0.1 Adaptive Multimedia Streaming over Wireless Links

Case Study: An Adaptive Audio-On-Demand Tool and Performance Evaluation using Real Testbed Experiments

Future integrated networks are expected to offer packetized voice and other multimedia conferencing services (data, images and video), to mobile users over wireless links. Wireless networks cannot easily support multimedia applications due to the media high probability, burstiness, persistence of errors and delay jitter suffered by packets. This is not only attributed to traditional variability in queuing delays due to connection multiplexing. The air interface is a channel extremely susceptible to noise, fading and interference effects and presents lower bandwidth limits than those seen on wired links. The broadcast nature of the wireless link, transient service outage, high error rates, burst error patterns demands LLC/MAC layer protocol retransmissions in order to provide the required services to the upper layers. Node mobility imposes extra requirements on routing introducing handovers, motion-induced effects, and heavy routing overhead. At the transport layer lower level services must be utilized, under minimum protocol overhead, in order to provide end-to-end QoS guarantees. Applying congestion and flow control is inherently more difficult when wireless links are used not only at end nodes, as in a multi-hop network. In this environment, the transport will need to be presented with a compressed form of the multimedia data, and thus will need to present to the upper layers a reliable service. Furthermore, seamless interaction with wired peers is generally required. Widely accepted transport protocols have been written for wired networks and will need middleware to achieve acceptable performance. Through this saga, the level of quality of service delivered to the application cannot be expected to be stable. To cope with this highly variable behavior associated with the wireless environment, it is widely recognized that protocols across layers should be able to adapt to a wide variety of situations. In supporting multimedia traffic at the application level, even though it is recognized that the details of the infrastructure support should be hidden from the user applications, the application needs to negotiate resource allocation with the network and adapt to the changing network condition by using adaptive compression. The QoS information fed back to the multimedia layer permits dynamic renegotiations of parameters so as to adjust to these changes. In this paragraph, after describing metrics and techniques that must be incorporated for enabling this kind of QoS adaptation, we proceed by describing and showing performance evaluation experiments on adaptive multimedia run on a real all-wireless (multi-hop test-bed). Such test-beds are very useful for performance evaluations. Even though large-scale experiments become very costly, smaller scale experiments are of great value since they are performed at the natural full complexity. Furthermore, the results can be used to validate and direct performance evaluation based on simulation. These simple experiments, presented in the case study, reveal the potential in perceptual improvement of a wireless multimedia conference, by applying application level monitoring and adapting to QoS techniques. The perceptual improvement is sketched in loss rate improvements combined with a graphical visualization of the possible thrashing from frequent layer switching.

0.2 How to be adaptive

As mentioned in the opening paragraph the application needs to negotiate resource allocation with the network and adapt to the changing network condition by using adaptive compression. The QoS information fed back to the multimedia layer permits dynamic renegotiations of parameters so as to adjust to these changes. This may hidden from user applications if made transparent by use of agent software or any other middle-ware e.g. in the form of an API. Adapting to layers starts at the application's flexibility to carry on its useful task (i.e. meaningful communication in a multimedia conference) in different grades of service. The network is assumed able to support the application's highest demand on light load. In a multimedia application one or more encoders may define the different grades of service, for example. CODEC technology has seen rapid development in recent years mainly due to the demand to deliver multimedia content to the personal computer through some storage media or a wired network. Current well-defined and highly used CODECs are defined in MPEG-3 (4 to be published December 1999) standards, for entertainment purposes since the compromise in quality for compression is not very obvious, and H.2/3XX for video/telephony. Future wireless mobile communication systems and networks present a different substrate under which such layering schemes may be redesigned. Important differences are: the invalidation of the assumption of a more or less reliable substrate, a much lower acceptable bit rate, and finally a stronger demand for multi-resolution (the ability to reproduce lower bit-rate streams from the original stream on the fly) and aggressive and finer layering. The latter will enable the flexibility to request different QoS support from the network, based on a compromise between the user perception and a reasonable future expectation from the network. If the application (or middleware) is to be able to switch between layers at any time, extra information need to be maintained, either encapsulated in an RTP-type packet or in a separate packet (even stream) indexed or synchronized to the data packet (or stream). This is because the application needs to be aware of the received stream characteristics. This required extra information introduces overhead which may prove damaging to performance. By combining the ability to switch between different streams (using one or more encoders) of different QoS requirements with monitoring and quantifying the underlying network conditions an adaptation strategy can be developed that will incorporate experience on how and when to choose layers.

