Adaptation Techniques for Ubiquitous Internet Multimedia

Margaritis Margaritidis [1] and George C. Polyzos [1,2]

{margarit,polyzos}@cs.ucsd.edu

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

[2] Department of Informatics
Athens University of Economics and Business
Athens, 10434, Greece

Abstract

A major trend in the Internet today is the push for ubiquitous access to rich multimedia content. Achieving this goal requires the incorporation of adaptation techniques, which will transform the media into various formats, suitable for a variety of devices with diverse communication and presentation capabilities. In this survey, we present the categories of adaptation techniques focusing on their characteristics and potential. We analyze various schemes in order to evaluate their effectiveness, performance, complexity, and applicability. We then describe the adaptation policies, methods and mechanisms that they use, along with the supporting mechanisms that aim in increasing the efficiency of the adaptation process. Finally, we present several case studies, commercial approaches and current trends in the area of ubiquitous access to Internet multimedia content.

1. Introduction

An enormous collection of information is available in electronic form and resides just a few clicks of the mouse away from the Internet user. From text and images, to rich multimedia content like audio and video, users are fascinated with the power of interactivity and the endless capabilities of the Web. This helped content providers realize the opportunities that the Internet can provide and they readily embrace it by modifying their products to take advantage of the power that the online interactivity promises.

At the same time, innovations in wireless technology dramatically improved the way people communicate untethered and on the move. Cellular telephones, Personal Digital Assistants (PDAs) and notebooks with wireless interfaces are experiencing sheer increases in sales. Recent studies showed that 1/3 of the earth's population will be a mobile subscriber by the year 2010 [1]. Inevitably, the power of the Internet could not leave the wireless community unaffected. Modern human life is moving in an increasingly fast pace and requires ubiquitous access to information. This in turn demands the incorporation of Internet access in any wireless communication device.

A current trend is undoubtedly the integration of all the aforementioned technologies. The user will only have to use a single device anywhere, anytime to access seamlessly all types of media existing in the Internet. In order for this convergence to occur, an important intermediate technology has to be developed that will adapt and formulate the content according to the transmission characteristics of the end-to-end communication path and to the capabilities of the displaying device. This middleware is the focus of this survey. We start by presenting the types of multimedia content and we continue by describing the characteristics of the current and future Internet infrastructure. We then discuss when, where, and how adaptation can be performed, followed by a thorough comparison of the available adaptation approaches. Next, we describe the adaptation policies, methods and mechanisms, along with the supporting mechanisms that increase the effectiveness of the adaptation process. Subsequently, we examine several case studies from the literature, we present a few existing commercial solutions and we take a look at some trends that are expected to dominate the area in the near future. We conclude this survey with a summary of the most important adaptation policies and mechanisms.

2. Multimedia Content

Text and (static) images were the main media of the Web from the start. The proliferation of richer and highly interactive media, like audio and video over the Internet, fully exposed the inadequacies of the Internet infrastructure and the existing protocols to operate equally well with all types of traffic. In order to discuss these problems and the need for adaptation, we briefly present next the various types of media carried over the Internet, along with their representation characteristics and their transmission requirements.

2.1. Text

In the current, rich multimedia content era, text remains significant and has some important applications. One-way time-critical text applications are very popular in today's mobile phones in the form of SMS messaging, and WAP traffic (e.g., for news, stock market information, etc.). Such type of traffic usually poses no serious demands for its display or transmission, except for some cases where immediate delivery is critical. More time-critical are some interactive, two-way text applications like chat, telnet and pager messages. However, text applications in general tend to have low demands and are easily displayed in the majority of devices.

2.2. Rich Web Content

As with text, we can categorize rich content based on whether timely delivery is critical or not. E-mail, ftp and off-line browsing are applications with low levels of network interactivity. In most cases, the user is expecting a fair delay during downloading, and accepts the best-effort nature of this type of traffic. Therefore, lower priority and low transmission demands characterize this kind of traffic.

On the other hand, rich interactive Web applications also include Web browsing and on-line gaming. These applications are able to operate adequately over slow speed connections, but they can also exploit any available bandwidth in order to achieve better quality presentation or faster response times. Critical parameters are the end-to-end delay, which significantly enhances quality when it is minimal and its variability is low, and the available bandwidth, which should be at least in the order of few tens of Kb/s.

2.3. Audio

Typical streaming audio applications are in use over the Internet for several years. Radio stations and sites that provide pay-per-listen services are very popular. In addition, the domination of the MP3 format [3] revolutionized the distribution methods of music over the Internet. Audio streams have moderate bandwidth requirements that can start from 8 Kb/s (or even lower) for telephone quality audio and rise up to 256 Kb/s for compressed CD quality audio. Timely and reliable delivery are very important, since unrecognizable sounds, due to delayed data, and audible gaps, due to errors and dropped packets (typically due to congestion), severely degrade the listening experience for the user.

Recently, interactive audio applications are attracting increased attention, especially in the form of Voice-over-IP (VoIP) [4, 5]. With VoIP, communications companies are able to provide ubiquitous telephony services over any kind of carrier infrastructure. IP is developing into the common layer that will effectively hide the transmission characteristics of the physical and link layers and possibly other underlying switching technologies. Thus, transmission of voice and data is feasible simultaneously through a single "transport" service. VoIP exploits the current Internet infrastructure and abandons the circuit-switching architecture of the telephony network in favor of the packet-switching architecture of the Internet. Consequently, the cost of telephony services is dramatically reduced. However, in the absence of specific Quality-of-Service (QoS) support mechanisms, this is realized at the expense of the (currently) delivered audio quality, which is affected by the delay variation of the arriving packets.

The transmission and playback characteristics of interactive audio applications are fairly similar to the streaming ones, with the additional requirement of the limited delay allowed, due to their interactive nature. In streaming applications it is a common tactic to buffer an amount of audio data (usually worth multiple seconds of playback), in order to absorb delay variations and possibly allow the retransmission of missing packets, if necessary. Buffering provides robustness and effectively relaxes the timing requirements, leaving the bandwidth requirement as the basic consideration for streaming audio. With interactive audio, on the other hand, significant amounts of buffering cannot be used, since audible gaps, e.g., during a telephone conversation, prove extremely annoying. Therefore, timing considerations are of equal, or even higher importance than bandwidth requirements for interactive audio.

