QoS tradeoffs using an Application-Oriented Transport Protocol (AOTP) for multimedia applications over IP

Songbin Wei, Vassilios Tsaoussidis, V.N. Venkatakrishnan
Email: [wsongbin, vassilis, venkat @ cs.sunysb.edu]

Abstract

Traditionally, the provided service of Internet transport protocols is reliable/ordered or unreliable/unordered service, respectively. It has been widely accepted during the last years, that a variety of applications (e.g. multimedia) can tradeoff reliability mechanisms to achieve higher throughput. Such a solution, which additionally respects the user preferences regarding the QoS characteristics of importance (e.g. reliability, cost, throughput, delay) is a current demand. We propose an Application-Oriented Transport Protocol (AOTP) to handle partially or completely reliable transport service favoring throughput at the expense of reliability, or dropping the reliability level in order to keep the cost at a desired level. The protocol can be applicable for cases where Resource Reservation is not possible or desired. Our approach does not use Forward Error correction mechanism attempting to save bandwidth, but uses instead, a receiver-based retransmission mechanism. Therefore, the protocol is appropriate for applications that tolerate losses and thus, the need for retransmission (additional RTTs) does not arise often. We present encouraging initial results tested over Ethernet links; we compare AOTP with TCP and a TCP-like protocol without congestion control (TCPWCC), contrasting throughput results for different levels of reliability requirements.

1 Introduction

Communication over a packet-switched network like the Internet, where resources are limited and congestion or delays are quite often the rule rather than the exception, requires adaptivity to network changes or resource reservation. Given the fact that Internet is the host of a variety of high-bandwidth demanding applications (e.g. multimedia), several mechanisms have been proposed to overcome the bandwidth limitations. Most of these mechanisms provide solutions by adapting the application Quality of Service (QoS) to network changes or by reserving resources to guarantee a specified level of QoS. However, the evolution of multimedia applications over the Internet during the last years has raised the demand of a new service class of transport protocols and the TCP/IP stack is not sufficient for a wide range of applications. When speed and/or throughput is the premium while reliability is not the greatest concern, reservation can be avoided and instead, tradeoffs of different QoS characteristics can apply. This new service class tries to offer application specific solutions. For example, video or audio streams can tolerate loss of packets and it should be the user's choice to favor bandwidth at the expense of reliability. Therefore, transport protocols that service such applications should be flexible to adapt to application requirements as well. This paper contributes a solution to this new generation of transport protocols. According to our approach, the level of reliability reflects the bandwidth usage or communication speed. Thus, reliability should be partially or completely offered. Unlike TCP, a receiver-based mechanism applies to decide about the need for retransmission; the criteria are (i) if the receiver has detected packet losses and (ii) if the level of reliability is satisfied (a user might desire a certain level of image quality for example, which can be reflected to the allowed packet loss rate). A receiver-based approach facilitates multi-party applications to decide about the required level/characteristic of Quality of Service depending on the user preferences and / or the local environment constraints. We present performance results of AOTP and TCP in order to emphasize the applicability of the protocol for multimedia applications. Since TCP has additionally a congestion control mechanism which affects significantly the measured performance (although recent proposals are evaluated[9], [10]), we contrast the same results with a TCP-like protocol without congestion control (TCPWCC). The testing environment was an Ethernet-based Network; we have used the x-kernel[8] for the protocol implementation and Virtual Protocols above IP to test the protocol behavior in cases of packet loss at different levels of reliability requirements.

2 Related Work

There are recently several research papers that discuss trading off QoS characteristics that are not significant, and favoring others of greater importance. There is a conflict between the variety of possible QoS requirements (e.g. reliability, speed, bandwidth, real-time playback) thus, leading research to application-specific protocols. Partial reliability - partial reordering is one solution as proposed by [3]. The authors see a possible advantage to the sender-based approach; In our opinion, a receiver-based approach might be more appropriate for multi-party applications when the receivers are flexible to decide about the level of the required QoS.

Additionally, Dempsey [2] supports the use of retransmission as effective way to handle loss of packets even for continuous media applications. This approach is widely rejected for real-time applications since it increases the RTT delay which is crucial for real-time multimedia especially for high-bandwidth links. The use of Forwarding Error Correction (FEC) mechanism reduces the RTT, but wastes more bandwidth than necessary if the application has no significant concern in reliability or if the link is reliable. Clearly, we propose avoidance of redundant data to facilitate error recovery when bandwidth is low, the application can tolerate losses, or the link is reliable, retransmitting, when the level of minimum reliability required has been exceeded. This is applicable for real-time applications over low-bandwidth links. Our protocol is designed to favor throughput for reliable links like Ethernet: in such a case avoiding FEC (that wastes bandwidth) is not a bad choice.

