University of Cyprus
Department of Computer Science
Networks Research Laboratory

Adaptive Methods for the Transmission of Video Streams in Wireless Networks

Deliverable 2.1

Video streams needs and Network Adaptation Techniques

Project Title	: Adaptive Methods for the Transmission of Video Streams in Wireless Networks
Deliverable Type	: Intermediate WP Progress report (M12)

Deliverable Number	: D2.1
Title of Deliverable	: Video stream needs and Network Adaptation Techniques
Work Package Title	: Network Adaptation Techniques
Work Package Number	: WP2
Internal Document Number	:
Contractual Delivery Date	:
Actual Delivery Date	:
Author(s)	: Pavlos Antoniou Andreas Pitsillides Vasos Vassiliou
Email(s)	: paul.antoniou@cs.ucy.ac.cy Andreas.Pitsillides@ucy.ac.cy vasosv@cs.ucy.ac.cy

Abstract

The explosive growth of compressed multimedia contents and repositories available and accessible worldwide, the recent addition of new video-related standards and the ever-increasing prevalence of heterogeneous, video-enabled devices such as computer, TV, mobile phones, and PDAs have multiplied the need for efficient and effective techniques for adapting compressed videos to suit better the different constraints, capabilities, and requirements of various transmission networks, applications, and end users. Network adaptation is an emerging field that offers a rich body of knowledge and techniques for handling the huge variation of resource constraints (e.g., bandwidth, display capability, CPU speed, and power) and the diversity of user tasks which often arise in pervasive media applications. To achieve universal multimedia access, various coding tools, such as bitstream switching, video transcoding, and scalable coding, have been developed to facilitate video content adaptation across wide varieties of systems and applications. This deliverable will first give a brief overview of video stream needs as well as techniques and standard supports for video adaptation to network parameters in order to meet various application requirements and user preferences.

Keywords: Network Adaptation Techniques, Application Adaptation Techniques.

Table of Contents

List of Figures

List of Tables

1. Introduction

The task of a communication network is to provide traffic transportation services between end users by conveying data from one tier to another in a reliable and timely fashion. This task means that a network is a shared entity; its services should not be dedicated to only a pair of communicating entities but to as many entities possible during its operational lifetime. The requirement that a network should service as many users possible simultaneously highlights the importance of the 'user capacity' concept. Network planners have always strived to design networks that achieve high user capacities but have always faced the same limitation: the finite amount of resources which a network has available for utilization in order to fulfill its task. Probably, the most valuable network resource in modern networks is link capacity and its value is even greatly appreciated in wireless networks because of the inherent difficulty to achieve efficient high speed modulation schemes for data transmission over the wireless medium. The limited frequency band size and the competition among various providers for each to acquire an operation license in a small range of this band also contribute to the problem of radio resources scarcity.

Assuming abundance of all kinds of resources and under ideal circumstances, the solution to resource scarcity would be the infinite upgrade of all insufficiently dimensioned network components. In this way, no resource shortages would ever occur and consequently, infinite number of users could be successfully supported by the network. However, real world limitations rule out such a solution as the concept of e.g. infinite capacity links is clearly theoretical and not realistic. A more realizable approach to this situation is network dimensioning at the maximum possible level provided modern expertise and available means e.g. use extremely high capacity links or high processing power servers, routers and switches. Although such a solution would be feasible, two factors appear as insurmountable obstacles to its implementation. The first factor has to do with the fact that a large inter-network comprised of smaller scale sub-networks is as efficient on an end-to-end point of view as the most inefficient sub-network by which it is composed. Even if large parts of a large network are super dimensioned, the achieved performance would still be lower than expected if smaller parts of the network are still limited by low resource availability. Since upgrading all existing networks is infeasible, this solution turns out impractical. The second prohibitive factor is the high degree of resource waste that would occur in such a super-dimensioned network since for most of the time the resources available in the network would remain unused and high under-utilization levels would be experienced. Many studies in the past have shown that congestion conditions in a network are not a constant phenomenon but rather occur at discrete time points during network's operational lifetime as a result of the unpredictable fluctuations of the amount of the offered load.

Bearing these in mind, the turn to intelligent resource management solutions seems inevitable. On the other hand a QoS Adaptation scheme can be implemented in order to address this problem. An effective adaptation scheme can be based on the observation that several applications can operate with acceptable performance when the provided QoS fluctuates within a range of values. By taking advantage of this fact, a methodology can be designed to adjust the QoS the network offers to an application in order to achieve better resource management and increase network capacity when possible, by allowing more concurrent user sessions. Apart from the above beneficial effect, the adaptation of delivered QoS (where this is possible) contributes to reducing the probability of call forced termination. It is well recognized that a trade-off exists between minimizing call blocking and forced termination probabilities. While a

network operator strives for good performance in both cases, a choice must be inevitably made between them and it is widely acknowledged (especially from user perspective) that minimization of forced termination probability is far more desirable.
In this deliverable we will refer to video architecture providing taxonomy of video applications. Furthermore we investigate some challenges in video streaming as well as video stream needs.
Moreover we will study Network Adaptation Techniques (NATs). The basic requirements of NATs are (1) to provide accurate information on the network load, (2) to distinguish between core congestion and wireless link errors, (3) to recognize a change in the possible bandwidth due to changes in the wireless link conditions, and (4) to adapt accordingly the transmission rate at the source.
These decisions will be based on feedback obtained by the receiver. We will examine the possibility of extracting useful information from feedback about the objective quality at the receiver. We expect that the estimation of objective quality will be made possible with the right feedback since the objective quality has a direct relationship with the nature of errors, the actual information content and the capability of concealment and reconstruction at the wireless mobile terminal.
The qualitative differentiation of errors is required so that we reduce the transmission rate only when it is justifiable (e.g. when we have heavy core congestion). In order for the estimations of network load and the type and quality of errors to be precise we need proper feedback from the receiver end of the stream.
The content adaptation techniques working in cooperation with the network adaptation techniques must be able to adapt the content to the necessary transmission rate without having to regenerate the information. Transmitting the information in multiple layers (one main and many sub-layers) which are added at the receiving end is deemed the most appropriate for cases of extreme conditions.
We introduced a new approach in creating layers (in the previous deliverable D1.2) which is based (1) on the estimation of the reduction in the objective quality if a layer is not sent and (2) on how easy it is for this piece of information to be reconstructed at the receiving end using already transmitted information. Adaptation is then possible with selective transmission of a subset of all layers which will reconstruct the stream at the best possible quality.
The classification of information blocks according to their contribution in the final objective quality provides another dimension in the protection of layers with FECs. We are provided with the advantage of applying better protection (more powerful codes) to the most important information and less or no protection to other layers. This way we introduce a control on the amount of overhead data we inject in the stream.
The existence of temporary memory at the receiver gives us the opportunity (1) to send information at rates larger than the required reconstruction rate when this is feasible, and (2) to employ retransmission techniques in cases of errors. However, we never have equal capabilities (computational power, memory, power autonomy) for all terminals in a network. Therefore we not only have the core and wireless network limitations, but also the limitations of the terminal of each user. The adaptation techniques that will be mentioned below will include provisions for these limitations as well.

2. Video Architecture

Video has been an important media for communications and entertainment for many decades. Initially video was captured and transmitted in analog form. The advent of

digital integrated circuits and computers led to the digitization of video, and digital video enabled a revolution in the compression and communication of video. Video compression became an important area of research in the late 1980's and 1990's and enabled a variety of applications including video storage on DVD's and Video-CD's, video broadcast over digital cable, satellite and terrestrial (over-the-air) digital television (DTV), and video conferencing and videophone over circuit-switched networks. The growth and popularity of the Internet in the mid-1990's motivated video communication over best-effort packet networks. Video over best-effort packet networks is complicated by a number of factors including unknown and time-varying bandwidth, delay, and losses, as well as many additional issues such as how to fairly share the network resources amongst many flows and how to efficiently perform one-to-many communication for popular content.

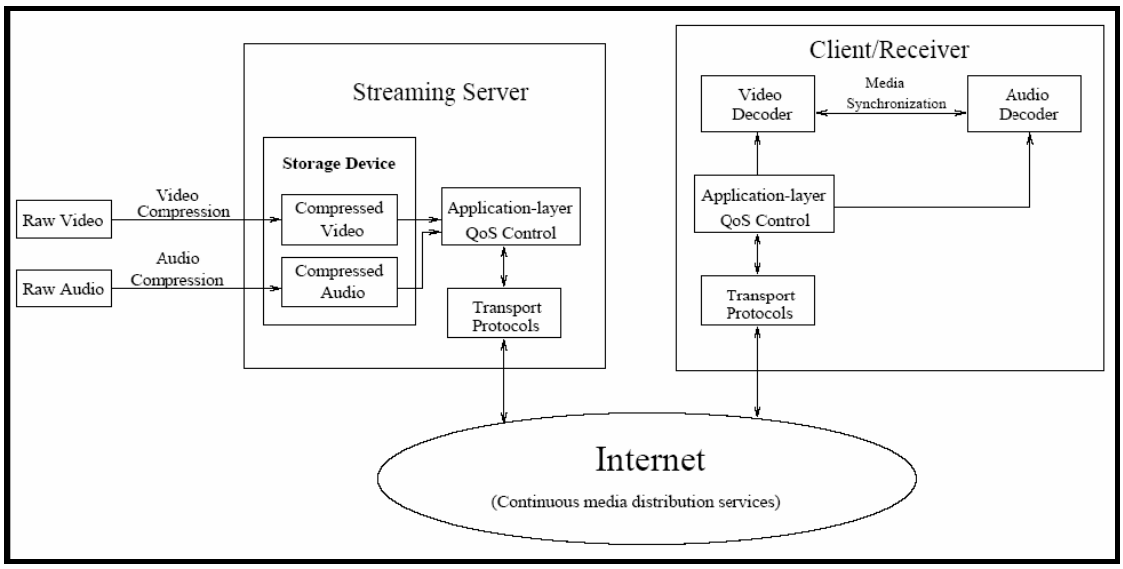

Figure 1: Video Streaming Architectures.

Figure 1 illustrates a video streaming architecture. Raw video and audio data are pre-compressed by video compression and audio compression algorithms and then saved in storage devices. Upon the client's request, a streaming server retrieves compressed video/audio data from storage devices and then the application-layer QoS control module adapts the video/audio bit-streams according to the network status and QoS requirements. After the adaptation, the transport protocols packetize the compressed bit-streams and send the video/audio packets to the Internet. Packets may be dropped or experience excessive delay inside the Internet due to congestion. For packets that are successfully delivered to the receiver, they first pass through the transport layers and then are processed by the application layer before being decoded at the video/audio decoder. To achieve synchronization between video and audio presentations, media synchronization mechanisms are required.

We continue by providing a brief overview of the diverse range of video streaming and communication applications. Section 4 identifies the three fundamental challenges in video streaming: unknown and time-varying bandwidth, delay jitter, and loss. Finally, Section 5 spots additional challenges due to the mobile environment.

3. Taxonomy of Video Applications

There exist a very diverse range of different video communication and streaming applications, which have very different operating conditions or properties. There exist

very diverse ranges of video communication. Video communication applications may be unicast, multicast, broadcast or anycast. The video may be pre-encoded (stored) or real-time encoded (e.g. videoconferencing applications). The communication channel may be static or dynamic, reserved or not, packet switched or circuit switched, may support some quality of service or may only provide best effort service. The specific properties of a video communication application strongly influence the design of the system. We continue by briefly discussing these properties.

3.1 Point-to-point, multicast, broadcast and anycast communications

Probably the most popular form of video communication is one-to-many (basically one-to-all) communication or broadcast communication, where the most well known example is broadcast television. Broadcast wastes bandwidth by sending the data to the whole network. It can also needlessly slow the performance of client machines because each client must process the broadcasted data whether or not the service is of interest. The main challenge for broadcasting is the scalability problem. Receivers may experience different channel characteristics, and the sender must cope with all the receivers. Another common form of communication is point-to-point or one-to-one communication, e.g. videophone and unicast video streaming over the Internet. In point-to-point communications, an important property is whether or not there is a back channel between the receiver and sender. If a back channel exists, the receiver can provide feedback to the sender which the sender can then use to adapt its processing. Unicast wastes bandwidth by sending multiple copies of the data. Another form of communication with properties that lie between point-to-point and broadcast is multicast. Multicast is a one-to-many communication, but it is not one-to-all as in broadcast. An example of multicast is IP-Multicast over the Internet. To communicate to multiple receivers, multicast is more efficient than multiple unicast connections (i.e. one dedicated unicast connection to each client), and overall multicast provides many of the same advantages and disadvantages as broadcast. The anycasting communication paradigm is designed to support server replications to easily select and communicate with the best server, according to some performance or policy criteria, in a group of content-equivalent servers.

3.2 Real-time encoding versus pre-encoded (stored) video

Video may be captured and encoded for real-time communication, or it may be pre-encoded and stored for later viewing. Interactive applications are one example of applications which require real-time encoding, e.g. videophone, video conferencing, or interactive games. In many applications video content is pre-encoded and stored for later viewing. The video may be stored locally or remotely. Examples of local storage include DVD and Video CD, and examples of remote storage include video-on-demand (VOD), and video streaming over the Internet (e.g. as provided by RealNetworks and Microsoft). Pre-encoded video has the advantage that it does not require a real-time encoding constraint, which enables more efficient encoding. On the other hand, it provides limited flexibility as, for example, the pre-encoded video can not be significantly adapted clients that support different display capabilities than that used in the original encoding.

3.3 Interactive versus Non-interactive Applications

Interactive applications have real-time data delivery constraints. The data sent has time bounded usefulness, after this time the received data is useless. *Figure 2*

illustrates a brief classification of packet video applications based on delay and loss requirements over packet switching network according to the ITU-T recommendation G.1010. This presents the classification of performance requirements. Various applications can be mapped onto axes of packet loss and one-way delay. The size and shape of the boxes provide a general indication of the limit of delay and information loss tolerable for each application class. The following classes of applications can be recognized:

- Interactive video applications. They need a few milliseconds of transfer delay such as conversational voice and video, interactive games, etc.
- Responsive video applications. Typically, these applications response in few seconds, so that human does not need to wait for a long time, such as voice and video messaging, transactions, Web, etc.
- Timely video application. The transfer delay can be about some second, such as streaming audio and video.
- Non-critical video application. The transfer delay is not a critical for those applications, such as audio and video download service.

