Chapter 1

UBIQUITOUS VIDEO STREAMING: A SYSTEM PERSPECTIVE

Mario Gerla, Ling-Jyh Chen, Tony Sun, and Guang Yang

Department of Computer Science, University of California at Los Angeles, Los Angeles, CA 90095, USA

{gerla,cclljj,tonysun,yangg}@cs.ucla.edu

Abstract Video streaming has become a popular form of transferring video over the Internet. With the emergence of mobile computing needs, a successful video streaming solution demands 1) uninterrupted services even with the presence of mobility and 2) adaptive video delivery according to current link properties. In this paper we study the need and evaluate the performance of adaptive video streaming in vertical handoff scenarios. We created a simple handoff environment with Universal Seamless Handoff Architecture (USHA), and used Video Transfer Protocol (VTP) to adapt video streaming rates according to the "Eligible Rate Estimates". Using testbed measurements experiments, we verify the importance of service adaptation, as well as show the improvement of user-perceived video quality, via adapting video streaming in the vertical handoffs.

Keywords: Adaptive Video Streaming, VTP, Seamless Handoff, Vertical Handoff

1. Introduction

As the demand, production and consumption of digitized multimedia has intensified in recent years, the latest application trends have created an increasing interest in providing practical multimedia streaming systems to meet the needs of mobile computing. In order to provide uninterrupted services and maximum user-perceived quality, a successful video streaming solution needs to adapt appropriately to mobile handoff scenarios.

Consider the scenario where a user is in the midst of monitoring her biological experiment through a multimedia streaming broadcast, on a

PDA device, via an 802.11b wireless connection in her office. Concurrently, she is informed of an urgent request for her immediate presence from her collaborators 20 miles away. Whereas she cannot afford to miss neither the streaming multimedia nor her meeting with her collaborators, an ideal ubiquitous computing solution would allow her to continue her current multimedia streaming session while in transit from her current location to her final destination. This involves leaving her office with her PDA (departing from an existing 802.11b high-capacity connection), take an express shuttle to her collaborator's location (during which time continuing her monitoring via a lower-capacity 1xRTT connection with the multimedia quality adapted to the changed capacity), and arrive at her collaborator's office (entering another 802.11b network and receiving multimedia of higher quality again). Although visions of such system have existed for some time [7] [20] however, an actual implemented system capable of handling the above scenario was not previous explored.

As previously identified by [7] [20], in order to provide a system that addresses quality of service in mobile computing environments, the following key issues need to be resolved: 1) seamless mobility across heterogeneous networks, 2) application adaptation to maximize the end user's perceived quality, and 3) adaptation to network dynamics such as wireless channel errors and congestion.

To accommodate mobile users switching between networks of different capacities, a seamless handoff technology, that preserves existing application sessions, is needed to tackle the first issue. Since mobile users may roam in an arbitrary pattern, an adaptive multimedia streaming technology, capable of maximizing the end user's perceived quality, is needed to address the second and third issues. Combining the criterion discussed above, a complete ubiquitous video streaming solution will undoubtedly incorporate both seamless handoff and adaptive multimedia streaming technologies.

For the purpose of this system, a simple seamless handoff environment is created with Universal Seamless Handoff Architecture (USHA)[6] to handle various handoff scenarios. An important feature of USHA is application session persistence. USHA can quickly adapt to user mobility while maintaining uninterrupted connectivity for established network sessions. Furthermore, USHA requires little modification to the current Internet Infrastructure, making it an attractive choice for a seamless handoff testbed. We will discuss USHA in more details in section 3.

A video streaming protocol, Video Transport Protocol (VTP) [2] is used for adaptive streaming applications. VTP adapts its sending rate, and thus quality, according to network conditions. Generally speaking, video streams encoded at higher rates have better quality over those encoded at lower rates, but they also demand more bandwidth. On the Internet where cross traffic is highly dynamic, bandwidth may not always be able to meet the demand. In such cases, the streaming server must lower its sending rate, or its packets would be heavily lost, severely impairing the quality perceived by the end user. On the other hand, the server should also raise its sending rate when bandwidth appears to be plentiful, and maximize the resource utilization and perceived quality. With the bandwidth estimation technique motivated by TCP Westwood (TCPW) [32], VTP satisfies all the above requisites. Details of VTP will be discussed in section 3.

