11

Multimedia Streaming Service Adaptation in IMS Networks

Tanır Özçelebi and Igor Radovanović
Eindhoven University of Technology, Vodafone Netherlands
The Netherlands

1. Introduction

Multimedia services such as video, gaming and music marked the close of the last century and have become inextricably linked with our lives in the current century. The success and popularity of these services was fuelled by the explosive expansion of the Internet and the furious penetration of broadband networks. In particular, the use of multimedia streaming services on portable devices has been popular whenever both the content and the perceived delivery quality have met the expectations of end users.
This chapter of the book does not address content aspects of multimedia streaming services. Such matters are left to media gurus and other researchers. Rather, this chapter focuses on the delivery quality of multimedia streaming services. Particular attention is paid to quality adaptation techniques intended to improve end users' experience of such services.
Our scope includes heterogeneous networks and devices. The solutions presented are applicable to the telecommunications industry.

2. Drivers of quality enhancement

Before we dive deep into quality adaptation of multimedia streaming services, we would like to address *variables* that affect quality and the objective *quality measures*. In order to avoid confusion with the mathematical variables, we will call these variables *drivers* of quality enhancement. Focusing on the drivers will help us identify issues that need to be tackled in order to provide solutions for increasing quality of multimedia streaming services, whereas focusing on the objective quality measures will help us check whether the provided solutions indeed can provide the desired result (which is quality enhancement).
Quality of a multimedia streaming service, as perceived by the end user, is defined as perceived Quality of Service (QoS) and is mainly driven by the two factors. The first factor is *the end-to-end resource availability* required to compose, transport, process and run multimedia streams. This includes the resources in all the devices and networks through which the multimedia stream is flowing. Here one can think of all kinds of end-devices (e.g. PC, gadgets and mobile phones), network devices (e.g. routers, switches and servers) and physical transport media (e.g. copper cable, fiber and ether). The second factor is the *distance*, i.e. the number of hops and the physical distance, between the source and the sink of the multimedia stream (see Fig. 1).

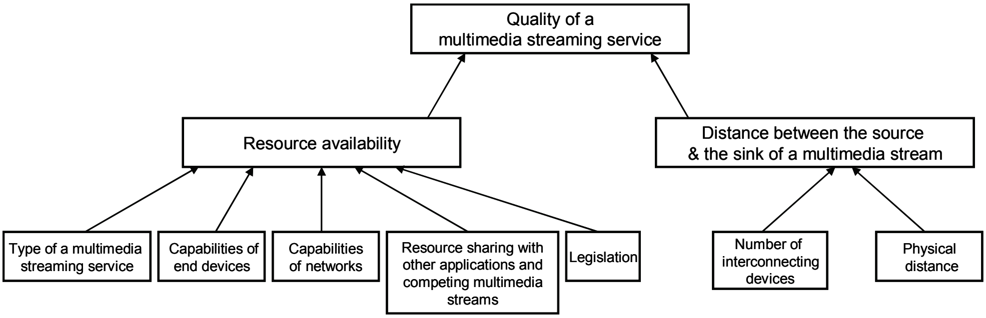

Fig. 1. Quality drivers for multimedia streaming.

The resource availability in turn, is mainly driven by the type of multimedia streaming service, the end-device capabilities, the network capabilities, the presence of other competing multimedia streams and applications, and the legislation regarding multimedia streaming. Let us now elaborate more on these five drivers of resource availability.

2.1 Drivers of resource availability

Not all multimedia streaming services have the same content nor do they all require the same amount of resources to achieve a certain quality. For example, voice, speech and music require throughput of no more than several 10’s or 100’s of Kbits per second. Other services, like streaming video, may easily require an order, or even two orders of a magnitude higher throughput. Therefore, resource availability is a *relative term* that can be measured only after taking resource requirements into account. Obviously, it is easier to provide better quality of services to those services requiring fewer amounts of resources. Moreover, more instances of such services can be supported simultaneously with the same quality, offering more benefits to both the providers and consumers of those services. Note that resource requirements are varying in time, as explained in Section 4.

Although portable devices like PDA’s and smart phones became very resource rich nowadays, the resource availability in those still differs substantially from the desktop PC’s and game consoles. Having more resources in the latter types of devices makes it easier to provide better quality for the end users.

Similar arguments hold for networks. Better network QoS can be achieved for packets transmitted over fiber links than for those transmitted over the air as a result of the different characteristics of these two types of physical transport media. For example, the available throughput is affected by the physical attributes of the media such as the amount of attenuation, dispersion and packet losses, the physical distance and the power of the transmitted signal. An important part of this driver is the type of hardware used in the interconnecting devices in the network. Increased packet processing delay in those devices will boost the total packet delay, inherently decreasing QoS in the network[1].

The presence of other competing streams and devices is also a very important quality driver, which affects *availability* of resources in end-devices and networks. After all, what’s the point of having ample amount of resources in the end-devices when they are all consumed by other services and applications leaving no room for an additional multimedia service to run?

[1] QoS in networks is defined in Section 3.

Finally, the legislation issue needs to be addressed as well. Having no regulation of media and traffic might lead to a situation where some multimedia streaming services are either unintentionally blocked or intentionally degraded in quality based on the multimedia content and ownership.

2.2 Drivers of distance

After describing the resource availability drivers, we would like to address the drivers of the distance between the source and the sink of the multimedia stream. The first driver, as depicted in Fig. 1, is the physical distance that directly determines the propagation delay in the network. The larger the distance, the larger the propagation delay and thus lower the perceived QoS of the multimedia streaming service. The more interconnecting devices we have the larger the total delay, which addresses the second driver.

3. Objective quality measures

For both providers and consumers of multimedia streaming service, high *perceived QoS* is of utmost importance. The perceived QoS is a subjective measure of the user experience, and is highly correlated with the measured *objective QoS*. The main objective measures used to describe QoS throughout a single multimedia service session are picture Peak Signal-to-Noise Ratio (PSNR), frame refresh rate (video smoothness) and continuousness of multimedia playback (Ozcelebi et al., 2007). Here, a service session is defined as a durable connection between the two end-devices or an end-device and a server. The more efficient the video compression algorithm, the less network bandwidth is needed for a video streaming service to achieve a certain perceived QoS level. Video codecs in the literature such as MPEG2 (ISO/IEC, 2000) and AVC/H.264 (Wiegand et al., 2003) try to achieve such efficient video compression (encoding) such that the perceived QoS of the decompressed (decoded) video is maximized.

Firstly, PSNR is used very commonly as an indicator of this perceived quality. Its value is determined by the power of the pixel noise introduced to the video as a result of lossy compression. As the average power of the pixel errors (mean square error) in video frames introduced during compression increases, the PSNR value decreases. Therefore, assuming ideal transport conditions, higher PSNR value means that a given decoded video at the receiving side is more similar to the original video, i.e. that it is of higher quality.

The second factor determining the perceived QoS is the video frame rate. The human eye perceives our surroundings in a continuous way instead of a frame-by-frame manner as introduced by video technology. This means that as the frame refresh rate of a video decreases, it gets further and further away from what the human eye would perceive. This issue becomes especially troublesome for video scenes with a lot of movement. For a smooth and natural video perception, it is necessary to operate at sufficient frame rates. In practice, 25 fps (frames-per-second) and above is thought to suffice for a decent video experience, although higher frame refresh rates are more suitable for the human perception.

