Adaptive Media Streaming in Heterogeneous Wireless Networks

Andreas Schorr*, Andreas Kassler† and Goran Petrovic*

*University of Ulm, 89069 Ulm, Germany. Email: schorr@informatik.uni-ulm.de
†Nanyang Technological University, SCE, Singapore 639798, Email: kassler@ieee.org

Abstract—In this paper we present a system for IP based networks which adapts media streams to the respective characteristics of different (wireless) access networks. Inside the mobile node, an Interface Selection Subsystem monitors characteristics of network interfaces and triggers vertical handovers based on policies supplied by the operator. A Service Subsystem validates media stream adaptation to match network resource availability and provider constraints. A Media Subsystem on mobile and correspondent node transmits the media streams and applies appropriate error correction mechanisms (like FEC, ARQ) depending on network characteristics. We evaluate vertical handover behaviour of our system in a combined GPRS - WLAN testbed and analyze the impact of several error correction mechanisms on the perceived video quality.

I. INTRODUCTION

In the near future many communication devices will offer the possibility to connect to several access networks like GPRS, UMTS, or WLAN. Although there is a clear trend towards all-over-IP networks, the characteristics of these networks (like coverage, bandwidth and QoS) are quite different. As the user always would like to be connected to the best access network at any given time for his service, this calls for adaptive applications or middleware that tailor the media streams according to the characteristics of the selected network. The decision 'when to switch to what access technology' depends not only on mobile terminal specific parameters (like user preferences, link bandwidth, price), but also on provider restrictions and user contracts. Therefore, a cooperative model between the terminal and the network is required in which the network provider supplies the mobile terminals with policy rules that guide this decision making process.

In [1], [2], we introduced a distributed multimedia system based on such a cooperative model. In [3], we demonstrated the system's ability to perform a seamless vertical handover (based on Mobile IPv4) triggered by policies during a running video session. Other researchers have studied the problem to find an optimal timing for handover trigger when the signal strength of the currently used network interface becomes weak [4], Nevertheless, there may be situations in which the signal strength drops so rapidly, that no action can be taken before the connection is totally lost. In this paper we show how our system recovers from such a suddenly lost connection. Similar architectures have been introduced by other researchers [5], [6], but these approaches do not address the problem that there may not be enough resources in the new network for maintaining the data rate of a running media session, which is very likely when switching between WLAN and GPRS.

There has been a substantial amount of prior work in the area of error protection (on application layer) for multimedia transmissions over lossy packet networks, using either Forward Error Correction (FEC) [7]-[9] or Automatic Repeat Request (ARQ) techniques [10], [11] or a combination of both [12]. Nevertheless, the approaches proposed are not equally suitable for all types of networks. In [11] the authors show similar measurements as we will present in this paper, depicting the influence of error correction mechanisms on the perceived video quality, but their work only considers packet losses caused by congestion, rather than losses caused by bit corruption, and they also consider only one specific mechanism. Our system differs from this work in that it allows to choose among different error correction techniques at runtime according to configurable policy rules and monitored network parameters (such as delay, jitter, packetloss). The rest of this paper is structured as follows. Section II gives an overview on our system. In section III we analyze the handover behaviour. Section IV evaluates the impact of several error correction mechanisms on the perceived video quality, and we conclude in section V.

II. ARCHITECTURE

Our system (Fig. 1) consists of Mobile Terminal Broker (MTB), Network Broker (NB), and VoD server. The VoD server provides video streaming services via the Realtime Transport Protocol (RTP) and content adaptation via stream switching to match terminal and network resource availability. The NB is operated by the network/service provider. It submits handover policies to the Mobile Terminals when they register at the NB. Furthermore, the NB performs access
control for all services requested by the Mobile Terminal by participating in the RTSP call setup process and evaluating the RTSP requests against the user contract. Inside the MTB, an Interface Selection Subsystem (ISS) is responsible for access network selection based on monitoring information and handover policies. A mobility management module in ISS enables the continuity of the session based on Mobile IPv4. The Service Subsystem of the MTB uses Realtime Streaming Protocol (RTSP) for session setup, adaptation and control. The Media Subsystem of the MTB recovers transmission errors, decodes, renders and displays the arriving video samples.