From loss point of view, we can find two types of applications:

- Error sensitive video applications such as highly compressed video.
- Error insensitive video applications such as non-compressed video.

The loss has a direct impact on the quality of the information finally presented to the user, whether it is voice, image, video or data. In this context, loss is not limited to the effects of bit errors or packet loss during transmission, but also includes the effects of any degradation introduced by media coding.

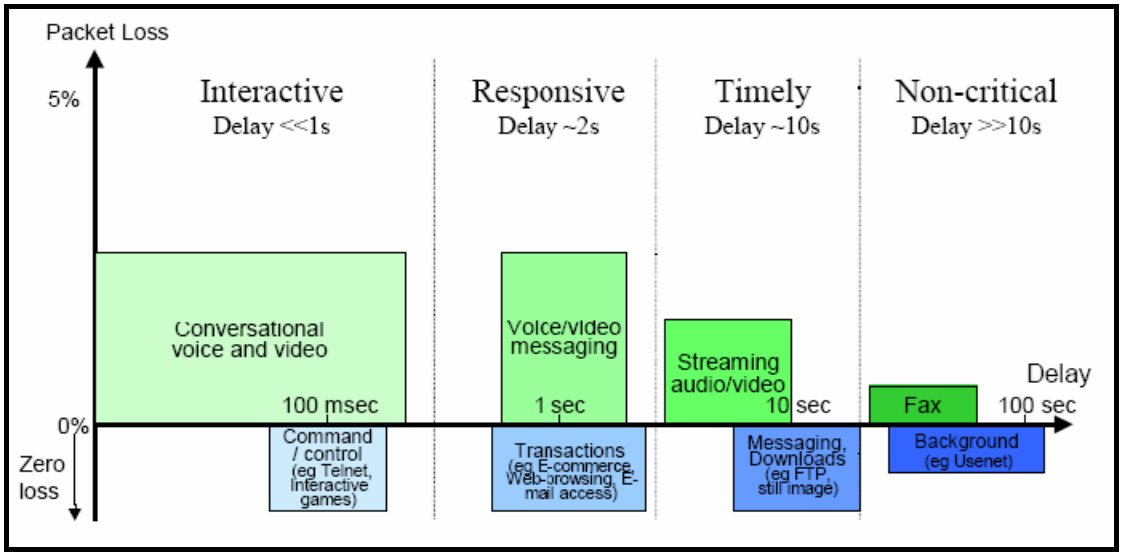

Figure 2: Classification of user application according to loss and delay requirements.

3.4 Static versus Dynamic Channels

The quality of the delivered video over networking environment is affected by the nature and the characteristic of the communication channel such as bandwidth, delay, and loss. These parameters may be static or dynamic. In the static channel, the bandwidth, delay, and loss are known in advance and bounded, whereas in the dynamic channel, it is difficult to determine their upper bound. Examples of static channels include ISDN (which provides a fixed bit rate and delay, and a very low loss rate) and video storage on a DVD. Examples of dynamic channels include

communication over wireless channels or over the Internet. Video communication over a dynamic channel is much more difficult than over a static channel. Furthermore, many of the challenges of video streaming, as are discussed later, relate to the dynamic attributes of the channels.

3.5 Constant-bit-rate (CBR) or Variable-bit-rate (VBR) Channel

Some channels support CBR, for example ISDN or DTV, and some channels support VBR, for example DVD storage and communication over shared packet networks.

3.6 Packet-Switched or Circuit-Switched Network

Packet-switched networks, such as Ethernet LANs and the Internet, are shared networks where the individual packets of data may exhibit variable delay, may arrive out of order, or may be completely lost. Alternatively, circuit-switched networks, such as the public switched telephone network (PSTN) or ISDN, reserve resources and the data has a fixed delay, arrives in order, however the data may still be corrupted by bit errors or burst errors.

3.7 Quality of Service (QoS) Support

QoS is a term used to express that the network provides some performance guarantees, e.g. guarantees on throughput, maximum loss rates or delay. Network QoS support can greatly facilitate video communication, as it can enable a number of capabilities including provisioning for video data, prioritizing delay-sensitive video data relative to other forms of data traffic, and also prioritize among the different forms of video data that must be communicated. Internet does not provide any QoS support, and it is often referred to as Best Effort (BE), since the basic function is to provide simple network connectivity by best effort packet delivery. Different forms of network QoS that are under consideration for the Internet include Differentiated Services (DiffServ) and Integrated Services (IntServ).

4. Challenges in Video Streaming – Video stream needs – Application Adaptation Techniques

This section discusses some of the basic approaches and key challenges in video streaming. The three fundamental problems in video streaming are briefly highlighted in the following three subsections.

Probably the most straightforward approach for video delivery of the Internet is by something similar to a file download, but we refer to it as video download to keep in mind that it is a video and not a generic file. Specifically, video download is similar to a file download, but it is a large file. This approach allows the use of established delivery mechanisms, for example TCP as the transport layer or FTP or HTTP at the higher layers. However, it has a number of disadvantages. Since videos generally correspond to very large files, the download approach usually requires long download times and large storage spaces. These are important practical constraints. In addition, the entire video must be downloaded before viewing can begin. This requires patience on the viewers part and also reduces flexibility in certain circumstances, e.g. if the viewer is unsure of whether he/she wants to view the video, he/she must still download the entire video before viewing it and making a decision.

Video Delivery via Streaming
Video delivery by video streaming attempts to overcome the problems associated with file download, and also provides a significant amount of additional capabilities. The basic idea of video streaming is to split the video into parts, transmit these parts in succession, and enable the receiver to decode and playback the video as these parts are received, without having to wait for the entire video to be delivered. Video streaming can conceptually be thought to consist of the follow steps:

1) Partition the compressed video into packets
2) Start delivery of these packets
3) Begin decoding and playback at the receiver while the video is still being delivered

Video streaming enables simultaneous delivery and playback of the video. This is in contrast to file download where the entire video must be delivered before playback can begin. In video streaming there usually is a short delay (called pre-roll delay and usually is on the order of 5-15 seconds) between the start of delivery and the beginning of playback at the client. Video streaming provides a number of benefits including low delay before viewing starts and low storage requirements since only a small portion of the video is stored at the client at any point in time. The length of the delay is given by the time duration of the pre-roll buffer, and the required storage is approximately given by the amount of data in the pre-roll buffer.

4.1 Basic Problems in Video Streaming

There are a number of basic problems that afflict video streaming. In the following discussion, we focus on the case of video streaming over the Internet since it is an important, concrete example that helps to illustrate these problems. Video streaming over the Internet is difficult because the Internet only offers best effort service. That is, it provides no guarantees on bandwidth, delay jitter, or loss rate. Specifically, these characteristics are unknown and dynamic. Therefore, a key goal of video streaming is to design a system to reliably deliver high-quality video over the Internet when dealing with unknown and dynamic:

- Bandwidth
- Delay and jitter
- Loss rate

The bandwidth available between two points in the Internet is generally unknown and time-varying. If the sender transmits faster than the available bandwidth then congestion occurs, packets are lost, and there is a severe drop in video quality. If the sender transmits slower than the available bandwidth then the receiver produces sub-optimal video quality. The goal to overcome the bandwidth problem is to estimate the available bandwidth and than match the transmitted video bit rate to the available bandwidth. Additional considerations that make the bandwidth problem very challenging include accurately estimating the available bandwidth, matching the pre-encoded video to the estimated channel bandwidth, transmitting at a rate that is fair to other concurrent flows in the Internet, and solving this problem in a multicast situation where a single sender streams data to multiple receivers where each may have a different available bandwidth.
The end-to-end delay that a packet experiences may fluctuate from packet to packet. This variation in end-to-end delay is referred to as the delay jitter. Delay jitter is a

problem because the receiver must receive/decode/display frames at a constant rate, and any late frames resulting from the delay jitter can produce problems in the reconstructed video, e.g. jerks in the video. This problem is typically addressed by including a playout buffer at the receiver. While the playout buffer can compensate for the delay jitter, it also introduces additional delay.

The third fundamental problem is losses. A number of different types of losses may occur, depending on the particular network under consideration. For example, wired packet networks such as the Internet are afflicted by packet loss, where an entire packet is erased (lost). On the other hand, wireless channels are typically afflicted by bit errors or burst errors. Losses can have a very destructive effect on the reconstructed video quality. To combat the effect of losses, a video streaming system is designed with error control. Approaches for error control can be roughly grouped into four classes: (1) forward error correction (FEC), (2) retransmissions, (3) error concealment, and (4) error-resilient video coding and are called Application Adaptation Techniques (AATs).

The three fundamental problems of unknown and dynamic bandwidth, delay jitter, and loss, are considered in more depth in the following section. Each sub-section focuses on one of these problems and discusses various approaches for overcoming it.

4.2 Application Adaptation Techniques

4.2.1 Transport and Rate Control for Overcoming Time-Varying Bandwidths

Congestion is a common phenomenon in communication networks that occurs when the offered load exceeds the designed limit, causing degradation in network performance such as throughput. Useful throughput can decrease for a number of reasons. For example, it can be caused by collisions in multiple access networks, or by increased number of retransmissions in systems employing such technology. Besides a decrease in useful throughput, other symptoms of congestion in packet networks may include packet losses, higher delay and delay jitter. As we have discussed in Section 4.1, such symptoms represent significant challenges to streaming media systems.

To avoid the undesirable symptoms of congestion, control procedures are often employed to limit the amount of network load. Such control procedures are called rate control, sometimes also known as congestion control. It should be noted that different network technologies may implement rate control in different levels. Nevertheless, for inter-networks involving multiple networking technologies, it is common to rely on rate control performed by the end-hosts.

4.2.1.1 Rate Control for Streaming Media

For environments like the Internet where little can be assumed about the network topology and load, determining an appropriate transmission rate can be difficult. Nevertheless, the rate control mechanism implemented in the Transmission Control Protocol (TCP) has been empirically proven to be sufficient in most cases. Being the dominant traffic type in the Internet, TCP is used for the delivery of web-pages, emails, and some streaming media. Rate control in TCP is based on a simple "Additive Increase Multiplicative Decrease" (AIMD) rule. Specifically, end-to-end observations are used to infer packet losses or congestion. When no congestion is inferred, packet transmission is increased at a constant rate (additive increase).

Conversely, when congestion is inferred, packet transmission rate is halved (multiplicative decrease).

4.2.1.2 Streaming Media over TCP

Given the success and ubiquity of TCP, it may seem natural to employ TCP for streaming media. There are indeed a number of important advantages of using TCP. First, TCP rate control has empirically proven stability and scalability. Second, TCP provides guaranteed delivery, effectively eliminating the much dreaded packet losses. Therefore, it may come as a surprise to realize that streaming media today are often carried using TCP only as a last resort, e.g., to get around firewalls. Practical difficulties with using TCP for streaming media include the following. First, delivery guarantee of TCP is accomplished through persistent retransmission with potentially increasing wait time between consecutive retransmissions, giving rise to potentially very long delivery time. Second, the "Additive Increase Multiplicative Decrease" rule gives rise to a widely varying instantaneous throughput profile in the form of a saw-tooth pattern not suitable for streaming media transport.

4.2.1.3 Streaming Media over UDP

Both the retransmission and the rate control mechanisms of TCP possess characteristics that are not suitable for streaming media. Current streaming systems for the Internet rely instead on the best-effort delivery service in the form of User Datagram Protocol (UDP). This allows more flexibility both in terms of error control and rate control. For instance, instead of relying on retransmissions alone, other error control techniques can be incorporated or substituted. For rate control, the departure from the AIMD algorithm of TCP is a mixed blessing: it promises the end of wildly varying instantaneous throughput, but also the proven TCP stability and scalability.
Recently, it has been observed that the average throughput of TCP can be inferred from end-to-end measurements of observed quantities such as round-trip-time and packet losses. Such observation gives rise to TCP-friendly rate control that attempts to mimic TCP throughput on a macroscopic scale and without the instantaneous fluctuations of TCP's AIMD algorithm. One often cited importance of TCP-friendly rate control is its ability to coexist with other TCP-based applications. Another benefit though, is more predictable stability and scalability properties compared to an arbitrary rate control algorithm. Nevertheless, by attempting to mimic average TCP throughput under the same network conditions, TCP friendly rate control also inherits characteristics that may not be natural for streaming media. One example is the dependence of transmission rate on packet round-trip time.

4.2.2 Playout Buffer for Overcoming Delay and Jitter

It is common for streaming media clients to have a 5 to 15 second buffering before playback starts. Critical to the performance of streaming systems over best-effort networks such as the Internet, buffering provides a number of important advantages:

1. Jitter reduction: Variations in network conditions cause the time it takes for packets to travel between identical end-hosts to vary. Such variations can be due to a number of possible causes, including queuing delays and link-level retransmissions. Jitter can cause jerkiness in playback due to the failure of same samples to meet their presentation deadlines, and have to be therefore skipped or delayed. The use of buffering effectively extends the presentation deadlines for all media samples, and in most cases, practically eliminates playback jerkiness due to delay jitter. The benefits

of a playback buffer are illustrated in *Figure 3*, where packets are transmitted and played at a constant rate, and the playback buffer reduces the number of packets that arrive after their playback deadline.

2. Error recovery through retransmissions: The extended presentation deadlines for the media samples allow retransmission to take place when packets are lost, e.g., when UDP is used in place of TCP for transport. Since compressed media streams are often sensitive to errors, the ability to recover losses greatly improves streaming media quality.

3. Error resilience through Interleaving: Losses in some media streams, especially audio, can often be better concealed if the losses are isolated instead of concentrated. The extended presentation deadlines with the use of buffering allow interleaving to transform possible burst loss in the channel into isolated losses, thereby enhancing the concealment of the subsequent losses.

4. Smoothing throughput fluctuation: Since time varying channel gives rise to time varying throughput, the buffer can provide needed data to sustain streaming when throughput is low. This is especially important when streaming is performed using TCP (or HTTP), since the server typically does not react to a drop in channel throughput by reducing media rate.