Finally, interruptions during video playback are quite undesirable in terms of perceived QoS and they must be avoided. The receiving side of a video streaming service does some video pre-buffering prior to starting the video playback in order to compensate for possible variations in network throughput and end-to-end delay (i.e. delay jitter). Video playback is started only after the receiving video buffer is at least partially filled (Ozcelebi et al., 2007).

Therefore, there is typically a delay between the time the first video packet is received and the time it is actually displayed on the receiver's screen, called the pre-roll delay, which remedies the negative effects of unpredicted channel behavior. However, if the video encoding rate is higher than the channel throughput for a long period of time, it is possible that the receiver buffer underflows (is emptied) and there are no more video data to display. In this case, the video playback is interrupted and the receiver needs to wait for another pre-roll delay before the video playback can continue, jeopardizing perceived QoS level.
Perceived QoS, in case of multimedia streaming services, is determined by the *intrinsic QoS*, i.e. QoS at the network level. In telecommunications networks, intrinsic QoS of a certain multimedia stream is measured using *throughput*, *delay*, *jitter* and *loss-rate*. In order to maximize intrinsic QoS, throughput has to be maximized, whereas delay, jitter and loss-rate have to be minimized, or at least, bounded to a certain limit depending on the service characteristics. The following table describes requirements regarding these intrinsic QoS parameters for 7 different types of services.

Application/ requirement	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

Table 1. Requirements for high intrinsic QoS of various applications

4. Maximizing perceived QoS

To achieve high perceived QoS, required resources in both the network and the end-devices must be available at *all times* during service provisioning. Moreover, the total delay has to be under a certain minimum level depending on the particular multimedia service. For a voice service it is typically below 150 mSec (on direction), whereas for video it is below 200 mSec. We specifically stress upon this issue since both resource availability and resource requirements fluctuate continuously and substantially during a single session. Resource requirements of a multimedia streaming service may vary drastically in time according to the encoding bitrate and the scene complexity (e.g. amount of spatial pixel variations, temporal movement) in each temporal segment.
Intrinsic QoS can be maximized by combining the usage of proper intrinsic QoS mechanisms with an overprovisioning in the network (Aurrecoechea et al., 1996). Intrinsic QoS mechanisms such as admission control, resource reservation and traffic engineering can be used for this purpose. The pre-requisite for using these intrinsic QoS mechanisms, however, is to have end-to-end control of resources, from the source to the sink of the multimedia stream. Note that this pre-requisite is easily satisfied by telecom operators as they have full control over their networks.

Despite such mechanisms, quality of a telecom multimedia service may still be jeopardized by several factors, e.g. Doppler Effect, shadow fading and multi-path interference in wireless networks. Furthermore, the sink of the multimedia stream is strictly speaking not in the network but in the end-devices, whose resources are not under control of the telecom operators. In the end-devices, resources such as battery, CPU power, memory size and storage space vary in real-time during a single multimedia session, making hard guarantees for a certain level of perceived QoS difficult to achieve.

This problem has become even more acute with the launch of the IP Multimedia Subsystem (IMS) framework as a telecommunications networks initiative (3GPP, 2006). The IMS framework allows usage of the multimedia services from unmanaged access networks that are beyond the operator's control (e.g. WiFi, xDSL). Thus, it is very difficult to control the entire physical bearer infrastructure of the end-to-end channel in a unified way, e.g. by a single operator. This, of course, makes QoS mechanisms such as admission control practically inapplicable. In principle, IMS operators can control call admission in all access networks by enforcing a Service Based Local Policy (SBLP) via the Policy Decision Function (PDF) at the access network border. However, this can be done only if the access network bearer reacts upon the commands from the signaling and control layers in the IMS core, which is generally not the case.

Moreover, even if all IMS networks were somehow controllable, the operators cannot give hard guarantees for resource availability during the whole session due to factors such as fading and interference as explained above. Interference may come from other sessions in the same network technology or signals from different sources that operate in the same frequency band. Some IMS access networks (i.e. 802.11 WiFi networks) utilize the shared Industrial, Scientific and Medical (ISM) frequency band, making them highly vulnerable to interference from other technologies as well, e.g. Bluetooth, microwave oven. These interfering technologies generate errors in real-time packet transmission that are mainly recovered by retransmissions. However, these retransmissions inherently increase latency and decrease effective throughput making provisioning of high perceived QoS extremely difficult.

5. The IMS framework

The IMS is an architectural framework for backward compatible Next Generation Network (NGN) as explained in (ITU, 2009). Backward compatibility implies that the new NGN technology can seamlessly integrate with the legacy one, i.e. 2G and 3G telecom networks and services. The IMS separates functionality of services from the underlying network architecture which means that services offered to the end users are network neutral. This in turn implies that perceived quality of a particular service must be the same irrespective of the network technology used. However, since unmanaged IMS access networks are allowed as well, it becomes much more difficult to achieve the original requirement. The IMS layered architecture is shown in Fig. 2.

The IMS infrastructure allows utilization of IMS services from any IPv6 capable end-device and access network possibly not owned by the telecom operator, as depicted in Fig. 3. These access networks can be unmanageable by telecom operators, meaning they may be unlicensed, non-dedicated and non-deterministic with no QoS control (3GPP, 2006). Therefore, end-to-end QoS reservations may be inapplicable and network bandwidth may

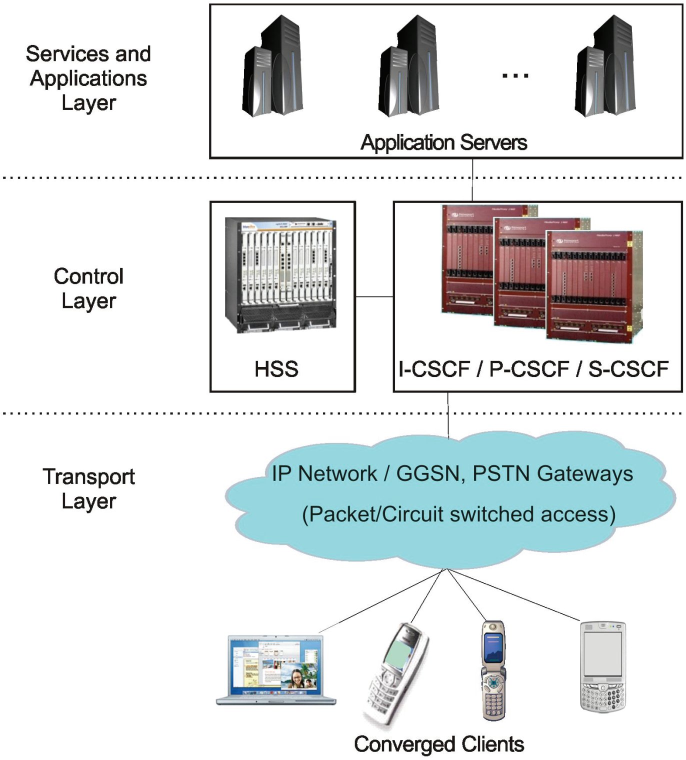

Fig. 2. The IMS layered architecture.

be reserved unnecessarily within the operator's network, since making the same reservation may not be possible in an unmanaged access network. This bandwidth could better be used for other sessions. For example, pure Ethernet-based and public broadband access networks are unmanageable. Therefore, it would not be possible to reserve network resources and guarantee their availability for the duration of a single session, causing perceived QoS to suffer when there is a shortage of network resources.