In this work, we have implemented a fundamentally adaptive, end-to-end multimedia streaming system that allows a mobile user to receive uninterrupted service of best possible quality multimedia, while roaming among multiple heterogeneous wireless networks. Although the general concepts of providing adaptive services are not new, we aim to provide insights on end-to-end dynamics of such system from an implementation perspective instead of a simulated one. Actual system measurements collected from our testbed show that the combination of USHA and VTP can indeed provide substantial improvements to streaming performance, in terms of perceived video quality (smooth video frame rate), and robustness against sudden changes in link capacities.

The rest of this paper is organized as follows. Section 2 presents some background and discusses related work in the area of seamless handoff and video streaming. Section 3 describes the novel system unification of our seamless handoff architecture (USHA) and VTP. Section 4 presents actual end-to-end measurement results of the system from our Linux testbed. Section 5 concludes the paper.

2. Background and Related Work

Seamless Handoff

Handoff occurs when the user switches between different network access points. Handoff techniques have been well studied and deployed in the domain of cellular system and are gaining a great deal of momentum in the wireless computer networks, as IP-based wireless networking increases in popularity.

Differing in the number of network interfaces involved during the process, handoff can be characterized into either vertical or horizontal [30], as depicted in Figure 1.1. A vertical handoff involves two different network interfaces, which usually represent different technologies. For example, when a mobile device moves out of an 802.11b network and into a 1xRTT network, the handoff event would be considered as vertical. A horizontal handoff occurs between two network access points that use the same technology and interface. For example, when a mobile device moves between 802.11b network domains, the handoff event would be considered as horizontal since the connection is disrupted solely by the change of 802.11b domain but not of the wireless technology.

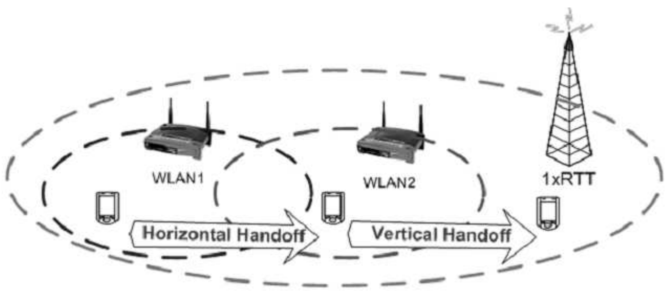

Figure 1.1. Horizontal and Vertical Handoff

A seamless handoff is defined as a handoff scheme that maintains the connectivity of all applications on the mobile device when the handoff occurs. Seamless handoffs aim to provide continuous end-to-end data service in the face of any link outages or handoff events. Low latencies and few packet losses are the two critical design goals. Low latencies require that path switches be completed almost instantaneously; service interruptions should be minimized. In case of an actual connection failure, the architecture should attempt to reconnect as soon as the service becomes available; packet losses due to the switch should also be minimized.

Various seamless handoff techniques [9] [17] [19] [22] have been proposed. These proposals can be classified into two categories: network layer approaches and upper layer approaches. Network layer approaches are typically based on IPv6 [8] or Mobile IPv4 [21] standards, requiring the deployment of several agents on the Internet for relaying and/or redirecting the data to the moving host (MH). Most upper layer approaches implement a session layer above the transport layer to make connection changes at underlying layers transparent to the application layer [12] [15] [23] [27] [28]. Other upper layer approaches suggest new transport layer protocols such as SCTP [29] and TCP-MH [24] to provide the necessary handoff support.

Previous seamless handoff solutions, whether mobile IP based or mobile IP-less, are often elaborate to implement and to operate. For the network layer solutions, deployment translates to upgrading every existing router without mobile IP capabilities. The cost imposed by these solutions is an existing barrier to wide deployment. For the upper layer solutions, a new session layer or transport protocol calls for an update to all existing applications and servers not supporting it, the potential cost is also discouraging. Consequently, even though many handoff solutions have managed to minimize both latency and packet loss, they are often not deployed in reality by the majority of service providers. With the proliferation of mobile applications and mobile users, a "simple" and "practical" seamless handoff solution with minimal changes to the current Internet infrastructure remains necessary.

USHA, an upper layer solution providing simple and practical handoff solution, is deployed in our experiments to handle seamless vertical handoffs. Details of USHA will be presented in section 3.

Video Streaming

Multimedia streaming, in particular video, has been growing as an important application on the Internet. However, the Internet is inherently not appropriate for such applications. Unlike conventional data transfers such as FTP, for which the Internet was designed decades ago, streaming usually has more strict QoS constraints on delay, bandwidth, etc. The best-effort Internet architecture lacks built-in schemes to guarantee these constraints. Thus enormous efforts have been put into research on streaming over IP networks.

