Multimedia Streaming over the Internet

Panagiotis Papadimitriou, Sofia Tsekeridou, Vassilis Tsaoussidis
Democritus University of Thrace, Electrical & Computer Engineering Department
Xanthi, 67100 GREECE
E-mail: {ppapadim,tsekerid,vtsaousi}@ee.duth.gr

Abstract

Multimedia streaming technology evolves in accordance with the growing needs of networked end-users for flexible and effective media delivery. The applicability of streaming media is further enhanced by the rapid evolution of the Internet. Our purpose is: (i) to review multimedia streaming technology with respect to the variety of multimedia applications, (ii) to assess the underlying *Quality of Service (QoS)* requirements and limitations - in order to evaluate the overall multimedia streaming performance and QoS under network heterogeneity, we investigate multimedia application requirements versus the QoS provided by the underlying network, focusing on those Internet functionalities and features that mostly influence multimedia streaming, such as its heterogeneity and its TCP protocol performance; and (iii) to finally discuss proposed solutions for efficient end-to-end QoS management of multimedia streaming applications over Internet - we present a comparative survey of remarkable approaches and research proposals functioning both on the application and the network layer.

1. Introduction

The growth of the *Internet* along with the increased bandwidth availability have drastically contributed towards the wide spread of huge amounts of multimedia content through the Internet. Multimedia data are usually of considerable size, requiring long and sometimes intolerable transfer times. The situation, for example, of *"live events"* transmission along with occasionally poor network support, calls for a more flexible and effective approach to deliver multimedia content.

Multimedia streaming enables delivery of video and audio content, directly over the Internet, without the need to download the entire file before playback initiates. During multimedia streaming, the client receives a regular flow of audio/video information. Multimedia content playback at the recipient's device starts after a short delay due to buffering. As a result, the user is able, almost from the beginning, to view the contents delivered to him, and control the download process according to his satisfaction. Further benefits of this method are the consumption of valuable storage space and the capability of delivering *"live events"* and large media files, which can be adapted to the client's bandwidth whenever desired.

The importance of efficient multimedia streaming over Internet is illustrated by its numerous applications, each imposing different *Quality of Service (QoS)* requirements. The sort of content that gets streamed (live and on demand) ranges from TV and radio programmes to videoconferences and meetings. Generally, multimedia streaming applies to entertainment, information, business and education. A large amount of Internet stations worldwide transmit live audio and video content including music, news, sports events, speeches, concerts, movies, and documentaries. Apart from live transmission, a client could gain access to on-demand or pay-per-view material with similar content. Businesses take advantage of streaming in conferences, meetings, seminars and speeches of their executives. The use of video, both live and pre-recorded, has become a very useful and important tool in education and training, as it enriches the learning experience and carries the power of multimedia into the classroom.

Despite these promising aspects, multimedia streaming, is still quite a long way from providing a satisfactory viewing experience. The low-data transmission rates limit the quality of multimedia content transmitted to the recipient. Although the backbone of the Internet is a high-speed data network, potential bottlenecks may appear within the path from the streaming server to the final client. Heavy traffic causes congestion resulting in variable delays and possibly packet dropping. Video is the most demanding type of content a stream can carry, as it has a higher data rate compared to audio, text or graphics. As a result, streaming applications which transmit video content have more severe network requirements. TCP is the dominant protocol for data transmission over the Internet. However, because it constantly alters the rate of the transmitted data, it is not preferable for multimedia streaming applications. As a result, many TCP protocol extensions have emerged to overcome the standard TCP limitations concerning efficient multimedia streaming. Alternatively, some streaming implementations are based on the *User Datagram Protocol (UDP)*. UDP is a fast, lightweight protocol without any transmission or retransmission control. Consequently, UDP appears to be more suitable for applications, such as multimedia implementations, that tolerate some packet losses. However, the lack of

a congestion control mechanism is a significant shortfall for UDP, especially as the Internetworking functionality evolves towards punishing free-transmitting protocols.
Apart from the variety of multimedia streaming applications and their specific QoS requirements, the picture is getting more complex by the diversity of client device and the heterogeneity of core and access networks. The reception of audio/video streams depends on the device the client uses and is affected by the type of the final network link, which delivers the packets directly to the recipient. Taking into account the user-centric character of multimedia services, user preferences may apply as additional quality requirements, related with video resolution, number of colors, compression bit-rate, audio/video duration and buffering time.
Our purpose is to present and analyze multimedia streaming, along with the relevant application and user QoS requirements, as well as their impact on content management and delivery, and network protocol design. Initially, we present an overview of existing streaming technologies and characteristics. Subsequently, we detail the QoS requirements and influencing factors related with the end-to-end application, network and the user/device. In this context, we present QoS management techniques for multimedia streaming. Finally, we comment on recent approaches to manage QoS focusing on context-awareness and content adaptation, as well as, network design and protocol support.