0.2.1 Quantifying and monitoring the network conditions

Quantification of the network conditions is performed by defining metrics that can adequately describe the QoS provided by the network. The decision of switching will be based solely on their value. Multimedia applications are sensitive to lost packets, delayed packets and jitter. Fortunately, in multimedia streaming it is easy to measure the loss rate, since the expected bit rate is known a priori. Jitter can be sampled by keeping information to perform an RTP-like measurement. Monitoring of these conditions is done along the path from the sender to the receiver, usually where the ability of changing layers is desired (at end-points for application level adaptation). The sampling is done periodically, and the quantification corresponds to a past interval. This past interval will be used to predict the future. In essence, the underlying assumption here is usually that in the near future the QoS to be provided by the network can be anticipated to be similar to the QoS provided in the near past. This implies that older QoS information must be given decreasing importance. Due to the high variability of the wireless link this implies in turn that the QoS information is very time-sensitive. Furthermore, since QoS information is generally quantified in a different node than the QoS user node, feedback delays imply different importance too. Another realization related to the feedback path delay is that, sadly, when we need the QoS information the most -that is when the network conditions are highly adverse- is exactly when they are usually received late, errored or lost altogether (proportional to how symmetric are the links.) Let us now look at evaluating performance gain by using a simple adaptation strategy on built from scratch audio on demand server/client application.

0.3 Experiments on an Audio on Demand System using Application Layer Adaptation over ad-hoc, multi-hop networks

An adaptation scheme for audio applications based on a QoS Notification Programming Model is described, in which senders use QoS feedback information from the receivers to dynamically select audio encoding parameters most appropriate to the reported network conditions. The selection of audio encoding is based on the principle of media scaling. By this principle, the bit-rate (and, hence, the quality) of an audio or a video stream is varied to be consistent with the available bandwidth. The effectiveness of this tool in enhancing end user perception is based on using QoS adaptation strategies and aggressive speech layering ideas like captioning. To improve end user perception, even at the face of extremely adverse network conditions (as reported in [Bolot], the perceived audio quality drops sharply as packet loss reaches 20%), an ultimate encoding layer is introduced. A text transcription is associated with the speech stream, either real-time using a speech recognition engine or from a caption file. The text traverses the path easier (smaller packets, very low bit rate) and more reliably (e.g. using redundancy). The data can be displayed at the receiver side in a caption window or, more importantly, when the QoS information suggests switching to this bottom layer, the speech can be reproduced using the transcription with a text-to-speech synthesizer. In this way the speech stream can still be comprehensible at the receiver. Switching into this layer must be properly indicated, so that it is triggered at the point where the layer above it, would not produce comprehensible speech. Switching to this "ultimate" layer is a local decision and is inherently different with decisions based on QoS information that have traveled through the reverse path. The QoS is measured at the receiver, exactly where the decision to use the synthesizer or the received audio stream is made. By running experiments in our wireless multi-hop testbed and reporting the measured QoS provided by the network we show that the communication quality is substantially upgraded even when adverse network conditions persist for a long time. As performance metrics we define the very same metrics monitored and used in the adaptation process.

0.3.1 Application Architecture

The application follows the principles of Application Level Framing (ALF) [7]. This way, the application can take advantage of the QoS parameter information provided by the network to adapt to changes in network's behavior [4]. A network-independent QoS notification programming model is implemented to provide the audio server with the ability to monitor network conditions at the client end and react adequately when congestion occurs. The model is designed so that future evolution of network layer services will not affect its generality and is comprised of three main parts: 1) the network layer API which provides the interface to network services; 2) the Network Monitor module which collects and analyses QoS information from the network, and 3) the Application QoS Notification API which accepts this QoS information from the Network Monitor and processes it. Such separation of the model into three modules allows the application to achieve the desired network independence.

The Network Monitor provides a quality of service abstraction for the application, so that it can always assume that the network provides QoS support, while in reality it may not. Its activity consists of the following three parts: 1) it monitors the multimedia stream from the server to the receiver; 2) it measures QoS parameters of this stream; and 3) it reports this QoS information to the application.

The protocol used to stream data to the receiver is similar to RTP but contains only the necessary functionality. The server generates the sequence number and timestamp information when it sends a packet over to the client. It stores the two numbers in the audio packet's transport header. When the packet arrives to the destination, the receiver extracts these parameters from the header and passes them to the Network Monitor.