2.4. Video

Similar to audio, video applications can be categorized as streaming and interactive. The first category consists of video-on-demand applications, using protocols like Active Streaming Format [6], Real Time Streaming Protocol [7] and Vivo [8], along with applications that display stored video clips in formats including MPEG [9], H.261 [10], H.263 [10], Video for Windows [11] and QuickTime [12]. Streaming video has significantly higher bandwidth requirements than audio. The amount of bandwidth required depends on several attributes, including the coding algorithm, the video resolution, the color depth and the frame rate. It can range from 50 Kb/s to 1.5 Mb/s, but most Internet video applications restrict video traffic to a maximum of 128 Kb/s - 512 Kb/s, in order to prevent network congestion.

Timing requirements on the other hand can be more relaxed for video. It was shown that small variations in interarrival delay between frames do not affect the perceptual quality of the stream’s presentation [13, 14, 15, 16]. The same happens with fairly small error rates, especially when they follow a uniformly random pattern. In this case, bit errors are spread throughout each frame and result in barely noticeable erroneous or missing spots that do not alter the perceptual value of the scene. However, if the error patterns are mainly bursty, they result in noticeable blackout periods that degrade the visual experience for the user.

Interactive video comes in the form of video-conferencing [17] and video-phony [18]. As in the case of interactive versus streaming audio, buffering techniques cannot be applied with the same effectiveness as in the streaming case, due to the more stringent timing requirements of the interactive video. Overall, video traffic proves to be the most demanding one, both in display and in transmission characteristics.

3. Internet Infrastructure Characteristics

During the early Internet years, low-speed modems with data rates in the order of just a few Kb/s were the prevailing means of accessing the Web. Only a small percentage of users had access to higher speeds, for example through 1.5 Mb/s T1 connections. Since then, the Internet scenery has changed dramatically and a broad spectrum of access technologies at various speeds has been introduced, adding up to the wide heterogeneity of the Internet infrastructure.

Initial low-speed modems were replaced by high speed ones, operating at 56 Kb/s, and ISDN lines that could combine two B channels to offer 128 Kb/s data rates. ATM stormed in the market promising speeds from 155 Mb/s to 620 Mb/s, but found a great competitor in the form of 100 Mb/s switched Ethernet and Gigabit Ethernet solutions. These technologies provide significantly enhanced Internet experience to the users that share large corporate or university networks. (It seems now that ATM technology will be limited to the backbone network and will have limited direct impact on the end users.) The enhanced connections at work increased the demand for similar personal services provided by Internet Service Providers (ISPs) as home networking solutions. Eventually, the modems will be replaced by xDSL lines capable of delivering 384 Kb/s worth of data and cable modems having performance characteristics similar to Ethernet LANs.

At the same time, wireless access to the Internet is gaining momentum. The first wireless devices offering data services were those using infrared technology for indoor use and CDPD protocols in the cellular context [19]. CDPD offered low quality connections with high error rates and low available bandwidth (up to 9.6 Kb/s). At the same time Wireless LANs (WLANs) were developed, offering data-rates of 1-2 Mb/s with relatively low error rates and the familiar "Ethernet" interface [20]. They were limited basically as in-building solutions for the office environment, providing, however, a very good solution that frees from the wiring burden of a typical LAN, while offering similar services.

Geosynchronous satellites provided download speeds in the order of 400 Kb/s and covered extremely large areas, offering continuous connectivity for mobile users. However, their susceptibility to atmospheric phenomena, their high propagation delay and their increased cost precluded them from becoming a prevailing wide area wireless solution.

The introduction of the second-generation wireless phones gave a great boost to the wireless phone industry and significantly increased the demand for wireless access to the Internet. GSM networks [2] provided text messaging and data services by using the Short Message Service (SMS) [21] and Unstructured Supplementary Service Data (USSD) [22] protocols, but the connection delays remained quite high and the link quality relatively low. General Packet Radio Service (GPRS) [23], High Data Rate (HDR) [85] and other solutions currently under deployment are expected to overcome these impairments and provide enhanced Internet services to mobile users. Concurrently, Wireless LANs are moving towards their second generation with the introduction of the IEEE 802.11 standard [24]. IEEE 802.11 utilizes sophisticated methods to compensate for the increased error rates of wireless links, while offering data-rates up to 11 Mb/s.

Users are able to select today among a great variety of connection types the one that fits best to their Web surfing style. This contributes towards an extremely heterogeneous Internet infrastructure, where the transmission characteristics of the terminals cover a broad range of values. Bandwidth ranges from 9.6 Kb/s cellular modems and 28.8 Kb/s wireline modems to LAN-like cable modems and Gigabit Ethernet solutions. Overall transmission delays begin from a few microseconds and can go up to several seconds for satellite-based connections. Packet error rates start at virtually 0% for wireline solutions, but they can reach a fluctuating 1-5% for wireless links. This heterogeneity builds up as new technologies are introduced to better cover parts of the domain and perplexes the effort for provision of ubiquitous access to Internet multimedia content.

4. Adaptation

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

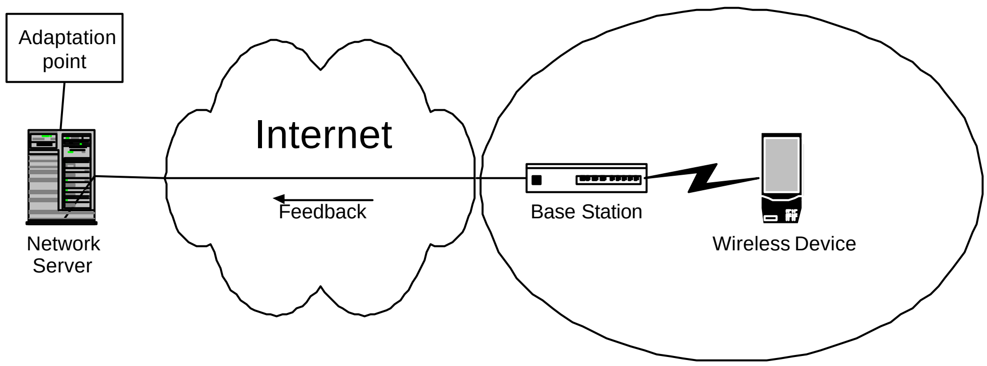

Figure 1: Source adaptation

customize the adaptation process according to his or her needs, through an explicit adaptation policy.

4.1. 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, (ii) at the destination, 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 wireline parts of a network.