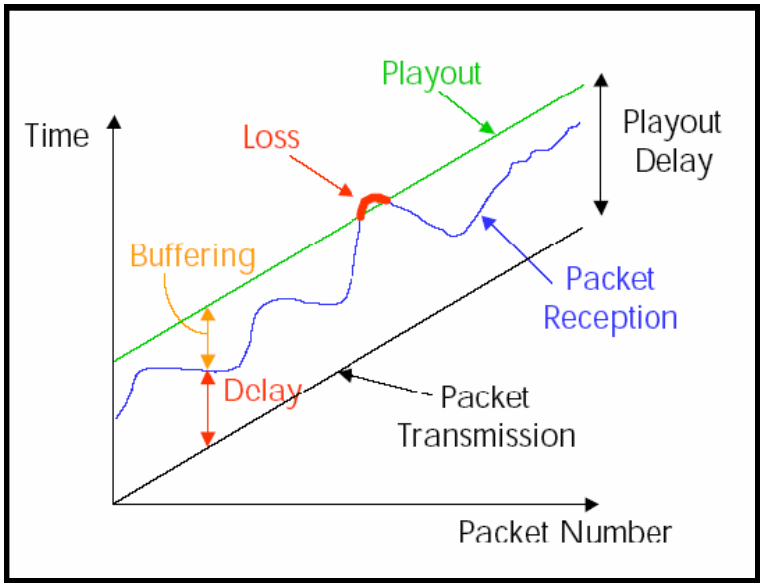

Figure 3: Effect of playout buffer on reducing the number of late packets.

The benefits of buffering do come at a price though. Besides additional storage requirements at the streaming client, buffering also introduces additional delay before playback can begin or resume (after a pause due to buffer depletion). Adaptive Media Playout (AMP) is a new technique that enables an valuable tradeoff between delay and reliability.

4.2.3 Error Control for Overcoming Channel Losses

The third fundamental problem that afflicts video communication is losses. Losses can have a very destructive effect on the reconstructed video quality, and if the system is not designed to handle losses, even a single bit error can have a catastrophic effect.

A number of different types of losses may occur, depending on the particular network under consideration. For example, wired packet networks such as the Internet are afflicted by packet loss, where congestion may cause an entire packet to be discarded (lost). In this case the receiver will either completely receive a packet in its entirety or completely lose a packet. On the other hand, wireless channels are typically afflicted by bit errors or burst errors at the physical layer. These errors may be passed up from the physical layer to the application as bit or burst errors, or alternatively, entire packets may be discarded when any errors are detected in these packets. Therefore, depending on the interlayer communication, a video decoder may expect to always receive "clean" packets (without any errors) or it may receive "dirty" packets (with errors). The loss rate can vary widely depending on the particular network, and also for a given network depending on the amount of cross traffic. For example, for video streaming over the Internet one may see a packet loss rate of less than 1 %, or sometimes greater than 5-10 %. A video streaming system is designed with error control to combat the effect of losses. There are four rough classes of approaches for error control: (1) retransmissions, (2) forward error correction (FEC), (3) error concealment, and (4) error-resilient video coding. The first two classes of approaches can be thought of as channel coding approaches for error control, while the last two are source coding approaches for error control. These four classes of approaches are discussed in the following four subsections. A video streaming system is typically designed using a number of these different approaches.

4.2.3.1 Retransmissions

In retransmission-based approaches the receiver uses a back-channel to notify the sender which packets were correctly received and which were not, and this enables the sender to resend the lost packets. This approach efficiently uses the available bandwidth, in the sense that only lost packets are resent, and it also easily adapts to changing channel conditions. However, it also has some disadvantages. Retransmission leads to additional delay corresponding roughly to the round-trip-time (RTT) between receiver sender and receiver. In addition, retransmission requires a back-channel, and this may not be possible or practical in various applications. In many applications the additional delay incurred from using retransmission is acceptable, e.g. Web browsing, FTP, telnet. In these cases, when guaranteed delivery is required (and a backchannel is available) then feedback-based retransmits provide a powerful solution to channel losses. On the other hand, when a back channel is not available or the additional delay is not acceptable, then retransmission is not an appropriate solution. There exist a number of important variations on retransmission-based schemes. For example, for video streaming of time-sensitive data one may use delay-constrained retransmission where packets are only retransmitted if they can arrive by their time deadline, or priority-based retransmission, where more important packets are retransmitted before less important packets.

4.2.3.2 Forward Error Correction

The goal of FEC is to add specialized redundancy that can be used to recover from errors. For example, to overcome packet losses in a packet network one typically uses block codes (e.g. Reed Solomon or Tornado codes) that take K data packets and output N packets, where N-K of the packets are redundant packets. For certain codes, as long as any K of the N packets are correctly received the original data can be recovered. On the other hand, the added redundancy increases the required bandwidth by a factor of N/K. FEC provides a number of advantages and disadvantages.

Compared to retransmissions, FEC does not require a back-channel and may provide lower delay since it does not depend on the round-trip-time of retransmits. Disadvantages of FEC include the overhead for FEC even when there are no losses, and possible latency associated with reconstruction of lost packets. Most importantly, FEC-based approaches are designed to overcome a predetermined amount of loss and they are quite effective if they are appropriately matched to the channel. If the losses are less than a threshold, then the transmitted data can be perfectly recovered from the received, lossy data. However, if the losses are greater than the threshold, then only a portion of the data can be recovered, and depending on the type of FEC used, the data may be completely lost. Unfortunately the loss characteristics for packet networks are often unknown and time varying. Therefore the FEC may be poorly matched to the channel -- making it ineffective (too little FEC) or inefficient (too much FEC).

4.2.3.3 Error Concealment and Error Resilient video coding

The basic goal of error concealment is to estimate the lost information or missing pixels in order to conceal the fact that an error has occurred. The key observation is that video exhibits a significant amount of correlation along the spatial and temporal dimensions. On the other hand the goal of error-resilient video coding is to design the video compression algorithm and the compressed bitstream so that it is resilient to specific types of errors. We referred to these two techniques in the previous deliverable (D1.2) so we can avoid saying more about this issue.

4.3 Additional Challenges due to the mobile environment

When the streaming path involves both wired and wireless links, some additional challenges evolve. The first challenge involves the much longer packet delivery time with the addition of a wireless link. Possible causes for the long delay include the employment of FEC with interleaving. For instance, round-trip propagation delay in the 3G wireless system is in the order of 100 ms even before link-level retransmission. With link-level retransmission, the delay for the wireless link alone can be significant. The long round-trip delay reduces the efficiency of a number of end-to-end error control mechanisms: the practical number of end-to-end retransmissions is reduced, and the effectiveness of schemes employing RPS and NewPred is also reduced. The second challenge is the difficulty in inferring network conditions from end-to-end measurements. In high-speed wired networks, packet corruption is so rare that packet loss is a good indication of network congestion, the proper reaction of which is congestion control. In wireless networks however, packet losses may be due to corruption in the packet, which calls for stronger channel coding. Since any end-to-end measurements contain aggregate statistics across both the wired and wireless links, it is difficult to identify the proper cause and therefore perform the proper reaction.

In the future, we will have access to a variety of mobile terminals with a wide range of display sizes and capabilities. In addition, different radio-access networks will make multiple maximum-access link speeds available. Because of the physical characteristics of cellular radio networks, the quality and, thus, the data rate of an ongoing connection will also vary, contributing to the heterogeneity problem. A related problem is how to efficiently deliver streamed multimedia content over various radio-access networks with different transmission conditions. This is achievable only if the media transport protocols incorporate the specific characteristics of wireless links, such as delays due to retransmissions of corrupted data packets.

Furthermore, in cellular networks, mobile devices often change base stations (handover) which they use for communicating with the network. Changing a cell during streaming causes a short break in the transmission and some data packets to be delayed or even lost. Also, available network resources may vary between the cells and thus the handovers may result in a drop of bandwidth if there are not enough resources available for streaming.

5. Network Adaptation Techniques (NATs)

Research towards a universal adaptation solution is the focus of the work conducted by several academic groups and commercial companies in order to achieve ubiquitous access to Internet multimedia content. The major issues to be addressed are the diversity of the multimedia content coupled with the variety of Internet connections utilized to access it.

There are two extreme solutions to this problem, which, however, can be easily disqualified. First, having multiple devices and multiple connections, one for each medium type, is far from an optimal solution. Users prefer to operate as few devices as possible, but they also have specific priorities for some of them (e.g., weight and power issues for mobile devices). Similarly, the other extreme solution would require one device to be capable of presenting every type of Internet content. This approach is also disqualified for similar reasons, in addition to form factor and cost issues that the production of such a device would introduce.

Solutions between these two extremes attempt to bridge this gap through adaptation. Instead of having each medium stored in multiple representations, each matching the characteristics of a probable end-device, the source can provide only one or a few representations and rely on the adaptation functionality to deliver the content in the appropriate form at the receiver. In addition, devices with transmission and display characteristics that prevent them from accessing certain types of media are able to do so by having the representation of the content adapted to a form suitable for them, with as small an impact to its perceptual value as possible. Thus, users gain access to a great variety of media, virtually with any possible combination of connection and device type. The adaptation process is flexible enough to both hide the adaptation details from the typical user and give the power user the means to customize the adaptation process according to his or her needs, through an explicit adaptation policy.

5.1 Locating the Adaptation Mechanisms

There are three different strategies for locating the adaptation mechanism on the end-to-end path from the source to the destination: (i) at the source (source-driven), (ii) at the destination (receiver-driven), and (iii) in between, somewhere in the network, with some special locations having special properties, e.g., at the boundary between the wireless and the wired parts of a network.

5.1.1 Source – driven Adaptation

The first strategy puts the mechanism at the source and requires periodic quality feedback from the receiver, as *Figure 4* shows. A typical implementation utilizes a (reliable or unreliable) signalling channel in the reverse direction, through which the receiver periodically transmits reports on the traffic that reaches it. These reports include the measured bandwidth, error rate, loss rate, average delay and average jitter of the packets that arrived at the receiver. The source evaluates the reports in order to

identify significant changes in the quality of the end-to-end path. If these changes are large enough to justify a change in the adaptation process, a new set of adaptation parameters are used or even different, more appropriate mechanisms are utilized. In the case of improvement of the communication path, the source upgrades the quality of the stream it transmits, while in the case of degradation, it throttles the transmission down (graceful degradation), or even temporarily pauses the transmission until the congestion dissolves.

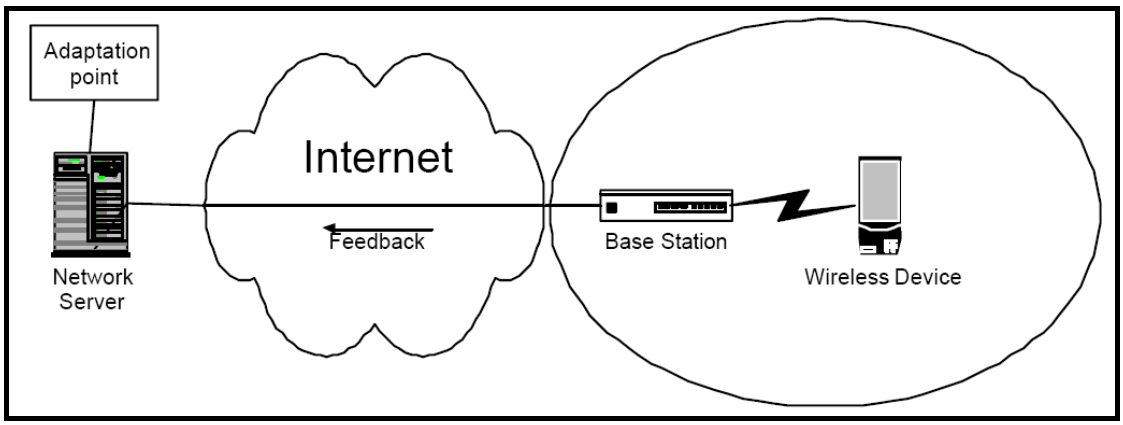

Figure 4: Source – driven adaptation.

Several existing schemes are utilizing this approach, mainly for its simplicity [3, 4, 5, 6, 7, 8, 9, 10, 11]. It can be easily implemented using existing signalling protocols, like RTP [12] and RSVP [13], although there are also solutions using proprietary protocols, like SCP [3, 14]. Since the feedback mechanism is at the transport layer, this strategy can be easily deployed over any kind of packet-switching network. However, the disadvantages are not negligible. Both the source and the receiver have to be altered in order to implement the feedback-based adaptation mechanism. While this is a moderate task for the receiver, it is an extremely costly and timely process when it comes to legacy Internet servers. In addition, having the source concurrently evaluating feedback from (potentially many) receivers substantially increases the amount of resources required, particularly because of the volume of information that needs to be processed. Moreover, in cases where the end-to-end distance is fairly large, the delay introduced through the reporting packets can severely affect the performance of the solution. This can be more profound when wireless access is involved, where the frequency and the severity of changes in the wireless channel are high. If the report reaches the source too late to inform about a potential congestion problem, the source will already have flooded the congested network, failing to adapt effectively to the link quality fluctuations.

Finally, putting the adaptation mechanism in the source precludes the utilization of this strategy in multicasting scenarios. Multicasting strives for each receiver to accept the stream with the best possible quality. Moving the adaptation mechanism at the source means that the same stream will reach all receivers and that it will be adapted to a sub-optimal quality, which the receiver with the fewer capabilities is able to accept. In an attempt to resolve the unfairness, the source might decide to transmit the stream with higher quality. In this case though, the unfairness moves towards the less capable receivers that are unable to receive any content at all. In conclusion, the simplicity of the source adaptation solution makes it efficient for simple cases with small variability, but cannot stand as a viable solution under extremely variable conditions or when multicasting is involved.

5.1.2 Receiver – driven Adaptation

The second strategy in locating the adaptation mechanism is to have it reside at the receiver, as depicted in *Figure 5*. The source remains unchanged and continues to transmit the same quality content to the receivers. The receiver, upon reception of the content, transforms it to a suitable form that it is capable of presenting. This scheme requires minimal changes at the receiver and can be extremely efficient in customizing the content to meet exactly his capabilities. It is mostly effective in situations where the transmission characteristics of the end-to-end path are less of a concern than the limitations of the displaying device [15]. For example, a PDA, connected with a WLAN to the Internet, can have a video stream scaled down in color depth and re-sampled to a lower resolution in order to match the characteristics of the device's small display.