2. Streaming Technologies

Streaming technology allows for real-time broadcast or pre-recorded data transmission, extending traditional multimedia applications such as news, education, training, entertainment and advertising. Streaming falls into two categories: real-time coverage and transmission of live events and streaming of on-demand material. The former application scenario typically involves the transmission of *"live events"* to significant numbers of viewers. Due to the viewer group sizes involved, such one-to-many applications are in practice best serviced by *multicast* networks. Unlike *unicast*, where the nodes establish two-way, point-to-point connections, in multicast, the server sends a single copy of the data over the network and a number of clients receive that data, resulting in more efficient bandwidth utilization. A multicast service, with the exception of playback actions, does not involve any high-level interaction and therefore, clients do not have control over the multimedia streams they receive.
The streaming production process consists of audio/video acquisition, compression and transmission. During compression the multimedia content is encoded using a data rate suitable for transmission over the Internet. If the target of delivery comprises of a wider and consequently heterogeneous audience, the media file can be encoded using multiple data rates allowing each recipient to choose the one most suited to his network states and device capabilities. Packetizing the compressed streams follows and the packets produced are transmitted over the Internet. The final stage consists of audio/video decoding. In order to present an overall picture of the field under study, we briefly discuss: (i) multimedia content acquisition and authoring, (ii) existing multimedia compression techniques and standards, emphasizing on layered coding schemes and (iii) existing streaming server technologies.

2.1 Producing multimedia data

Regardless of the type of multimedia content to be transmitted, this content should be properly acquired, before the compression process takes place. Considering the scenario where data is pre-recorded (on-demand), the appropriate media file(s) should be simply placed in the streaming server. However, in live broadcasts, the requirements are strict and the whole acquisition process gets more complex, as the data to be streamed should be acquired in real-time. Furthermore, the content produced can be edited and special effects can be added using appropriate software packages. The overall goal of this process is to create multimedia of relatively high quality. Apart from the creation of the original multimedia content, this phase also involves the generation of metadata descriptions, a valuable asset and tool nowadays. The purpose is to efficiently describe content. There are numerous metadata standardisation efforts, such as the *MPEG-7* and currently the *MPEG-21* initiatives [1, 2].

2.2 Compressing multimedia data

The next phase in the streaming production process is compression. Generally compression reduces the volume of streaming data for distribution on the Internet while retaining the highest quality possible. In the case of multimedia, data are usually of considerable size. Encoding techniques apply to all kinds of multimedia streaming content, although they are usually implemented in audio and video data.
The most important compression methods for streaming video that are currently used in both compression standards and hardware implementations are based mainly on the *Discrete Cosine Transforms (DCT)*. These lossy compression algorithms highly contribute in the adaptation process, used to account for changing

network conditions, device capabilities, and user preferences. *DCT* is the compression method used in the *MPEG (Moving Picture Experts Group)*, *H.261* and *H.263* standards [1]. An advantage of the *DCT-based* approach is its compatibility with current, as well as imminent draft compression standards. Due to their compatibility and interoperability, the *DCT-based* standards are the standards of choice in implementing practical video transmission systems.

Scalability and layered coding

The term *scalability* is associated with the manipulation of a compressed stream that satisfies constraints on parameters, such as bit rates, display, resolutions or frame rates [1]. The video stream is encoded into a set of cumulative sub-streams (layers), where each layer is a refinement of the previous layers - i.e. layered coding. The so-called base layer contains the most important data of the video. Additional layers are called enhancement layers. The base layer is fundamental for video decoding. Decoding only the base layer, results in a playable video of low quality. Scalable compression allows the compressed stream to be manipulated, even after compression. This is very important, since various applications do not know in advance, during the compression stage, the constraints on resolution, bit rate or decoding complexities.

The perceptual quality of a frame-based video sequence depends on the frame rate, frame resolution and frame quality. By scaling each one of them we achieve *temporal*, *spatial* and *quality scalability* respectively [3, 4, 5]. *Temporal scalability* is the most common scalability technique used in several video compression standards, such as *H-263* and the *MPEG* family. These standards support combinations of such scalabities. Therefore, new hybrid scalable schemes can emerge which would be more efficient for specific applications. A common hybrid scalable scheme is *spatial-temporal scalability* providing 2D space for scaling [1].

2.3 Streaming multimedia data

A streaming server provides streaming services and runs along with a Web server. The streaming server, running in the background, operates similarly to a Web server, whereas it handles and oversees the distribution and access of the multimedia content. In accordance to a Web server, a streaming server can use the HTTP/TCP protocols. However, streaming servers have the extended capability of a more efficient transmission based on the UDP protocol. In order to provide high-quality services, a streaming server confines processing of multimedia content within strict timing constraints, in order to eliminate undesired artifacts, such as jerkiness in video motion and pops in audio during playback [1].

However, the task of multimedia data delivery can be alternatively achieved without the contribution of a streaming server. According to this approach, a regular Web server is used for streaming. It is obvious that the streaming server based approach is the overall best solution for every serious and demanding provider of streaming multimedia content.

3. QoS Management in Multimedia Streaming

Quality of Service is increasingly important for multimedia communication systems. This section provides an overview of QoS requirements and issues that are closely related to the streaming process. Among other applications, multimedia streaming applications require a more sophisticated management of QoS. That is, apart from certain application and network parameters, the QoS requirements of a multimedia application are usually determined by specific user preferences and the receiving device characteristics. Thus, we separately discuss: (i) the end-to-end multimedia streaming application requirements, (ii) the Internet QoS parameters, (iii) associated user preferences, and (iv) influencing device characteristics.