This is the reason why QoS cannot be guaranteed, and the best-effort QoS model has to be adopted (3GPP, 2006). However, the best-effort QoS model used in the Internet does not support operator's business model that is based on the user's high expectation of the perceived QoS. We strongly believe that in order to satisfy user's expectation, the operators need to adopt *the best of the best-effort QoS model*. This model implies that the best-effort perceived QoS has to be maximized at all times. The way to achieve this goal is described in this chapter. The proposed solution enhances the perceived QoS of multimedia services in the application layer, in situations where the real-time multimedia service QoS cannot be guaranteed in the network layer. The perceived QoS is adapted according to the system overall resource availability. The quantitative value of the system overall resource availability is updated using a signaling mechanism during a single multimedia session. Since the current IMS architecture lacks such a mechanism, as described in the following section, the authors proposed a solution, which is presented in Section 8.

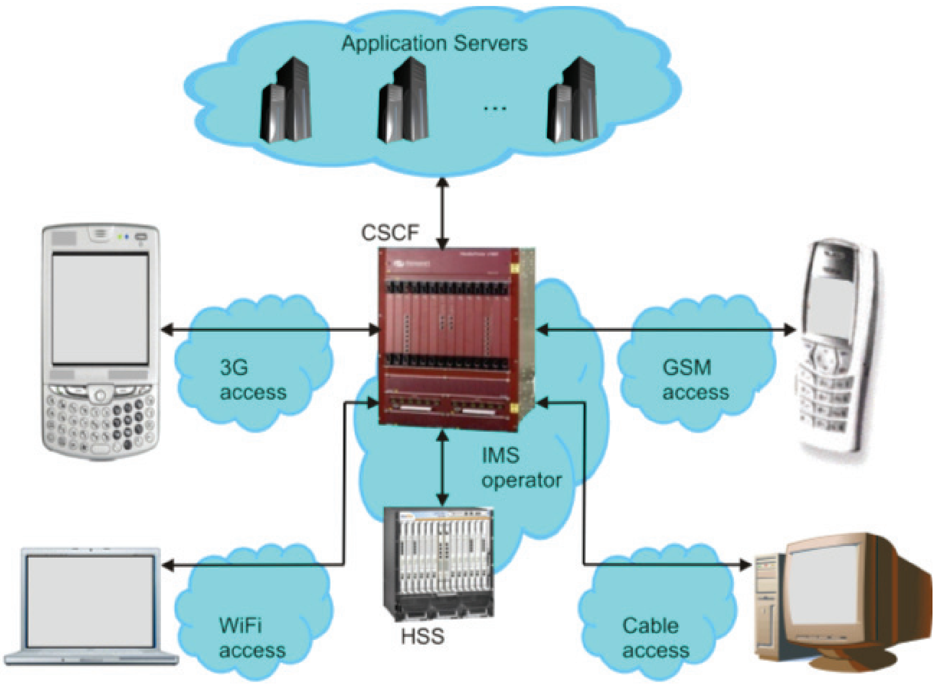

Fig. 3. Architecture of the converged IMS based network.

6. QoS negotiation during the IMS session establishment

In IMS, a call session control is implemented in the signaling and control layers (3GPP, 2006). With the call session control, it is possible to negotiate the QoS parameters of a given session in IMS standard specifications at the beginning of the session using the SIP control messages. QoS parameter negotiation is necessary for the support of variety of end-device types, media codecs, access network technologies and types of user subscriptions. The message flow diagram for such negotiation is shown in Fig. 4.

The caller user equipment (*UE*1) sends a SIP INVITE (Rosenberg et al., 2002) message to the call receiving user equipment (*UE*2) including her proposal for the QoS parameter values. This proposal is checked against the users' subscription credentials and modified if necessary in the Serving-Call Session Control Functions (S-CSCF) serving the two users. The information about the user subscription is sent to S-CSCF by the Home Subscriber Server (HSS). The reason to check for the user's subscription credentials is that the user might not be allowed to use some QoS parameter values due to the type of her subscription.

The reply message (SDP answer) from the call receiving user equipment (*UE*2) includes a set of QoS parameter values that are either the same as the ones received from the S-CSCF (originally sent from *UE*1), or different, depending on the availability of resources in the access network and terminal, media codec type used and the type of user's subscription. This parameter value set is then again checked in the S-CSCF's, which in turn might modify those. Finally, the SIP reply message is forwarded to *UE*1.

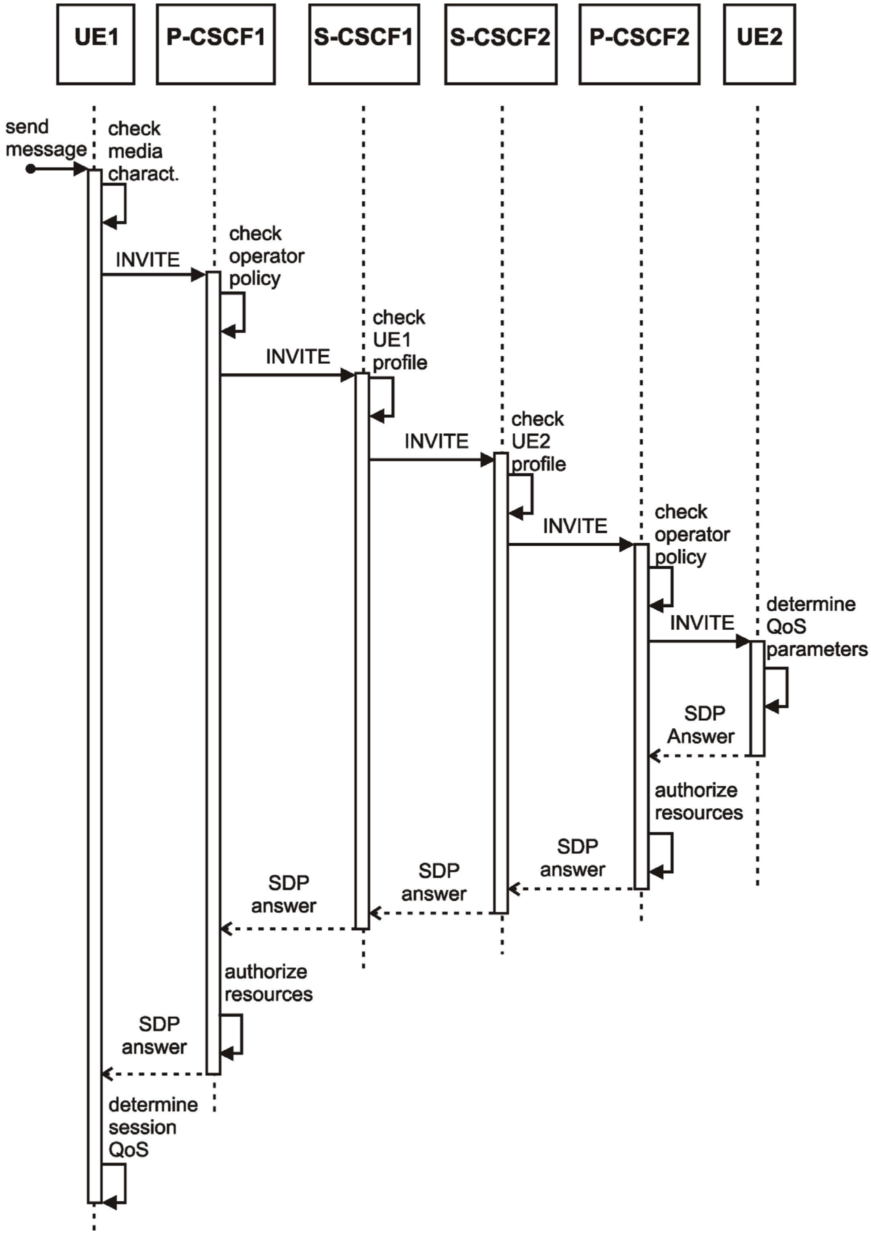

Fig. 4. Message Sequence Diagram showing pre-session QoS negotiation in an IMS network.

