STREAMING AGENT FOR WIRED NETWORK / WIRELESS LINK RATE-MISMATCH ENVIRONMENT

Gene Cheung, Wai-tian Tan *Takeshi Yoshimura*

Hewlett-Packard Laboratories NTT DoCoMo, Inc.

ABSTRACT

It has been shown that an agent located at the junction of wired and wireless links can help streaming media systems identify where packet losses occur and therefore maintain proper end-to-end congestion control. In this paper, we further expand the functionality of such agents in two ways. First, they allow streaming servers to identify the allowed transmission rate in both the wired and wireless parts of the path. Second, they serve as a relay to exploit the difference in transmission rate of the two parts. Simulation results show that PSNR improvement of over 2 dB can be achieved without extra bandwidth usage in the bottleneck links.

1. INTRODUCTION

Much work on streaming media systems has been on enhancing the intelligence of the endpoints only [1]. Nevertheless, for streaming systems involving both wired and wireless links, the effectiveness of endpoint intelligence alone is limited due to the lack of information to elect proper actions (Section 2.1). As a result, architectures involving agents located at the junction of the wired and wireless networks have been proposed to enhance the information available to the endpoints (Sections 2.2-2.3). Examples include (1) the *RTP Monitoring Agent* of Section 2.2, which provides additional information so that streaming servers can identify where packet losses occur and thereby perform proper congestion control, and (2) the *Streaming Agent version 1* of Section 2.3 which provide fine grained packet reception feedback so that streaming servers can perform optimization as to which packets to transmit or retransmit.

In this paper, we expand the capability of the agent in two ways. First, it provide additional information so that the allowed transmission rate at both the wired and wireless parts of the network can be determined. Second, instead of sending feedback information only, the agent actively act as a relay between the streaming server and client to exploit any additional available bandwidth to reduce overall perceived distortion. Specifically, in the typical case when the available wired bandwidth, R_1, is higher than the available wireless bandwidth, R_2, the agent can coordinate with the streaming server to use the excess wired bandwidth of $R_1 - R_2$ to reduce the effective packet loss rate of the wired links. One possible way to achieve such effective reduction of wired packet loss rate is through retransmission between the streaming server and the agent. On the other hand, when the available bandwidth in the wireless link is higher, we can similarly exploit the excess wireless bandwidth of $R_2 - R_1$ to reduce the effective wireless loss rate.

The rest of this paper is organized as follows. Related works in using agents at the junction of wired and wireless networks are given in Section 2. Limitations of the related works are summarized in Section 3. Our enhanced agent is then presented in Section 4. Simulation results, discussion and conclusion are finally given in Section 5, 6 and 7.

2. RELATED WORK

2.1. End-to-end Approach

Conventional practice for streaming media over last-hop wireless network is to ignore the effects of the last hop wireless link and employ endpoint media adaptations based solely on observable endpoint statistics. Since endpoint statistics are aggregated across all wired and wireless links, it is impossible to distinguish the respective conditions of the wired and wireless networks.

The problem with the conventional practice is that the server can confuse losses due to wireless link failure as losses due to wired network congestion given only endpoint statistics. If losses are due to wired network congestion, then the server should perform endpoint congestion control and reduce sending rate. Typical congestion control algorithms [2] decrease the sending rate with increasing round trip time and packet loss rates between two endpoints. On the other hand, if losses are due to wireless link failure, then server should increase the error resiliency of the transmitting media stream while keeping the same sending rate. By not being able to distinguish the two different types of losses, in the case when losses are caused by wireless link failure, the server confuses them as wired network congestion losses and will unnecessarily perform congestion control and reduce the sending rate. The end result is a decrease in performance.