4.1.1. Source Adaptation

The first strategy puts the mechanism at the source and requires periodic quality feedback from the receiver, as Figure 1 shows. A typical implementation utilizes a (reliable or unreliable) signaling 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.

Several existing schemes are utilizing this approach, mainly for its simplicity [25, 26, 27, 28, 29, 30, 31, 32, 82]. It can be easily implemented using existing signaling protocols, like RTP [33] and RSVP [34], although there are also solutions using proprietary protocols, like SCP [25, 35]. 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

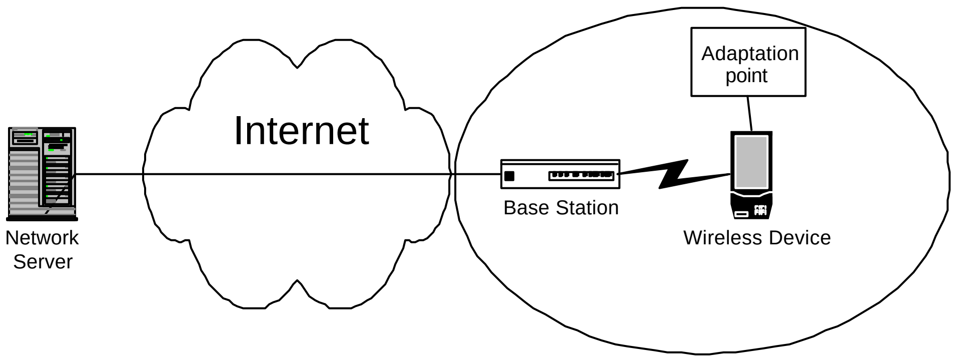

Figure 2: Receiver adaptation

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.

4.1.2. Receiver Adaptation

The second strategy in locating the adaptation mechanism is to have it reside at the receiver, as depicted in Figure 2. 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

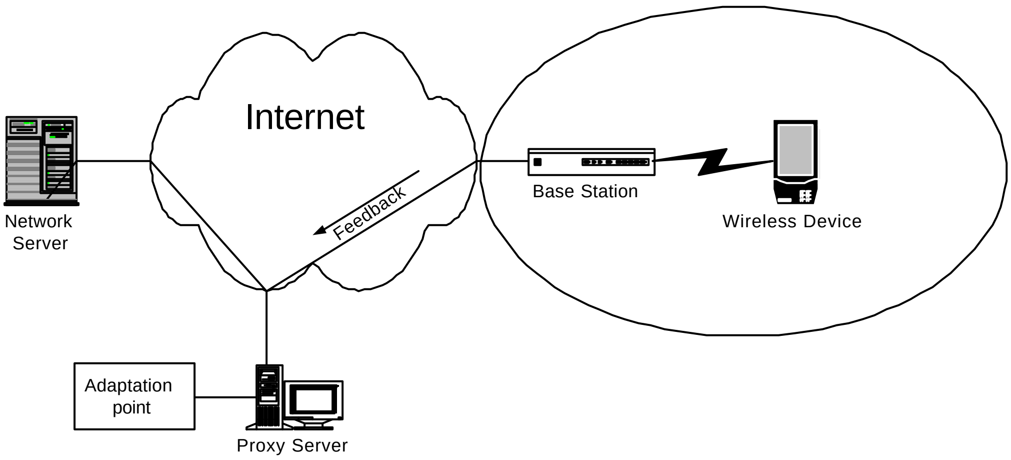

Figure 3: Proxy adaptation

situations where the transmission characteristics of the end-to-end path are less of a concern than the limitations of the displaying device [36]. 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.

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.

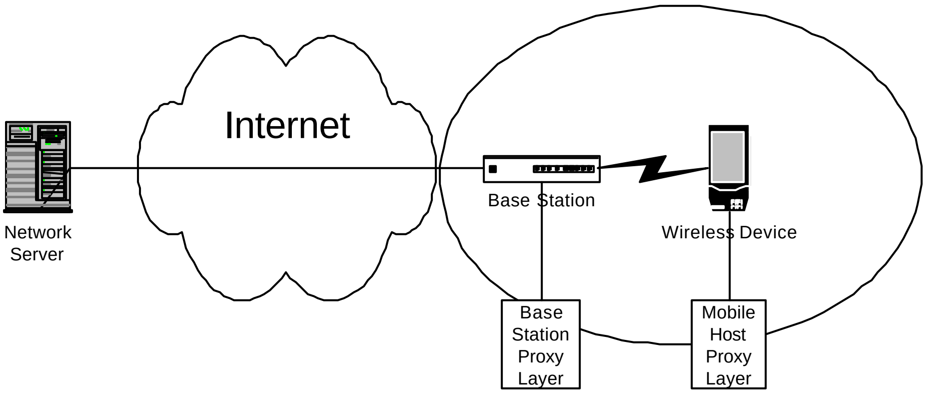

Figure 4: Last Hop Proxy Adaptation

4.1.3. 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 [37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 80, 81]. The intermediate node, usually denoted as *proxy* or gateway (see Figure 3), 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 [41, 48, 49], to move up and down the tree structure [47, 50, 51, 52, 53]. 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 ac[illegible]se where a stationary device uses a wireless link to co[illegible] wireless interface), which in turn uses a wireline connec[illegible] the last (wireless) hop is exp[illegible]d [illegible]e the bottlene[illegible]roxy will perform ideally if it [illegible] router or as [illegible]Figure 4). Finally, c[illegible]der [illegible] a PDA acces[illegible]oaming into a hierarc[illegible]ally s[illegible]y [4[illegible]]. In th[illegible] functionality doesn't h[illegible] to be [illegible] instea[illegible] locati[illegible]ntroller will yield similar performa[illegible] avoiding the h[illegible] time the mobile device hands-over to a new micr[illegible]flexibility