On the video compression side, popular standard algorithms such as MPEG-4 [26] and H.263 [18] produce encoded streams in a wide range of rates. On the networking side, the key issue is to estimate an eligible rate at which the server should send in order to maximize the utilization of network bandwidth while effectively sharing it with other flows. There are two classes of techniques to estimate eligible rates: with network feedback and end-to-end. On the Internet, due to various scalability and deployment issues, end-to-end techniques seem more practical.

Several solutions based on TCP congestion control have been proposed for the transport of video over the Internet. For instance, SCP [5], a TCP Vegas [4] -like rate adjustment method, suffers from the same problems as TCP Vegas, thus remains inherently unfriendly to other TCP flows in many scenarios. RAP [31], a protocol employing AIMD to adapt the sending rate as TCP, does not take retransmission timeout into account, and therefore may result in poor performance when the impact of timeout is significant. TFRC [10] is a popular equation based solution built upon the model of TCP Reno, and aims to provide good smoothness and

TCP friendliness. However, efficiency of TFRC is susceptible to random losses at wireless links, a legacy problem from TCP Reno.

Additionally, TCP based streaming approaches also suffer delayed reactions to network dynamics in mobile scenarios (e.g. the maximum increase in TFRC's sending rate is estimated to be 0.14 packet/RTT and 0.22 packets/RTT with history discounting [10]). Consider a scenario where a video client handoffs from a low capacity link to a high one. A TCP based approach would use the "congestion avoidance" technique to linearly (and slowly) probe the available bandwidth on the new link. Such a slow reaction to network dynamics is unsatisfactory and can easily impair the overall experience of the client. As a result, a fast adaptive streaming technique is clearly a requisite for mobile needs.

Some of the commercial products claim to support adaptive video streaming, e.g. Helix Universal Server [16] and Microsoft Media Server [25]. However, lack of product disclosure and related analysis hinders independent efforts to verify the claims or to evaluate the streaming performance.

On the research side, some ongoing projects utilize packet pair/train measurements to estimate the end-to-end capacity and/or available bandwidth (or residual capacity), and adapt the sending rate accordingly. For example, inspired by TCP Westwood and its Eligible Rate Estimate (ERE) concept, SMCC [1] and VTP [2] are capable of adapting the sending rate to existing path conditions and resulting in both efficiency, i.e. high utilization of the bottleneck link, and friendliness to legacy flows. This enables faster responses in mobile handoff scenarios as well as achieving TCP friendliness. As a result of VTP's capabilities, it is used in this paper to evaluate the benefits of video adaptation in handoff scenarios. The VTP overview will be presented in section 3, and the experiments will be presented in section 4.

3. Proposed Approach

Universal Seamless Handoff Architecture

Universal Seamless Handoff Architecture (USHA), is a simple handoff technique proposed in [6] to deal with both horizontal and vertical handoff scenarios with minimum changes to current Internet infrastructure (i.e., USHA only requires deployment of handoff servers on the Internet.) USHA is a mobile IP-less solution; however, instead of introducing a new session layer or a new transport protocol, it achieves seamless handoff by following the middleware design philosophy [11], integrating the middleware with existing Internet services and applications. The simplicity of USHA makes it an attractive choice for a seamless handoff test bed.

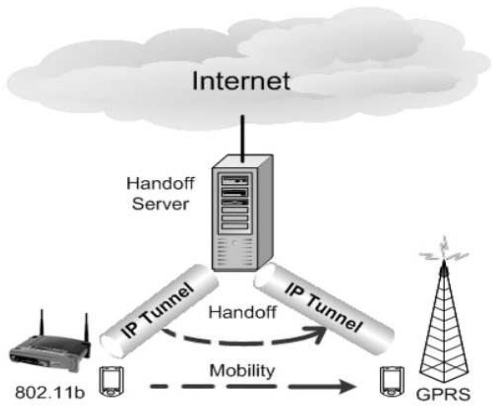

Figure 1.2. Universal Seamless Handoff Architecture

USHA is based on the fundamental assumption that handoff, either vertical or horizontal, only occurs on overlaid networks with multiple Internet access methods (e.g. soft handoff), which translates to zero waiting time in bringing up the target network interface when the handoff event occurs. If coverage from different access methods fails to overlap (e.g. hard handoff), it is possible for USHA to lose connectivity to the upper layer applications.

In Figure 1.2, a handoff server (HS) and several mobile hosts (MHs) are shown. USHA is implemented using IP tunneling techniques (IP encapsulation), with the handoff server functioning as one end of the tunnel and the mobile host as the other. An IP tunnel is maintained between every MH and the HS such that all application layer communications are "bound" to the tunnel interface instead of any actual physical interfaces. All data packets communicated through this IP tunnel are encapsulated and transmitted using the connectionless UDP protocol.