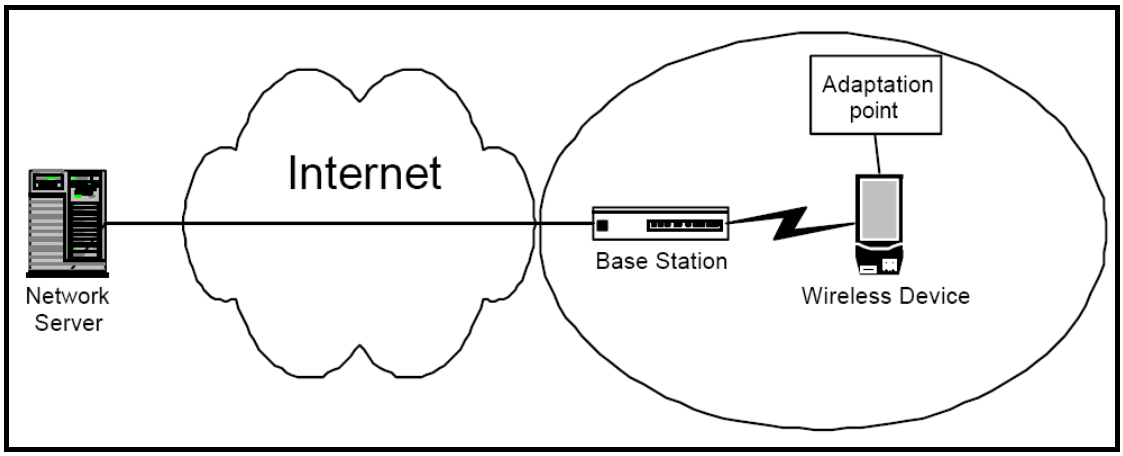

Figure 5: Receiver – driven adaptation.

The reverse situation, however, unveils the limitations of this approach. In situations where the transmission characteristics of the end-to-end path are the point of consideration, rather than the displaying ones, this solution performs poorly. Having a desktop PC connected to the Internet using a low speed modem, for example, renders the receiver adaptation mechanism useless, since it is residing after the critical part of the end-to-end path; the adaptation mechanism is applied to the stream after the stream has already congested the low-speed modem link. Therefore, whatever transformation it applies, it cannot prevent the ISP's modem from dropping a significant amount of packets of this stream which overflowed the link. Unfortunately, the occasions where the transmission characteristics are of greater importance constitute the majority of cases, significantly limiting the applicability of this solution as a generic adaptation mechanism. Another characteristic, which also hinders the wide acceptance of this approach, is the usually high complexity of the adaptation mechanisms. Since a significant proportion of the devices used to access the Internet have limited CPU power, memory, energy, and storage, implementing such a resource-demanding process on them might not be feasible or very efficient. In particular, the complexity of algorithms for transformations of text and images are fairly low, but transcoding algorithms for audio and video are complex and demanding, making their implementation on small and not so powerful devices, such as PDAs and mobile phones, extremely difficult.

5.1.3 Proxy Adaptation

The third strategy in locating the adaptation mechanism is a compromise between the two extremes discussed above. It places the mechanism within the end-to-end path, at an intermediate node identified as the most appropriate for performing the most effective adaptation [16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28].

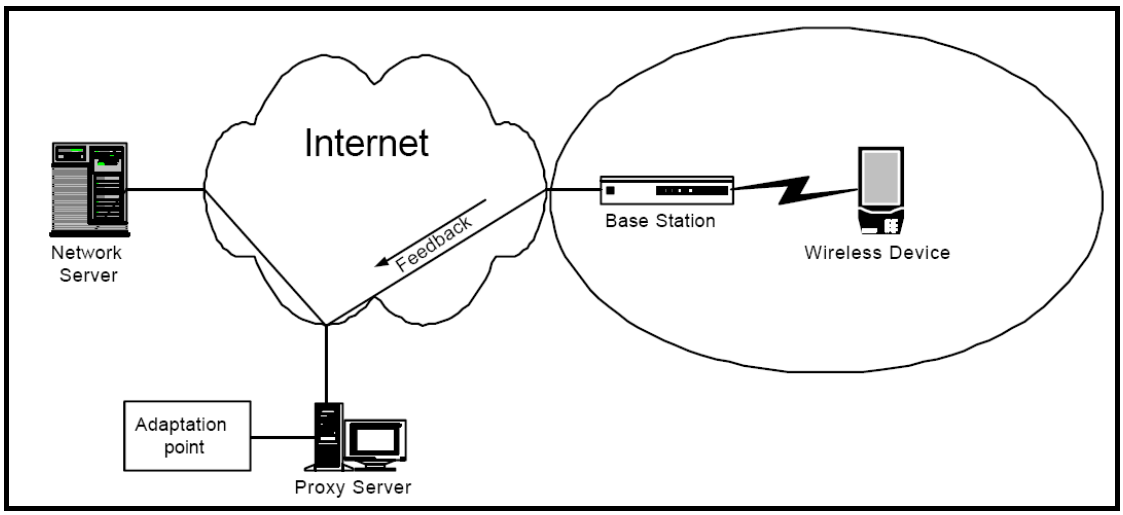

Figure 6: Proxy adaptation.

The intermediate node, usually denoted as proxy or gateway (see *Figure 6*), receives instructions from the receiver prior to the stream's initiation, regarding the parameters of the adaptation process. It then intercepts the stream coming from the source and transforms it according to the user's preferences, before passing it on towards him. There are several advantages that come with the adoption of the proxy solution. First, the proxy can be relocated and can be located at the most critical position in the end-to-end path. We saw earlier that the stationary location of the adaptation mechanism in the previous adaptation solutions led to inefficiencies under different topological scenarios. These problems are alleviated with the flexibility of the proxy architecture. In a multicasting scenario, the major concern is bandwidth conservation and reception of the best possible stream from each receiver. To satisfy such a contradicting requirement, the multicasting tree allows the adaptation mechanisms, filters in this case [29, 30, 31], to move up and down the tree structure [32, 33, 34, 35, 36]. The algorithm tries to move the filter as close to the source as possible until it reaches a node where the children have incompatible requirements.

As another example, consider the wireless access to the Internet case where a stationary device uses a wireless link to connect with a router (with a wireless interface), which in turn uses a wired connection to the Internet. In this case, the last (wireless) hop is expected to be the bottleneck, which indicates that the proxy will perform ideally if it resides at the router or as close to it as possible (see *Figure 7*).

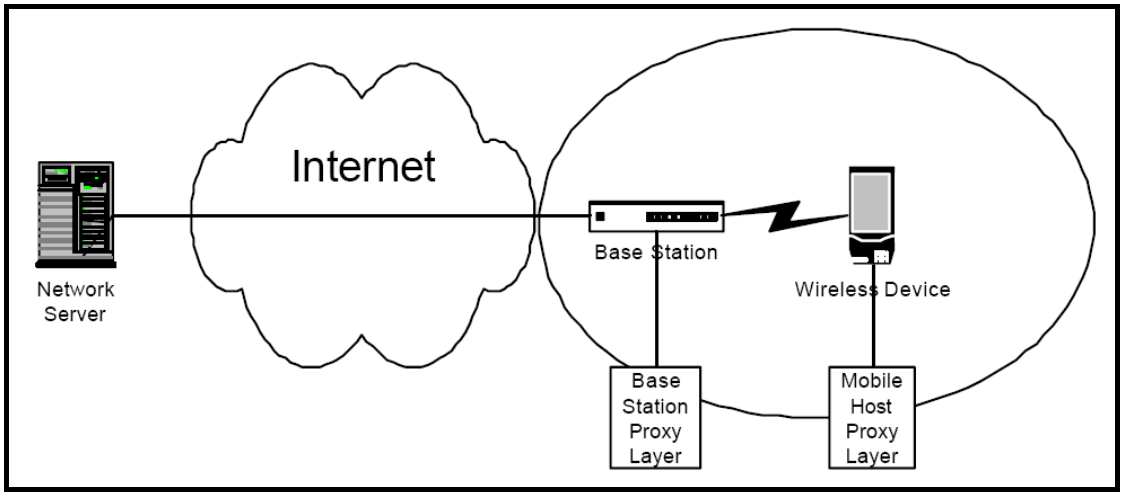

Figure 7: Last hop Proxy adaptation.

Finally, consider a mobile multimedia PDA accessing the Internet while roaming into a hierarchically structured cell array [37, 38]. In this case, the proxy functionality doesn't have to be at the last hop. Instead, locating it at the macro-cell controller will yield similar performance, while avoiding the burden of relocating it every time the mobile device hands-over to a new micro-cell (see *Figure 8*). Therefore, the flexibility of locating the adaptation mechanism at the best position in the end-to-end path is an important advantage of the proxy solution over the other two. Moreover, the proxy solution has the ability to react to changes with minimal delay and with near perfect accuracy. A monitoring mechanism can continuously scan the performance of the bottleneck link and immediately react to any significant changes.

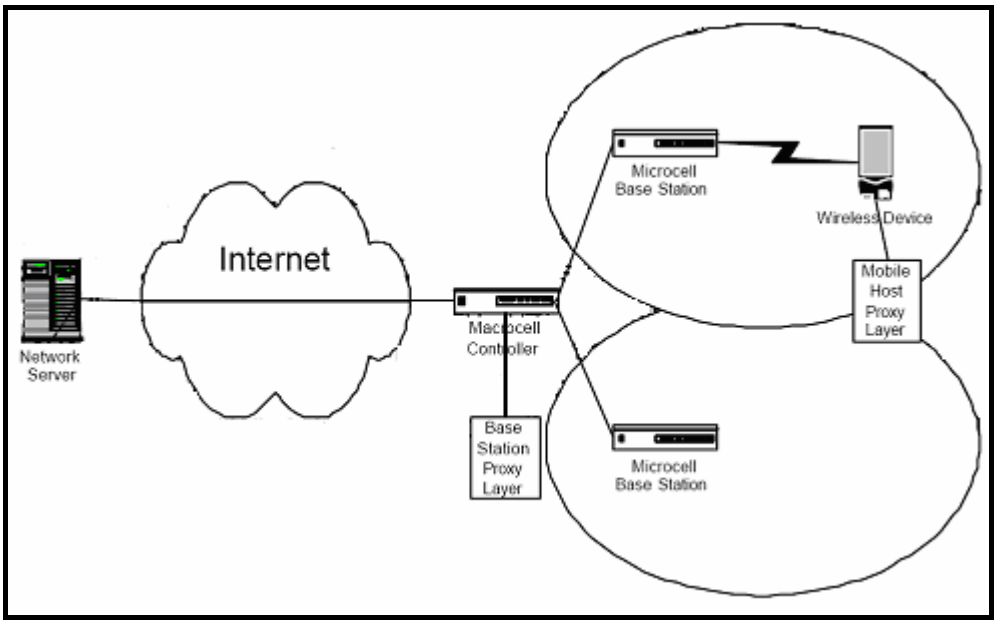

Figure 8: Macrocell Controller Proxy adaptation.

This way the proxy solution is able to adapt the multimedia streams efficiently, even in situations where unpredictable and highly variable wireless links are involved.
Naturally, the proxy solution also presents some disadvantages. First, having an intermediate intercepting the stream raises security issues. The (usually) third party operating the proxy must be trusted by the receiver and the source. In the absence of a

trusted third party the proxy architecture has no way of adapting the multimedia stream. In addition, the third party would typically require to be compensated for the services it provides and the resources it employs to perform the adaptation for the receiver. Therefore, accounting mechanisms should be incorporated in the proxy solution in order to keep track of the amount of resources utilized. Finally, the complexity of the proxy architecture is significantly higher than that of the other two solutions and requires gateways with powerful CPUs and a lot of memory.
The degree of the receiver's participation in the adaptation process can dictate the applicability and the effectiveness of the proxy adaptation scheme.

5.1.4 Comparison of the Adaptation Approaches

We attempt now to quantify the effectiveness of each adaptation solution. We emphasize their strengths and weaknesses and identify the fields where they perform best. Comparing their ease in deployment, the receiver – driven adaptation approach requires the fewer changes in the current Internet infrastructure. Only the receiver is aware of the adaptation process and all modifications apply to a single node in the network. The source adaptation approach requires changes in only two nodes, but one of them is the source providing the content. Since the modification of a legacy Internet server is an expensive and time-consuming process, content providers will be extremely reluctant to adopt this solution. Finally, the proxy adaptation approach demands adaptation awareness from some (or many) of the nodes on the end-to-end path, especially in multicasting scenarios. Modifying intermediate nodes and adding simple filtering functionality is significantly less of a burden than changing a legacy content provider, but the extent of the changes makes the proxy solution similarly expensive.
Comparing CPU processing power requirements, the receiver adaptation approach is again the less demanding. However, in cases where the receiver is resource poor, even the low complexity of this scheme can have a great negative impact. As *Table 1* describes, the complexity of the source adaptation approach increases proportionally to the increase in requested streams. The source adapts streams that are directed to several unrelated receivers, thus the streams have a low correlation between them, since it is improbable that they will compete for the resources of the same limited link. Therefore, the events that force a stream to adapt will most probably leave the remainder of the streams transmitted from the same source unaffected.

Adaptation Method	*Growth*
Source – driven	Linear
Receiver – driven	Constant
Proxy (all streams)	Exponential
Proxy (single stream)	Additive

Table 1: CPU Processing power growth with traffic.

On the other hand, when we consider one proxy feeding receivers on the same end IP network, the proxy adaptation approach is at the opposite end of source adaptation with respect to complexity. Here, all streams have to compete over the same path and adapting one directly affects the performance of the others. The schemes that force all streams over the same link to adapt whenever fluctuations in the quality of the link occur have an exponential increase in their complexity as the number of streams increases. However, the proxy solution can significantly lower its complexity if prioritization of the streams and ordering of the application of adaptation is used. In

this case adaptation occurs on only one stream at a time, on-demand and as long as additional resources need to be freed or are available. With this approach the resource utilization goals are achieved, the streams satisfy their demands according to the priorities set, while the adaptation complexity is kept to a minimum.