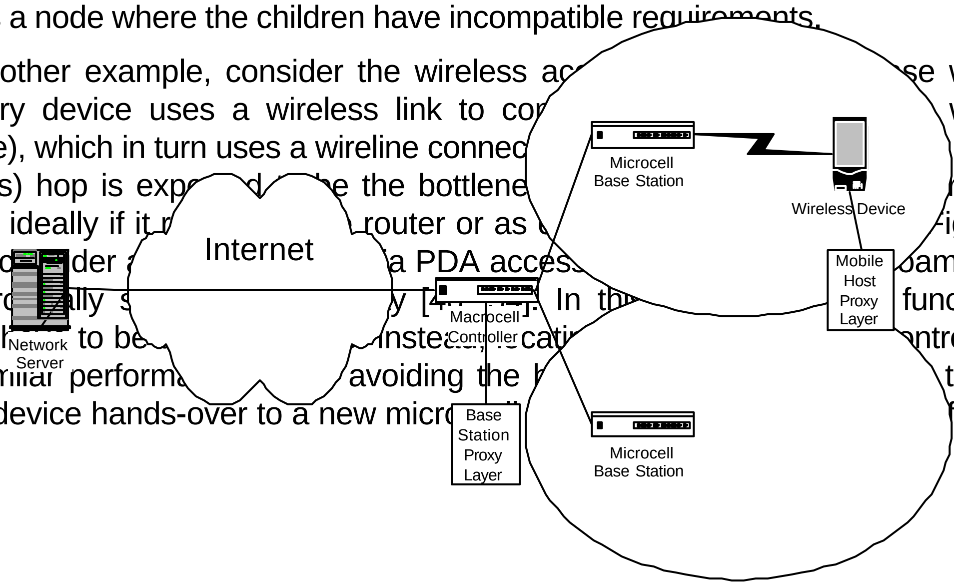

Figure 5: Macrocell Controller Proxy Adaptation

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. 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. Next we present the three categories of proxy adaptation based on the degree of the receiver's participation.

4.1.3.1 Receiver-Unaware Proxy Adaptation (Thin Clients)

The term "thin clients" describe small "dumb" devices that have very limited communication resources and processing power. They are usually capable of reproducing only a few types of media, thus the adaptation process has to be geared towards their implementation characteristics. In this scenario the receiver is unaware of any transformation of the data before they reach their destination. It can only accept and display a single type of each different medium, so it is solely the responsibility of the proxy server to transcode the stream to match with the capabilities of the receiving device.

A well-known example of a thin client is the Infopad project [86], where a wireless pad is designed to receive and display text, graphics, audio and video with the help of specialized Pad servers. Each Infopad has to register with a Pad server, which manages the traffic towards the Infopad. The Pad server cooperates with Type servers that are responsible for the transcoding of media into the specialized formats that the Infopad supports.

The advantage of the receiver-unaware proxy adaptation is that it minimizes the client's complexity in design and implementation. In addition, the resource and power consumption is reduced, which prolongs the battery life and the autonomy of the device. On the other hand, the implementation of such a scenario requires many specialized servers in the Internet infrastructure tuned into serving Infopads. Since the

Infopad can display only a few media types, most of the current Internet content will have to be transcoded before it reaches its destination. This adds significantly to the processing power requirements of the Internet infrastructure. Finally, the wireless portion of the path between the transcoding server and the Infopad quickly becomes saturated since the media are transmitted in raw format without compression. Therefore, this approach has scalability problems.

4.1.3.2 Receiver-Aware Proxy Adaptation

In this scenario the receiver is aware of the adaptation process, but does not actively participate in it. Instead, it notifies the proxy server with the user's preferences and its own displaying capabilities and lets the proxy server decide on the adaptation process. The proxy server manages the existing resources according to each client's adaptation policy and notifies them in the event of a change in the adaptation process. The client then simply adjusts the presentation of the medium to the new format.

Many of the existing solutions follow this approach in designing a proxy adaptation scheme, mainly due to its versatility and effectiveness [37, 39, 40, 42, 46, 47]. The complexity induced in the Internet infrastructure is still high, but the processing power and the communication resources consumed are drastically reduced compared to the receiver-unaware solution. The clients can be versatile and display any kind of media without requiring specialized transcoding in doing so. The only drawback is that this solution cannot enhance the performance of existing non-adaptive applications, since their cooperation is imperative.

4.1.3.3 Receiver-Active Proxy Adaptation

In this third scenario, the receiver actively participates in the adaptation process. The proxy functionality consists of two symmetric parts, usually residing at the two edges of the wireless link. Depending on the direction of the data stream, one of them encodes or transcodes the data in a form that requires fewer network resources, while the other part decodes them in a similar to the initial format before they are delivered to the client. The client itself does not have to be modified; however it must be versatile enough to allow the redirection of the data through the proxy server.

Several examples that follow this scenario can be found in the literature, with the most common ones being those that transcode web pages to reduce the redundancy of the transmitted information [38, 43, 45]. Typical functions applied are header redundancy elimination, text compression, protocol reduction, differencing and caching. The transmitting end of the proxy sends redundant information only once across the wireless link and the receiving end caches it. Then, the transmitting end omits the redundant data from subsequent transmissions, while the receiving end attaches the cached information to the data stream before it is delivered to the client.

TABLE 1
CPU PROCESSING POWER GROWTH WITH TRAFFIC

Adaptation method	Growth
Source	Linear
Receiver	Constant
Proxy (all streams)	Exponential
Proxy (single stream)	Additive

TABLE 2
EFFECTIVENESS WITH INCREASING END-TO-END DELAY

Adaptation method	Effectiveness
Source	Reversibly Proportional to Delay
Receiver	Low, Constant
Proxy	High, Constant

The major advantage of this scenario is that the delivered stream is in its initial format, so the application doesn't need to change in order to take advantage of the adaptation functionality. However, the two ends of the proxy have to use only loseless compression techniques, which severely limits the versatility and effectiveness of the adaptation process.

4.1.4. 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 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.

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.

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.

4.2. 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.