3.1 Not all applications require the same service: Requirements and Taxonomy

The network status reflected by its QoS parameters significantly affects the multimedia application performance. For example, long delays in the network path have a direct impact on the application latency. However, a user cannot directly evaluate the network QoS performance and should not be concerned with the specifics of how a network service is implemented. If a user receives streaming video with jitter (e.g. freezing frames), he is not certain whether the network (delay variation) or the application (e.g. the video is not decoded adequately due to application or hardware constraints) is responsible for this inconvenience. The relation of application-level QoS parameters with network QoS parameters depends basically on the type of the application and the type of multimedia content involved. After a thorough study of the last two aspects, we hereby present a taxonomy of this nature [6], where the application domain is classified in the following categories:

Elastic vs. Inelastic

Elastic applications can tolerate delay and throughput variations, without considerable performance degradation. Although unfavourable network conditions, such as long delays, usually degrade application performance, the actual outcome of the data transfer is not affected. Traditional data transfer applications, such as http traffic, e-mail service and file transfer, are typical elastic applications.
Inelastic are real-time applications which are comparatively intolerant to delay and variations of throughput and delay. They are also affected by reliability parameters, such as packet loss and bit errors. Inelastic applications deliver satisfactory performance only under certain QoS provisions, which may vary depending on the application task and the type of media involved.

Tolerant vs. Intolerant
Tolerant are usually inelastic applications which can tolerate certain levels of QoS degradation and can operate satisfactory within a range of QoS values. Most multimedia streaming applications fall into this category, as they have specific QoS requirements, but they are not extremely sensitive to delays, jitter and packet loss. For example, a video streaming application can tolerate a specific level of packet loss with minimal visual impairments.
Intolerant are applications which operate only under strict QoS requirements. If these QoS demands are not met, the outcome is unacceptable and the application task fails. While intolerant applications do not tolerate the distortion of delay adaptivity, they may be able to take advantage of rate adaptivity. These applications are called rate-adaptive, since they are able to adapt their rate to instantaneous changes in throughout. For example, there are specific video coding algorithms which are able to trade off bit-rate versus quality.

Multimedia Traffic QoS
Multimedia traffic has special properties and characteristics, differing from traditional data traffic. These properties impose challenging QoS requirements for multimedia communication systems. The efficiency of the QoS management is strongly dependent upon the nature of the traffic to be accommodated. In general, multimedia traffic is classified, as: (i) constant bit-rate *(CBR)* and (ii) variable bit-rate *(VBR)*. Although *VBR* multimedia traffic has certain advantages (better quality, shorter delay and lower average bandwidth), it is delay-sensitive, bursty and long-range dependent [7]. Hence, *CBR* traffic is often more desirable, especially when stabilized transmission is the target. Furthermore, *CBR* traffic eases resource allocation and network management, because of its predictable traffic patterns [3].

3.2 Well-defined QoS: Internet QoS
A highly determinant feature of the Internet is its heterogeneity. Besides the transmission channel diversity, new types of devices are also gaining popularity. Not only the underlying network components exhibit different characteristics, but also user devices pose their distinct limitations. Since TCP, is designed for wired Internet, it does not perform well in heterogeneous environments. We outline three major shortfalls of TCP: (i) ineffective bandwidth utilization, (ii) unnecessary congestion-oriented responses to wireless link errors (e.g. fading channels) and operations (e.g. handoffs), and (iii) wasteful window adjustments over asymmetric, low-bandwidth reverse paths [10]. Hence, the applicability of TCP for multimedia applications over heterogeneous networks is limited, since throughput is degraded, and data transmission cannot always conform to the time constraints of some applications. We refer to the major network QoS parameters and we discuss how they affect the performance of multimedia streaming applications.

Bandwidth
Bandwidth is the available capacity of a network channel. A channel with high bandwidth cannot always guarantee high throughput for a specific flow, due to multiplexing. Taking into account on the one hand that most multimedia applications require relatively high data rates and on the other hand that along the path, there may be links with limited bandwidth availability, we observe that the application performance is drastically affected. Wireless links typically offer limited data rates.

Delay
Network delay is expressed by the summation of propagation and transmission delays, as well as, variable queuing and processing delays at the intermediate routers along the path. Propagation delay is fixed for a specific network path and transmission delay is basically determined by the amount of the transmitted data. However, the queuing and processing delays are varying, since routers store and forward packets of multiple applications simultaneously. Generally, increased delays dramatically affect the data delivery, and in some situations may cause data unavailability.

Delay Variation
Delay variation is usually caused by the variable queuing and processing delays on routers during periods of increased traffic and sometimes by routing changes. Delay variation is responsible for the phenomenon called network jitter, which has unpleasant effects in a multimedia application, as packets often reach the receiver later than required. Furthermore, delay variation can result in temporal inconsistency of the multimedia presentation (e.g. freezing video frames). Most applications have specific delay requirements and afford a certain level of delay variation.

