using the mechanism explained next, the client components have the best knowledge of bandwidth available and user preferences. Client's current knowledge of bandwidth is sent to the server. The server will periodically send some control packet at higher bandwidth than reported by the client. Depending on the rate at which the client is able to receive data, any decrease or increase in bandwidth will be detected

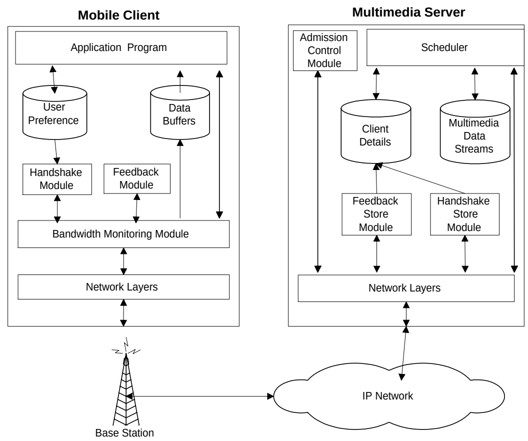

Fig. 1: Adaptive Multimedia System Architecture

by the client.The server components have the best knowledge of the levels of data fidelity stored in the database. Hence a system with participation of both client and server components should yield better results. The two end-systems (client and server) can be relatively easily modified and updated. With the current size of the Internet, it is a quantum task to effect any change in the network. Based on this intuition we have designed the system which does not have any dependency on the underlying network.

Video Standard Features The MPEG and H.26x video standards [6] offer a generalized scalable framework supporting temporal, spatial, and SNR scalability. SNR scalability allows video streams to be divided into two types of layers - base layer and enhancement layer. Multiple enhancement layers can be used to improve the quality of multimedia playback. This division offers a means of gracefully degrading the quality when the bandwidth and other resources are limited and change frequently (Figure 2). With the declining cost of storage, the multimedia server can easily store multiple streams of data encoded at different fidelity levels.

Perceptual-Value Based Analysis In our perceptual-value based analysis we determine the value of data based on user perception and not on quantity of data received. For example, viewing the slides is more important in the case of presentation, and hearing the speech is more important in the case of news. The properties of streaming data received: audio quality, resolution, color, and frame-rate, are compared with the user preferences to compute the perceptual-value. Larger perceptual-values are assigned for data that match user expectations. *Expected Data (ED)* is multimedia data (audio, base layer, and enhancement layer) user expects based on user-preference provided to the system. *Received Data (RD)* is multimedia data actually delivered to the client. *Received-Expected Match Ratio (REMR)* is defined as ratio of bytes matching the user-preference ($B_{RD \cap ED}$) and bytes of RD (B_{RD}), and is used to determine how closely does the RD match the user-preference.

$$REMR(\%) = \frac{B_{RD \cap ED}}{B_{RD}} * 100$$

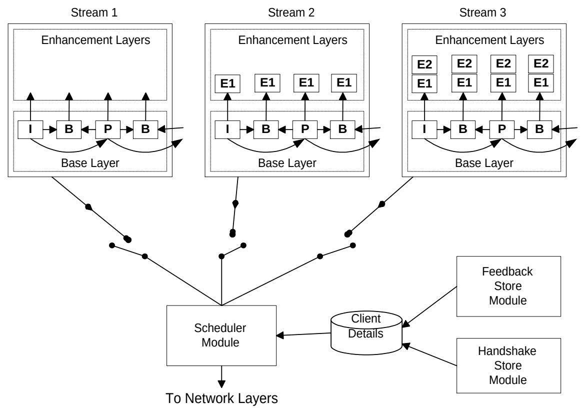

Fig. 2: Scheduling Base and Enhancement Layers

Perceptual value (PV) of the data received is the aggregate number of bytes that match the user-preference.

$$PV = \Sigma B_{RD \cap ED}$$

For example, for viewing a presentation user preference choosen is video resolution and when bandwidth falls to 32 Kbps, ED is enhancement video layer and base video layer (Table 1). When user preference are not considered RD is audio layer and base video layer . The intersection of RD and ED is base video layer and with equal bandwidth for each layer, REMR value of 50% is achieved. When adaptation takes user preference into consideration, RD is enhancement video layer and base video layer, which results in REMR value of 100%. Comparison between RD and ED is done for various user preferences to obtain the complete Perceptual-Value Table 1 for 32 Kbps bandwidth. Similar tables can be obtained for other bandwidth values.