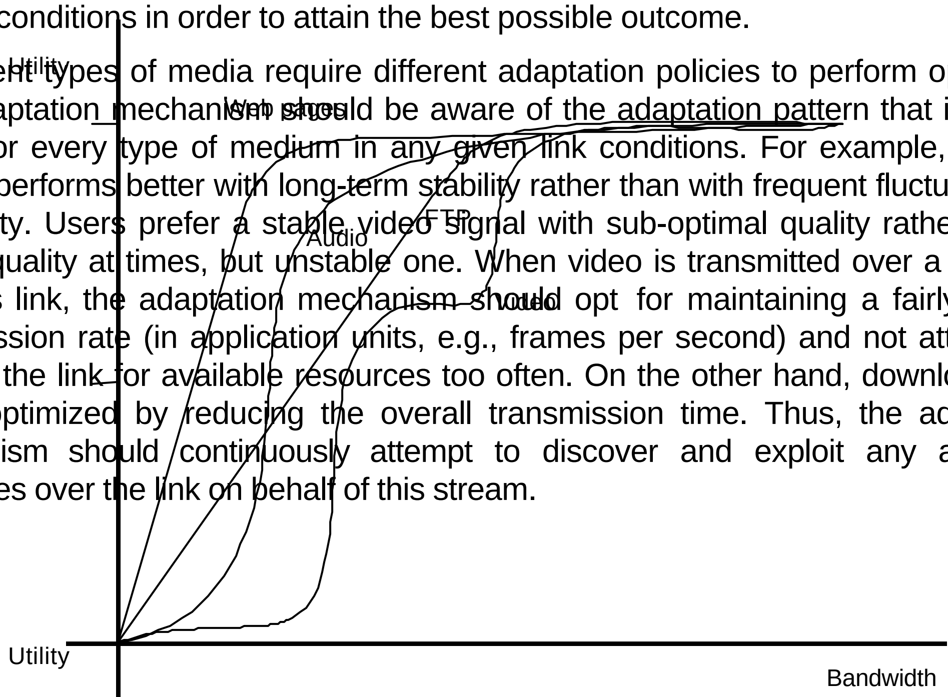

Figure 6: Bandwidth Utility Functions

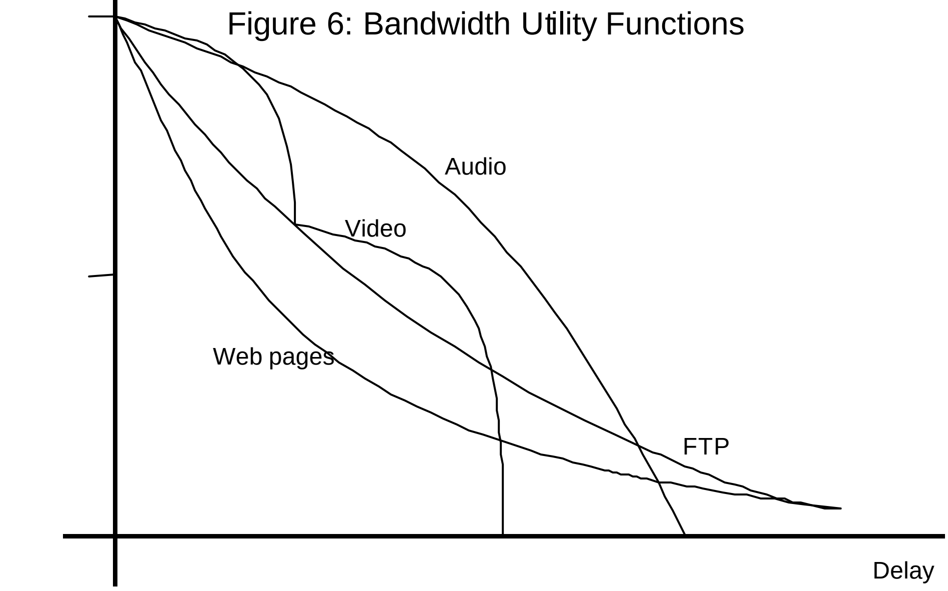

Figure 7: Delay Utility Functions

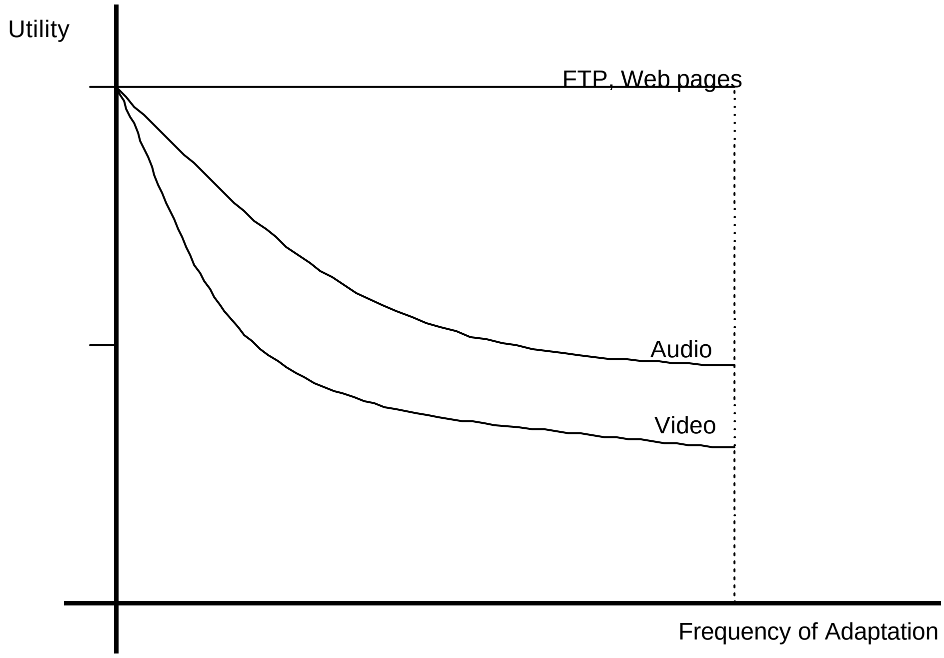

Figure 8: Impact of Adaptation Aggressiveness on Utility

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 6 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 6.

Figure 7 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 [44, 45, 51, 83]. 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.

Finally, Figure 8 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 behavior of each stream throughout a range of possible link conditions [44, 55]. In *MobiWeb*, a set of specialized timers is dictating the frequency and aggressiveness of the adaptation process in a wireless environment [37]. 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 [56]. 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 [57, 58]. 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.

4.3. 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 [38, 41, 42, 43]. Images can be transcoded into another format with different color depth and resolution, but still remain recognizable [41, 59]. Web pages can be filtered to loose redundant objects [38, 43] or to translate the HTML code into another markup language (e.g., WML) that can be interpreted by the displaying device [38, 43, 45, 60]. 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 [61, 82]. 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 [11]. 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 [34, 62], 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 [9, 10] 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 variation 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 regeneration 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 undecodable frames, the adaptation method should strive to accommodate them efficiently. Two methods for doing that are by associating them with a higher priority [26, 32, 37, 44, 63, 67, 80, 81] and by applying an initial buffering scheme that will be able to absorb most of the variations in the inter-frame delay [30, 64, 65, 83].

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 [13, 28, 29, 30, 36, 66].

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 [41, 48, 49]. 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 [37, 39, 40] 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 [26, 39, 40, 41, 42, 63]. 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 [37, 58]. 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 [56, 58, 67]. The original stream is decomposed in several sub-streams containing a portion of the information necessary for displaying it [80]. 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.