Adaptation Method	*Effectiveness*
Source – driven	Reversibly proportional to delay
Receiver – driven	Low, Constant
Proxy	High, Constant

Table 2: Effectiveness with increasing end-to-end delay.

Another important characteristic of adaptation solutions is their adaptation effectiveness, measured as the reaction time after a change in the quality of the link. *Table 2* illustrates that the proxy adaptation solution retains the same effectiveness regardless of the end-to-end delay. When the adaptation point is strategically located close to the bottlenecked link, the reaction time is kept to minimum and it is unrelated to the distance from the source. The source adaptation approach on the other hand is directly related to this distance. An increase in the end-to-end delay quickly decreases the effectiveness of this solution since the source can no longer react in time. This allows the stream to flood the bottleneck link and to force a significant amount of dropped packets. Finally, the receiver adaptation approach is responsible for the adaptation of the stream after its transmission. Therefore, its effectiveness remains at a constant low value, since it is unrelated to the end-to-end distance or the location of the bottleneck link.

The last concept that we introduce in this comparison is the range of applicability of each solution. The receiver adaptation approach is the most limited one, since it performs well only when the display characteristics of the device are the concern rather than the transmission characteristics of the network path. Since this is the case only for a small percentage of the cases, the receiver adaptation approach cannot be considered as a generic adaptation solution (even though this is basically the current situation in the Internet with inflexible sources and essentially no support for adaptation in the network). The source adaptation approach applies to all kinds of situations, except from multicasting scenarios where the purpose of multicasting is defeated, as explained previously. In addition, for wireless access to the Internet, the overhead associated with mobility and hand-offs is increased compared to the receiver adaptation approach, since the new end-to-end path has to be re-evaluated before the streams are successfully adapted to the new conditions. The proxy adaptation approach performs exceptionally well in all situations and is ideal for both multicasting (assuming multiple proxies are possible if required) and wireless access scenarios. However, a drawback of this solution in the case of mobility and with the placement of the proxy as close to the wireless link as possible is the increased network overhead incurred in the case of high frequency of hand-offs. The reason for this is the need to transfer the proxy functionality and the current state between different routers every time a hand-off occurs. Overall, the three adaptation approaches provide a fair trade-off between complexity and quality of adaptation. The receiver adaptation approach provides limited effectiveness for small complexity. The source adaptation approach is frequently effective while keeping the complexity at a reasonable level. Finally, the proxy adaptation approach is remarkably effective and efficient for most scenarios, but it typically induces increased complexity.

5.2. Adaptation Policies

Locating the adaptation mechanism at the right place accounts for only half the effort required to achieve efficient and effective adaptation. The second half consists of the policies associated with the adaptation mechanism in conjunction with the link or path conditions in order to attain the best possible outcome.

Different types of media require different adaptation policies to perform optimally. The adaptation mechanism should be aware of the adaptation pattern that it should follow for every type of medium in any given link conditions. For example, a video stream performs better with long-term stability rather than with frequent fluctuations in its quality. Users prefer a stable video signal with sub-optimal quality rather than a higher quality at times, but unstable one. When video is transmitted over a variable wireless link, the adaptation mechanism should opt for maintaining a fairly steady transmission rate (in application units, e.g., frames per second) and not attempt to explore the link for available resources too often. On the other hand, downloading a file is optimized by reducing the overall transmission time. Thus, the adaptation mechanism should continuously attempt to discover and exploit any available resources over the link on behalf of this stream.

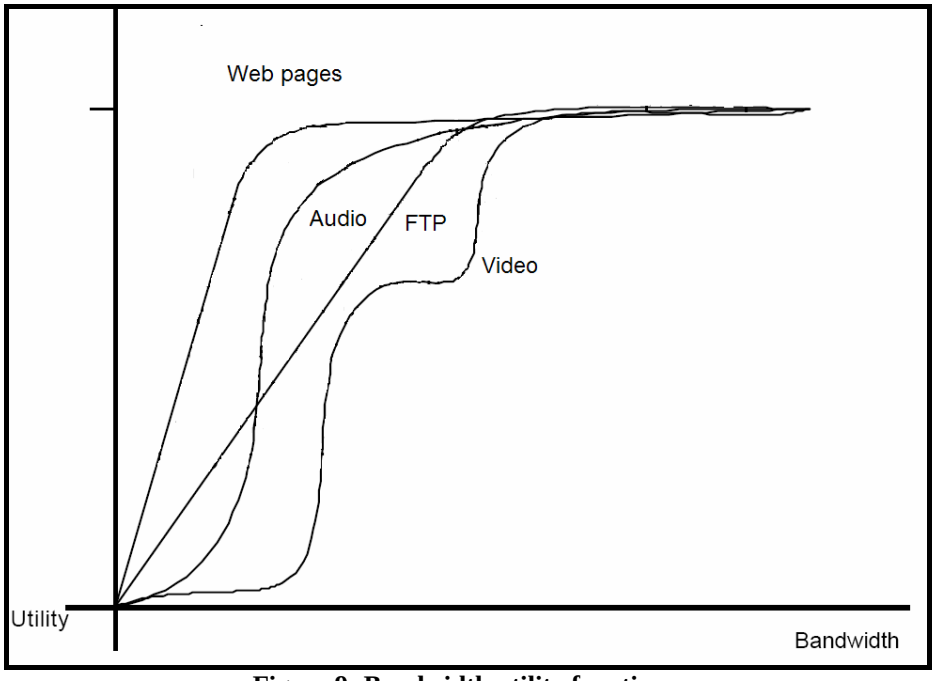

Figure 9: Bandwidth utility functions.

The most useful input for an adaptation policy is a utility function for a stream. A utility function associated with a certain medium describes how different values of a link (path) transmission characteristic affect the perceived quality (i.e., the utility) of the displayed stream. For example, *Figure 9* depicts how bandwidth availability might affect the performance of various media. FTP transfers take advantage of the extra bandwidth in a direct way, while Web page transfers, having to transmit lots of small objects, increase their utility fast but up to a certain point, after which any excessive bandwidth remains unutilized. Rate-adapted audio requires a minimum amount of bandwidth to start transmitting and quickly advances its utility thereafter. Similar to Web page transfers, audio reaches fairly quickly a point where additional bandwidth is not needed. Finally, a two-layered video stream requires a significant amount of bandwidth to start transmitting with the base layer. After this, the utility curve

remains flat until a point in bandwidth availability is reached where there is enough bandwidth for the second layer to be accommodated over the link. This results in the stepwise curve shown in *Figure 9*.

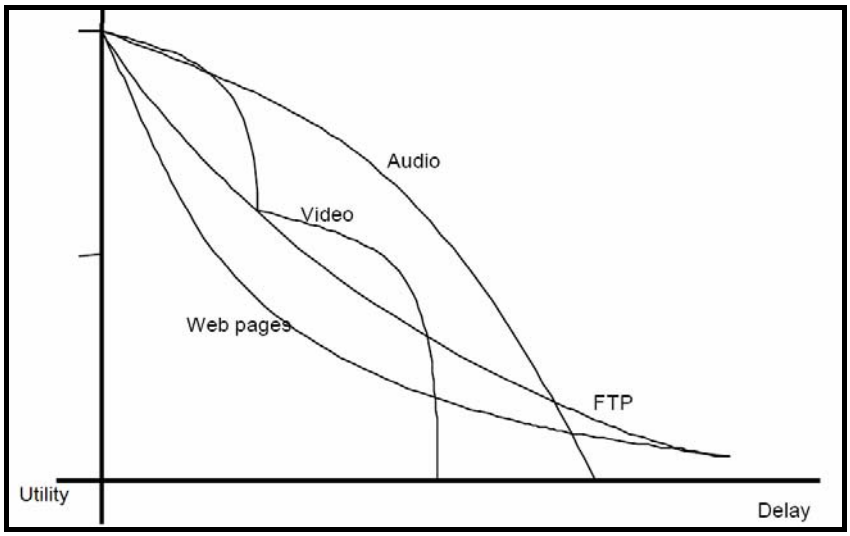

Figure 10: Delay utility functions.

Figure 10 represents the utility curves for the same media versus variations in the end-to-end fixed propagation delay. For small file transfers, increased delay translates almost linearly into lower utility. For real-time media, like video and audio, delay is less a factor when it is small due to the effects of pre-fetching and buffering [23, 24, 33, 39]. At the beginning of the transmission of a real-time stream, the receiver buffers a certain amount of data prior to initiating the stream's presentation, in order to be able to absorb small-scale fluctuations in delays in the arrival of successive frames. The result is a utility curve that starts fairly flat and drops steeply later, once for an audio stream and twice for a two-layered video stream.

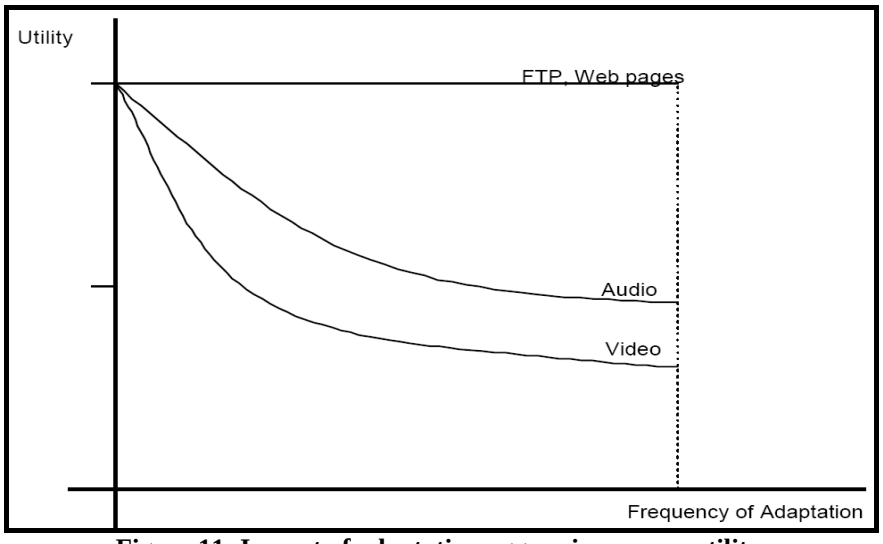

Figure 11: Impact of adaptation aggressiveness on utility.

Finally, *Figure 11* illustrates how aggressiveness in applying the adaptation process can affect the utility of these media. Since a file transfer can take advantage of any available resources, being aggressive in exploring and utilizing them does not impact their quality and, therefore, their utility remains maximal. On the other hand, real-time

continuous media perform better with stability rather than frequent oscillations between different levels of quality. Thus, a high frequency of adaptation has a negative effect on their utility. For a rate-adapted audio stream the adaptation steps are fairly small, so the effects are mild. For a layered video stream, however, the adaptation steps are coarser and the effects of frequent adaptation can be severe.

Having to provide the adaptation policy is an added burden for the user and, thus, it should remain an optional feature. A high percentage of users prefer simplicity of operation that traditional devices like TV and radio tuners provide. This is evident in the area of personal computers, where their popularity has skyrocketed when the operating systems became more user-friendly. Therefore, the adaptation mechanism should always include a set of default adaptation policies, like those described previously, which can offer acceptable performance in the "common" case.

On the other hand, automation should not preclude the users from customizing their adaptation process to better meet their requirements. For example, considering a video stream, different content might require a different approach adaptation policy. If the stream is a surveillance video, quality is not as important factor as stability is.

However, if the stream is a sports video, then quality becomes the major factor, since the details dictate the value of the content for such a stream. Providing the user with the potential to tweak the adaptation policy allows optimization of the quality of the content received.

There are several methods for an application to represent and communicate the adaptation policy for each stream. In Mobiware, utility functions are used to capture the adaptation behaviour of each stream throughout a range of possible link conditions [23, 40]. In MobiWeb, a set of specialized timers is dictating the frequency and aggressiveness of the adaptation process in a wireless environment [16]. When the conditions are fairly stable the timers push the exploration process for those streams that are willing to exploit more resources. In contrast, when the channel conditions vary frequently, the timers protect the streams from rapidly and continuously adapting to short-term changes. In Receiver-driven Layered Multicasting (RLM), a collaborative system between receivers accessing the same multicasting tree distributes the knowledge concerning the current network conditions [41]. When a receiver experiments with a different quality stream, it broadcasts the result of the experiment to all interested parties, for them to decide their actions without having to repeat the experiment. Finally, Layered Video Multicast with Retransmission (LVMR) uses the shared-experiment approach that RLM introduced, but stores the information in strategically located nodes high in the multicasting tree, instead of distributing it to all the receivers [42, 43]. This way traffic is reduced and each new receiver can still have access to previous experiments simply by requesting the appropriate information from a repository higher in the multicasting tree.

5.3. Adaptation Methods and Mechanisms

There are two different methods for adapting media content: continuously and discretely. Selecting which to use depends merely on the medium type and on the capabilities of the encoding format. Media that do not impose timing constraints are usually adapted continuously. Text, images, file transfers and web page downloads adaptation consists of regulating the allowed portion of bandwidth available to them. This adaptation method does not modify the content of the stream. Instead, it might delay the transmission according to the channel characteristics, which is generally acceptable when timing relationships are not a consideration. Continuous adaptation

is used in this fashion when the receiver opts for better quality at the expense of transmission delay.

Non real-time media can also be adapted discretely. In this case, the receiver elects to reduce the transmission delay at the expense of presentation quality. The discrete adaptation alters the medium representation format in order for it to fit the transmission characteristics of the link, while it strives to preserve the perceptual value of the original content. For example, text can be transformed to a different format (e.g., from postscript, to PDF, or rich text format, or plain text format), where it looses some aesthetic characteristics (fonts, font sizes, alignment, etc.), but retains the content value [17, 20, 21, 22]. Images can be transcoded into another format with different color depth and resolution, but still remain recognizable [20, 44]. Web pages can be filtered to loose redundant objects [17, 22] or to translate the HTML code into another markup language (e.g., WML) that can be interpreted by the displaying device [17, 22, 24, 45]. This way, some information is necessarily omitted, but there is usually an option for the application to fetch it on demand.