Preference	Expected	**No user preference**		**With user preference**	
	Data	Received Data	REMR (%)	Received Data	REMR (%)
Audio	Au+BL	Au+BL	100%	Au+BL	100%
Frame-rate	BL+2EL (@ 12 fps)	Au+BL (@ 25 fps)	50%	BL+2EL (@ 12 fps)	100%
Resolution	BL+EL	Au+BL	50%	BL+EL	100%

Au: Audio Layer BL: Base Video Layer
EL: Enhancement Video Layer fps: frames per second

Table 1: Perceptual-Value of Data Received for 32 Kbps

3.1 System Components

The following are the modules that constitute the system architecture (Figure 1).

- *Client Application* accepts the client preferences in a user friendly interface. Two simple choices have to be made: the first is the preference between audio and video, and the second is the preference between frame rate and picture resolution. The client application stores these preferences in a database for later use.
- *Handshake Module* accesses the preference database and sends the information in MSG_HANDSHAKE message to the server.
- *Bandwidth Monitoring Module* keeps a check on the current network state by keeping track of the amount of data being received by the client. The information is used to determine the bandwidth available to the client device. It periodically invokes the services of feedback module to update the server about the bandwidth variation.
- *Feedback Module* is periodically invoked by bandwidth monitoring module to update the server (by MSG_FEEDBACK message) about the bandwidth available to the mobile client.
- *Admission Control Module* decides whether the multimedia server has sufficient free resources to service a new request from a client. Based on available resources, the server decides to accept or reject the connection request.
- *Handshake Store Module* processes the MSG_HANDSHAKE message received from client during the initialization phase. It then stores the client preferences received in the message in the *client details* database.
- *Feedback Store Module* processes the MSG_FEEDBACK message received from the client and stores the bandwidth availability value in the *client details* database.
- *Scheduler Module* uses data from client details database to select appropriate stream of multimedia data. Both client preferences and bandwidth available to the client are used to decide the appropriate stream. The scheduler also prepares data packets for transmission to the client. Figure 2 shows the scheduler modules switching between three streams of multimedia data based on *client details* which include user preference and bandwidth available.

The server and client interaction starts with the connection initialization phase in which the client requests streaming data from the server. The server accepts or rejects the request based on admission control. In the handshake phase the user preferences are transferred to the server and stored in client details database for future use. The handshake message, MSG_HANDSHAKE, has four bits for user preferences (audio, video, resolution, and frame rate). The available bandwidth is monitored by the client and reported to the server. The feedback message, MSG_FEEDBACK, has four values of 1 byte each for bandwidth over the previous 30 seconds (Bw30), 60 seconds (Bw60), 120 seconds (Bw60) and 180 seconds (Bw180). Thus, the size of MSG_HANDSHAKE is 29 bytes and of MSG_FEEDBACK is 32 bytes, with IP header 20 bytes and UDP header 8 bytes. The server adapts the fidelity of data in the adaptation phase and transmits the adapted video stream.

Overhead on Mobile Client Mobile devices have limited capabilities in terms of power, computation power, memory, and storage. Hence it is important to discuss the overhead of the proposed architecture. Minor changes are required in the application program and the network layer, hence the overhead introduced is negligible. User preference needs two bits of storage and memory, one bit to indicate choice between audio and video, and second bit to store choice between resolution and frame-rate. Handshake module requires few (around 50) cycles and one network packet to send the information to the server, and does not require extra storage or memory. Bandwidth Monitoring module is invoked for each received packet to calculate the current bandwidth available to the client. It stores the bytes received during the past few seconds to calculate the bandwidth available. Hence, the memory requirement is 16 bytes, four bytes each for Bw30, Bw60, Bw120, and Bw180. Computation overhead of both Feedback module and Bandwidth Monitoring module is less than 100 cycles. Hence, the system does not have much of overhead on the mobile client.