For the purpose of this paper, we consider only RTP based MPEG-4 video streaming using [13], although the implemented system is able to handle other encodings, too.

III. HANDOVER BEHAVIOUR

The ISS constantly monitors the signal strength of all available interfaces and immediately triggers a forced handover when it detects a signal loss at the currently active interface. If multiple air interfaces are available it selects the best one based on the evaluation of policy rules supplied by the NB (policy-based handover). We defined several rules like: WLAN is the preferred interface between 8:00am and 1:00pm, but after 1:00pm trigger a policy-based handover to the GPRS interface if available. As shown in [3] our system is able to perform an almost seamless policy-based vertical handover with only a small disruption time of 300-600 milliseconds and a frame loss of at most one single frame. In this paper, we now focus on our system’s behaviour during a forced handover.

Our testbed (Fig. 2) consists of three subnetworks and runs the Dynamics Mobile IP implementation [14]. The Home Network of the Mobile Terminal contains the Home Agent (HA), NB and VoD server. For the Mobile Node (MN), we used a laptop with a WLAN interface (Lucent Orinoco PCMCIA card) and a GPRS interface (by connecting a Siemens S45 mobile to the serial port). The MN can connect with Foreign Network 1 (FN1) via the WLAN interface or with Foreign Network 2 (FN2) via GPRS. In FN2 we used a complete GPRS test network (including BTS, BSC, GGSN and SGSN). No cross-traffic was generated during our test sequences.

The router connecting to FN2 runs a traffic control module with two separate packet queues. A classifier was used to differentiate between control traffic (Mobile IP, RTSP) and video traffic, which was directed to the low priority queue. We limited the bandwidth of the low priority queue to 30 kbps (the GPRS link in our testbed offered 32 kbps) and excess traffic was dropped. Without this traffic control module, forced vertical handover from WLAN to GPRS was never possible while high video traffic (more than 300 kbps) was streamed to the Mobile Terminal. The reason is the following. When the ISS of the Mobile Terminal initiates a handover, it registers its care-of address for FN2 with the HA. The final response message from the HA to the MN was always dropped/delayed; because after the HA created the new tunnel, the video traffic is flooding FN2. As we are in the forced handover case, we could not adapt the video before the WLAN link got lost. Since the datarate of the video traffic by far exceeds the available bandwidth of the GPRS link, the airline is congested before the MN can issue an adaptation request to the VoD server.

We evaluated performance for forced vertical handover from WLAN to GPRS during a running video session by pulling out the network cable which connected the WLAN Access Point with the FN1 router. Then we plugged the cable in again and the system switched back to WLAN according to the installed policy rules. Then we tested forced vertical handover from GPRS to WLAN by first switching to GPRS manually (installing a policy rule to prefer GPRS) and then disconnecting the mobile phone from the serial port of the laptop (which forces a handover). We repeated this test sequence 20 times. On the MN the ISS needed 4-6 seconds (5.16 seconds on the average) for the handover procedure from WLAN to GPRS.

This includes: detecting that the current WLAN connection is lost (Layer 2 measurement), checking whether the usage of GPRS is allowed (policy check), and establishing a new Mobile IP tunnel. For the handover procedure from GPRS to WLAN, the ISS needed 1-2 seconds (1.5 seconds on the average). We also measured the time between sending an RTSP adaptation message from the Service Subsystem of the MTB to the VoD server and the reception of the corresponding RTSP Acknowledge. This time varied between 1814 and 3274 milliseconds (average: 2473 milliseconds) when transmitting the RTSP request over GPRS, and it varied between 656 and 1050 milliseconds (average: 795 milliseconds) when transmitting the request over WLAN. The reason is in the different roundtrip times on the WLAN route and on the GPRS route (which includes whole GPRS infrastructure).