Currently, the XTP Forum [7] has brought to the market a set of protocols that offer high-throughput compared to TCP and even UDP especially over high-bandwidth links. The XTP stack is flexible, widely portable and can adapt to application specific requirements[6] although it aims mainly at high-bandwidth links.

3 AOTP Design issues

In this paper we outline the design, implementation and evaluation of the proposed Application-oriented transport protocol (AOTP). The main concern of AOTP is to achieve maximum throughput and offer the required level of reliability. Therefore, the sender keeps the pipe full, initially without worrying of packet loss or reordering. The receiver adapts the timeout as appropriate to handle the delayed packets, and the sender is informed of dropped packets using NACKs. However, the decision is taken by the receiver since according to our algorithm, the receiver computes the percentage of packet loss, and then when the constraint is violated it sends back NACKs. This approach reduces the NACKs reports since not all dropped packets are reported, but instead, only the ones that will be necessary to bring back the packet loss to an affordable rate (reflecting an image quality for example).

As shown in Figure 1, AOTP uses receiver-based packet loss recovery mechanism. The application can set the desired (or minimum) percentage of the message it wishes to receive. The sender

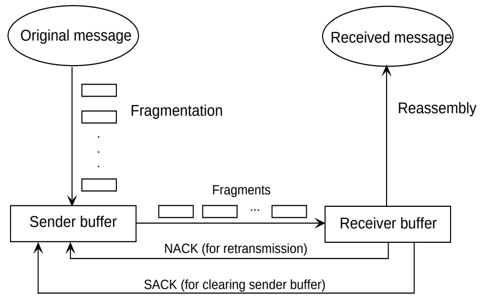

Figure 1: AOTP peer interactions

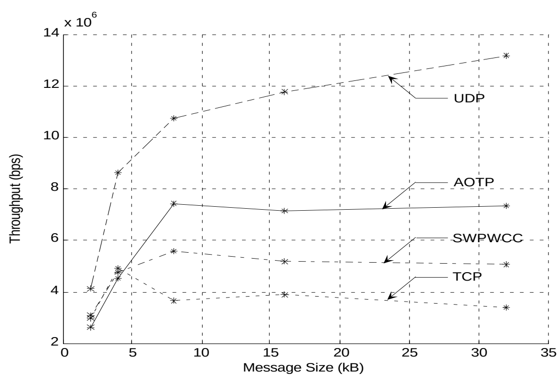

Figure 2: AOTP, TCP, UDP and TCPWCC measured performance

fragments the message when its size exceeds MTU. The fragmentation is optimized for Ethernet. Each message is identified by a unique message ID and each fragment is assigned with a sequence number. The sender stores a copy for every packet it sends. The receiver starts a timer upon receipt of the first packet of a certain message. When the timer expires, it checks the percentage received which is the of number of packets currently received divided by the total number of packets of the message.

```
(receiver)
        if RP < DP then
            send NACKs
        else reassemble
```

If the Receiving Percentage (RP) is less than the Desired Percentage (DP) as it is set by the application, the receiver sends NACKs for those packets which has not been received yet. Otherwise, the receiver reassembles the packets and sends a SACK (Selective Acknowledgment) to the sender in order to clear the sender buffer. We let the receiver send such kind of packets twice to guarantee that the sender will receive this message and clear the buffer in time. The sender, upon receipt of the NACKs, retransmits the lost packets as indicated by the NACKs.

4 Evaluation

AOTP is implemented on the x-kernel platform[8]. We test AOTP, UDP, TCP and TCPWCC without configuring VDELAY or VDROP(virtual protocols in the xkernel suite). Figure 2 shows the results of this experiment. From this figure, we see that the throughput performance of AOTP

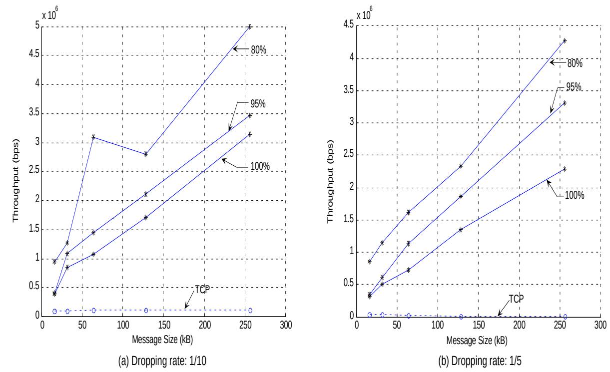

Figure 3: Throughput comparison between AOTP and TCP with a dropping rate of (a) 1/10, and (b) 1/5