Note that, not all the messages such as acknowledgements (ACK) are shown in Fig. 4, for the sake of simplicity. It may take a while before the resource reservation can be finalized at the beginning of a media session, e.g. when the network resources are not available over a specific bearer. Furthermore, after the first QoS offer (see Fig. 4), it is possible that *UE*1

decides to change some session QoS parameters during the same session (e.g. choosing another codec). Therefore, only one QoS negotiation round before the session start may not be enough. *UE*1 can update its QoS offer by sending a SIP UPDATE message that includes the new QoS parameter values. Similarly, *UE*2 can make a new offer with an SIP UPDATE message as well. Therefore, *UE*2 must not inform the caller (by ringing) until the required resources for the session are surely negotiated and reserved. The final values of QoS parameters are set only after *UE*2 sends a 200 OK response to the SIP INVITE message. The quantified QoS parameter values are communicated through the Session Description Protocol (SDP) inside these SIP UPDATE messages.
In reality, the above pre-session QoS negotiation cannot offer a good perceived QoS due to lack of bearer control, nondeterministic networks and interference. In order to achieve enhanced QoS, session QoS parameters need to be renegotiated and modified very often during a single session, adapting to the varying system resource availability in real-time.

7. Literature review on several QoS signaling mechanisms in the IMS

There are several RFCs published by the Internet Engineering Task Force (IETF) that propose resource/capability signaling among end-devices. In (Camarillo et al., 2002), a method that integrates resource management (specifically RSVP) and SIP signaling is introduced in order to make network resource reservation *before the session is established*, i.e. before the called end-device is alerted such that session establishment failures are avoided. However, this RFC proposes no signaling *during* a single session. In our proposed solution it is assumed that reservations may not be possible in the access networks, and therefore the relation with the reservation protocols, like RSVP, is not considered in this work. Furthermore, in our solution, signaling is done during a single session. Another difference is that, we propose an architecture in which information about the resource availability of *both* the end-devices and networks can be transmitted to other interested parties, whereas (Camarillo et al., 2002) gives information about the network resource availability only.
In (Nomura et al., 2006), Internet Media Guides (IMG), i.e. multimedia session descriptions, which can use the SDP format are introduced. However, it is denoted in (Nomura et al., 2006) that SDP syntax causes a huge amount of overhead in delivering IMG metadata over the network and SDP can carry only a small subset of IMG metadata in practical cases (e.g. codec type).
The bandwidth modifier of (Westerlund, 2004) notifies the receiving end-device on the maximum media codec rate to be used and the communication bit-rate required for the bit stream. Thus, (Westerlund, 2004) aims to convey bit-rate information only, without conveying any information about end-device resource availability.
In (Casner, 2003), bandwidth modifiers for RTP Control Protocol (RTCP) are introduced to SDP such that the amount of bandwidth allocated to RTCP in an RTP session is adapted (typically kept below 5% of the overall data rate). We envision that SIP resource availability signaling is preferable for protecting the privacy of resource availability data compared to transport layer protocols (e.g. RTCP), which lack to provide means for authentication, encryption and billing.
An extended SDP protocol for capability declaration (e.g. codec) amongst end-devices to be used in multimedia sessions is introduced in (Andreasen, 2002). It is declared that such capability declarations can be intended for session negotiation, but such session negotiation mechanisms are not described.

In the IMS, it is envisioned that the end-to-end QoS negotiation and resource allocation should be reevaluated during the session depending on requests from the application, network load and link quality (3GPP, 2006). On the other hand, the implementation specifics of such a *QoS renegotiation* mechanism are not provided.

8. Resource availability signaling during a single session and service quality management

As explained in the previous section, the session QoS parameter values have to be renegotiated and modified very often during a single multimedia session in order to adapt the multimedia content quality to the varying resource availability in the system in real-time, maximizing the perceived QoS. This section gives a solution to this problem.

To solve the problem of monitoring resource availability in real time and sending the data which quantitatively describes it, we have introduced two software components for resource availability monitoring and resource availability signaling in real time. Those are the Resource Management (RM) module, and the Resource Availability Server (RAS), respectively. To adapt the multimedia content to the resource availability, we have introduced a Service Quality Management (SQM) software component based on which a real-time adaptation of multimedia communication streaming and stored multimedia data streaming (e.g. video-on-demand) services in the IMS network is made.

The RM module is a crucial part of the proposed resource availability signaling framework. It is responsible for tracking the available device and network resources in real-time. At the receiving device, the RM module publishes the resource availability information in the RAS server in which this information can be accessed by the remote transmitting device. At the transmitting device, the RM is responsible for gathering the resource availability data of the remote receiving device from the RAS. Based on this information, the content quality of the multimedia streaming service can be adapted.

It is the proposed RAS server that is responsible for collecting resource availability information from the receiving end-devices and delivering this information to the transmitting end-devices. Note that in a multimedia communication scenario, e.g. video-conferencing, an end-device can be transmitting multimedia, receiving multimedia, or both.

The proposed SQM module is responsible for adapting the perceived quality of the streamed multimedia service according to the real-time measured resource availability, such that the resource requirements of the streaming service is always kept below or equal to the actual resource availability in the end-devices and in the network. In this way, users are presented with a more reliable and higher quality user experience, where the perceived QoS is boosted when plenty of resources are available and reduced when the resource availability drops. Without adaptation of the perceived QoS, a multimedia session can experience variations in picture and sound quality and re-buffering events (interruptions in playback).

The solution presented here, which includes the introduction of the three software components must be IMS compliant in order to be practically implemented. For the sake of having an IMS compliant architectural solution, we have employed the existing SIP call session control protocol of IMS for resource availability signaling in the architecture we propose. Here it is assumed that the user equipments *UE*1 and *UE*2 have registered to each other's resource availability information at the RAS server. The proposed resource availability signaling and the proposed software components (i.e. RM, RAS and SQM) are depicted in Fig. 5 and Fig. 6.

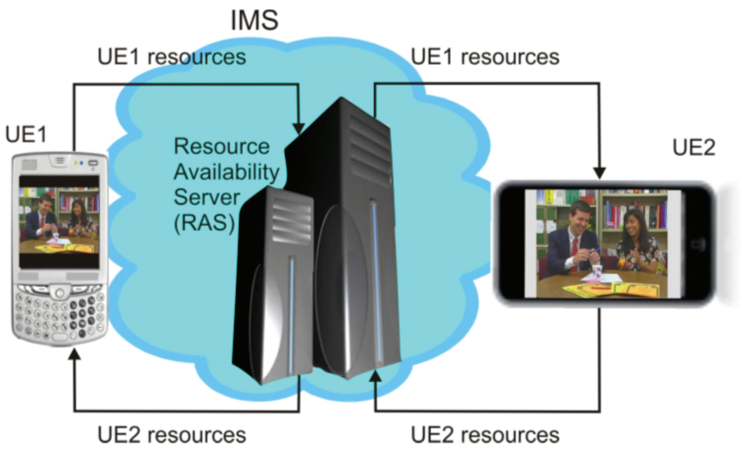

Fig. 5. The proposed system architecture for the mid-session SIP-based resource availability signaling.