The IP tunnel above utilizes two pairs of virtual/fixed IP addresses, one on HS and one on MH. The fixed IP addresses are necessary for an MH to establish a physical connection to the HS. When the handoff event occurs and the physical connection from MH to HS changes, the MH is responsible for automatically switching the underlying physical connection of the virtual tunnel to the new interface, as well as notifying the HS of its change in physical connection. Upon handoff notification, the HS immediately updates its IP tunnel settings so that any subsequent data packets will be delivered to MH's new physical link.

Since all data packets are encapsulated and transmitted using UDP, there is no need to reset the tunnel after the handoff. Therefore, end-to-end application sessions (e.g. TCP) that are bound to the IP tunnel are kept intact. This provides handoff transparency to upper layer applications.

A simple USHA testbed is implemented. Experiments and evaluation of adaptive video streaming in vertical handoff scenarios on this testbed will be discussed in section 4.

VTP

Bandwidth Estimation. VTP is a video streaming protocol aiming to adapt its rate and quality according to network conditions. The core of VTP is its bandwidth estimation technique. It estimates the Eligible Rate Estimate (ERE) by applying an Exponentially Weighted Moving Average (EWMA) to the achieved rate, which is in turn calculated as the number of bytes delivered to the client during a certain time interval, divided by the length of the interval. Assume we use packet trains of length k to measure the achieved rate. Denote d_i as the number of bytes in packet i, t_i as the time when packet i arrives at the client. The sample of achieved rate measured when packet j is received, denoted as b_j, is

$$b_j = \frac{\sum_{l=0}^{k-1} d_{j-l}}{t_j - t_{j-(k-1)}} \tag{1.1}$$

The EWMA is needed to smooth achieved rate samples and eliminate random noise. Denote B_i as the available bandwidth estimate after getting sample b_j, then

$$B_j = \alpha \cdot B_{j-1} - (1-\alpha)(\frac{b_j + b_{j-1}}{2}) \tag{1.2}$$

The reason of using both b_j and b_{j-1} is to further reduce the impact of randomness in the achieved rate samples.

Rate Adaptation. Current VTP implementation works with pre-stored streams but can be extended to live video. Multiple streams of the same content are encoded *discretely* at different rates. Compression algorithms such as MPEG-4 can adjust parameters, such as the Quantization Parameter (QP), to achieve different encoding rates. For example, a movie trailer may be encoded at 56Kbps, 150Kbps and 500Kbps, targeting users with different access capacities. VTP chooses from multiple encoding levels of the same content the best rate according to ERE. Figure 1.3 illustrates with a finite state machine how rate adaptation is performed in VTP. Three video encoding levels, namely Q0, Q1 and Q2 with ascending rates, are shown. IR0 through IR2 are the "increasing rate" states while DR is the "decreasing" rate state.

VTP starts from state Q0. Upon receiving an ACK from the client, VTP server compares its current sending rate with the recently updated bandwidth estimate B. If the sending rate is less than or equal to B, VTP regards it as an indication of good network condition and makes a transition to IR0, where VTP linearly increases its sending rate to probe the available bandwidth. The amount of rate increase is limited to 1 packet/RTT, same as in TCP. On exiting IR0, VTP may move to state Q1 when the rate is high enough to support the level 1 stream, i.e. quality upgrade; or return to Q0 otherwise. Thus Q0 only implies the server is sending the level 0 stream; it says nothing about the actual sending rate. This process repeats itself, with possible quality upgrades, until the bandwidth estimate drops below current sending rate.

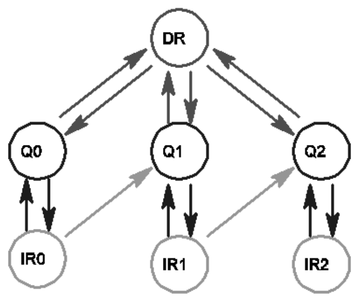

Figure 1.3. Rate adaptation in VTP

Rate decrease happens immediately when the measured bandwidth estimate drops below the sending rate. A transition from the current encoding level, say Q2, to DR is made. In DR, sending rate is decreased to the bandwidth estimate. If this rate is no longer able to support the current encoding level (level 2 in this example), one or more level decreases, i.e. quality downgrade, will occur until the level that the new sending rate can support. If the sending rate is below Q0, the lowest level, the streaming service will either be stopped or send at this lowest level, depending on administration policies.