Packet loss
Packet loss is typically the result of excessive congestion on the network. Congestion is associated with the condition of the buffers of the routers which are located in the specific network path. A series of mechanisms have been proposed for congestion control, including Congestion Avoidance [8], Slow Start, Fast Retransmit and Fast Recovery. Although packet loss is more likely to be caused by congestion, if the path from the source to the destination contains exclusively wired links, hardly can we be sure for the cause of packet loss, if the network path contains wireless links. In a heterogeneous wired/wireless environment, apart from congestion, hand-offs and fading channels can result in packet loss. Generally TCP is unable to successfully detect the nature of the errors in such a network environment. As a result, TCP does not use the appropriate error-recovery strategy and this has a negative impact on the performance of the multimedia application.

Reliability
TCP is carefully designed to provide reliable transmission. TCP uses a variety of techniques to achieve reliability. Generally the protocol combines retransmission in conjunction with the *sliding window* mechanism. In standard TCP, *sliding window* adjustments are implemented according to the *Additive Increase Multiplicative Decrease (AIMD)* algorithm proposed by Chiu and Jain in [9]. Although the retransmission mechanisms offered by TCP achieve reliability, multimedia applications struggle to operate adequately, since they require a regular flow while transmitting data. Consequently, under these awkward conditions the requirements of streaming implementations are not met.

Contention
Internet is designed to allocate the resources of a network channel equally to each application which uses this channel. The *AIMD* algorithm [9], used in standard TCP versions, achieves stability and converges to fairness when the demand of competing flows exceeds the channel bandwidth. Although TCP incorporates mechanisms in order to eliminate contention, contention conditions between different flows often arise. Practically, TCP mechanisms do not always converge to fair use of resources. Competing flows with aggressive mood can easily have undesirable implications for the network, as they can lead to congestion. In this situation, TCP initiates congestion control which inevitably limits the network resources of the irresponsible applications, as well.

3.3 Users' perception of QoS differs
Management of QoS for a multimedia application includes aspects further than content and network parameters. A notable issue is how the end-user perceives the quality of the multimedia application. It is obvious that user-oriented QoS requirements arise and whenever the application priorities strictly converge to the client satisfaction (in user-centric multimedia services), user requirements should be taken seriously into consideration. The most common user perceived QoS parameters [11] are: picture/video detail, picture color accuracy, audio quality, video frame rate, video smoothness, and video/audio stream synchronization.
Different users are not expected to have equal levels of perception. A user's perception may be more sensitive to video smoothness than video/audio synchronization. Furthermore, users may exhibit tolerance to a high degree of impairments for some cases, but not for others.

3.4 Device Limited Service
The client device is another significant parameter in the multimedia streaming process, as the variety of available devices nowadays results in a plethora of characteristics and in several constraints that should be seriously considered. The device selection is basically imposed by the working environment and the applications to be mainly accessed by and consumed with it. However, other factors, such as business considerations, cost or just personal preferences, may also affect this choice. In order to account for the distinct characteristics of each terminal, terminal profiles carrying information such as size and portability,

display, battery life, data input, processing power and memory could be defined, in a similar manner as the MPEG-21 DIA (digital adaptation part) defines. Automatic awareness and adaptation of content, services and network behaviour may thus be feasible, as explained later on.

4. Recent Proposals

In this section, we review several approaches and research proposals which attempt to provide a more efficient and reliable end-to-end service and support to current multimedia streaming applications.

4.1 Tailoring Multimedia Content

A large number of pervasive devices, such as hand-held computers, PDA, smart phones, TV browsers and wearable computers, are gaining access to the Internet. The latter, as already mentioned, implies network heterogeneity, an additional factor that makes the picture more complex. Diverse devices have variable capabilities for receiving, processing and displaying multimedia content. Taking further under consideration the diversity of user interests in and perception of quality, it is difficult for an Internet provider to tailor multimedia content to the needs and capabilities of all individual devices, all available networks and all groups of end-users. In order to enable universal access of multimedia content and thus the desired end-to-end QoS, a system should be developed that adapts this content according to the characteristics of the device, the network condition and user preferences. Such ubiquitous access can be achieved provided that network services offer adaptation capabilities so as to handle dynamically changing environments. Content adaptation is directly related to context-awareness. Either a system should allow the user to specify his context or the device should by itself have the ability to sense context. In order to achieve the optimal version of the content under the current conditions, the device system should be able to leverage this information. Apart from client-side context awareness and content adaptation, such mechanisms may be implemented at the network level based on monitoring mechanisms of its status and on network protocol design dedicated to multimedia applications, described in more detail in subsequent sections. Below we define context and context-awareness and we discuss about proposed content adaptation mechanisms.

4.1.1 Context-Awareness

In pervasive computing, context is often related simply with location. However, this concept is much more complex [12]. Context is a broader concept and according to [13, 14] may include elements, such as: device capabilities, location time, user's tasks (spontaneous activity, engaged tasks, general goals), physical conditions (noise, light, pressure, temperature), network status (congestion, traffic, packet loss, etc.), movement relative to surroundings, proximity to other users, user habits. The above elements are either specified by the user, or they are sensed by the device and network. Furthermore, when computing, user, network and physical contexts are recorded across a time span, a context history is obtained, which can be used by certain applications (e.g. as a conjectural proceeding to the next context) [15].