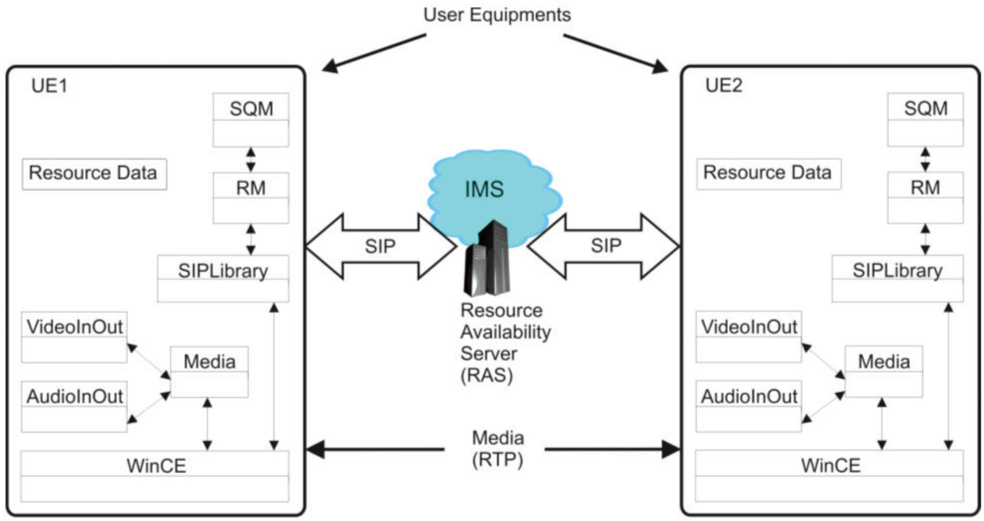

Fig. 6. Deployment view of the user devices and the IMS core.

The resource availability data is carried from *UE*1 and *UE*2 to the RAS and back in the SIP event notification (SIP NOTIFY) messages, as shown in Fig. 6. A resource update is signaled whenever the availability of local resources (e.g. memory, CPU, storage etc.) or the network resources (e.g. bandwidth, jitter etc.) at one end cross critical boundary threshold values. When the resource availability is not changing very fast, resource availability update messages can be quite infrequent. Therefore, the amount of end-device and network resources spent on the resource monitoring and signaling is negligible in the proposed

framework. A worst case scenario is investigated in the next section, where the maximum bitrate adaptation frequency (i.e. inversed minimum response time) of a video encoder is considered as an upper bound on the resource availability update signaling frequency.
The proposed message flow diagram from the end-device to the RAS for the resource availability signaling is depicted in Fig. 7 and our additions to the SIP/SDP parameters as resource indicators are shown in Table 1.

r	::=	"memory" | "CPU" | "storage" | "throughput" | "battery"
t	::=	"Mbytes" | "Kbytes" | "kbps" | "seconds" | "percentage"
a	::=	<resource availability measure>

Table 1. Proposed Additional Resource Data in SDP

After the addition of the proposed resource availability parameters, an example of the SDP resource availability update message is shown in Table 2. Here, both users will be aware of each other's local and network resources and SQM can use this information to adapt the multimedia content in order to maximize the perceived QoS.

```
NOTIFY sip:abc.somename.com SIP/2.0
Via: SIP/2.0/UDP abc.tue.nl:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
To: Bob <sip:server.somename.com>
From: Bob <sip:abc.somename.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 NOTIFY
Contact: <sip:abc@192.0.2.4>
Expires: 7200
Content-Type: application/sdp
Content-Length: 131

v= RM/1.45
o= abc 5876768686868 7698798797979 IP 1.2.3.4
s= 123456789
i= Resource update to presence server
c= IN IP4 1.2.3.4
b= 100 kbps
k= none

r= memory
i= free memory status
t= kbytes
a= 12450

r= battery
i= battery charge remaining
t= percentage
a= 86
--msg ends--
```

Table 2. Example SDP with Resource Data.

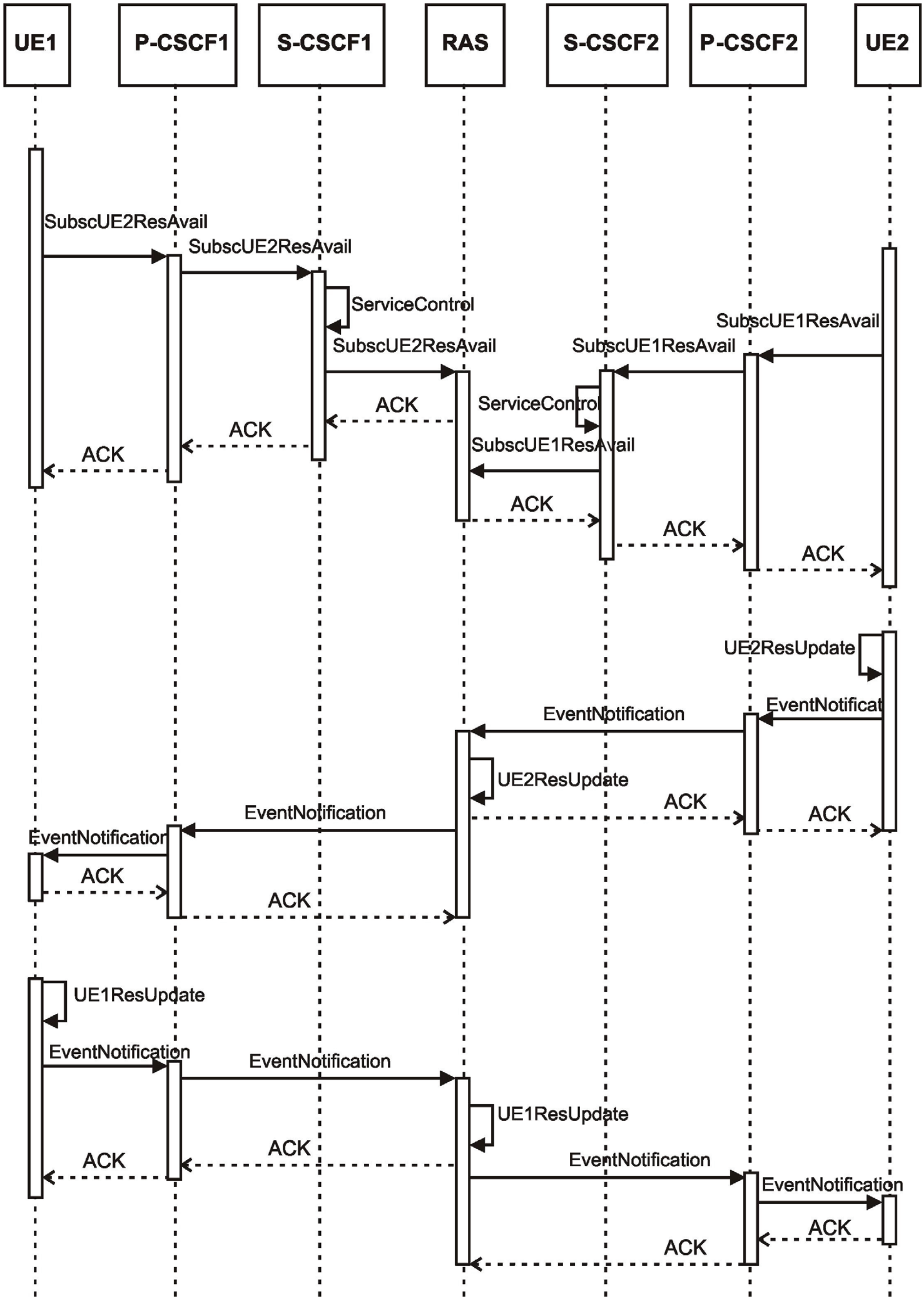

Fig. 7. Message Sequence Diagram showing SIP based resource availability update signaling.