2.2. RTP Monitoring Agent

The solution proposed in [3] to solve this problem is the RTP monitoring agent — a network agent, placed at the intersection of wired network and wireless link, monitors existing streaming flows and periodically sends statistical feedback in the form of RTCP reports back to the senders of the flows. As part of the RTP monitoring agent design, a *shaping point,* placed in front of the RTP monitoring agent, "adjusts the outgoing rate of all packet traffic to the rate of the radio link". Essentially, the shaping point does the following. Let R_1 be the permissible bandwidth of the wired network as determined by a standard wired network congestion control algorithm such as [2], based on wired network endpoint observable round trip time and packet loss rate. Let R_2 be the maximum sending rate permissible for the wireless link, as determined by the base-station during wireless link resource allocation phase of the connection setup. In the event that $R_1 < R_2$, the shaping point does nothing, and the streaming server sends at rate R_1 as a result

0-7803-7713-3/02/$17.00 © 2002 IEEE

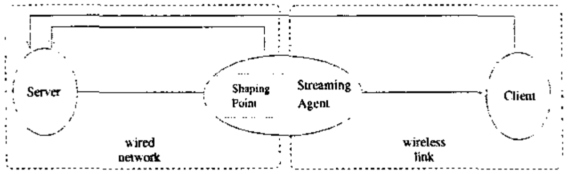

Fig. 1. RTP Monitoring Agent / Streaming Agent version 1

of the default behavior of the wired congestion control algorithm. In the event that $R_1 > R_2$, the shaping point drops packets before packet arrival at the RTP monitoring agent until the server reacts to the drops by reducing the sending rate to R_2 due to perceived wired network congestion control. See Figure 1 for an illustration.

2.3. Streaming Agent version 1 (SA1)

Because RTCP reports are sent in mid-term (on the order of of seconds) and do not contain information unique to individual packets, it is argued in [4] [5] that RTCP reports from RTP monitoring agent is neither timely nor specific enough for certain application-level optimizations such as format-adaptation [4] or application-level retransmission [5]. The solution is an enhancement to RTP monitoring agent so that *timely feedbacks* are sent in short-term (within a second), with each feedback packet containing information unique to the most recent K packets in the media stream. The enhanced agent is called *streaming agent* — denoted as streaming agent version 1 (SA1) to avoid confusion with this paper's proposal. Using these fine-grained feedbacks from SA1, the sender can now perform above mentioned application-level optimizations more efficiently than statistical feedbacks from RTP monitoring agent or timely feedbacks from client alone.

3. NETWORK UNDER-UTILIZATION OF SHAPING POINT

Looking closely at the resulting behavior stemming from the use of the shaping point in [3] [4] [5], we see that in the case when $R_1 < R_2$, the shaping point does not drop packets, and hence the feedbacks from the agent are reliable statistics of the wired network only. The information that the server can gather using feedbacks from agent and client are:

1. Permissible sending rate in the wired network R_1.
2. Wired loss condition due to congestion.
3. Wireless loss condition due to wireless link failure.

In the case when $R_1 > R_2$, the shaping point drops packets before packet arrival at the agent to reduce sending rate to R_2. After the sending rate has converged to the steady state R_2, the information that the server can gather in this case are:

1. Permissible sending rate in the wireless link R_2.
2. An upper bound of wired loss condition (the result of wired loss plus drops at the shaping point).
3. Wireless loss condition due to link failure.

Of course, at any given time the server does not know whether it is in the first or second case. In the absence of that knowledge, the server can only induce the following information:

1. $\min\{R_1, R_2\}$, which is the resulting sending rate.
2. An upper bound of the wired packet loss condition.
3. Packet loss in the wireless link only.

Focusing on the first item, we see that although the available network resources are R_1 for the wired part and R_2 for the wireless part, we are forced to send at the minimum of the two. This means available $R_1 - R_2$ bandwidth is left unused in the wired part when $R_1 > R_2$, and $R_2 - R_1$ bandwidth is left unused in the wireless part when $R_1 < R_2$. We identify this undesirable condition from the end application point of view as *network under-utilization of the shaping point*.