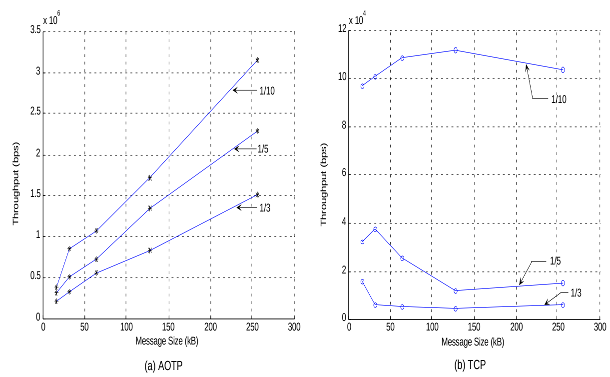

Figure 4: Throughput of (a) AOTP, (b) TCP with a dropping rate of 1/10, 1/5 and 1/3, respectively.

is much better than the ones of TCP and TCPWCC, but worse than UDP. This can be attributed to the fact that AOTP is a partially reliable protocol while UDP is not reliable and while TCP and TCPWCC are reliable. In addition, TCP uses a congestion control mechanism which reduces the throughput even further.

In our next experiment, AOTP is configured above VDROP(virtual protocol that drops packets) and VDELAY(protocol that delays packets) which are configured right above IP. We did various tests setting the dropping rate to 1/10, 1/5, 1/3, 1/2, and receiving percentage to 100%, 95% and 80%, respectively. The results are compared with TCP which is configured at the same dropping rate, as well as UDP. The graphs show that AOTP has much higher throughput than TCP, especially for large message sizes. Below, we present some of these test results:

Figure 3 (a) and (b) illustrate the throughput of AOTP versus TCP with a dropping rate of 1/10 and 1/5, respectively. For AOTP in each figure, three throughput curves corresponding to a receiving rate of 100%, 95%, and 80% are plotted. These figures show that the throughput of AOTP is much higher than TCP for each value of the dropping rate.

Figure 4 shows the throughput variations of AOTP and TCP under different dropping rate. While in both cases the throughput decreases as the dropping rate increases, for TCP the throughput drops much more dramatically than AOTP. This is due to the fact that TCP interprets the loss of packets as congestion and slows down the transmission rate.

5 Conclusion and Future Work

AOTP, as seen from the above discussion, is an appropriate protocol in the current scenario of the Internet. It can be used to serve video/audio applications that can tradeoff reliability to achieve higher throughput. It adapts flexibly to the application's requirements. The test results demonstrate the efficiency of AOTP as compared to the existing protocols.

Our future work includes addition of appropriate congestion control, further optimization, and functionality features addition to satisfy the requirements of real-time multimedia applications. Test results that compare AOTP with current protocols used for multimedia applications (e.g. RTP, RTCP etc.) are being evaluated. In our recent evaluation of reliability / bandwidth tradeoff, we are developing a priority based policy algorithm at the receiver; the schema is in accordance with the MPEG frame classification (I, P, B frames) and we use a message stream generated by the sender. Delay, Bandwidth, and Packet loss rate will be evaluated as the Quality of Service Requirements on the trade.

6 References

[1] P. Amer, C. Chassot, T. Connolly, P. Conrad, M. Diaz "Partial order transport service for multimedia and other applications" *IEEE/ACM Trans on Networking,* 2(5), Oct 1994.

[2] B. Dempsey. "Retransmission-Based Error Control For Continuous Media Traffic In Packet Switched Networks", *Ph.D. Thesis,* University of Virginia, 1994.

[3] R. Marasli, P. Amer, P. Conrad, "An analytic Study of Partially ordered Transport Services", *Computer Networks and ISDN Systems* 1998.

[4] R. Marasli, P. Amer, P. Conrad, "Retransmission-Based Partially Reliable Transport Service: An Analytic Model" *IEEE INFOCOM '96,* CA, March 1996

[5] R. Marasli, P. Amer, P. Conrad, "Partially Reliable Transport service" *ISCC '97,* Alexandria, Egypt, 1997

[6] Espen Klovning, Olivier Bonaventure, "Behaviour of XTPX in the European ATM Pilot" *European Transactions on Telecommunications,* Special issue on "ATM Field Trials and Experiments", Vol. 7, No. 5, September/October 1996,

[7] XTP: The Xpress Transport Protocol: www.ca.sandia.gov/xtp/xtp.html

[8] x-kernel: www.cs.arizona.edu/xkernel

[9] D. Y. Kim, "Enhanced Communications Transport Service Definition", Internet-draft of Network Working Group

[10] V. Paxson, "Known TCP Implementation Problems", Internet-draft of Network Working Group