The media that impose timing constraints are mostly adapted with the discrete method. The drawback of the continuous method in this case is that it alters the transmission rate of the stream without altering the volume of data transmitted. The result is typically an unacceptable perceptual representation of the displayed stream. For example, reducing the transmission rate of a video stream by half without reducing the transmitted frame rate will force it to be displayed in slow motion. The discrete method, on the other hand, changes the transmission rate for continuous media by altering the content representation. Typical types of media that perform well with the discrete adaptation method are audio and video. In audio, the encoding algorithm dictates the amount of bandwidth required for the real-time transmission of the stream. Changing the encoding parameters, such as the number of bits per sample, the sampling frequency and the encoding algorithm, the transmission characteristics of the audio stream can be adjusted to the available resources of the transmission path [11, 46]. The drawback of the discrete adaptation method is the lack of continuity in the required bandwidth between different representations of the stream. Streams discretely adapted cannot always fully exploit all the available resources.

Similar to audio, motion video is usually adapted discretely. A video stream is displayed in frames, which consist of several packets worth of data. The packets constituting each frame have a specific deadline to meet in order for the frame to be displayed properly. If some of those are missing, the frame will be displayed with visible gaps or with inferior quality. Even if these packets arrive later on, they will be useless and in most cases will have to be discarded. Therefore, an important consideration for the adaptation process of a video stream is to preserve the stream's timing characteristics.

Video is undoubtedly the most bandwidth-consuming medium and transmitting it efficiently over limited links is extremely challenging. Initially, video was captured in formats that were using intra-frame encoding--each frame was coded based on the information included only in the frame itself [47]. This resulted in high data-rate video streams with little variation in the frame size. The low variation is important for the accuracy of resource reservation schemes [13, 48], but the high data-rate precludes the transmission of this stream over a large portion of Internet connections that are severely bandwidth limited.

In order to overcome this limitation, researchers developed inter-frame encoding algorithms [49, 50] that take advantage of the redundancy in the information between neighboring frames. In a video scene, where only a few objects move or change, like

in news broadcasting, it is more efficient to transmit only the difference between successive frames rather than a whole new frame. The gain in bandwidth can be between 50%-90%, which substantially lowers the average bandwidth required for this stream to be transmitted. Such encoding allows the transmission of real-time video even over low bandwidth wireless links.

The bandwidth gained with inter-frame coding does not come without a price. Having frames with different sizes significantly increases the variations in arrival delay, putting the frames more at risk for missing their deadlines. Part of the problem is the introduction of intra-coded frames within streams, which usually have much larger sizes than the inter-coded frames. Their presence in the stream is necessary for several reasons. First, in case a frame is lost, those that follow cannot be decoded correctly since they depend on the information carried by the lost frame. Instead, the intra-coded frame is self-contained and is a re-generation point, restoring the display after a lost frame. Second, they are important for random accessing the video stream in a non real-time fashion and for using functions like pause, fast forward, fast rewind and slow motion. However, due to their large size and thus their increased transmission delay, they are prime candidates for arriving late. Since their loss translates into a series of un-decodable frames, the adaptation method should strive to accommodate them efficiently. Two methods for doing that are by associating them with a higher priority [4, 10, 16, 23, 51, 52, 27, 28] and by applying an initial buffering scheme that will be able to absorb most of the variations in the inter-frame delay [8, 53, 54, 55].

Reducing the data rate of a video stream can also be done with the continuous method, when the adaptation process is located at the source (or when transcoding is used at a proxy). The encoder gathers performance information from the receiver through a feedback channel. This information can include the error rate of each link that the stream crossed during the transmission. The encoder utilizes the highest error rate in order to alter the stream's signal-to-noise ratio (SNR). This effectively makes the stream more or less robust to errors, while increasing or decreasing the consumed bandwidth, respectively [56, 6, 7, 8, 57, 58].

Except from adapting the data rate, video streams allow multi-dimensional adaptation by adjusting most of the encoding parameters. A filter applied to a video stream in order to change its presentation characteristics is a solution frequently encountered in the literature [20, 30, 31]. The filter can adjust the color depth of the stream, from 32 to 16 or 8-bit color or to grayscale (typically 8 bits-per pixel or less). It can also resample it to a lower resolution (very easily if scaling it down by factors of two). It can also change the frame rate of the delivered stream by selectively discarding frames out of the frame sequence. A common implementation of a frame discarding filter takes as input an MPEG stream and discards the B frames or both P and B frames, depending on the requested quality.

Filtering mechanisms adapt the stream discretely. The result is a set of possible levels of quality [16, 59, 60] that the adaptation mechanism can select from to adjust its resource demands. The adaptation mechanism has to continuously monitor the link quality in order to decide whether it is more appropriate for the stream to switch to another level of quality. The aggressiveness of this process is dictated by the adaptation policy, as mentioned earlier, which is provided by the user or the adaptive application. There are two different approaches in deciding whether to change the current level of quality. The first one utilizes bandwidth bounds, which trigger the adaptation process when the transmitting stream exceeds them [4, 59, 60, 20, 21, 51]. Although it is fairly simple to implement and it performs well in situations with infrequent but long-term changes, this approach suffers when highly variable links are

considered. Over a fluctuating wireless link, short but abrupt changes in the quality might force this adaptation method into a series of unnecessary changes in the quality of the stream.
The second approach uses timing limits instead of bandwidth ones [16, 42]. It allows the adaptation process to relax for a short period of time giving the stream a chance to recover from a short, abrupt or not, fluctuation in link quality. After an adequately long time period, the decision whether the change was permanent or not is based on a better collection of data and, thus, is more accurate than an immediate response. As the conditions of the link become fairly stable, the adaptation mechanism should limit the frequency of probing performed by the existing streams for more resources as long as the link is close to full utilization. This avoids the short, but unnecessary experiments for discovering additional resources, which are inevitably going to fail in this case, permitting instead the streams to reach a steady state operation. When resources are or become available, however, the adaptation mechanism notifies the existing streams to increase their probing frequency in order to utilize the additional resources quickly.
Finally, discrete adaptation is also used with layered multimedia streams (typically video), initially introduced in multicasting scenarios [41, 42, 52]. The original stream is decomposed in several sub-streams containing a portion of the information necessary for displaying it [27]. The first layer, called the base layer, is self-contained, in the sense that it doesn't require any other layer in order to be displayed, and represents the lowest quality for this video stream. The remaining layers are designed to add to the base layer, gradually enhancing the quality of the displayed stream. In multicasting scenarios, the source transmits each layer over a different connection (stream) and the receivers have the opportunity to subscribe to as many of them as their display and transmission characteristics allow. The adaptation method for layered multicasting operates somehow similarly to a frame-dropping filter. Each receiver communicates with a layer-dropping filter and indicates the layers it is interested in receiving. The filter is then responsible for selecting only those layers and redirecting them to the receiver.

5.4. Supporting mechanisms

5.4.1 Priorities

The order with which the streams adapt after a change in the link/path quality is of great importance for the efficiency of the adaptation process. The basic approach requires all streams to adapt at the same time [31, 57, 18, 59, 20, 21, 60, 27, 28].
This solution has some undesirable side effects. When all streams adapt to a new level of quality they would have often freed up more than enough resources, if they have adapted backwards, or they would have utilized more than the available resources, if they have upgraded. In both situations a new cycle of adaptation is initiated in the reverse direction, forcing eventually all streams to oscillate between levels of quality. Such oscillations have a degrading effect on quality, especially for real-time continuous media streams, like audio and video.
Therefore selecting an order for the adaptation of the streams is imperative. Each stream can be associated with a priority value that indicates the stream's importance according to the user (compared to the other streams). The adaptation process can then force the streams with the lowest priority to adapt first, protecting the rest of them from adapting at the same time [61, 50, 62, 63, 64, 35]. If the freed resources are not adequate, adaptation of the next lowest priority stream will be invoked and this

process will continue as necessary, possibly until some streams have to terminate their transmission, temporarily or permanently.

Using a static prioritization scheme, unfairness issues can arise mainly for the lowest prioritized streams. If the priority value does not change throughout the lifetime of the stream, those who start with higher priority will eventually utilize all the available resources forcing the rest of them to effectively shut down their transmission [4, 10, 23, 51, 52]. Conversely, using a dynamic prioritization scheme allows all streams to fairly compete for the resources of the link. Such a scheme increases the priority of a stream whenever it adapts backwards and decreases it whenever it adapts forward [16]. The result is that the stream initiated with high priority can no longer dominate the link. As soon as it advances a few levels of quality, its priority drops to an equal or lesser value than other streams. When link degradation occurs, the stream that utilized excessive resources, but has now a low priority, will be the first to be forced to give up some of them. Thus, a stream that adapted once will not be forced to adapt again before all other streams that previously had the same priority adapt at least once [16].

Assigning priorities to different streams is an important task that should reflect user preferences without introducing unfairness between streams. When the link under consideration is not shared by many users or different applications, the solution is straightforward. The user responsible for the streams can assign individual priorities to them that reflect his preferences. In addition, a default prioritization scheme can complement the adaptation mechanism, which can assume the responsibility of assigning priorities to those streams that the user did not care to prioritize. A sample prioritization scheme will put different types of media in order of importance and use this mapping to assign priorities for each stream. For example, file transfer might have the lowest priority, followed by images, Web pages (html), real-time video and finally audio will receive the highest priority.

When the link is shared among many users the prioritization process requires an arbitrating third party in order to retain fairness. Lack of an arbitrating authority could allow users to abuse the prioritization scheme by setting their streams to the highest priority. The obvious solution to this problem is the introduction and enforcement of a charging mechanism, which will charge each user according to the resources utilized over the link or some other pricing scheme [4, 65, 66].

5.4.2 Admission Control and Resource Reservations

Admission control and resource reservations are necessary complementary mechanisms to the adaptation process, particularly when resource-intense streams with minimum resource demands for acceptable performance are transmitted over resource-poor links. The adaptation process relies on information about the availability of the resources over the link in order to adapt the streams to optimally utilize it. By allowing uncontrolled admission of new streams, without reserving resources for the existing and new ones, the adaptation process can be rendered useless.

Deciding whether to admit a new stream or not is based on the available resources and the relative priority of this stream against the existing, already admitted ones. Imposing strict admission control is a rather difficult task, since optimal admission control can be extremely hard to perform in real-time [69, 70, 84]. Instead, heuristic methods are favoured, which perform well, while requiring minimal computational power [84]. A common heuristic method admits streams based on their average resource requirements [26, 63, 69, 70]. With discrete adaptation, admission can also be done based on the characteristics that the stream presents when it transmits with its

base level of quality [16]. In both cases, the monitoring mechanism informs the admission control about the approximate amount of available resources and the admission control decides whether the stream will be accepted or not. It is a common tactic to leave a small percentage of resources unutilized in order to tolerate occasional variations in demand for resources from existing streams.
The importance of the new stream can also play a key role in the admission process. If the new stream is of higher priority than some admitted ones, the admission control might allow it to initiate transmission, even if there are not enough resources available over the link. In this case the adaptation mechanism will identify the contention for resources and will force the least important stream to adapt backwards.

5.4.3 Hand-off Notifications

In wireless networks supporting mobility, mobile hosts may be forced to hand-off frequently between cells, both horizontally (between cells of a wireless network with the same technology and similar parameters) and vertically (between cells belonging to different wireless networks potentially using diverse technologies and different parameters) [25, 38]. Upon hand-off, the streams initiated by the mobile device encounter a new environment with potentially different transmission characteristics and different traffic load. In order for the migrating streams to seamlessly integrate with the existing ones, admission control must be applied upon hand-off. Note that the admission process is slightly different in this case than described previously. This time the streams to be admitted have already been initiated (before the hand-off) at some previous cell, and their interruption or discontinuation will be unpleasant and undesirable for the user. The admission process must instead make every effort to accept all these (existing and migrating) streams, even if this leads to over-utilization of resources temporarily. The next step then would be to notify the adaptation mechanism about this discrepancy between required and available resources, so that it does initiate the adaptation process immediately, in order to minimize the effects of the hand-off in the perceptual quality of the streams [4, 25, 67, 68].
A hand-off notification can also initiate the migration of the adaptation mechanism in the proxy adaptation approach. The proxy is usually residing close to the base station at the end of the wireless link and a potential hand-off requires it to transfer the current state and filters in use to a new location closer to the new base station. Lack of transfer of state typically results in several seconds of inferior quality media before the new proxy, located close to or at the new base station, reaches the appropriate adaptation setup.

6. Case Studies

We present here briefly three different and complete recent adaptation proposals as case studies in order to better illustrate the problems addressed and the proposed solutions.

6.1 SCP

SCP (Streaming Control Protocol) [3, 14] is a flow and congestion control scheme for real-time streaming of continuous multimedia data across the Internet. It addresses two issues associated with real-time streaming. First, it uses a congestion control policy that allows it to fairly share network bandwidth with both TCP and other SCP streams. Second, it improves smoothness in streaming and ensures low, predictable latency, both of which enhance the representation quality of the displayed stream.

SCP resembles TCP in most of its features, but strives to avoid TCP's jittery behaviour in its congestion avoidance technique when it reaches steady-state. SCP is designed for streaming real-time continuous multimedia, which perform significantly better with stable amounts of available resources. Thus, when the network is close to fully utilized, SCP avoids continuous probing for more bandwidth that would have been followed inevitably by a back-off period when the network would have become congested. Instead, SCP enters a steady-state phase, where it slightly adjusts the actual transmission rate based on estimates of the current RTT and the available bandwidth. Moreover, when the application pauses its transmission, SCP remembers the steady-state point it had reached before pausing and uses it when the application starts transmitting again, instead of invoking the slow-start mechanism. The slow-start mechanism though is invoked at the initialization of the stream, while the exponential back-off mechanism is applied when network congestion occurs. SCP tries to also ensure a low, predictable latency by avoiding the retransmission of lost packets, since real-time applications are usually capable of sustaining such losses without noticeable changes in their quality.