4 Simulation

To test the performance of the system architecture described in Section 3, we implemented the modules in network simulator-2 (NS-2) for simulation experiments. The setup consists of grid of 1000m by 1000m. The base stations have range of 50m and can provide maximum bandwidth of 64 Kbps to mobile clients. Base stations are placed such that mobile client can communicate with minimum one base station at any point in the topology. We used simulated H.263 streams over RTP and UDP protocols to perform the experiments. In the simulation model, multimedia server stores various combination of the audio, base video, and two enhancement video layers. Each layer requires 16 Kbps of bandwidth, and hence, the best quality stream (one audio, one base and two enhancement layers) can be streamed at 64 Kbps. We randomly vary the bandwidth available in the new cell within the range of 16-64 Kbps to mimic the real life scenario. Each simulation of 500 seconds is repeated ten times to obtain the average values used in the graphs. Randway Algorithm is used to generate mobility patterns for the client node. Three scenarios were executed on top of a simulated dynamic wireless environment. In the first scenario, no feedback mechanism is used and the server streams data at 64 Kbps. When available bandwidth declines, the client will experience long starvation periods. During the starvation periods no data is presented and the user experiences pauses or gaps in playback. Such a scenario is also observed in the standard media players and the status shown during the pause or gaps is “buffering” or “waiting for data”. Playback time is defined as the amount of time for which audio/video is played to the user and used as a metric in the simulation experiments. In the second scenario, feedback mechanism is employed without using user preferences. The server is able to adapt to the bandwidth variation and selects appropriate stream for the client. In the third scenario, both feedback mechanism and user preferences are used to dynamically schedule data transmission resulting in better perceptual values of received data for the client. Since with feedback, we are considerably reducing the starvation period, the audio/video playback is smoother for the user.

Figure 3 shows playback time corresponding to the simulation of the three scenarios. Without feedback, the playback time is considerably reduced because of the mismatch in the playback rate and the reduced bandwidth. Less playback time means more starvation resulting in breaks during playback. Feedback increases the playback time because the server is able to adapt to lower fidelity multimedia stream matching the transmission bandwidth and the playback rate. The overhead of using client preferences causes slight reduction in the playback time as shown in Figure 3. Without feedback multimedia data is played for 21% - 26% of time (see Table 2). With feedback the playback time increase to 87% - 98% of the time. Figure 4 shows perceptual-values for the three scenarios with the user preference as video and resolution. Feedback with user preference results in maximum perceptual-value for the corresponding data received among all the three scenarios. Figure 5 shows perceptual-values for the three scenarios with the user preference as video and frame-rate. The results are similar to the previous simulation. In the above two scenarios, adaptation with user preference has shown improvement of 47%-56% over adaptation without user preference. Figure 6 shows perceptual-values for the three scenarios with the user preference as audio. As the default adaptation is now same as the one explicitly chosen, the results of the two scenarios are similar. Still there is marked improvement from the base case which does not use feedback for adaptation.