9. Overhead of the signaling mechanism

For the proposed solution to be practically implementable, it is important that the introduced signaling overhead is negligible when compared to the original multimedia data. The maximum overhead caused by the proposed signaling mechanism will be in case the resource availability is done at the maximum adaptation speed of the multimedia codec used. Video encoders are slow in changing their encoding rates compared to channel variations due to limitations imposed by buffer management strategies of latest video encoders such as (VBV) model (Zhao & Kuo, 2003) of MPEG and the Hypothetical Reference Decoder (HRD) model (Ma et al., 2003) in AVC/H.264. Since we cannot adapt the video quality at a speed higher than the adaptation response time of the codec, resource availability signaling frequency must not be higher than the codec response time. Video adaptation algorithms need at least one group of pictures (GOP) - a frame sequence in a given structure - in order to converge to a target bit rate every time the video is adapted and at most 1 GOP per second is taken as a rule of thumb in order to achieve high compression efficiency. Such an approach would allow 1 adaptation per second in the worst case. Therefore, we assume that resource availability is done on a one update per second basis in the worst case, and the network overhead caused by this signaling is measured to be around 8 kbps, as shown in Fig. 8. The signaling overhead is zero before the session starts, and it increases to 8 kbps on average per session after the session is initiated. Note that this overhead of 8 kbps is very small compared to the average encoding bitrate of a decent quality video.

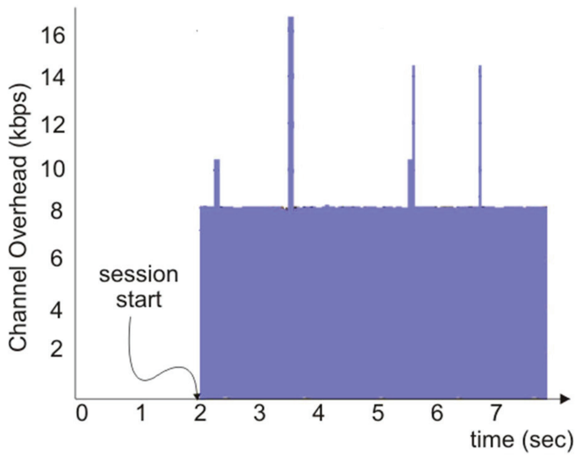

Fig. 8. Resource availability signaling channel overhead in the worst case scenario (signaling period: 1 sec.).

For an analysis of how many IMS sessions can be supported with the introduced signaling framework, it is time to look at some literature. In (Vingarzan et al., 2005), the design and implementation of an Open IMS core has been done and the load on the IMS core network and the proxies due to SIP message flow is investigated. In an IMS network with 100,000 subscribers, 1/3 of the subscribers are assumed to be online and registered simultaneously at a typical instant and 1/16 of these subscribers are assumed to be engaged in a multimedia session with an average duration of 180 seconds. In this case, the system would have to support 11,57 calls per second and the Open IMS core would have to process around 81 SIP messages per second (7 SIP messages for each multimedia call setup). In their experimental results, it was shown that a simple Intel Pentium 4 processor running at 3GHz (HyperThreaded) is enough to do the tasks of all IMS core components at once (i.e. I-CSCF, S-CSCF, P-CSCF and HSS) and still handle 120 SIP messages per second (around 17 calls per second). Considering the above data, in a worst case scenario of the proposed architecture, i.e. when each and every one of the active users has to adapt their multimedia within a given second, around 2000 SIP messaging events would need to be handled by the IMS core. This is quite realizable in a real-life deployment of the IMS core network since i) all components (CSCF's and HSS's) of the system normally reside on different hardware nodes in a deployed IMS core, ii) using multiple instances of the same component (e.g. multiple S-CSCF's) is very common for load-balancing, and iii) the state-of-the-art processors of today (e.g. multi-core processors) are much more powerful than a 3GHz Intel Pentium 4 processor. Clearly, in a more realistic case, the resource availability signaling overhead decreases even further when the resource signaling is done based on critical thresholds as described in the previous section.

The RAS is an additional server unit that can be implemented as an AS and it is independent of the IMS CSCF. Therefore, the existence of RAS does not put any computational overhead on the CSCF's.

10. Experimental results

In our experiments, we have separately demonstrated signaling from the multimedia transport. The reason for this is that running multimedia on the PCs rather than on the PDAs offered more freedom for experimenting with the adaptation of the perceived QoS.

10.1 Signaling scheme

For demonstrating our signaling scheme we have implemented the RAS server in a PC, whereas the RM software components were installed in the 2 PDAs running the WinCE operating system. Between the two PDA's we have established a multimedia communication session (see Fig. 5).

To start the session, the PDA applications use the SIP INVITE message. The multimedia flow is started after the ACK is received from the caller. The multimedia communication session ends up with a SIP BYE message which terminates the media session. Resource availability data from each PDA is transported to the RAS module using our SIP NOTIFY messages with the new header fields introduced in Table 1.

An example snapshot of the RAS interface is shown in Fig. 9. Here the real-time resource availability data of both end-devices is shown on a display, quantifying a remaining battery power (BT in %), storage space (ST in MBytes) and the available dynamic memory size (MM in MBytes).

Fig. 10 shows a snapshot of the User Interface (UI) of the client test application in the PDA. The top left menu is used to make a call or to exit the application. The first line shows the local IP address and the port number. The local and the remote resource availability data are displayed in the second and the last line, respectively.

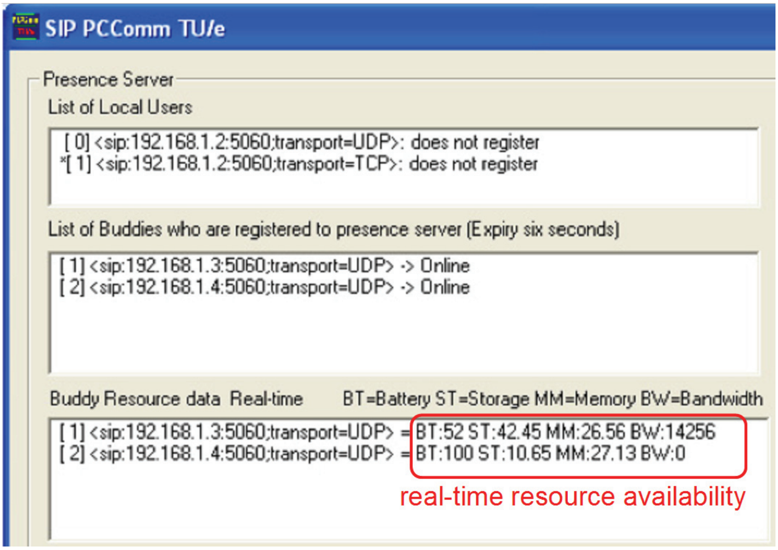

Fig. 9. A snapshot of the RAS log interface.