The Network Monitor uses the sequence number, the timestamp, and the local time information to determine two QoS parameters of the stream: packet loss rate lr and delay jitter jt. It divides time into measuring periods of duration $t_{mp} = 1$ sec. During each measuring period, it counts the total number of packets received n_{total}, and the number of packets lost nlost. It also records the arrival and send times of the last packet in the measuring period: $t_{LastArrival}$ and $t_{LastSend}$. The arrival time is taken to be the system time when the packet arrives to the receiver, while the send time is extracted from the packet header. At the end of every period k, the Network Monitor computes the two QoS parameters with the following simple calculations:

$$l_r(k) = \frac{n_{lost}(k)}{n_{total}(k)} \tag{1}$$

$$j_t(k) = \frac{[InterArrivalTime(k) - InterSendTime(k)]}{InterSendTime(k)} \tag{2}$$

where

$$InterArrivalTime(k) = tLastArrival(k) - tLastArrival(k-1) \tag{3}$$

$$InterSendTime(k) = tLastSend(k) - tLastSend(k-1) \tag{4}$$

In order to de-couple the jitter measurement from packet losses alternative metrics may be assumed. For example, difference in amount of playing time queued as seen by 2 chosen packets. The two parameters are then reported to the receiver application. A structural representation of the application objects and the data QoS paths can be seen in Fig. 1.

0.3.2 Source adaptation to QoS change

Upon receiving a QoS update, the server makes a decision on whether to change the current audio sampling rate or leave it intact. This decision is based upon the following heuristics:

If $lr > LR_{UpperThreshold}$ then

$$SamplingRate_{Current} = OneStepDownSamplingRate(SamplingRate_{Current})$$

$$PacketSize_{Current} = OneStepDownPacketSize(PacketSize_{Current})$$

If $lr \geq LR_{LowerThreshold}$ then

$$SamplingRate_{Current} = OneStepUpSamplingRate(SamplingRate_{Current})$$

$$PacketSize_{Current} = OneStepUpPacketSize(PacketSize_{Current})$$

If $jt > JT_{UpperThreshold}$ then

$$SamplingRate_{Current} = OneStepDownSamplingRate(SamplingRate_{Current})$$

If $jt \leq JT_{LowerThreshold}$ then

$SamplingRate_{Current} = OneStepUpSamplingRate(SamplingRate_{Current})$ whre lr is the loss rate , jt is the jitter,$LR_{UpperThreshold}$ and $LR_{LowerThreshold}$ and $JT_{UpperThreshold}$ and $JT_{LowerThreshold}$ are threshold vaules OneStep[Down/Up][SamplingRate/PacketSizze]() are functions that take a current parameter and returns the next lower/upper value of the parameter.

Figure 1:

This heuristic is based on the assumption that the primary cause of packet loss is congestion. Hence, when the audio server decreases the audio sampling rate, and therefore, its transmission rate, the packet loss should decrease and the perceived speech quality should increase. Similarly, the audio packet is decreased in size, in order improve packet loss characteristics of the channel.

0.3.3 Testbed

The multi-hop testbed used for the experiments consists of a server, a client, a gateway and a (packet level) interfering station. A Wavelan I [ref] is used for wireless networking, which has a closed configuration range of a 100 feet, works at 915Mhz and uses a CSMA/CA MAC protocol.

0.3.4 Performance Evaluation

Based On Experimental Results Our goal is both to show that by adapting to network conditions the application will have on average a higher quality of perception and that by introducing an ultimately compressed layer perception can be improved more even at most adverse network conditions. In the lab we can hear the differences. We may quantify the quality of the audio delivered in the different experiments by using two facts: 1. The loss rate (meaning both actually lost and packets that arrived after the playback point) is a good indication of perception. Theoretically, if the stream could be finely layered, the QoS directed exactly on time switching to the appropriate layer -assuming that the lower layer is an ultimate compression that will go through even at most adverse conditions- then by adapting we would achieve the highest quality possible, and an insignificant loss rate. 2. However, anomalies from frequent switching in and out of different layers would limit perception. Consequently, we move to the second fact. In order to evaluate perception a visualization of the layer switching is needed. For this reason, on each experiment we show the measured loss rate over time and a graph indicating the switching between the audio and the text stream. We show three experiments, one without using any adaptation mechanism which would require a QoS feedback path, one with using adaptation in the same environment and one with adaptation and more intense interference to emulate behavior in extremely adverse conditions.

0.3.5 Discussion of Experimental Results

Figure 2: (a): Loss Rate when no adaptation is performed, and with standard experiment interference. Audio stream is packetized in 960 byte packets and sampled at 22000 samples per second and 8 bit per sample. Interference is 'ping' packets of 40 bytes attempting to fill the channel
(b): Loss rates when adapting to QoS, and with standard experiment interference. Audio stream is packetized in 240-960 byte packets and sampled at 8-22Khz . Interference is 'ping' packets of 40 bytes attempting to fill the channel
(c):Loss rates when adapting to QoS with extremely adverse network conditions. Interference is 'ping' packets of 160 bytes attempting to fill the channel The distance between the stations here is larger and the larger propagation delays result in higher loss rates due to the MAC protocol.
(d): Visualization of text synthesizer use: switching between audio (even number) and text stream (odd number) with different TTS-thresholds in first experiment
(e): Visualization of text synthesizer use: switching between audio (even number) and text stream (odd number) with different TTS-thresholds in second experiment.
(f): Visualization of text synthesizer use: switching between audio (even number) and text stream (odd number) with different TTS-thresholds in third experiment