A multimedia application can adapt to context provided that it is able to successfully detect this context. According to [15], there are essentially two ways for an application to use context. The application may automatically adapt the behaviors according to discovered context *(active context)*, or may present the context to the user on the fly and/or store the context for the user to retrieve later *(passive context)*. A series of applications which leverage and use contextual information (either *active* or *passive* context), have been developed in the framework of relevant research work. Authors in [15, 21] provide an overview of such efforts. MPEG-21 finally presents the current standardization efforts towards describing context information.

4.1.2 Content Adaptation

Adapting multimedia content is a critical process in order to enable universal access to it by a variety of devices. The efficient leverage and use of context, in the framework of context-aware computing, is the key for content adaptation. Most content adaptation techniques take advantage of the available context information and target at producing the optimal version of the content according to context elements, such as device capabilities, network characteristics and user preferences.

Related research proposals include the development of systems which deal with content adaptation. Authors in [22] propose the implementation of such a content adaptation system. This system incorporates a component, called "Decision Engine", which in two phases (pre-processing and real-time processing) determines the optimal content version for representation. In the pre-processing phase a list of candidate content versions is formed, while in the real-time processing phase the optimal content version is determined based on appropriate negotiation algorithms.

Content adaptation is concluded with the generation of the selected content version. This task is performed by either transcoding: a set of independent encoders re-encodes the stream using different parameters for rate control, or transmoding: usually on-the-fly re-formatting, without any re-encoding.

Transcoding Techniques
Numerous techniques have been proposed and developed for transcoding. Most of them require complex computations. A simple transcoding technique is spatial domain processing [3]. According to this approach, the video stream is decompressed, processed and then recompressed. Because processing is implemented on the original pixels, many operations can be supported (e.g. pixel downsampling or color reduction). The basic shortfall of this method is that the involvement of full decoding and encoding of the video stream is computationally intensive.
Rate adaptation is a class of transcoding techniques which can be efficiently applied to multimedia streams and assess the whole process of multimedia streaming. Rate adaptation techniques attempt to adjust the rate of traffic generated by the encoder according to the current network conditions. In order to detect changes in the network and control the rate of the video encoder, a number of feedback mechanisms are used. We discuss the most remarkable rate adaptation techniques, mostly applied to video streams.

Bandwidth negotiation
Several streaming applications use a rate adaptation technique, called bandwidth negotiation. Before the video transmission, the system estimates the available capacity of the network path. The transmitted stream is adjusted to the capacity characteristics of the specific path. Although this method is very simple, it is not sufficiently flexible, as the available capacity is subject to change at any time.

Simulcast
Simulcast is a technique that uses multiple versions of the stream, encoded at different bit-rate. The versions of streams used are often limited in order to avoid high redundancy. The server switches to the stream version that best matches the client's capacity. Some streaming protocols support dynamic switching among multiple streams.
A sample simulcast protocol is *Destination Set Grouping (DSG)* protocol [16, 17]. In *DSG* the source maintains only three streams carrying low, medium and high-quality versions of the original video. The receiver subscribes to a stream version according to the network states. *DSG* permits the receiver to change to another stream when the current one cannot satisfy its requirements [3].
The main drawbacks of simulcast rely on the fact that it complicates the encoding process and requires extra redundant storage for the multiple encoded versions. In addition, the available rate adaptation options are strictly limited to the number of available streams.

Scalable (layered) adaptation
Scalable adaptation has been proposed as a solution to bandwidth redundancy caused by simulcast. This approach is based on information decomposition. Rate adaptation can be performed by adding or dropping enhancement layers that are transmitted according to the network conditions. A representative layered adaptation approach is prioritized transmission [18, 19]. According to this approach, the packets carrying base layer data are assigned with high priority, whereas the packets carrying enhancement layers data are assigned with progressively lower priorities. Therefore, when congestion takes place, packets containing data from an enhancement layer are dropped. Consecutively, the base layer data are prioritized for transmission. The main drawback of this technique is that this priority-based packet scheduling policy on routers is quite complex (far more complex than *FIFO* policy) and impractical for implementation over the Internet [3]. In contrast to prioritized transmission, a receiver-driven layered multicast *(RLM)* protocol has been proposed [20], which is a practical approach, as it simply requires *FIFO* drop-tail routers. Generally, scalable adaptation methods are particularly useful for networks that employ some form of flow prioritisation and QoS support. A remarkable adaptive multimedia implementation is *MIT's View Station* [23].

Transmoding
Transmoding (or translation) is the process of converting a data object from one representation into another representation without though re-encoding and consists of the following processes: (i) format conversion, when a client device cannot support a format type, e.g. conversion of a HTML page to a WML page for representation on a mobile phone, and (ii) tailoring the multimedia presentation, according to the receiving terminal or user interests, e.g. when the multimedia contents of an HTML page cannot be fully presented on

a small size display and need to be tailored by reducing image and graphic sizes, converting images to text, change navigation guides, etc.. Such a system is able to adapt various types of data, such as video, images, audio and text to individual devices. Translation is implemented based on a framework for on-the-fly summarizing, translation and conversion of the content.