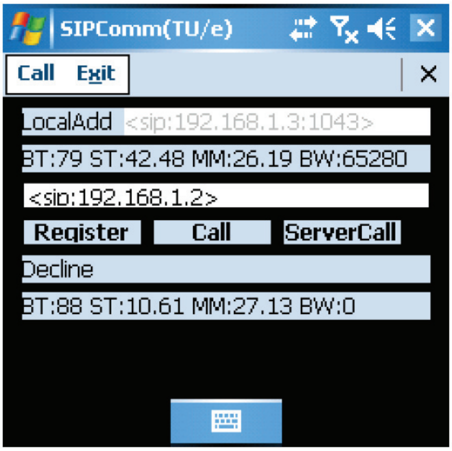

Fig. 10. A snapshot of a user interface of the PDA.

10.2 Multimedia adaptation

To demonstrate the effect of the multimedia content quality adaptation during a single session in order to increase the perceived QoS, we have installed the source of the multimedia stream in a PC and the sink in a laptop, both running the Windows XP operating system (see Fig. 11). To emulate the effects of bandwidth variation during a single multimedia session we have introduced a Linux Router on which Linux Advanced Routing and Traffic Control is implemented. Finally, we have compared the perceived QoS of the received multimedia with and without content quality adaptation during a single session. The multimedia session between the PCs is carried over Real-time Transport Protocol (RTP).

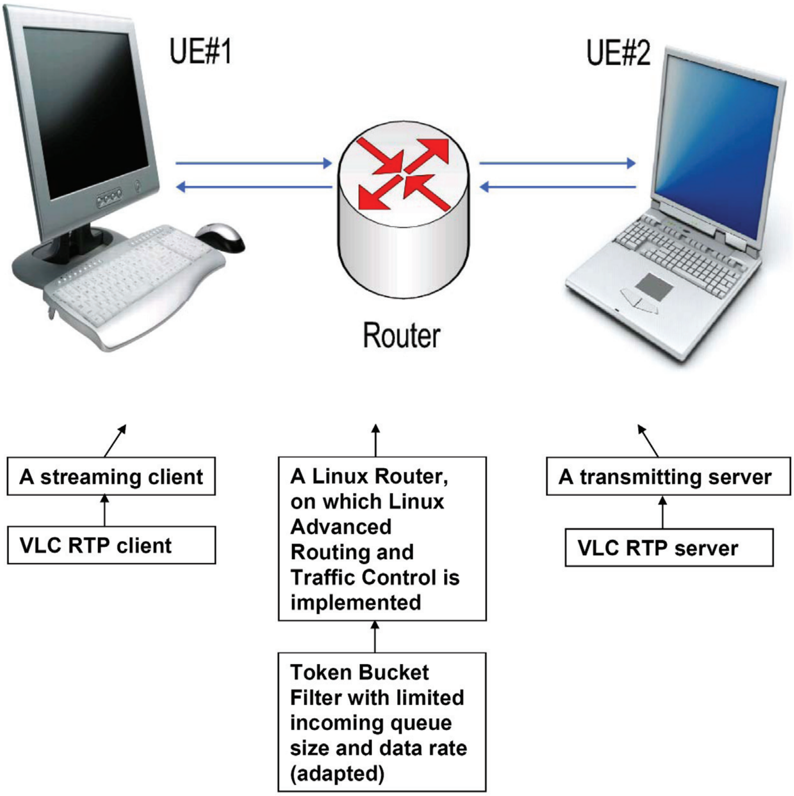

Fig. 11. Traffic controlled video streaming system setup.

Based on the signaling information, quality of the multimedia stream content has to be adapted to maximize the perceived QoS. However, such adaptation cannot be done arbitrarily, since multimedia codecs have their own limitations in changing the encoding bit rate on-the-fly even in the case of scalable codecs or the bit stream switching. Therefore, in multimedia content quality adaptation, adaptation speed should not be higher than that of the multimedia codec. In (DeVito et al., 2006), it is argued that video adaptation algorithms in the literature need up to 3 groups of pictures (GOP) in order to converge to a target bit rate every time the video is adapted. Here a GOP is defined as a frame sequence of a given structure in a video stream, whose first frame is an intra-coded (I) frame. Furthermore, it is also denoted in (DeVito et al., 2006) that the size of a GOP has to be kept large in an encoded video bit stream in order to attain reasonable compression efficiency and 1 GOP per second is taken as a rule of thumb, which would allow 1 adaptation in every 3 seconds for the other rate controllers in the literature and 1 adaptation per second for the advanced rate controller of (DeVito et al., 2006). Therefore, we assume that the maximum video adaptation frequency is 1 adaptation per second for typical videos. Updating resource availability at the speed higher than the adaptation speed would result in *no* improvement of the perceived QoS. Moreover, resource consumption in the network and end-devices will be higher.

We controlled the instantaneous maximum throughput from *UE*2 (a laptop computer) to *UE*1 (a desktop computer) by means of a Linux router in the middle as shown in Fig. 11, on which a Linux Advanced Routing and Traffic Control (LARTC, 2009) script is run to employ a Token Bucket Filter (Perez & Valenzula, 2005) as the traffic shaper (bandwidth limiter).
For our experiments, we encode the video at different target bitrates at different time intervals according to the channel throughput determined by the router. Let the maximum data throughput through the router be *R* kbps in a given time interval. We encode the corresponding video segment at a lower target bitrate than *R* considering a typical 25% − 30% channel overhead due to Ethernet/IP/UDP/RTP headers involved.

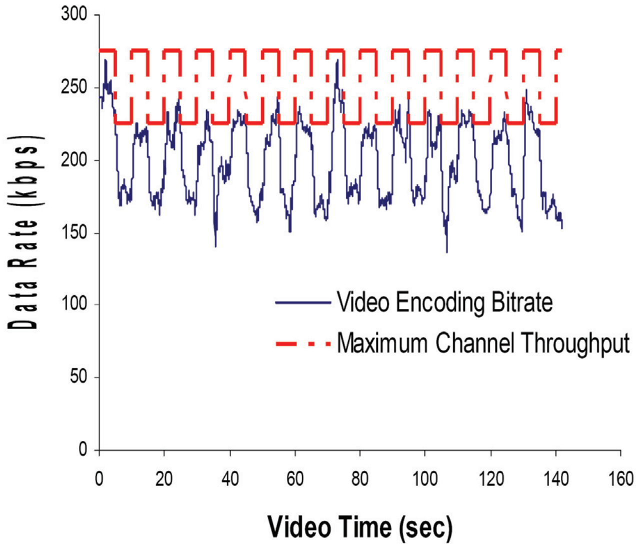

Fig. 12. Maximum channel throughput vs proposed adaptive video encoding bitrate graph (an example case)