4. STREAMING AGENT VERSION 2 (SA2)

To alleviate the under-utilization problem, we propose a new version of the streaming agent (SA2) in this paper. Similar to previous work [3] [4] [5], we place SA2 at the intersection between the wired network and the wireless link. Let R_2 be the maximum permissible bandwidth of the wireless link, as determined by the wireless infrastructure during the wireless link resource allocation phase of the wireless session setup. R_2 is not likely to change during the course of the streaming session, unless the mobile client moves to a new coverage area that uses a new base-station [6]. The first task of SA2 is then to obtain R_2 from the wireless infrastructure and inform the streaming server before the start of the streaming session.

The second task of SA2 is to provide timely feedback, in the form of packet acknowledgment packets (ACKs), to the server so that it can determine the maximum permissible bandwidth of the wired network R_1 using a congestion control algorithm such as [2]. SA2's timely feedback together with the client feedback also allows the server to perform timely application-level optimizations such as format-switching [4] and rate-distortion optimized retransmission [5].

As we will show next, knowledge of both R_1 and R_2, instead of $\min\{R_1, R_2\}$ obtained using SA1, along with some assistance from SA2 such as adding error control codes to packets, enables the server to deploy error control mechanism that exploits the channel mis-match condition and maximally utilizes available resource. We consider two cases separately: when wired network bandwidth is smaller than wireless link bandwidth, $R_1 < R_2$, and otherwise.

4.1. Fat Pipe Thin Pipe: $R_1 > R_2$ Case

In the case when $R_1 > R_2$, server prepares a media stream at R_2 with the appropriate error resiliency for the wireless link loss process. During the streaming session, the server sends the media stream at rate R_2 plus packet retransmission that uses the surplus $R_1 - R_2$ in the wired network. At the edge of the wired network, SA monitors the RTP sequence number space of the media stream, sends ACKs back to the server to acknowledge packet arrivals, and drops duplicated packets to keep outgoing rate no larger than R_2. Using SA2's ACKs, the server can perform application-level retransmission scheme such as one in [5].

The reason retransmission is used instead of FEC in the wired network is twofold. First, delay is generally much smaller than that of the wireless link means retransmission is plausible even for small client buffer. Second, the performance of retransmission based schemes are less sensitive to burstiness in the packet loss process than FEC based schemes.

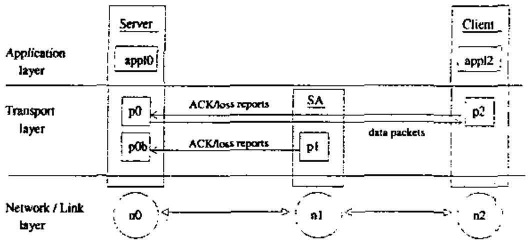

Fig. 2. *Simulation Setup.*

4.2. Thin Pipe Fat Pipe: $R_1 < R_2$ Case

In the case when $R_1 < R_2$, SA2 can use the surplus $R_2 - R_1$ bandwidth in the wireless link to correct losses in the wireless link. If the clients are unaware of the existence of SA2, the choices available to SA2 to correct losses are limited. One example is selective repetition where SA2 selectively chooses more important packets and transmits multiple copies. In contrast, for SA2-aware clients, the effectiveness of using the excess bandwidth to lower perceived loss rate can be significantly improved. One example is the use of forward error correction (FEC). FEC is preferred on the wireless link, since it is well known that the wireless link delay for today's cellular network is quite large — on the order of 100ms [7]. As a result, retransmission schemes are not very effective over the wireless link [5], particularly when the wireless client buffer is small.

As an implementation example, for every k correctly received packets at SA2, SA2 uses a (n, k) systematic Reed-Solomon code to produce $n - k$ additional parity packets. A packet correctly received at SA2 is then lost at the client only if it is lost in the wireless link *and* at least $n-k$ other packets are also lost in the wireless link. Let α and β be the packet loss probability of the wired network and wireless link respectively. The end-to-end packet loss probability ϵ is then:

$$\epsilon = \alpha + (1-\alpha)\beta \sum_{i=n-k}^{n-1} \binom{n-1}{i} \beta^i (1-\beta)^{n-1-i} \quad (1)$$

Note that by simply providing added FEC protection on packets at the wired/wireless junction, it is not necessary for the network agent SA2 to decode and interpret the payload of the packet. Hence, security protocol such as secure RTP [8] can still be deployed. This payload-ignorant security feature is kept in the $R_1 > R_2$ case as well, and it is a fundamental network agent design feature that differs from proposals that advocate real-time video transcoding at the edges of wired networks [9]. The simple addition of FEC is also lightweight in implementation, which is important when a large number of streams traverse the same base-station simultaneously.