As the adaptation message from the MN to VoD can only be sent after the new Mobile IP tunnel is established, there is some interval during which the video traffic is sent through the new interface before the server adapts the rate. This is a problem when switching from GPRS to WLAN, because WLAN has enough bandwidth to accommodate with low bandwidth streams tailored for GPRS network. However, after a forced handover from WLAN to GPRS, the GPRS link is congested and thus the perceived disruption time for the user is higher than the time needed by the ISS to switch the interface. As the traffic control module ensures that some of the traffic which is flooding the GPRS link is discarded, it depends on the maximum length of the router queues, how long this disruption time lasts. If e.g. the data rate transmitted over the WLAN was 320 kbps and 1s of the data stream...
is stored in the router queue after the handover to GPRS, it will take 10s to transmit this data over a 32 kbps GPRS link. We illustrate this influence of the maximum length of the router queue in Fig. 3 and 4. Fig. 3 depicts the situation when we used a router queue with a maximum length of 10 IP packets. It depicts the delay in milliseconds (left y-axis) when displaying consecutive frames. As the framerate was adapted after the handover the x-axis is not proportional in time. We used a video clip encoded at 15 fps for streaming over WLAN, otherwise 3 fps. The diagram shows a disruption time of more than 10 seconds for the WLAN to GPRS case (at frame 250). However, almost no disruption is visible during policy-based handovers (at frame 350 and 2650) and during forced handover from GPRS to WLAN (at number 2700), because the client application had a playout buffer of max. 5 seconds. A similar situation is depicted in Fig. 4 with a maximum queue length of 5 packets. In this case the disruption time during the forced handover from WLAN to GPRS is shorter (less than 6 seconds). Such a short router queue would lead to increased packet drop if cross traffic would be used. We also performed several tests using longer router queues, but the maximum delay was always around 11 seconds. The figures also depict the number of lost frames (right y-axis), which is similar in both test sequences. We lost around 15 frames during the forced handover from GPRS to WLAN.

IV. ERROR CORRECTION

The optimal selection of an appropriate error correction mechanism for multimedia streaming depends on several factors: session type (interactive or on-demand), characteristics of the network (e.g., bandwidth, bit error rate, delay), and the current network load. At the MTB we monitor dynamic parameters (such as packetloss, delay, jitter) using RTCP. If a single error correction mechanism is not equally appropriate for all hops on the end-to-end path, a proxy node as proposed in [15] could be used to adapt the media stream to the specific requirements of the next network link. In the following analysis we concentrate on a single (wireless) network link in a video-on-demand scenario.

We have implemented three different mechanisms to cope with packet loss. The first one is an implementation of the generic forward error correction (FEC) for RTP streams defined in [7]. The second scheme uses unequal erasure protection (UXP) introduced in [9]. The third is based on RTP retransmission (ARQ) style, similar to [10]. Nevertheless, our system is designed in such a flexible way that other error correction mechanisms can be easily integrated. Since we wanted to analyse the behaviour of the mechanisms under different network conditions, we installed the NISTNet emulation software [16] on a Linux router connecting a sender and a receiver terminal. NISTNet allows to emulate bandwidth, packetloss, delay, and jitter on IP level. For the following test sequences we emulated a similar airlink as in our real GPRS testbed with 64kbps in the downlink and 9.6kbps in the uplink direction of the receiver, a one-way delay of 400 milliseconds, and a delay standard deviation of 50 milliseconds.

We streamed a compressed MPEG-4 video and varied the packetloss rate and the applied error correction technique. Because of the relatively low bandwidth of the GPRS link, we used a video with a framerate of 3fps. Finally, we compared the received video with the original uncompressed video, and calculated the average PSNR of the luminance component for the whole scene. We also calculated the average PSNR of the encoded video with the original video and thus compared the PSNR reduction due to packet loss. We used three different video clips (with low and high motion scenes) and averaged the PSNR reduction for all three videos. The result is depicted in Fig. 5 under varying packetloss rates. The line denoted as 'ARQ_1s.buffer' shows the result when applying the implemented ARQ mechanism and using a playout buffer at the receiver terminal of 1 second. The UXP mechanism allows to specify three parameters: The payload size per RTP packet, the redundancy added for each I-frame, and the
redundancy added for each P-frame (we didn’t use B-frames in this measurement). The line denoted as ‘UXP 50.45.25’ shows the result when applying UXP with RTP payload size 50 Bytes, 45% I-frame redundancy, 25% P-frame redundancy. In Fig. 6 we plot the percentage of the frames in the video clips, which could be decoded at the mobile terminal.