4.2 Dealing with delay variation: Playback

Apart from traffic rate adjustments, there are numerous adaptation techniques which deal with delay variation. Any data of a real-time application arriving at the receiver after the associated play-back point is useless in reconstructing the real-time signal [6]. In order to overcome variable end-to-end delays while transmitting audio or video, a playback buffer is used. The depth of this buffer may vary according to the type of application. Interactive applications, generally, have relaxed delay variation requirements; therefore, they afford a limited playback buffer depth. Other multimedia applications, such as live streaming applications, have strict requirements regarding delay variation, so a larger playback buffer is needed.
Practically, instead of playing-out media data as soon as they arrive, the playback is delayed appropriately, using a suitable buffer, in order to have smooth media playback. The amount of delay, introduced by the playback buffer, is called playback delay and can be either fixed, where each data unit is delayed for a fixed time-interval, or adaptive, where the play-back point of each data unit is adaptively changed according to the experienced delay. The applications that can efficiently adjust their playback point are called delay-adaptive.

4.3 Improving TCP Real-Time Capabilities

A streaming implementation has the option to run over TCP or either UDP. UDP does not incorporate any congestion control mechanism at all, and consequently may provide a regular data rate which is in accordance with most multimedia application requirements. However, congestion episodes do happen frequently in the Internet, due to flows contention. Under these awkward conditions the services provided by UDP are inadequate. Furthermore, the design principles of UDP do not anticipate fairness, thus, any applications running over UDP are not being fair. The above observations render UDP as a protocol with limited potential regarding the streaming process. Therefore, we focus exclusively on the perspective of TCP and especially in selective TCP implementations with enhanced real-time capabilities.

4.3.1 Congestion Avoidance

A congestion episode might have a negative impact on the performance of a multimedia application, regardless of the effectiveness of the TCP congestion control mechanisms. Based on this observation, an approach, dealing with congestion from another perspective, has been proposed. The goal of this approach, called congestion avoidance, is to estimate the level of congestion before it takes place, and hence avoid it.
According to this technique, network traffic loads is constantly monitored in an effort to anticipate and avoid congestion at common network bottlenecks. Congestion avoidance may be achieved through packet dropping. Among the more commonly used congestion avoidance mechanisms is *Random Early Detection (RED)*. *RED* takes advantage of TCP's congestion control mechanism. By randomly dropping packets prior to periods of high congestion, *RED* imposes the sender to decrease its transmission rate until all the packets reach their destination, indicating that the congestion is cleared. Virtually, *RED* triggers congestion control in advance, aiming at limiting the loss to one packet and avoiding a potential congestion collapse or a bursty packet drop.
Authors in [24] propose *ECN (Explicit Congestion Notification)*, a more promising congestion avoidance mechanism. Unlike *RED Gateways*, *ECN* marks than rather drops packets when congestion is about to happen. The benefit is obvious: TCP functionality regarding congestion is enhanced without the need to drop and eventually retransmit packets. However, this method comes at a cost: both TCP and the routers need the appropriate modifications in order to support *ECN*. In heterogeneous wired/wireless environments, *ECN* might have limited functionality: by not receiving an explicit notification the TCP sender will not be able to safely assume that a detected drop was not caused due to congestion [25]. *ECN* would be definitely more beneficial in a more sophisticated TCP, able to distinguish between congestion and wireless losses.
WRED (Weighted Random Early Detection) is a congestion avoidance technique which aims at anticipating and avoiding congestion. *WRED* combines the capabilities of the *RED* algorithm with *IP Precedence* to provide for preferential traffic handling of higher priority packets. *WRED* applies a selective drop packet policy based on *IP Precedence* during a congestion event and provides differentiated performance characteristics for different classes of service. The sender receives an eventual packet dropping as a feedback and directly reduces its transmission rate. If the *Resource Reservation Protocol (RSVP)* is used, *RSVP* flows are prioritized and *WRED* drops packets among the other flows.

A well-designed, congestion avoidance mechanism is *TCP Vegas*. Every *RTT (Round Trip Time)* the sender calculates the throughput rate which subsequently is compared to an expected rate [25]. Depending on the outcome of this comparison the transmission rate of the sender is adjusted accordingly. Based on [26] admissions, *Vegas* achieves better transmission rates than *TCP Reno* and *TCP Tahoe*. Although the protocol is compliant to the rules of fairness (*AIMD* algorithm), according to [27], *Vegas* can not guarantee fairness. Another shortfall of this protocol is that it is not able to distinguish the nature of error. However, this constraint is common to the most TCP versions.