Transmission Scheduling for VBR Video. Constant Bit Rate (CBR) video continuously adjusts QPs to maintain the target bit rate of the stream. This simplifies transmission scheduling, but results in varying video quality from frame to frame, which is unpleasant to the viewer. On the other hand, Variable Bit Rate (VBR) video produces streams with varying bit rates; and with more consistent quality.

Due to space limit we will not cover VTP transmission scheduling in detail in this paper. Briefly speaking, VTP divides a video clip into a number of segments. For each segment, VTP computes a target rate, at which neither buffer overrun or underrun should occur. Since video streams are pre-stored, instantaneous sending rates are available beforehand, and so are the target rates of the segments. VTP then applies these target rates to the finite state machine in Figure 1.3 for rate adaptation.

In the next section we will evaluate the performance of adaptive video streaming in seamless handoff scenarios of our integrated USHA + VTP testbed.

4. Experiments

In this section, we present measurement results of adaptive video streaming in vertical handoff scenarios using a 2-minute movie trailer encoded in MPEG-4 at three discrete levels. We denote them as levels 0, 1, and 2, corresponding to the encoding rates (VBR) of below 100, 100 $\sim$ 250, and above 250 Kbps, respectively. The VTP server is implemented on a stationary Linux desktop; the client is on a mobile Linux laptop. The USHA system is also set up in Linux, with custom configured NAT and IP tunneling. Both the VTP server and client are connected to the handoff server, the former via 100 Mbps Ethernet; the later via 802.11b and 1xRTT provided by Verizon Wireless. The 802.11b is set at the 11 Mbps mode; the bandwidth of 1xRTT varies with cross traffic, the typical value is around tens of Kbps.

We have tested two handoff scenarios, from 1xRTT to 802.11b (low capacity to high) and vice versa. In all experiments, one-time handoff occurs at 60 sec after the start of the experiment. In each scenario, we have tested both non-adaptive and adaptive video streams. In the non-adaptive case, video of fixed quality is sent throughout the experiment regardless of ERE, while in the adaptive case the video quality adapts accordingly.

Handoff from 1xRTT to 802.11b

In the first set of experiments, we evaluate the performance of video streaming when the mobile host performs handoff from the lower-capacity interface of 1xRTT to the higher-capacity interface of 802.11b.

Non-adaptive Video Streaming. First, we run "non-adaptive" experiments one for each encoding level. Since the coding rates of levels 1 and 2 are both above the capacity of 1xRTT, the corresponding experiments "died" shortly after started simply because of the inability of 1xRTT to handle such high rates. Results are not reported. Only video of level 0 made it through as the results show below. More specifically, Figure 1.4 shows the frame rate received by the mobile client, and Figure 1.5 shows the sending rate at the VTP server. In Figure 1.5, "Reference Rate" means the source rate of the video stream (note that the source rate is variable, even within a given encoding scheme), whereas the "Sending Rate" means the instantaneous transmission rate of the data, which depends on the link capacity and thus may exceed the source rate.

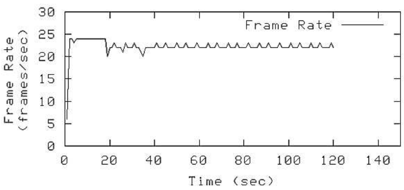

Figure 1.4. Frame Rate received at the Mobile Host

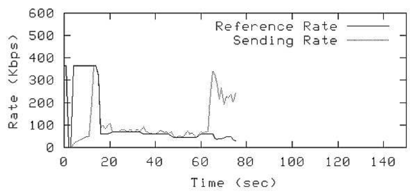

Figure 1.5. Sending Rate at the Video Server

In Figure 1.4, the video frame rate is stable and consistently between a visually pleasing range of 20 and 25 frames/sec (fps) shortly after it is started. Even in the presence of a handoff from LOW to HIGH at 60 sec, the frame rate remains unaffected. This proves our USHA to be transparent to applications. The video quality is overall very good in terms of smoothness. However, Figure 1.5 reveals more insightful information. In this non-adaptive experiment, the reference rate and video quality remain low after the handoff at 60 sec, where they could increase to take the advantage of the increased "sending rate" and bandwidth. This justifies the exploration of adaptation in video streaming applications. Note that after the handoff, the actual sending rate is much higher than the reference rate, so the server finishes sending quickly (before 80 sec).