4.4. Supporting mechanisms

4.4.1. 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 [25, 36, 39, 40, 41, 42, 68, 80, 81]. 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 [26, 32, 37, 44, 63, 67]. 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 [26, 32, 44, 63, 67]. 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 [37]. 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 [37].

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 [26, 78, 79].

4.4.2. 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 favored, 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 [37]. 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.

4.4.3. 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) [46, 54]. 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 overutilization 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 [26, 46, 71, 72].

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.

5. 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.

5.1 SCP

SCP (Streaming Control Protocol) [25, 35] 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 ressembles TCP in most of its features, but strives to avoid TCP's jittery behavior 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.

5.2 Mobiware

Mobiware [44, 55] 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.

5.3 MobiWeb

MobiWeb [37] 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 real-time 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* instills 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. Commercial Solutions

The importance of adaptation has been recognized by the Internet community and several commercial products are emerging that try to incorporate the concepts and techniques of adaptation in order to enhance their performance. We describe here a few representative ones and analyze their performance.

The most favorable media candidates for adaptation are audio and video. Microsoft introduced the active streaming format (ASF) for storing and transmitting adaptive multimedia streams [6]. A single audio stream and multiple encodings of the same video stream are stored in the same file. The source initiates transmission with the best available quality and adapts the transmission to a lower quality encoding whenever feedback indicates degradation of the end-to-end path. The different video streams have different bit-rate encodings, which constitute one dimension of adaptation. Another one takes the form of adapting the transmitted frames-per-second (FPS) within a given bit rate encoding. By adapting the FPS, the active streaming format can tolerate small fluctuations in quality without having to frequently adapt between different bit-rates.

The advantages of the active streaming format is the seamless adaptation, without any intervention from the user, and the multiple levels and dimensions of adaptation that add versatility and efficiency in the adaptation process. In addition, the source shows preference in transmitting the audio stream over any video stream. The disadvantages of it are, however, several. The encoding of the stream into multiple bit-rates is an extremely computational-intensive process. Since the source has to identify the degradation of the end-to-end path, the adaptation delay can become significant and the adaptation process rendered ineffective. Finally, the format has potential for up to six different video encodings, but only one audio encoding.

In the same context, RealNetworks introduced the RealPlayer G2 [7] for the adaptive transmission of multimedia content over the Internet. RealPlayer G2 supports a variety of media including text, images, MIDI files, VRML, SMIL, MPEG audio and video. Similar to ASF, the RealPlayer G2 stores multiple encodings of a single stream in the same file in order to select and transmit the appropriate one, based on the end-to-end transmission characteristics. The source receives periodic feedback from the receiver indicating the lost packets during the previous period. The source then decides whether adaptation is in order and whether some of the lost packets can or should be retransmitted. This decision is based on a default prioritization scheme built-in at the transmitting source. The similarities with ASF mean that RealPlayer G2 has basically the same advantages and disadvantages as ASF, except from the fact that it allows more media types and multiple encodings for all of them.

IBM is currently offering a commercial transcoding proxy for multimedia adaptation in order to be displayed in small devices, like PDAs and mobile phones [73]. The transcoder defines a pyramid of discrete representations for each of the supported media (text, images, audio, video), within which it navigates during the adaptation process [74]. The product is currently operating to adapt Internet traffic for PDAs and mobile phones. For PDAs, text is summarized, images are transcoded into grayscale equivalents and audio and video is analyzed and presented as text whenever possible. For mobile phones, text is summarized into a single headline, images are omitted and audio and video are transcoded into speech describing them. The proxy can be installed onto any desktop PC and transcodes content on a demand basis.

Finally, Palm Computing introduced “Web clipping” as an HTML transformation process for its product Palm VII [75]. Web clipping tries to transcode the Web content into a fitting representation for the small text-based display of Palm VII and aims in minimizing the energy consumed by the air interface. During an HTML access, the request is redirected to the central server of the company, which acts as a proxy for the PDA. The proxy retrieves the web page, discards the redundant objects and transforms the content to fit in the PDA’s screen. It then starts transmitting only the objects that will immediately be displayed, while it holds the rest of them for later transmission upon request. This way the air interface is used only when the transmission of an object is necessary and, thus, the power consumption is reduced to minimum. Palm VII is currently limited in displaying only text, which narrows the domain of application of this adaptation process. In order to increase the popularity of the device, Palm Computing has come into agreements with major Internet companies that will provide their content in a suitable format for easy and fast adaptation to the characteristics of Palm VII.

7. Current Trends

Demand for ubiquitous access to the Internet is increasing significant and wireless access emerges as an important access mode. The potential for providing access to Web content for the many millions of mobile phone users is just overwhelming. All mobile communications companies have intensified their efforts to provide the best Internet services with the upcoming third-generation (3G) mobile systems. 3G wireless communications will bring a broad range of multimedia content onto new small portable devices, effectively merging mobility with availability. There are currently several proposals concerning the implementation of the 3G air interface: NTT's DoCoMo's wideband CDMA, UMTS, hybrid Time Division/CDMA, wideband CDMA1, and Qualcomm’s High Data Rate (HDR) [76, 1, 77, 85]. All the proposed solutions offer data-rate targets starting from 384 Kbps for unrestricted mobility up to 2 Mbps for low or no mobility. The network layer has been decided to be IP (IPv6), which enables these devices for direct Internet access. The amount of available bandwidth that is offered by 3G wireless communications is expected to move the adaptation focus at a different equilibrium point between the transmission characteristics of the link and the display characteristics of the device.

HDR was announced recently and it is designed based on the 3G CDMA2000-1x technology, also known as IS-2000. It initially offers a 153 Kbps data rate, with the potential to go up to 2.4 Mbps for the forward link and 307 Kbps for the backward one. The interface is compatible with the existing IS-95 CDMA and, since it offers a de-coupled voice-data interface, it can be deployed in parallel with the current wireless infrastructure. HDR devices will be using IP as the Internet protocol, which will allow them to easily interface with the existing Internet infrastructure and gain access to rich multimedia Web content.