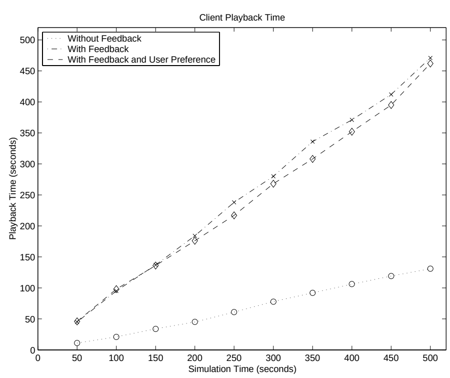

Fig. 3: Client Playback Duration

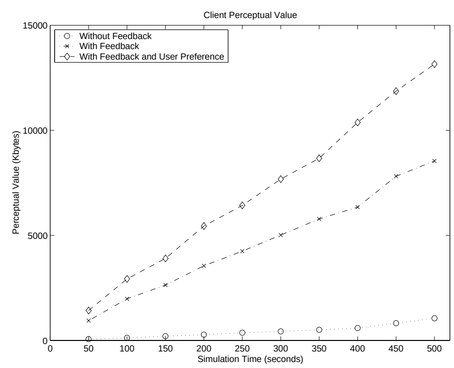

Fig. 4: Perceptual-value of Data with Video and Resolution as user preference

Time (s)	Without Feedback	With Feedback Without User Preference	With Feedback With User Preference
100	21.0	95.0	98.0
200	22.5	92.0	88.0
300	26.0	93.33	89.33
400	26.5	92.75	88.0
500	26.2	94.2	92.4

Table 2: Average playback percentage of time

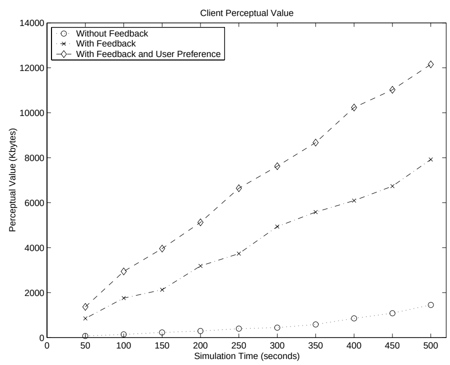

Fig. 5: Perceptual-value of Data with Video and Frame-rate as user preference

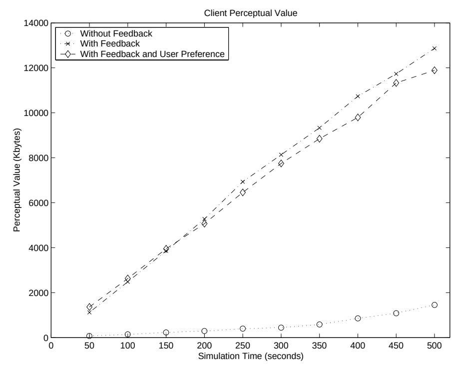

Fig. 6: Perceptual-value of Data with Audio as user preference

5 Conclusion

Significant improvement in playback time and perceptual-value of data are obtained by using the proposed adaptive multimedia system. Improvement of 47%-56% in perceptual values have been observed over the traditional adaptation techniques. The system does not have much of overhead which makes it suitable for less resourceful mobile client. The "perceptual-value based" system takes the user preference into account resulting in better adaptation. Our system does not have any dependency on the underlying network, making its implementation possible in any wireless network including the future 3G wireless networks. The system adapts to both the user preferences and network resources to improve the perceptual value of the data delivered to the user. To test the proposed system in real life conditions, we are working to implement it on test bed with actual wireless devices.

References

1. Margaritidis, M., Polyzos, G.: MobiWeb: Enabling adaptive continuous media applications over 3G wireless links. IEEE Personal Communications Magazine **7** (2000) 36–41
2. Noble, B., Satyanarayanan, M.: Experience with adaptive mobile applications in odyssey. Mobile Networks and Applications **4** (1999) 245–54
3. Bahl, P.: Supporting digital video in a managed wireless network. IEEE Communications Magazine Special Issue on Wireless Video **36** (2000) 94–102
4. Bharghavan, V., Gupta, V.: A framework for application adaptation in mobile computing environments. Computer Software and Applications Conf., Bethesda, MD (1997)
5. Joshi, A.: On proxy agents, mobility, and web access. ACM/Baltzer Journal of Mobile Networks and Nomadic Applications (MONET) **5** (2000) 233–41
6. Puri, A., Eleftheriadi, A.: MPEG-4: An object-based multimedia coding standard supporting mobile application. Mobile Networks and Applications **3** (1998) 5–32