Having a limited incoming queue size, the router drops any UDP packet that overflows its incoming queue. The multimedia sender is not aware of the bandwidth variation causing video packet losses. If there are no retransmissions at the sender, perceived video quality loss will be considerable unless the video encoding rate is kept to the reasonable values. In Fig. 12, the encoding bitrate for a video of 142 seconds length can be seen as it changes over time together with the network resource availability (i.e. maximum channel throughput). As the maximum channel throughput alternates between 275 kbps and 225 kbps in 5 second intervals, the video encoding rate is changed accordingly (alternated between 220 kbps and 180 kbps) in the proposed scheme, achieving an average encoding rate of 200 kbps.
The traffic control script applied for this purpose is given in Table 2 where the uplink throughput limit for the outgoing eth0 interface of the LINUX router is alternated between 225 kbps and and 275 kbps as explained above. The parameter limits and burst denote the maximum number of bytes that can be queued waiting for tokens (the queue size) and the bucket size (maximum possible number of tokens instantaneously available).

```
tc qdisc add dev eth0 root tbf rate 275kbit limit 150kb burst 10kb
sleep 5
tc qdisc change dev eth0 root tbf rate 225kbit limit 150kb burst 10kb
sleep 5
tc qdisc change dev eth0 root tbf rate 275kbit limit 150kb burst 10kb
...
tc qdisc change dev eth0 root tbf rate 225kbit limit 150kb burst 10kb
sleep 5
tc qdisc del dev eth0 root tbf rate 225kbit limit 150kb burst 10kb
```

Table 3. Linux Advanced Routing and Traffic Control script applied for the adaptive streaming experiment

The comparison of an example frame from the original video, the proposed solution and the nonadaptive solution can be seen in Fig. 13-15. When the perceived QoS of the standard nonadaptive video (average 200 kbps) solution with the proposed adaptive video (average 200 kbps) solution, the perceived QoS is enhanced in the proposed scheme as shown in Fig. 13-15 and the continuity of the video playback at the client side is satisfied only in the adaptive encoding case.

11. Summary

Thanks to a rapid increase in the number of both fixed and mobile Internet users, networked multimedia services have become an important part of our daily lives over the last decade. The availability of broadband connectivity in both homes and offices has allowed users to access multimedia content without bandwidth concerns and with a high degree of perceived quality. Furthermore, the wide availability of powerful and relatively inexpensive end user equipment through which high quality multimedia services can be enjoyed (such as personal computers) has considerably diminished concerns relating to device capability and resource availability.

Fig. 13. Original uncompressed video

Fig. 14. Constant bitrate coding and transmission (avg: 200 kbps)

Fig. 15. Proposed adaptive coding and transmission (avg: 200 kbps)

The recent introduction of "fast" telecom data services (e.g 3G) made it possible to introduce the experience of streaming multimedia content into the telecom community. This is however not straightforward from a technical perspective and a number of challenges need to be addressed. In contradistinction to the fixed broadband situation, bandwidth in mobile telecoms networks is expensive. This makes it difficult in such networks to utilize overprovisioning methods in support of multimedia streaming services. In addition, resources in mobile user equipment (battery, CPU etc) are often quite limited. It is therefore difficult to guarantee adequate resource availability throughout a single multimedia session. Despite this, telecom users are now able to stream multimedia content to their equipment depending on the real-time availability of their network and end device resources. The introduction of IMS networks has made it possible for users residing in unmanaged access networks (such as WiFi, xDSL) to access these services. However, it is not possible for the operator to apply network-level QoS mechanisms (such as admission control) on such unmanaged access networks. Perceived QoS guarantees become infeasible in this case. This chapter presents a framework for monitoring and signaling resource availability and for QoS adaption in support of devices, used over heterogeneous, IMS-based networks, which have variable resource availability and which lack QoS support. This enables the user experience of accessing real-time multimedia services on such devices and across such networks to be enhanced. An extended IMS architecture with additional system components is proposed in the context of the framework presented. Network and end device resource availability during a single session is monitored and communicated in real time and the encoding bitrate is adapted accordingly.

The system components proposed in this context are the Resource Manager, the Resource Availability Server and the Service Quality Manager. The proposed enhanced systems architecture is IMS compliant. The real time resource availability data is conveyed by means of the SIP protocol. Experimental results have shown that the proposed method can substantially improve the perceived quality of real-time streaming over IMS networks. The same approach can be applied to other real-time services having similar requirements, e.g. online gaming.

12. References

3GPP. (2006) Digital Cellular Telecommunications System (Phase 2+), Universal Mobile Telecommunications System (UMTS), IP Multimedia Subsystem (IMS), *Stage 2, V7.6.0, TS 23.228, 3GPP*, December 2006.

Andreasen, F. (2002). Session Description Protocol (SDP) Simple Capability Declaration, RFC 3407, October 2002.

Aurrecoechea, C. ; Cambell, A. & Hauw, L. (1996). A survey of QoS architectures, *Proceedings of the 4th IFIP International Conference on Quality of Service*, Paris, France, March 1996.

Camarillo, G. ; Marshall, W. & Rosenberg, J. (2002). "Integration of Resource Management and Session Initiation Protocol (SIP)", *RFC 3312*, October 2002.

Casner, S. (2003). Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth, *RFC 3556*, July 2003.

De Vito, F. ; Ozcelebi, T. ; Sunay, O. ; Tekalp, M. ; Civanlar, R. & De Martin, J. C. (2006). Per-GOP Bitrate Adaptation for H.264 Compressed Video Sequences, *in L. Atzori et al. (Eds.): LNCS 3893*, pp. 198-206, Springer-Verlag Berlin Heidelberg 2006.

ISO/IEC. (2000). ISO/IEC 13818: 'Generic coding of moving pictures and associated audio (MPEG-2)', 2000.

ITU. (2009). http://www.itu.int/ITU-T/ngn/index.phtml, June 2009.

LARTC. (2009). http://lartc.org/, June 2009.

Ma, S. ; Gao, W. ; Wu, F. & Lu, Y. (2003). Rate control for JVT video coding scheme with HRD considerations, *Proceedings of the IEEE International Conference on Image Processing (ICIP),* vol. 3, pp. III_793-III_796, September 2003.

Nomura, Y. ; Walsh, R. ; Luoma, J-P. ; Asaeda, H. & Schulzrinne, H. (2006). A Framework for the Usage of Internet Media Guides (IMGs), *RFC 4435*, April 2006.

Perez, D. & Valenzuela, J.L. (2005). Performance of a token bucket traffic shaper on a real IEEE 802.11 test-bed, *Proceedings of the 16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol.3, pp.1931-1935, 11-14 September 2005.

Rosenberg, J. ; Camarillo, G. ; Johnston, A. ; Peterson, J. ; Sparks, R. ; Handley, M. & Schooler, E. (2002). "SIP: Session Initiation Protocol," *RFC 3261*, June 2002.

Vingarzan, D. ; Weik, P. & Magedanz, T. (2005). Design and Implementation of an Open IMS Core, *in T. Magedanz et al. (Eds.) LNCS 3744*, pp. 284-293, Springer-Verlag Berlin Heidelberg 2005.

Westerlund, M. (2004). A Transport Independent Bandwidth Modifier for the Session Description Protocol (SDP), *RFC 3890*, September 2004.

Wiegand, T.; Sullivan, G. & Luthra, A. (2003). Draft ITU-T recommendation and final draft international standard of joint video specification, *ITU-T Rec.H.264|ISO/IEC 14496-10 AVC*, May 27, 2003.

Zhao, L. & Kuo, C.-C. J. (2003). Buffer-constrained R-D optimized rate control for video coding, *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, Hong Kong, pp. III_89 -III_92, April 6-10, 2003.