5. RESULTS

5.1. Simulation Setup

We performed simulations using Network Simulator 2 [10]. The simulation setup is shown in Figure 2. It has a transport layer duplex connection (p0-p2) from the sender node n0 to the client node n2, and a simplex connection (p1-p0b) from the SA node n1 to the sender node n0.

wired loss	wireless loss	w/o SA2	w/ SA2
0.03	0.005	34.37	35.64
0.05	0.005	32.81	34.40
0.07	0.005	31.54	33.38
0.03	0.01	33.87	35.13
0.05	0.01	32.44	34.00
0.07	0.01	31.26	33.03

Fig. 3. Results for $R_1 > R_2$ Case

wired loss	wireless loss	w/o SA2	w/ SA2
0.03	0.03	32.71	34.59
0.05	0.03	31.99	33.47
0.07	0.03	31.43	32.29
0.03	0.05	31.56	33.49
0.05	0.05	31.01	32.84
0.07	0.05	30.39	31.89

Fig. 4. Results for $R_1 < R_2$ Case

In our simulation, the links n0-n1 and n1-n2 have constant delay and uniform loss rates. An instance of the application, app0, sits at sender node and sends packets to the client using the first connection. Each packet has a sequence number in the packet header. There is a filter at the link from n1 to n2 that sniffs out the packets targeted to the client and forwards it to p1, who sends ACKs back to the sender using the second connection. The sender performs the optimization above based on received ACKs.

The objective measure we are using is average PSNR, calculated relative to the uncompressed original video sequence. When the receiver is unable to decode frame i, the most recently correctly decoded frame j is used for display for frame i, and so we calculate the PSNR using original frame i and encoded frame j instead. If no such frame j is available, then PSNR is 0.

For real video data, we use H.263 video codec to encode the first 50 frames of the carphone sequence into a video stream, encoded at QCIF size, 230kps, 20frames/s and at I-frame frequency of 25 frames. The resulting average PSNR for the compressed stream is 37.01dB. We assume an 1s delay between server start time and client playback time. The delays on the wired network and the wireless link are 50ms and 100ms respectively.

5.2. Fat Pipe Thin Pipe: $R_1 > R_2$ Case

For the more common $R_1 > R_2$ case, we assume a 20% bandwidth excess $R_1 - R_2$ is available for retransmission in the wired network. SA2 detects and drops duplicated packets at the wired / wireless network intersection by scanning at the RTP sequence number of the media flow. The retransmission scheme is a rate-distortion optimized discussed in [5]. We assume a layer of FEC is built into the media stream at the sender to combat wireless loss so the resulting wireless loss is fairly small (less than 1%). The results between the retransmission scheme not using SA2 and using SA2 are shown in Figure 3. The PSNR improvements range from $1.26dB$ to $1.84dB$, with the large improvement taking place when the wired network is most poor and the wireless link is most clean.

5.3. Thin Pipe Fat Pipe: $R_1 < R_2$ Case

For the case when wireless link has a higher throughput than the wired network, recall we employ FEC of rate $R_2 - R_1$ at SA2 for the wireless link. We first assume the raw packet loss rates for the wired network and wireless link α and β are as shown in Figure 4. We assume a $(11, 10)$ Reed-Solomon code is used at SA2, causing the wireless packet loss rate to drop using (1). Performing the rate-distortion optimized application-level retransmission discussed in [5], using timely feedbacks from the SA2 only, the PSNR is increased in all cases as shown in Figure 4 with the addition of the $(11, 10)$ RS code. The most dramatic improvement takes place when the wired and wireless network loss are the smallest.