Adaptive Video Streaming. The setup of adaptive streaming experiment is similar to the non-adaptive one described above except that now the video quality level adapts to the network conditions. In Figure 1.6, we show the frame rate received by the mobile client. Still it is stable and consistently in a range that gives good perceived quality. No dips in frame rate are found when the handoff event occurs.

Figure 1.7 shows the quality level of the video sent by the VTP server (averaged over 1-sec intervals). Level 2 is highest and 0 is lowest. Prior to the handoff at 60 sec, most frames are sent at the lowest quality level (i.e. 0); after handoff the average quality jumps to about 1.5. This is consistent with our experiment setup where the available bandwidth increases drastically when moving from 1xRTT to 802.11b.

Figure 1.8 shows the reference and sending rates on the VTP server. Prior to the handoff at 60 sec, Figure 8 looks very similar to Figure 1.5. The difference emerges after the handoff. The reference rate jumps up and strives to match the sending rate (300 Kbps), indicating that high quality video is now being transmitted across the 802.11b channel. In other words, VTP successfully detects (within fractions of a second) the change in available bandwidth and adapts its video encoding level to maximize the perceived quality of the mobile user.

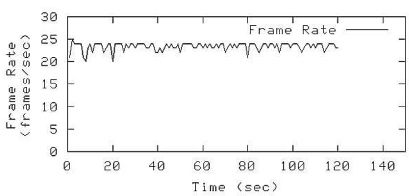

Figure 1.6. Frame Rate received at the Mobile Host

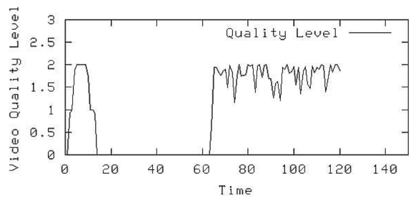

Figure 1.7. Video Quality sent at the Video Server

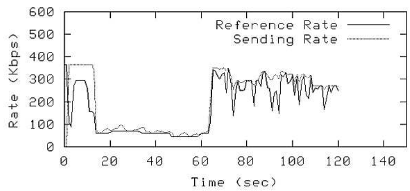

Figure 1.8. Sending Rate at the Video Server

Handoff from 802.11b to 1xRTT

In the second set of experiments, we evaluate the performance of video streaming when the mobile host performs handoff from the high-capacity interface of 802.11b to the low-capacity interface of 1xRTT. To make results comparable to the previous experiments, the one-time handoff is also generated 60 sec after the experiment is started.

Non-adaptive Video Streaming. Similar to the experiments that we have done in the case where handoff occurs from 1xRTT to 802.11b, we have also tested non-adaptive streaming at all three quality levels, respectively. Unlike the previous experiments, this time handoff occurs from the high-capacity interface to the low-capacity one, thus all three levels are feasible initially and can be tested. As expected, after the handoff, experiments with levels 1 and 2 virtually "died".

We show the experiment results with level 2, i.e. the highest quality in Figure 1.9 (video frame rate received by the mobile client) and Figure 1.10 (sending rate on the VTP server). Before the handoff, the frame rate received by the client is high and stable, and the reference and sending rates at the server are both high and close to each other, an obvious sign of high quality video. These metrics drop sharply at 60 sec when the handoff occurs, the reason being that 1xRTT is not able to handle the video of highest quality as we have explained. The frame rate drops to an unacceptable level of 10 fps; the sending rate becomes less than half of the reference rate. In the experiment we have found that the video virtually "froze" after the handoff. This experiment confirms the claim that adaptive multilevel video codes are a must in heterogeneous roaming.

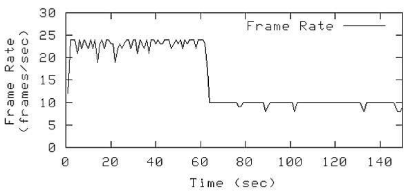

Figure 1.9. Frame Rate received at the Mobile Host

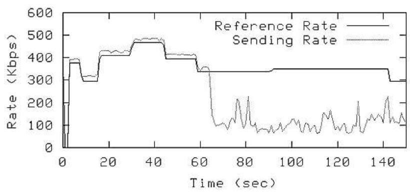

Figure 1.10. Sending Rate at the Video Server

Adaptive Video Streaming. Moving on to the adaptive video experiments, Figure 1.11 shows the video frame rate received at the mobile client - high and stable as we have seen in Figure 1.6. Note that there exists a small dip in the frame rate shortly after the handoff event at 60 sec, but the recovery is within a couple of seconds. This again proves the effectiveness of seamless handoff and rate adaptation of our proposed solution.