SCP is an end-to-end adaptation scheme that uses feedback messages from the receiver back to the source. For reliability, it uses TCP for its control messages since SCP does not use retransmissions for lost packets. The deployment of SCP requires significant changes to existing Web and media servers. In addition to the new transport protocol, the servers are required to change their media content since SCP requires them to be stored in multiple resolutions and qualities in order to perform the adaptation process. Its coexistence with TCP results in sharing the available bandwidth based on the configuration of TCP. The more aggressive TCP is configured to be, the fewer resources are left available for SCP to utilize. Finally, since SCP expects different qualities of the same stream to be stored in different files at the server, the adaptation process consists of only switching between files for retrieving data. Thus, SCP adds minimally to the required CPU processing.

SCP strives primarily for stability by enhancing TCP's features, but it remains a best-effort protocol, without admission control and resource reservations. Each stream operates with a certain quality as long as the consumed resources are between an upper and a lower bound. As soon as any of those is crossed, the adaptation mechanism is initiated and the server switches to retrieving and transmitting data from a different file. The protocol supports high level representations of the different qualities that the stream can assume, so that the user can select the desired one as a starting point. Finally, during handoffs, all applications are reset and use the slow-start mechanism in order to adapt their transmission to the new environment.

6.2 Mobiware

Mobiware [23, 40] is a programmable, active middleware toolkit that provides a set of open programmable interfaces and objects for adaptive mobile networking. It runs on mobile devices, wireless access points and mobile capable switches/routers allowing applications to probe the network for resources and adapt based on their availability, according to predefined utility functions and adaptation policies.

Mobiware uses the receiver-active proxy adaptation model. Each application on the mobile device utilizes Mobiware's API to communicate with a QoS Adaptation Proxy (QAP) and a Routing Anchor Proxy (RAP). Each QAP gathers the utility functions of all traversing flows along with the adaptation policies defined by the user's preferences for the specific stream. Each RAP, on the other hand, bundles all flows to/from a single mobile device and performs routing functions, during handoff, only

once for the bundle and not for each stream individually. Mobiware does not use reservation of resources in order to guarantee a certain quality to streams. Instead the streams constantly probe the network for more resources, but it is the QAP that make the adaptation decisions. After the QAP collects all the probes, it splits the available resources between bundles according to the requested resources and their availability. Furthermore, it splits the reserved bandwidth for the bundle between individual streams according to their utility function and adaptation policies. By refreshing the allocation of resources only through probing, the QAP retains a soft state of the existing streams, which enhances the robustness of the scheme. The streams do not have to explicitly remove their state from the QAP when they cease their operation, thus a garbage collection mechanism is redundant.

Both RAP and QAP proxies can be located anywhere in the network. However the QAP proxy performs better when it resides closer to the mobile device, since it can estimate more accurately the available resources over the limited and variable wireless link. The QAP filters the incoming stream in a fashion that conforms to the associated utility function. The specification of both the utility function and the adaptation policy is solely the application's responsibility; they should both adhere to the performance requirements of the transmitting stream. For example, a real-time video stream requires a less aggressive, but robust (in the sense of ensuring uninterrupted, smooth delivery of at least the lower levels of quality of the video stream) policy, while an FTP stream performs optimally with an aggressive, best-effort one.

During handoffs, the RAP reroutes the stream bundle associated with the mobile device performing a single set of routing functions. This significantly reduces the routing overhead and latency for all the streams. In addition to routing parameters, the RAP also propagates the filters, which the streams of the bundle were using in the previous QAP, to the new one, so that it will also speed up the restoration of the streams to their previous quality level. However, since the environmental conditions are expected to be different in the new cell, it is possible that they will not be enough resources for all the streams to be reinstated to their previous condition. In this case, Mobiware allows users to state their relative preferences for each individual stream and then uses this knowledge to give priority during the handoff process to those streams with higher preferences. If the resources are insufficient for all streams, the ones with lowest priority will eventually terminate their transmission.

6.3 MobiWeb

MobiWeb [16] is a proxy-based architecture designed to enhance the performance of adaptive real-time streams over wireless links. MobiWeb uses the receiver-aware proxy adaptation model, where the receiver initializes the adaptation process but does not interfere with it throughout the duration of the transmission. The proxy, residing next to the wireless link for maximum efficiency in adapting streams to the current link conditions, admits incoming real-time streams, initializes their transmission with the assistance of the adaptive application and adapts them according to the specified user preferences and adaptation policies. MobiWeb does not require legacy Internet servers to change their content. Instead, the proxy performs filtering of the incoming stream to match the transmission characteristics of its current target Level of Quality.

MobiWeb has several features that assist the adaptation process and cope with the peculiarities of wireless links. Since the resources over a wireless link are limited and variable, MobiWeb performs admission control and soft reservations only for realtime streams. Reservations are performed based on the resources that the stream needs in

order to operate with its base Level of Quality. The reserved resources can still be utilized by other (best-effort) streams whenever the associated application does not use them. When a stream does not utilize its allocated resources for a long period of time, the reservation eventually times out and the rest of the streams can contend for the freed resources.

In an unpredictably variable environment simply reserving resources might not be enough to ensure a certain quality for a stream. Thus, MobiWeb accompanies the admission control process with a dynamic prioritization scheme. When conditions degrade, MobiWeb chooses the stream with the lowest priority as the first candidate for adaptation, while leaving the remainder of the streams intact. After the selected stream adapts, if the conditions are still not stable (i.e., the resources still not adequate for this set of Levels of Quality for all streams), MobiWeb will continue to adapt streams, one at a time, until the link reaches stability. Thus, streams are protected from entering the adaptation process and fluctuating their perceptual quality, unless it is absolutely necessary.

The user specifies at the initiation of a stream his relative preference for it in the form of a priority value. The stream initiates transmission with its lowest quality and as soon as it advances to a higher Level of Quality its priority value drops and vice versa. This way, lower quality streams are expected to have higher priority and thus be protected from adaptation during degradation. Another way to look at it is that a single stream will not be forced to adapt twice, before all other streams with the same priority adapt at least once. The dynamic prioritization scheme is able to incorporate adaptation unaware traffic by assigning a default priority value to all incoming streams falling into this class.

To avoid the initiation of the adaptation process too often in a rapidly fluctuating environment, like a wireless one, MobiWeb instils tolerance against short-term link variations by means of a set of specialized timers. These timers set a time limit in which the link can recover before they trigger the initiation of the adaptation process. On the other hand, whenever resources become available, they promptly identify the change and hasten the adaptation process. Finally, when the link reaches a fairly stable condition, they try to retain a steady-state operation for all streams by exponentially spacing the probing of the link for more resources.

In the case of handoff, the current state of all streams associated with the mobile device is transferred to the new proxy, along with the appropriate filters. Since the new environment is likely to have a different amount of resources available, admission control is performed again for the newly arrived streams, giving preference to those with higher priority.

6.4 Receiver – driven Layered Multicast (RLM)

Receiver-driven Layered Multicast is one of the first end-to-end receiver-oriented rate-adaptation algorithms for real-time multicast flows. RLM can be deployed in current networks and its performance can be incrementally optimized by adding new but “lightweight” functionality to routers.

Layered coding is the basis of the Receiver-driven Layered Multicast (RLM) scheme, pioneered by McCanne [1] and others. McCanne’s multicast backbone (MBONE) tool *vic* implements RLM by coding video in multiple layers and broadcasting (actually, multicasting) each layer to a different multicast group.

A typical configuration for multicast video consists of a heterogeneous set of hosts receiving video from some number of sources all within one multicast session. But because the receivers are connected to the network at different rates, the source cannot

adjust its transmission to simultaneously satisfy the rate requirement of each receiver. We solve this problem by moving the burden of rate adaptation from the source to the receivers in a scheme we call Receiver-driven Layered Multicast (RLM). Under RLM, a layered signal is transmitted over multiple IP Multicast groups —each layer of a hierarchical media stream is distributed on an individual IP multicast group—and receivers adjust their rate of reception by controlling the number of groups they subscribe to.

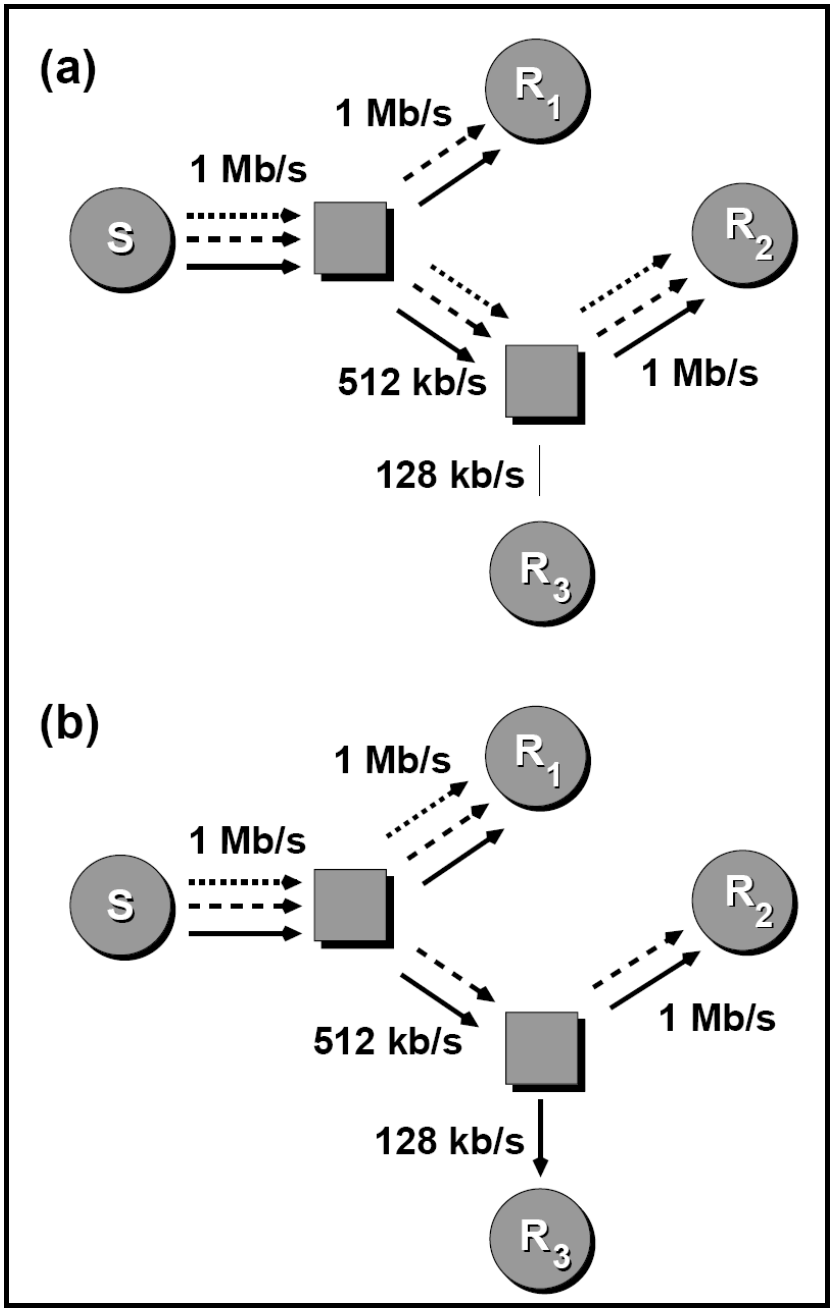

Figure 12: End-to-end Adaptation. Receivers join and leave multicast groups at will. The network forwards traffic only along paths that have downstream receivers. In this way, receivers define multicast distribution trees implicitly through their locally advertised interest. A three-layer signal is illustrated by the solid, dashed, and dotted arrows, traversing high-speed (1 Mb/s), medium-speed (512 kb/s), and low-speed (128 kb/s) links. In (a), we assume that the 512 kb/s is oversubscribed and congested. Receiver R2 detects the congestion and reacts by dropping the dotted layer. Likewise, receiver R3 eventually joins just the solid layer. These events lead to the configuration in (b).

Under RLM, each receiver individually adapts to observed network performance by adjusting its level of subscription within the overall layered multicast group structure. Moreover, all the members in a session share control information across the group to improve the convergence rate of the adaptation algorithm.

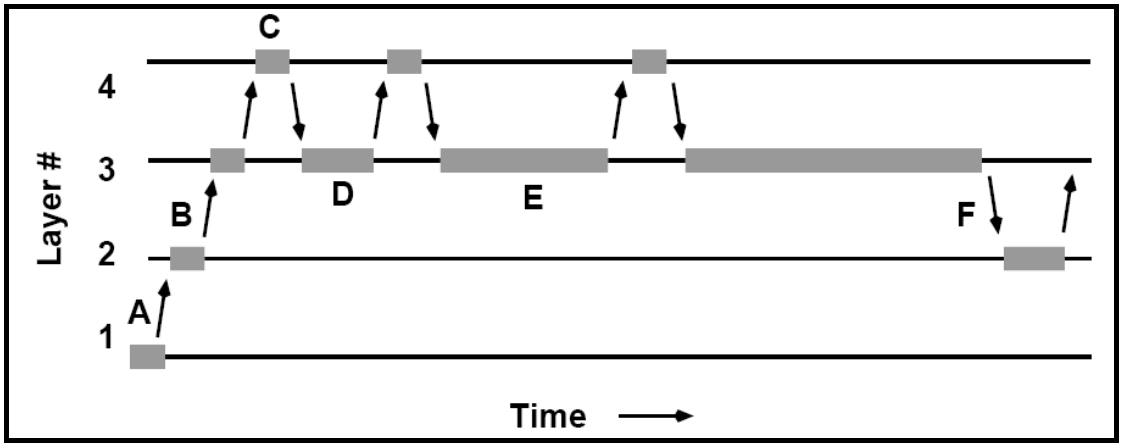

Figure 13: An RLM "Sample Path". This diagram illustrates the basic adaptation strategy from the perspective of a given receiver. Initially, the receiver joins the base layer and gradually adds layers until the network becomes congested (C). Here, the receiver drops the problematic layer and scales back its join-experiment rate for that level of subscription.

Each receiver estimates its available bandwidth and joins a number of these multicast groups to fill that available bandwidth. The available bandwidth is estimated by measuring packet losses.