Towards the development of the 3G wireless networking, vendors attempt to adapt the Internet content in suitable forms to be displayed in current 2G wireless devices. Following the example of Palm VII, four major companies founded the Wireless Application Protocol (WAP) forum, with the purpose of providing seamless packet services over a variety of air interfaces [60]. The WAP architecture defines a protocol stack starting at the transport layer and going up, which relies on the interface provided by the carrier to implement the addressing and transmission functionality. The advantage of WAP is that it can operate over a great variety of link layer protocols, regardless of the use of IP as the network protocol or not. Applications designed to work with WAP can be easily ported onto any device implementing the WAP stack. In addition, WAP introduces a new markup language called Wireless Mark-up Language (WML), which is suitable for wireless devices with small and depth-limited displays. HTML replies from a Web server are intercepted by a proxy agent which translates HTML into WML and discards objects that cannot be presented by the device before forwarding the transformed content to it over the air interface. WAP is still in its initial stages of development and it is currently supporting only HTML adaptation. The next version, however, intends to provide access to rich multimedia content, including audio and video.

Finally, the continuing trend in traditional telephony favors the migration of telephony services from the old circuit-switching architecture to the packet-switching Internet-based one. Encoding voice for transmission over the Internet has been part of videoconferencing applications for several years now. However, there is an increased interest recently from Internet companies to provide telephony services directly to telephone devices through the Internet infrastructure (Voice-over-IP--VoIP) [4, 5]. This approach can dramatically reduce the charges for long distance calls, and may eventually lead to fixed monthly fees for long-distance telephone services throughout the world, much like today's policy for local telephone services. Taking advantage of packet-switching allows a significant increase in the utilization of the available bandwidth through statistical multiplexing, but contention of voice traffic with traditional Internet traffic might introduce excessive delay, variation in voice quality, and occasionally dropped calls. Therefore, the implementation of a VoIP solution has to also consider the deployment of an accompanying adaptation and resource reservation mechanism that will assure the Quality-of-Service.

8. Conclusions

We identified the importance of adaptation for the ubiquitous access to Internet multimedia content. The broad variety of media types combined with the diversity of Internet connection characteristics raises momentous challenges to the achievement of this goal. With adaptation, the characteristics of various media can be adjusted to better match those of the specific network path used and the end device. As a result, adaptive streams enjoy superior robustness and provide substantially better presentation quality than traditional, non-adaptive ones, especially in variable, capacity-limited environments. Thus, adaptation enables users to efficiently access any type of media content with any type of device from anywhere over the Internet.

There are several important factors that need to be taken into consideration to design and optimize the adaptation architecture. First, the location of the adaptation mechanism on the end-to-end path must be chosen. This choice has significant impact on the effectiveness of the scheme and the operational cost of the solution, mainly in the form of the computational resources required. Second, the adaptation policy must be specified. This is supplied either by the transmitting multimedia application (source) or the user (receiver, client) and describes the agility with which the relevant streams adapt their transmission characteristics in response to variations in the Quality-of-Service provided by the network path. Third, the adaptation mechanism needs to be aware of the detailed attributes of the media streams it handles: the different representations each stream can take, the resources each representation needs in order to be transmitted, and the value of the perceptual quality that each representation provides to the user. This information can be represented in the form of a utility function associated with each stream, describing the utility levels achieved versus the amount of resources utilized.

We also need to emphasize the role of several supporting mechanisms enhancing the adaptation process, namely, prioritization, admission control and hand-off notifications. Prioritizing streams allows the users to identify the relative importance of each stream, which in turn helps the adaptation process to maximize the efficacy (i.e., the sum of all utilities) over the transmitting link. Admission control is crucial for the transmission of real-time streams, since they usually require a certain minimum amount of resources in order to perform at a minimum acceptable level of quality. Finally, hand-off notifications are valuable in high mobility scenarios, where hand-offs are frequent, because they can dramatically reduce the overhead associated with relocating, re-admitting and adapting several media streams onto the new wireless link.

Even though adaptation exhibits many significant advantages discussed above, there are still several steps that need to be taken and open issues that need to be addressed before an adaptation architecture is accepted and widely deployed in the Internet. One is the demonstration of the importance of such a solution in practice, which will convince the otherwise reluctant ISPs to implement the necessary and potentially costly upgrades to their network infrastructure. Another open issue is the introduction and standardization of pricing and charging mechanisms that will provide the appropriate incentives for users to make choices with direct impact on the amount of resources utilized by each stream. This is critical for shared environments, i.e., the vast majority of Internet links, including new wireless technologies such as

GPRS and 3G systems. Finally, there are important security considerations that arise from the interception and reformatting of media streams in the case of proxy adaptation. The communicating entities have to be aware of the security provided by the intercepting proxy in order to decide whether it can be trusted.

References

1. The UMTS Forum. http://www.umts-forum.org; September 2000.

2. GSM World. http://www.gsmworld.com/technology/index.html; September 2000.

3. MPEG Layer III. http://www.mp3.com; September 2000.

4. dialpad.com. http://www.dialpad.com; September 2000.

5. Net2Phone. http://www.net2phone.com. September 2000.

6. Microsoft Inc. Active Streaming Format. http://msdn.microsoft.com/workshop/imedia/windowsmedia/abc.asp; September 2000.

7. RealNetworks Inc. RealPlayer G2. http://www.real.com; September 2000.

8. Vivo Software Inc. VivoActive Player. http://www.vivo.com; September 2000.

9. Moving Picture Experts Group. The MPEG Format. http://cselt.it/mpeg; September 2000.

10. International Telecommunication Union (ITU). Recommendations H.261 and H.263. http://www.itu.int; September 2000.

11. Microsoft Inc. Video for Windows. http://www.jmcgowan.com/avi.html; September 2000.

12. Apple Inc. QuickTime. http://www.apple.com/quicktime; September 2000.

13. Chaddha N, Wall GA, Schmidt B. An End-to-End Software Only Scalable Video Delivery System. Proceedings of the 5th NOSSDAV 1995; 139-150.

14. Boyce JM, Gaglianello RD. Packet Loss Effects on MPEG Video Sent over the Public Internet. Proceedings of the ACM Multimedia 98; New York, USA, 1998; 181-190.

15. Ghinea G, Thomas JP. QoS Impact on User Perception and Understanding of Multimedia Video Clips. Proceedings of the ACM Multimedia 98; New York, USA, 1998; 49-54.

16. 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.

17. Hendricks CE, Steer JP. Videoconferencing FAQ. http://www.bitscout.com/faqtoc.htm; September 2000.

18. Video Contact Inc. Videoconferencing & Videophones. http://www.visualcontact.co.uk; September 2000.

19. Agosta J, Russell T. CDPD: Cellular Digital Packet Data Standards and Technology. McGraw-Hill Computer Communications Series; September 1996.