6. DISCUSSION

In a nutshell, SA2 exploits on the difference in characteristics of the two channels, in particular, the different maximum sending rates in wired network and wireless link. It is interesting to note, however, that SA2 provides improvement only with non-zero network losses. Consider the following scenarios.

Suppose the wired network transmits at R_1 without loss and the wireless link transmits at R_2 without loss. Then no packet recovery mechanisms such as retransmission or FEC are necessary, and it is clear that the sender sends media streams optimally at $\min\{R_1, R_2\}$. This means SA2 is not useful in lossless network environment.

We will next provide an idealized intuition of the gain in using SA2. Suppose the wired network transmits at R_1 with loss probability l_1 and the wireless link transmits at R_2 with loss probability l_2. Suppose however that there is no delay constraint, i.e. a packet is useful as long as it is eventually delivered to the client. This is obviously no longer a streaming scenario; it can instead be a download-and-playback type application. Then, the capacity for the wired and wireless parts of the network are $(1 - l_1)R_1$ and $(1 - l_2)R_2$, respectively.

Consider now that a network agent is used as a relay at the intersection of the wired / wireless intersection, where a sufficiently large initial buffer is used to buffer packets from the sender before transmitting packets to the client. Suppose that retransmission is performed separately between the sender and the agent, and between the agent and the client. By sufficiently large buffer, we mean the likelihood that buffer is empty (buffer underflow) due to retransmission in the wired network is very small. In this case, it is obvious that system can achieve an overall capacity, C_{agent}, given by:

$$C_{agent} = \min\left[(1 - l_1)R_1, (1 - l_2)R_2\right]$$

When no such special network agent is used at the wired / wireless network intersection, the server will transmits at a rate of $\min(R_1, R_2)$, with end-to-end packet loss rate of $1 - (1 - l_1)(1 - l_2)$. Hence, the capacity when the above network agent is not used, C_{NoAg}, is given by:

$$\begin{aligned} C_{NoAg} &= (1 - l_1)(1 - l_2)\min(R_1, R_2) \\ &= \min\left[(1 - l_1)(1 - l_2)R_1, (1 - l_1)(1 - l_2)R_2\right] \end{aligned}$$

Since both $(1 - l_1)$ and $(1 - l_2)$ are less than one in the non-trivial case, we have $C_{Agent} > C_{NoAg}$ in general. Hence we can conclude that a network agent in the intersection of wired / wireless intersection improve performance, even for non-streaming applications.

7. CONCLUSION

In this paper, we show that using a new network agent, situated at the wired / wireless network intersection, streaming media quality can be improved by exploiting the difference in maximum rates between the wired network and the wireless link. The actual improvement depends on the extent to which the rates are mismatched, and in our experiments, we show a PSNR improvement of up to 2dB in video quality.

8. REFERENCES

[1] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," in *submitted to IEEE Trans. MM*, February 2001.

[2] S. Floyd et. al., "Equation-based congestion control for unicast applications," in *SIGCOMM*, 2000.

[3] T. Yoshimura et. al., "Rate and robustness control with rtp monitoring agent for mobile multimedia streaming," in *IEEE ICC 2002*, April 2002.

[4] G. Cheung and T. Yoshimura, "Streaming agent: A network proxy for media streaming in 3g wireless networks," in *Packet Video Workshop*, 2002.

[5] G. Cheung, W. t. Tan, and T. Yoshimura, "Rate-distortion optimized application-level retransmission using streaming agent for video streaming over 3g wireless network," in *(to appear) in ICIP*, 2002.

[6] H. Holma and A. Toskala, Eds., *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*, Wiley, 2001.

[7] G. Montenegro et. al., "Long thin networks," January 2000, IETF RFC 2757.

[8] R. Blom et. al., "The secure real time transport protocol," February 2001, IETF Internet-draft.

[9] G. Tomlinson et. al., "A model for open pluggable edge services," July 2001, IETF Internet-draft: draft-tomlinson-opes-model-00.txt.

[10] "The network simulator ns-2," June 2001, release 2.1b8a, http://www.isi.edu/nsnam/ns/.