4.3.2 TCP-Friendly Protocols

The disqualification of standard TCP versions to meet the requirements of the multimedia applications outlines the need for a new set of protocols with improved performance and effectiveness. In [28, 29, 30, 31] a family of protocols are proposed, called TCP Friendly. These are TCP compatible protocols which satisfy two primary objectives: (i) achieve smooth window adjustments by reducing the window decrease ratio during congestion and (ii) compete fairly with TCP flows by reducing the window increase factor according to a steady state TCP throughput equation. A significant admission which affected the design principles of the TCP-Friendly protocols is that TCP can achieve application-oriented improvements. This can be approached by using a gentle backward adjustment upon congestion which results in favoring smoothness. However, this modification has a negative impact on the protocol responsiveness. Taking into account that under these conditions TCP has become less responsive, the implementation of an error detection and classification strategy is eventually required [25]. In the following of this section, we present three representative TCP-Friendly protocols which are highly advisable for streaming multimedia applications.

TFRC (TCP-Friendly Rate Control) is a TCP-Friendly, rate-based congestion control protocol. According to TFRC, the transmission rate is adjusted in response to the level of congestion as it is indicated by the loss rate [32]. Unlike standard TCP, the instantaneous throughput of *TFRC* has a much lower variation over time and consequently only smooth adjustments are needed. Furthermore, multiple packet losses in the same *RTT* are considered as a single loss event by *TFRC* and hence, the protocol follows a more gentle congestion control strategy. *TFRC* eventually achieves the smoothing of the transmission gaps and therefore, is suitable for applications requiring a smooth sending rate, such as streaming media. However, this smoothness has a negative impact, as the protocol becomes less responsive to bandwidth availability [33]. *TFRC* has another major constraint: it is designed for applications transmitting fixed sized packets and consequently its congestion control is unsuitable for applications which use packets with variable size. In order to overcome this inconvenience, a *TFRC* variant, called *TFRC-PS (TFRC-PacketSize)*, has been alternatively proposed.

TCP-Real is a TCP-friendly, high-throughput transport protocol that incorporates congestion avoidance mechanism in order to minimize transmission-rate gaps. Therefore, this protocol is suited for real-time applications, as it enables better performance and reasonable playback timers. *TCP-Real* [34, 35] employs a receiver-oriented and measurement based congestion control mechanism that significantly improves TCP performance over heterogeneous networks and over asymmetric paths. In *TCP-Real*, the receiver decides with better accuracy about the appropriate size of the congestion window. Slow Start and timeout adjustments are present, but they are only used whenever congestion avoidance fails. However, rate and timeout adjustments are aborted whenever the receiving rate indicates sufficient availability of bandwidth [10]. In the scenario of multimedia streaming over heterogeneous networks with time-constrained traffic, wireless link errors and asymmetric paths, *TCP-Real* achieves improved performance over standard TCP versions.

TCP Westwood is a sender-side only modification of *TCP Reno* congestion control, which exploits end-to-end bandwidth estimation to properly set the values of slow-start threshold and congestion window after a congestion episode. *TCP Westwood* significantly improves fair sharing of high-speed networks capacity. The protocol performs an end-to-end estimate of the bandwidth available along a TCP connection to adaptively set the control windows after congestion [36]. Although *TCP Westwood* does not incorporate any mechanisms to support error classification and the corresponding recovery tactics for wired/wireless networks, the proposed mechanism appears to be effective over symmetric wireless links due to its efficient congestion control.

5. Conclusions

Based on our discussion, we reach the conclusion that the relation of application-level QoS parameters with network QoS parameters depends basically on the type of the application and the type of multimedia content involved. Furthermore, we have described several approaches both on the application and network layers to

meet QoS requirements, like context awareness and content adaptation with emphasis on scalable adaptation and transmoding, as well as Internet protocol re-design and network support.