20. Tuch B. Development of WaveLAN, an ISM Band Wireless LAN. AT&T Technical Journal; July-August 1993; 72(4):27-37.

21. Short Message Service (SMS). http://www.mobileSMS.com; September 2000.

22. Unstructured Supplementary Services Data (USSD). http://www.mobileUSSD.com; September 2000.

23. General Packet Radio Service (GPRS). http://www.mobileGPRS.com; September 2000.

24. Bing B. Measured Performance of the IEEE 802.11 Wireless LAN. Proceedings of the 24th Conference on Local Computer Networks LCN'99; Los Alamitos, CA, 1999; 34-42.

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

26. 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.

27. 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.

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

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

30. 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.

31. 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.

32. 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.

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

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

35. 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.

36. 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.

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

38. Housel BC, Lindquist DB. WebExpress: A System for Optimizing Web Browsing in a Wireless Environment. MOBICOM'96 Demonstration Session; November 1996.

39. 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.

40. Bharghavan V, Gupta V. A Framework for Application Adaptation in Mobile Computing Environments. Proceedings of the IEEE Compsoc'97; November 1997.

41. 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.

42. Zenel B, Duchamp D. A General Purpose Proxy Filtering Mechanism Applied to the Mobile Environment. Proceedings of MOBICOM'97; Budapest, Hungary, September 1997.

43. 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.

44. 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.

45. 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.

46. Brewer EA, Katz RH, et al. A Network Architecture for Heterogeneous Mobile Computing. IEEE Personal Communications Special Issue; October 1998; 8-24.

47. Murkejee A. Supporting Online Services in Environments Constrained by Communication. Ph.D. Thesis, CMU CS Tech Report CMU-CS-98-172; November 1998.

48. 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.

49. Yeadon N, Garcia F, Hutchison D, Shepherd D. Filters: QoS Support Mechanisms for Multipeer Communications. IEEE JSAC; September 1996; 14(7):1245-1262.

50. Pasquale JC, Polyzos GC, Anderson EW, Kompella VP. The Multimedia Multicast Channel. Proceedings of the 3rd NOSSDAV; La Jolla, CA, November 1992; 197-208.

51. 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.

52. 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.

53. Wolf LC, Herrtwich RG, Delgrossi L. Filtering Multimedia Data in Reservation-Based Internetworks. Kommunikation in Verteilten Systemen, 22-24.02; TU Chemnitz-Zwickau, 1995.

54. Stemm M, Katz RH. Vertical Handoffs in Wireless Overlay Networks. Mobile Networks and Applications; ACM Press, 1998; 3(4):335-350.

55. 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.

56. McCanne S, Jacobson V, Vetterli M. Receiver-driven Layered Multicasting. Computer Communication Review; ACM, October 1996; 26(4):117-130.

57. 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.

58. Li X, Paul S, Ammar M. Layered Video Multicast with Retransmissions (LVMR): Evaluation of Hierarchical Rate Control. Proceedings of INFOCOM 1998.

59. 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.

60. The WAP Forum. Wireless Application Protocol. http://www.wapforum.org; September 2000.

61. Steinmetz R, Nahrstedt K. Multimedia: Computing, Communications and Applications. Prentice Hall, July 1995.

62. Topolcic C. ST II. Proceedings of the 1st NOSSDAV; Berkeley, USA, 1990; 5-132.

63. Floyd S, Jacobson V. Link-Sharing and Resource Management Models for Packet Networks. IEEE/ACM Transactions on Networking; August 1995; 3(4):365-386.

64. 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.

65. Sen S, Rexford J, Towsley D. Proxy Prefix Caching for Multimedia Streams. IEEE INFOCOM '99; Piscataway, USA, 1999; 1310-1319.

66. Eleftheriadis A, Anastassiou D. Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video. Proceedings of the 5th NOSSDAV; April 1995.

67. Bajaj S, Breslau L, Shenker S. Uniform Versus Priority Dropping for Layered Video. Computer Communication Review; October 1998; 28(4):131-143.

68. Sisalem D, Schulzrinne H. The Loss-Delay Adjustment Algorithm: A TCP-friendly Adaptation Scheme. NOSSDAV; Cambridge, UK, July 1998.

69. Shenker S. Fundamental Design Issues for the Future Internet. IEEE JSAC; September 1995; 13(7):1176-1188.

70. Breslau L, Shenker S. Best-Effort Versus Reservations: a Simple Comparative Analysis. Computer Communication Review; October 1998; 28(4):3-16.

71. 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.

72. Liao RRF, Campbell AT. On Programmable Universal Mobile Channels in a Cellular Internet. MobiCom'98; New York, USA, 1998; 191-202.

73. IBM Inc. Internet Transcoding for Universal Access. http://www.research.ibm.com/networked_data_systems/transcoding/index.html; September 2000.

74. Smith JR, Mohan R, Li CS. Transcoding Internet Content for Heterogeneous Client Devices. Proceedings IEEE International Symposium on Circuits and Systems (ISCAS), Special session on Next Generation Internet; June 1998.

75. 3Com Inc. Palm VII. http://www.palm.com/products/palmvii; September 2000.

76. America's Network. 3G wireless: Will the best technology win? http://www.americasnetwork.com/issues/98issues/980201/980201_3gwls.html; September 2000.

77. IMT-2000 3G Wireless. http://www.3gwireless.com; September 2000.

78. 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.

79. MacKie-Mason JK, Varian HR. Pricing Contestable Network Resources. IEEE Journal on Selected Areas in Communications; September 1995; 13(7):1141-1149.

80. Bahl P. Supporting Digital Video in a Managed Wireless Network. IEEE Communications Magazine; June 1998; 94-102.

81. 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.

82. 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.

83. 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.

84. IEEE Communications Magazine, Special Issue on Bandwidth Allocation in ATM Networks; May 1997; 35(5).

85. Qualcomm Inc. HDR - High Data Rate. http://www.qualcomm.com/cda/tech/hdr; September 2000.

86. Narayanaswamy S, Seshan S, Brewer E, Brodersen R, Burghardt F, Burstein A, Chang YC, Fox A, Gilbert JM, Han R, Katz RH, Long AC, Messerschmitt DG, Rabaey J. Application and Network Support for InfoPad. IEEE Personal Communications Magazine; March 1996.