In Fig. 2a we can observe the loss rates measured by the application QoS monitor at the receiver side. The MAC (CSMA/CA) layer prevents any node fromconstantly transmitting, resulting in a highly variable and oscillatory loss rate for our real-time application as expected from a highly utilized wireless channel. In Fig. 2b it is clear that by employing adaptation the loss rates are reduced (averaging 13% as opposed to 40%) and a more stable channel is seen by the application (notice also the difference in scale). In Fig. 2c we show the loss rate the application suffers with very intense (packet level) interference. The audio stream is corrupted at these high loss rates averaging 77%. In this environment the ultimately compressed layer takes over. With almost cut-off loss rates of 99%, as long one packet goes through in the 2-second predefined time window and the transcription is delivered, with the redundant scheme used, and the conference is kept alive. The remaining graphs (Fig. 2d, 2e, 2f contain the same information.

They show clearer the switching between the audio and the text stream for different threshold values. A 'high; on each line indicates that the speech is reproduced from the TTS synthesizer. Different lines are shown corresponding to different thresholds from 10% (lower line) to 40%. In Fig. 2d we can observe the behavior of multiplexing the layers when the loss is oscillatory (experiment with no adaptation). In Fig. 2e the switching is much more frequent resulting in a degradation of the level of perception. Fig. 2f, where the loss rate is constantly very high, shows that our scheme produced a very high level of perception in an environment where the audio stream would not comprehensible at all. Combining these measurements we may conclude that since the loss rate is reduced and thrashing can be avoided, adaptation increases quality of perception.

References

[1] Manthos Kazantzidis, Tsuwei Chen, Yuri Romanenko, Mario Gerla, "An ultimate encoding layer for real-time packetized voice streaming and experiments over a multi-hop wireless network" - Second Annual UCSD Conference on Wireless Communications, San Diego, California, March 1999

[2] "Experiments on QoS Adaptation for Improving End User Speech Perception Over Multi-hop Wireless Networks" - Tsu-Wei Chen, Manthos Kazantzidis, Mario Gerla, Ilya Slain - Quality of Service Mini-conference ICC'99

[3] "WaveLAN -II: A High-Performace Wireless LAN for the Unlicensed Band" - Ad Kamerman and Leo Monteban - Bell Labs Technical Journal, Summer 1997 pages 118 - 133

[4] "An Extensible Framework for RTP-based Multimedia Applications" - J.Du and et al., Network and Operating System support for Digial Audio and Video May 1997 pages 53-60

[5] "QoS routing performance in multi-hop, wireless networks" - T.-W. Chen and J.T. Tsai and M. Gerla,IEEE 6th ICUPC Oct 1997

[6] "End-to-End Quality of Service Control using Adaptive Applications" - D. Sisalem -IFIP Fifth International Workshop on QoS 1997

[7] "Architectural considerations for a new generation of protocols" -D. Clark and D. Tennenhouse- Computer Communication Review vol20, num4, Sep 1990, pages 200-208

[8] "RTP: A Transport Protocol for Real-Time Applications" -H. Schulzrinne et al.- RFC 1889 Jan 1996 M. Gerla and T.-C. Tsai,

[9] "Multicluster, Mobile, Multimedia Radio Network," ACM Baltzer J. Wireless Networks, vol. 1, no. 3, 1995, pp.255-265 D. C. Cox,

[10] "Wireless Personal Communications: What Is It?" IEEE Personal Commun., Apr. 1995 A. Shen et al., "A Robust and Variable-Rate Speech Coder," Proc. Int'l, Conf. Acoustic Speech Sig. Proc. (ICASSP), vol.1, 1995, pp.249-252

[11] S. Furui, Digital Speech Processing, Synthesis, and Recognition, New York: Marcel Dekker, 1989 A. Gersho, "Advances in Speech and Audio Compression," IEEE Proc, June 1994

[12] A. Alwan, R. Bagrodia, N. Bambos, M. Gerla, L. Kleinrock, J. Short, and J. Villasenor, "Adaptive Mobile Multimedia Networks," IEEE Personal Commun., April 1996

[13] "End-To-End Arguments in System Design" by Saltzer et al ACM Trans. On Computer Systems, Nov '84

[14] "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism" by Dave Clark et al in the proceedings of SIGCOMM '92

[15] "vic: A flexible Framework for Packet Video" by Steve McCanne et al in the proceedings of ACM Multimedia conference, Nov. '95 J.-C. Bolot and A. Vega-Garcia, "The Case for FEC-Based Error Control for Packet Audio in the Internet" - ACM Multimedia Systems 1998