References

1. K. R. Rao, Z. S. Bojkovic, D. A. Milovanovic, *"Multimedia Communications Systems: Techniques, Standards and Networks"*, Prentice Hall, 2002
2. J. M. Martínez, R. Koenen and F. Pereira *"MPEG-7: The Generic Multimedia Content Description Standard"*, IEEE Multimedia, pp. 78-87, April-June 2002
3. J. Liu and Ya-Qin Zhang, *"Adaptive Video Multicast over the Internet"*, IEEE Multimedia, pp. 22-33, 2003
4. B. Vandalore, W. Feng, R. Jain, S. Fahmy, *"A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia"*, Real-Time Imaging, Vol. 7, No. 3, pp. 221-235, 2001
5. J. Hunter, V. Witana and M. Antoniades, *"A Review of Video Streaming over the Internet"*, DSTC TR97-10, August 1997
6. D. D. Clark, S. Shenker, L. Zhang, *"Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism"*, In Proc. of SIGCOMM '92, pp. 14-26, August 1992
7. E. Knightly, *"H-BIND: A New Approach to Providing Statistical Performance Guarantees to VBR Traffic"*, In Proceedings of IEEE, INFOCOM '96, Vol. 3/3, pp. 1091-1099, 1996
8. V. Jacobson, *"Congestion avoidance and control"*, In Proceedings of the ACM SIGCOMM '88, August 1988
9. D. Chiu, R. Jain, *"Analysis of the increase/decrease algorithms for congestion avoidance in computer networks"*, Journal of Computer Networks, 1989
10. C. Zhang and V. Tsaoussidis, *"TCP Real: Improving Real-time Capabilities of TCP over Heterogeneous Networks"*, In Proceedings of the 11th IEEE/ACM NOSSDAV, June 2001
11. D. Chalmers, M. Sloman, *"A Survey of Quality of Service in Mobile Computing Environments"*, IEEE Communication Surveys, 1999
12. Peter Tarasewich, *"Towards a Comprehensive Model of Context for Mobile and Wireless Computing"*, In Proceedings of AMCIS 2003 Conference, pp. 114-124
13. A. Schmidt, M. Beigl and H. Gellersen, *"There is More to Context than Location"*, Computers & Graphics Journal, Elsevier, Vol. 23, No. 6, pp. 893-902, December 1999
14. A. Pashtan, S. Kollipara and M. Pearce, *"Adapting Content for Wireless Web Services"*, IEEE Internet Computing 7(5), pp. 79-85, 2003
15. G. Chen and D. Kotz, *"A Survey of Context-Aware Mobile Computing Research"*, TR2000-381, Dartmouth College, 2000
16. S. Cheung, M. Ammar and X. Li, *"On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribusion"*, Proc. of Ann. Joint Conf. IEEE Computer and Comm. Soc. (Infocom 96), IEEE CS Press, pp. 553-560, 1996
17. X. Li, M. Ammar and S. Paul, *"Video Multicast over the Network"*, IEEE Network Magazine, Vol. 13, No. 2, pp. 46-60, April 1999
18. S. Bajaj, L. Breslau and S. Shenker, *"Uniform Versus Priority Dropping for Layered Video"*, In Proceedings of ACM Sigcomm Conf, ACM Press, pp. 131-143, September 1998
19. B. Vickers, C. Albuquerque and T. Suda, *"Source Adaptive Multi-Layered Multicast Algorithms for Real-Time Video Distribution"*, IEEE/ACM Trans. Networking, Vol. 8, No. 6, pp. 720-733, 2000
20. S. McCanne, V. Jacobson and M. Vetterli, *"Receiver-Driven Layered Multicast"*, Proc. ACM Sigcomm Conf., ACM Press, pp. 117-130, 1996
21. A. Dey and G. Abowd. *"Towards a Better Understanding of Context and Context-awareness"* Technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing, June 1999
22. W. Lum and F. Lau, *"A Context-Aware Decision Engine for Content Adaptation"*, Pervasive Computing 1 (2002) 3 (July-Sept), IEEE, pp. 41-49, 2002
23. D. Tennenhouse, *"The ViewStation: a software-intensive approach to media processing and distribution"*, Multimedia Systems, 1995
24. K. Ramakrishnan, S. Floyd, *"A proposal to add explicit congestion notification (ECN) to IP"*, RFC 2481, January 1999
25. V. Tsaoussidis and I. Matta, *"Open issues on TCP for Mobile Computing"*, Journal of Wireless Communications and Mobile Computing, Wiley Academic Publishers, Issue 2, Vol. 2, Feb. 2002
26. L. Brakmo and L. Peterson, *"TCP Vegas: End to End Congestion Avoidance on a Global Internet"*, IEEE Journal on Selected Areas of Communications, October 1995
27. U. Hengartner, J. Bolliger, and T. Cross, *"TCP Vegas Revisited"*, In Proceedings of IEEE INFOCOM 2000, March 2000
28. S. Floyd, M. Handley and J. Padhye, *"A Comparison of Equation-based and AIMD Congestion Control"*, May 2000 URL: http://www.aciri.org/tfrc/
29. S. Floyd, M. Handley, J. Padhye, and J. Widmer, *"Equation-Based Congestion Control for Unicast Applications"*, In Proceedings of ACM SIGCOMM 2000, August 2000
30. Y.R. Yang and S.S. Lam, *"General AIMD Congestion Control"*, In Proceedings of the 8th International Conference on Network Protocols", Osaka, Japan, November 2000
31. Y.R. Yang, M.S. Kim and S.S. Lam, *"Transient Behaviors of TCP-friendly Congestion Control Protocols"*, In Proceedings of IEEE INFOCOM 2001, April 2001
32. L. Mamatas and V. Tsaoussidis, *"Protocol Behavior: More Effort, More Gains?"*, To appear in the Proc. of 15th IEEE Inter. Symposium On Personal, Indoor And Mobile Radio Communications (PIMRC), Barcelona, 2004
33. C. Zhang and V. Tsaoussidis, *"The interrelation of TCP Responsiveness and Smoothness"*, Proc. of 7th IEEE ISCC, July 2002
34. C. Zhang and V. Tsaoussidis, *"TCP Real: Improving Real-time Capabilities of TCP over Heterogeneous Networks"*, In Proc. of the 11th IEEE/ACM NOSSDAV, June 2001
35. V. Tsaoussidis and C. Zhang, *"TCP Real: Receiver-oriented congestion control"*, Computer Networks, 40(4), November 2002
36. S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, and R. Wang, *"TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links"*, In Proceedings of the MobiCom'01, July 2001