Figure 1.12 and Figure 1.13 show the video quality level and the reference and sending rates at the VTP server. Prior to the handoff at 60 sec when the system is running over the 802.11b connection, video quality is high (i.e. 2), so is the reference rate, matching the sending rate. Exactly at 60 sec the system is able to detect the handoff event and to adapt the video quality to the reduced bandwidth. Throughout the experiment the sending rate is always ahead of the reference rate so that there is no backlog build up at the sender.

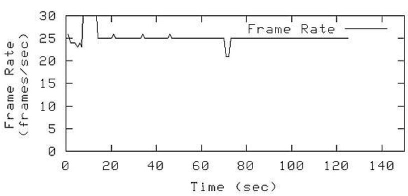

Figure 1.11. Frame Rate received at the Mobile Host

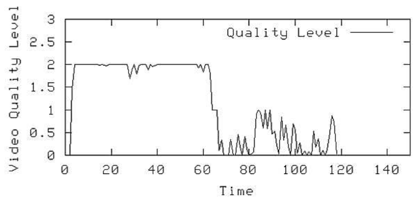

Figure 1.12. Video Quality sent at the Video Server

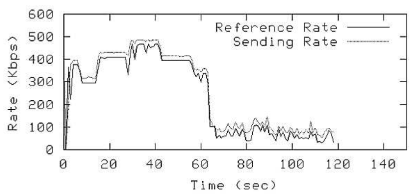

Figure 1.13. Sending Rate at the Video Server

Discussions

In a handoff-enabled environment, drastic changes in the link capacity are often associated with vertical handoff events. For instance, handoff from 1xRTT to 802.11b can easily witness a 100-fold increase in the link capacity (from 100 Kbps to 11 Mbps). Some traditional approaches (e.g. TFRC) incorporate the well-known *slowly-responsive congestion control* (SlowCC) [3] and thus can smoothly adjust the sending rate. However, SlowCC cannot take aggressive advantage of the rapid change of resources in emerging vertical handoff scenarios [13]. In order to utilize the bandwidth resources well, and maximize the user-perceived quality, a well-designed adaptive streaming scheme must take into account the effect of drastic capacity changes in both up and down directions.

From the experiment results presented in above, it is evident that VTP is one such scheme. Using the eligible rate estimate, VTP can properly and rapidly adapt its sending rate and video quality to available bandwidth, and hence is successful in handling vertical handoffs. This is not small feat. In fact, in most AIMD-based streaming protocols inspired to TCP, the adaptation process adjusts slowly to capacity changes. For example, when handoff occurs from LOW to HIGH (i.e. 1xRTT to 802.11b), no congestion loss is detected. A TCP based scheme will remain in congestion avoidance and linearly increase its congestion window (and rate) to probe the available bandwidth.

In the opposite direction, where handoff occurs from high (e.g. 802.11b) to low capacity (e.g. 1xRTT), there is immediate packet loss at the moment of the handoff, so AIMD protocols will react promptly to such loss. In fact, they tend to overreact causing oscillatory behavior and slower convergence to the new (lower) encoding rate.

In general, application performance would benefit if the server could predict the imminent handoff (e.g. MAC layer feedback from fading signals of one connection and strengthening signals of the other) and thus slow down its sending rate just before the handoff. We plan to address this issue in our future work.

5. Conclusion

In this work, we have studied the need and evaluated the performance of adaptive video streaming in vertical handoff scenarios. We have proposed an integrated solution of seamless handoff and adaptive video streaming, and implemented it on a Linux testbed, consisting of a USHA server and a VTP streaming system. Experiments on both non-adaptive and adaptive video applications, with handoffs from 1xRTT to 802.11b and vice versa, have been carried out to evaluate the performance of our proposed solution. From the measurements results we have seen that the USHA/VTP solution can effectively hide handoff events from the application and provide uninterrupted transport and application sessions during handoffs. Moreover, the adaptive streaming system is able to detect available bandwidth changes and adjust the video quality and sending rate accordingly. In summary, such a combination of adaptive video streaming and seamless vertical handoff will become very desirable in the emerging ubiquitous mobile computing environment.

References

[1] Aboobaker, N. et al. " Streaming Media Congestion Control using Bandwidth Estimation," in Proc. of IFIP/IEEE International Conference on Management of Multimedia Networks and Services, 2002.

[2] Balk, A. et al. "Adaptive Video Streaming: Pre-encoded MPEG-4 with Bandwidth Scaling,"" submitted for publication. www.cs.ucla.edu/NRL/hpi/tcpw/tcpw_papers/mpeg-cnj2003.pdf

[3] Bansal, D. et al. "Dynamic Behavior of Slowly-Responsive Congestion Control Algorithms," In Proc. of ACM SIGCOMM 2001.