ARQ (with a buffer of 3s) or FEC seem to be the optimal choice at all error rates regarding visual quality, since no significant reduction of PSNR is visible. But of course, both mechanisms have some drawbacks. The FEC mechanism adds more than 100% redundancy to the data stream. When applying ARQ, the effective data rate is lower (see Table I). Nevertheless, the type of application or the current RTT may not allow a large playback buffer. As can be seen in Fig. 5 and Fig. 6, ARQ performs worse when using a playback buffer of only 1s. We can also see that with 45% redundancy for I-frames and 25% redundancy for P-frames the UXP mechanism produces good results. Finally, we can see that UXP performs slightly worse than FEC and ARQ at low packet loss rates, but a reduction of the PSNR of less than 0.4 might be acceptable for the user. In summary, UXP might be a good choice in cases where the current RTT doesn’t allow the use of ARQ and the available bandwidth doesn’t allow the use of FEC. But notice that the effective data rate for UXP depends not only on the amount of redundancy which is added to the payload of the RTP packets. One configurable parameter when applying the UXP mechanism is the size of the RTP packets. As shown in Table I, the IP-, UDP-, and RTP-header overhead is higher when using small RTP packet sizes. On the other hand, the measurements show that UXP performs better with lower packet sizes (and the same redundancy) in terms of visual quality and the number of decodable frames.

### TABLE I

<table>
<thead>
<tr>
<th>Effective Data Rate</th>
<th>0.01</th>
<th>0.05</th>
<th>0.1</th>
<th>0.15</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Error Correction</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>ARQ (3s buffer)</td>
<td>10%</td>
<td>11%</td>
<td>13%</td>
<td>14%</td>
<td>15%</td>
</tr>
<tr>
<td>UXP 50.45.20</td>
<td>20%</td>
<td>20%</td>
<td>20%</td>
<td>20%</td>
<td>20%</td>
</tr>
<tr>
<td>UXP 50.45.20</td>
<td>16%</td>
<td>16%</td>
<td>16%</td>
<td>16%</td>
<td>16%</td>
</tr>
<tr>
<td>UXP 50.45.25</td>
<td>21%</td>
<td>19%</td>
<td>21%</td>
<td>21%</td>
<td>21%</td>
</tr>
<tr>
<td>FEC</td>
<td>25%</td>
<td>25%</td>
<td>25%</td>
<td>25%</td>
<td>25%</td>
</tr>
</tbody>
</table>

### V. CONCLUSION

We have introduced a system which applies the "always best connected" paradigm based on policies supplied by the operator. We have demonstrated how the system recovers from forced handovers. Nevertheless, there remains a noticeable disruption time especially when switching from fast to slow networks (like WLAN to GPRS). An essential part of our system was the traffic control module to provide differentiated treatment to signalling and data transport in a Mobile IP based handover mechanism, if the current datarate before the handover exceeds the available bandwidth in the target network. In future implementations, the disruption time may be further reduced if the data packets are marked and hence can be dropped according to their importance (e.g. drop all ps-frames). Furthermore, we are currently enhancing our system to change the parameters and error correction mechanisms based on monitoring data to match network resource availability, network type and perceived QoS. To this extent we are also considering to switch the codec dynamically once the new network is selected.

### ACKNOWLEDGMENT

This work was partially funded by the DFG within the AKOM framework and by Siemens AG, Department of ICN Networks. The authors would like to thank their colleagues from Siemens AG who hosted the GPRS test network.

### REFERENCES