Essentially, if the packet losses are frequent, then the transmission rate is deemed too high for the available bandwidth and the receiver leaves some of the multicast groups. If the packet losses are sufficiently rare, then the receiver joins some of the multicast groups. In this way the receiver can track the available bandwidth even if it is varying. *Figure 5* illustrates the exponential back-off strategy from the perspective of a single host receiving up to four layers. Initially, the receiver subscribes to layer 1 and sets a join-timer (A). At this point, the timer duration is short because the layer has not yet proven problematic. Once the join-timer expires, the receiver subscribes to layer 2 and sets another join-timer (B). Again, the timer is short and layer 3 is soon added. The process repeats to layer 4, but at this point, we assume that congestion occurs (C). A queue then builds up and causes packet loss. Once the receiver detects these lost packets, it drops back to layer 3. The layer 3 join-timer is then multiplicatively increased and another timeout is scheduled (D). Again, the process repeats, congestion is encountered, and the join-timer is further increased (E). Later, unrelated transient congestion provokes the receiver to drop down to layer 2 (F). At this point, because the layer 3 join-timer was never increased in response to congestion, the layer is quickly reinstated.

Because each receiver determines its own transmission rate, the scheme is said to be receiver-driven. Commercial systems do not use RLM, primarily because efficient layered video coding is difficult to achieve. Commercial systems typically simulcast their content at a variety of rates. Because each receiver determines which multicast group it will join, this is also a receiver-driven scheme. However, it does not achieve the network bandwidth efficiencies of RLM, nor is it as bandwidth adaptive as RLM.

References

[1] S. R. McCanne, “Scalable compression and transmission of Internet multicast video”, Ph.D. dissertation, Univ. California, Berkeley, Dec. 1996.

[2] M. Margaritidis and G. C. Polyzos, “Adaptation Techniques for Ubiquitous Internet Multimedia”.

[3] Inouye J, Cen S, Pu C, Walpole J. “System Support for Mobile Multimedia Applications”, Proceedings of the 7th NOSSDAV; May 1997.

[4] Bharghavan V, Kang-Won L, Songwu L, Sungwon H, Jin-Ru L, Dwyer D. “The TIMELY Adaptive Resource Management Architecture”, IEEE Personal Communications; August 1998; 5(4):20-31.

[5] Hsu CY, Ortega A, Khansari M. “Rate Control for Robust Video Transmission over Burst-Error Wireless Channels”, IEEE Journal on Selected Areas in Communications; May 1999; 17(5):756-773.

[6] Busse I, Deffner B, Schulzrinne H. “Dynamic QoS Control of Multimedia Applications Based on RTP”, Computer Communications; January 1996; 19(1):49-58.

[7] Bolot JC, Turletti T. “Experience with Control Mechanisms for Packet Video in the Internet”, Computer Communication Review; ACM, January 1998; 28(1):4-15.

[8] Rejaie R, Handley M, Estrin D. “Quality Adaptation for Congestion Controlled Video Playback over the Internet”, Computer Communication Review; ACM, October 1999; 29(4):189-200.

[9] Chen Z, Tan SM, Campbell RH, Li Y. “Real Time Video and Audio in the World Wide Web”, World Wide Web Journal; vol. 1, January 1996.

[10] Wood R, Fankhauser G, Kreula R, Monni E. “QoS Mapping for Multi-Media Applications in a Wireless ATM”, Proceedings of ACTS Mobile Summit '97; October 1997.

[11] Kazantzidis M, Wang L, Gerla M. “On Fairness and Efficiency of Adaptive Audio Application Layers for Multi-hop Wireless Networks”, IEEE MOMUC'99; San Diego, November 1999; 357-362.

[12] Schulzrinne H, Casner S, Frederick R, Jacobson V. “RTP: a Transport Protocol for Real-Time Applications”, Internet Engineering Task Force RFC 1889; January 1996.

[13] Zhang L, Deering S, Estrin D, Shenker S, Zappala D. “RSVP: a New Resource ReSerVation Protocol”, IEEE Network; September 1993; 7(5):8-18.

[14] Cen S, Walpole J, Pu C. “Flow and Congestion Control for Internet Media Streaming Applications”, Proceedings of the SPIE – The International Society for Optical Engineering 1997; (3310):250-264.

[15] Nandagopal T, Kim TE, Sinha P, Bharghavan V. “Service Differentiation through End-to-End Rate Control in Low Bandwidth Wireless Packet Networks”, Proceedings of the 6th International Workshop on Mobile Multimedia Communications; San Diego, CA, USA, November 1999.

[16] Margaritidis M, Polyzos GC. “MobiWeb: Enabling Adaptive Continuous Media Applications over Wireless Links”, IEEE International Conference in 3G Wireless 2000; San Francisco, June 2000.

[17] Housel BC, Lindquist DB. “WebExpress: A System for Optimizing Web Browsing in a Wireless Environment”, MOBICOM’96 Demonstration Session; November 1996.

[18] Satyanarayanan M, Noble B, Narayanan D, Tilton JE, Flinn J, Walker KR. “Agile Application-Aware Adaptation for Mobility”, Proceedings of the 16th ACM Symposium On Operating Systems Principles; December 1997; 31(5):276-287.

[19] Bharghavan V, Gupta V. “A Framework for Application Adaptation in Mobile Computing Environments”, Proceedings of the IEEE Compsoc’97; November 1997.
[20] Fox A, Gribble SD, Brewer EA, Amir E. “Adapting to Network and Client Variability via On-Demand Dynamic Distillation”, Proceedings of the Seventh International ACM Conference on ASPLOS; Cambridge MA, October 1996.
[21] Zenel B, Duchamp D. “A General Purpose Proxy Filtering Mechanism Applied to the Mobile Environment”, Proceedings of MOBICOM’97; Budapest, Hungary, September 1997.
[22] Liljeberg M, Alanko T, Kojo M, Laamanen H, Raatikainen K. “Optimizing World Wide Web for Weakly Connected Mobile Workstations: An Indirect Approach”, Proceedings of the 2nd International Workshop on Services in Distributed and Networked Environments 1995; 132-139.
[23] Angin O, Campbell AT, Kounavis ME, Liao RRF. “The mobiware Toolkit: Programmable Support for Adaptive Mobile Networking”, IEEE Personal Communications; August 1998; 5(4):32-43.
[24] Freytag C, Kumpf C, Neumann L. “Utilization of User and Device Profiles for Adaptive WWW Access”, Proceedings of the 5th International Workshop on Mobile Multimedia Communications; October 1998.
[25] Brewer EA, Katz RH, et al. “A Network Architecture for Heterogeneous Mobile Computing”, IEEE Personal Communications Special Issue; October 1998; 8-24.
[26] Murkejee A. “Supporting Online Services in Environments Constrained by Communication”, Ph.D. Thesis, CMU CS Tech Report CMU-CS-98-172; November 1998.
[27] Bahl P. “Supporting Digital Video in a Managed Wireless Network”, IEEE Communications Magazine; June 1998; 94-102.
[28] Das SK, Chatterjee M, Kakani NK. “QoS Provisioning in Wireless Multimedia Networks”, Wireless Communications and Networking Conference (WCNC '99); New Orleans, October 1999; 1493-1497.
[29] Fox A, Gribble SD, Brewer EA, Amir E. “Adapting to Network and Client Variability via On-Demand Dynamic Distillation”, Proceedings of the Seventh International ACM Conference on ASPLOS; Cambridge MA, October 1996.
[30] Balachandran A, Campbell AT, Kounavis ME. “Active Filters: Delivering Scaled Media to Mobile Devices”, Proceedings of the 7th NOSSDAV; New York, USA, 1997; 125-134.
[31] Yeadon N, Garcia F, Hutchison D, Shepherd D. “Filters: QoS Support Mechanisms for Multi-peer Communications”, IEEE JSAC; September 1996; 14(7):1245-1262.
[32] Murkejee A. “Supporting Online Services in Environments Constrained by Communication”, Ph.D. Thesis, CMU CS Tech Report CMU-CS-98-172; November 1998.
[33] Pasquale JC, Polyzos GC, Anderson EW, Kompella VP. “The Multimedia Multicast Channel”, Proceedings of the 3rd NOSSDAV; La Jolla, CA, November 1992; 197-208.
[34] Pasquale JC, Polyzos GC, Anderson EW, Kompella VP. “Filter Propagation in Dissemination Trees: Trading off Bandwidth and Processing in Continuous Media Networks”, Proceedings of the 4th NOSSDAV; Lancaster, UK, November 1993; 269-278.
[35] Wittmann R, Zitterbart M. Towards “Support for Heterogeneous Multimedia Communications”, Proceedings of the 6th IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems; Los Alamitos, CA, 1997; 336-341.

[36] Wolf LC, Herrtwich RG, Delgrossi L. “Filtering Multimedia Data in Reservation-Based Internetworks”, Kommunikation in Verteilten Systemen, 22-24.02; TU Chemnitz-Zwickau, 1995.
[37] Brewer EA, Katz RH, et al. A Network “Architecture for Heterogeneous Mobile Computing”, IEEE Personal Communications Special Issue; October 1998; 8-24.
[38] Stemm M, Katz RH. “Vertical Handoffs in Wireless Overlay Networks”, Mobile Networks and Applications; ACM Press, 1998; 3(4):335-350.
[39] Zhao W, Willebeek-LeMair M, Tiwari P. “Efficient Adaptive Media Scaling and Streaming of Layered Multimedia in Heterogeneous Environment”, IEEE ICMCS'99 - International Conference on Multimedia Computing and Systems; Florence, Italy, June 1999.
[40] Bianchi G, Campbell AT, Liao RRF. “On Utility-Fair Adaptive Services in Wireless Networks”, Sixth International Workshop on Quality of Service (IWQoS'98); New York, USA, 1998; 256-267.
[41] McCanne S, Jacobson V, Vetterli M. “Receiver-driven Layered Multicasting”, Computer Communication Review; ACM, October 1996; 26(4):117-130.
[42] Li X, Paul S, Pancha P, Ammar M. “Layered Video Multicast with Retransmission (LVMR): Evaluation of Error Recovery Schemes”, Proceedings of the 7th NOSSDAV; New York, USA, 1997; 161-172.
[43] Li X, Paul S, Ammar M. “Layered Video Multicast with Retransmissions (LVMR): Evaluation of Hierarchical Rate Control”, Proceedings of INFOCOM 1998.
[44] Han R, Bhagwat P, LaMaire R, Mummert T, Perret V, Rubas J. “Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing”, IEEE Personal Communications; December 1998; 5(6):8-17.
[45] The WAP Forum. “Wireless Application Protocol”, http://www.wapforum.org; September 2000.
[46] Steinmetz R, Nahrstedt K. “Multimedia: Computing, Communications and Applications”, Prentice Hall, July 1995.
[47] Microsoft Inc. Video for Windows. http://www.jmcgowan.com/avi.html; September 2000.
[48] Topolcic C. ST II. Proceedings of the 1st NOSSDAV; Berkeley, USA, 1990; 5-132.
[49] Moving Picture Experts Group. The MPEG Format. http://cselt.it/mpeg; September 2000.
[50] International Telecommunication Union (ITU). Recommendations H.261 and H.263. http://www.itu.int; September 2000.
[51] Floyd S, Jacobson V. “Link-Sharing and Resource Management Models for Packet Networks”, IEEE/ACM Transactions on Networking; August 1995; 3(4):365-386.
[52] Bajaj S, Breslau L, Shenker S. “Uniform Versus Priority Dropping for Layered Video”, Computer Communication Review; October 1998; 28(4):131-143.
[53] Rexford J, Sen S, Dey J, Feng W, Kurose J, Stankovic J, Towsley D. “Online Smoothing of Live, Variable-Bit-Rate Video”, Proceedings of the 7th NOSSDAV; New York, USA, 1997; 235-243.
[54] Sen S, Rexford J, Towsley D. “Proxy Prefix Caching for Multimedia Streams”, IEEE INFOCOM '99; Piscataway, USA, 1999; 1310-1319.
[55] Zhao W, Willebeek-LeMair M, Tiwari P. “Efficient Adaptive Media Scaling and Streaming of Layered Multimedia in Heterogeneous Environment”, IEEE ICMCS'99 - International Conference on Multimedia Computing and Systems; Florence, Italy, June 1999.

[56] Chaddha N, Wall GA, Schmidt B. “An End-to-End Software Only Scalable Video Delivery System”, Proceedings of the 5th NOSSDAV 1995; 139-150.
[57] Nandagopal T, Kim TE, Sinha P, Bharghavan V. “Service Differentiation through End-to-End Rate Control in Low Bandwidth Wireless Packet Networks”, Proceedings of the 6th International Workshop on Mobile Multimedia Communications; San Diego, CA, USA, November 1999.
[58] Eleftheriadis A, Anastassiou D. “Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video”, Proceedings of the 5th NOSSDAV; April 1995.
[59] Bharghavan V, Gupta V. “A Framework for Application Adaptation in Mobile Computing Environments”, Proceedings of the IEEE Compsoc’97; November 1997.
[60] Sisalem D, Schulzrinne H. “The Loss-Delay Adjustment Algorithm: A TCP-friendly Adaptation Scheme”, NOSSDAV; Cambridge, UK, July 1998.
[61] dialpad.com. http://www.dialpad.com; September 2000.
[62] Watson A, Sasse MA. “Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications”, Proceedings of the ACM Multimedia 98; New York, USA, 1998; 55-60.
[63] General Packet Radio Service (GPRS). http://www.mobileGPRS.com; September 2000.
[64] Pasquale JC, Polyzos GC, Anderson EW, Kompella VP. “Filter Propagation in Dissemination Trees: Trading off Bandwidth and Processing in Continuous Media Networks”, Proceedings of the 4th NOSSDAV; Lancaster, UK, November 1993; 269-278.
[65] Shenker S, Clark D, Estrin D, Herzog S. “Pricing in Computer Networks: Reshaping the Research Agenda”, Computer Communication Review; April 1996; 26(2):19-43.
[66] MacKie-Mason JK, Varian HR. “Pricing Contestable Network Resources”, IEEE Journal on Selected Areas in Communications; September 1995; 13(7):1141- 1149.
[67] Polyzos GC, Xylomenos G. “Enhancing Wireless Internet Links for Multimedia Services”, Proceedings of the 5th International Workshop on Mobile Multimedia Communications; October 1998; 379-384.
[68] Liao RRF, Campbell AT. “On Programmable Universal Mobile Channels in a Cellular Internet”, MobiCom'98; New York, USA, 1998; 191-202.