[4] Brakmo, L., O'Malley, S., and Peterson, L. "TCP Vegas: New techniques for congestion detection and avoidance," in Proc. of the SIGCOMM '94 Symposium (Aug. 1994) pages 24-35.

[5] Cen, S., Pu, C., and Walpole, J. "Flow and Congestion Control for Internet Streaming Applications," Proc. of Multimedia Computing and Networking, 1998.

[6] Chen, L.J. et al. "Universal Seamless Handoff Architecture in Wireless Overlay Networks," Technical Report TR040012, UCLA CSD, 2004.

[7] Davies, N. et al. "Supporting adaptive video applications in mobile environments," IEEE Communications Magazine, 36 (6):138–143, June 1998.

[8] Deering, S., and Hinden, R. "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, Dec. 1998.

[9] Dommety, G. et al. "Fast Handovers for Mobile IPv6," draft-ietf-mobileip-fast-mipv6-04.txt, IETF Internet draft, Mar. 2002.

[10] Floyd, S. et al. "Equation-Based Congestion Control for Unicast Applications," In Proc. of ACM SIGCOMM 2000.

[11] Fox, A. et al. "Cluster-based Scalable Network Services," SOSP'97, 1997.

[12] Ghini, V. et al."Smart Download on the Go: A Wireless Internet Application for Music Distribution over Heterogeneous Networks," in Proc. of ICC, 2004.

[13] Gurtov, A., and Korhonen, J. "Measurement and Analysis of TCP-Friendly Rate Control for Vertical Handovers," submitted for publication, http://www.cs.helsinki.fi/u/gurtov/papers/vho.html

[14] Handley, M. et al. "TCP Friendly Rate Control (TFRC): Protocol Specification," RFC 3448, January 2003

[15] Handley, M. et al. "SIP: Session Initiation Protocol," RFC 2543, March 1999.

[16] Helix Universal Server, http://www.realnetworks.com

[17] Hsieh, R., Zhou, Z. G., and Seneviratne, A. "S-MIP: a seamless handoff architecture for mobile IP," in Proc. of IEEE INFOCOM 2003.

[18] ITU-T Recommendation H.263, Feb. 1998

[19] Johnson, D. B., Perkins, C., and Arkko, J. "Mobility Support in IPv6," draft-ietf-mobileip-ipv6-17.txt, IETF Internet draft, May 2002.

[20] Lu, S., Lee, K.-W., and Bharghavan, V. "Adaptive service in mobile computing environments," in Proc. of IWQOS 1997, New York.

[21] Perkins, C. Ed. "IP Mobility Support for IPv4," RFC 3344, Aug. 2002.

[22] K. El Malki et al, "Low Latency Handoffs in Mobile IPv4," draft-ietf-monileip-lowlatency-handoffs-v4-03.txt, IETF Internet draft, Nov. 2001.

[23] Maltz, D., and Bhagwat, P. "MSOCKS: An architecture for transport layer mobility," In Proc. of IEEE Infocom, p.p. 1037-1045, March 1998.

[24] Matsumoto, A. et al. "TCP Multi-Home Options," draft-arifumi-tcp-mh-00.txt, IETF Internet draft, Oct. 2003.

[25] Microsoft Media Server, http://www.microsoft.com/windows/windowsmedia

[26] MPEG-4 Industry Forum, http://www.m4if.org

[27] Schlaeger, M. et al. "Advocating a Remote Socket Architecture for Internet Access using Wireless LANs," Mobile Networks & Applications, vol. 6, no. 1, pp. 23-42, January 2001.

[28] Snnoeren, C. "A Session-Based Approach to Internet Mobility," PhD Thesis, Massachusetts Institute of Technology, December 2002.

[29] Stewart, R. et al. "Stream Control Transmission Protocol," RFC 2960, Oct. 2000.

[30] Stemm, M. and Katz, R. H. "Vertical Handoffs in Wireless Overlay Networks," ACM MONET, 1998.

[31] Rejaie, R., Handley, M., and Estrin, D. "RAP: End-to-end Rate Based Control for Real Time Streams in the Internet," In Proc. of IEEE INFOCOM'99, 1999

[32] Wang, R., Valla, M., Sanadidi, M. Y., and Gerla, M. "Adaptive Bandwidth Share Estimation in TCP Westwood," In Proc. of IEEE Globecom 2002.