Studying Multi-rate Multicast Congestion Control with Explicit Router Feedback

Kiyohide Nakauchi Katsushi Kobayashi
National Institute of Information and Communications Technology
{nakauchi, ikob}@nict.go.jp

Abstract

Congestion control with the positive use of explicit notification that indicates internal network conditions is a promising way to address the performance issues of congestion control in high-speed networks. We previously proposed SIRENS, a scalable, robust, and flexible fine-grained explicit notification framework where each router captures a snapshot of the various kinds of downstream link status along the IP-level path from a sender to a receiver and notifies a receiver of the status. A receiver can find out the overall path status by assembling all the cumulative notifications that indicate the status in the single hop, and a sender can share the path status using feedback from the receiver. Such per-hop information is needed by end-hosts to flexibly design novel congestion control mechanisms or to significantly enhance the performance of traditional forms of congestion control in specific situations. In this paper, as a typical application of SIRENS, we study receiver-driven congestion control for multi-rate multicast communications. We address the traditional issues of TCP-friendliness and low responsiveness using SIRENS. We evaluate the proposed multi-rate multicast congestion control mechanism using simulations, and show the implementation status.

1. Introduction

Congestion control with the positive use of explicit notification from routers that indicates internal network conditions is a promising way to address the performance issues of congestion control in high-speed networks. For example, TCP Quick-Start [4] and XCP [5] significantly improve bandwidth utilization using the explicit feedback quantitatively indicating the available bandwidth and the target congestion window size, respectively. Binary explicit feedback, such as Explicit Congestion Control (ECN) [14], AntiECN [7], Variable-structure congestion Control Protocol (VCP) [17], and Explicit Transport Error Notification (ETEN) [6], can also improve bandwidth utilization in specific situations, respectively.

In current explicit notification mechanisms, a receiver is notified of only the maximum or the minimum value of the requested information along the path. We consider that the information is insufficient and instead per-hop information is needed by end-hosts to flexibly design novel congestion control mechanisms or to significantly enhance the performance of traditional forms of congestion control in specific situations. For example, for congestion control like TCP where there is some possibility that the microscopic transmission rate exceeds the link bandwidth when using the high-speed network interface card without any packet pacing mechanism, not only the minimum available bandwidth but also the queue size on the bottleneck link along the path is essential to avoid burst packet drops. As the router with the minimum available bandwidth does not necessarily have the minimum queue size, router feedback indicating only the minimum value of such information through the path is meaningless. Cumulative per-hop router feedback that can capture a snapshot of the path status on a per-hop basis is therefore useful.

In the previous work, we proposed a scalable, robust, and flexible explicit notification framework, called Simple Internet REsource Notification Scheme, or SIRENS [12]. SIRENS is a per-hop and in-band notification scheme where each router captures a snapshot of the various kinds of downstream link status such as link utilization, link bandwidth, delay, queue size, etc, along the IP-level path from a sender to a receiver and notifies a receiver of the status. A receiver can find out the overall path status by assembling all the cumulative notifications that indicate the status in the single hop, and a sender can share the path status using feedback from the receiver. SIRENS is inspired by TCP Quick-Start and Performance Transparency Protocol (PTP) [16]. Though SIRENS requires router modifications, we showed its feasibility in terms of the router processing overhead through the development and evaluation of a network-processor-based high-performance network emulator [12].

In this paper, as a typical application of SIRENS, we study multi-rate multicast congestion control with explicit router feedback, called Explicit Multi-rate Multicast, or EMcast. Multi-rate multicast [9, 10] is an attractive solution to cope with heterogeneous receivers. Multi-rate multicast congestion control mechanisms traditionally suffer from the issues of TCP-friendliness and low responsiveness due to receiver-driven rate adaptation without a feedback to a sender. We apply SIRENS to address these issues.

In EMcast, each receiver is informed of the available bandwidth along the path from a sender to the receiver and independently determines optimal subscription level. This explicit router feedback enables EMcast receivers to quickly grasp the remaining bandwidth (leave the current multicast group and join the higher-bandwidth one) when the available bandwidth suddenly increases due to sudden increase of the link bandwidth (e.g. in the case of handoff in the mobile environments) or the termination of competing traffic. In addition, EMcast receivers decrease the receiving rate (leave the current multicast group and join the lower-bandwidth one) in accordance with TCP's steady-state throughput model. This congestion avoidance policy enables EMcast receivers to achieve TCP-friendliness.

The main contribution of this paper is that we show multi-rate multicast congestion control with SIRENS can cope with the traditional issues of TCP-friendliness and low responsiveness.

This paper is organized as follows. We provide a design overview of SIRENS in Section 2. In Section 3, we show its application to multi-rate multicast congestion control and describe EMcast congestion control. We also show a set of simulation results and implementation status. We conclude in Section 4.

2. SIRENS Framework

SIRENS is a per-hop and in-band notification scheme where each router captures a snapshot of its downstream link status along the IP-level path from a sender to a receiver and notifies a receiver of the status. In this section, we briefly show the design rationale and architecture of SIRENS.

2.1. Design Rationale

The fine-grained explicit notification framework is required to be applicable to a broad class of transport protocols and be able to provide the various kinds of fine-grained feedback information . In addition, the framework must be scalable and robust. Considering these requirements, we designed SIRENS with the following properties.

(1) Per-hop feedback
(2) Interpretation of QoS semantics at end-hosts

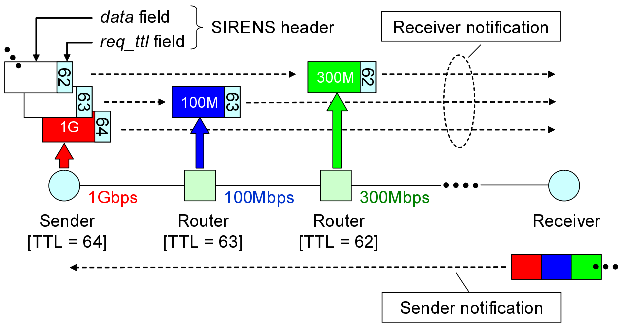

Figure 1. Basic Behavior of SIRENS for a LINK Profile

(3) Fine-granularity of feedback information
(4) Per-packet processing
(5) Maintenance of no per-flow state at a router
(6) Applicability to multicast congestion control

The concept of explicit notification itself is not novel. The most notable features of SIRENS that differentiates SIRENS from the current explicit notification schemes are per-hop feedback and interpretation of QoS semantics at end-hosts. TCP Quick-Start and XCP provide simpler explicit notification mechanisms for the specific purposes. In these schemes, a receiver is informed of just the minimum value for the specific information (the available bandwidth in Quick-Start and the optimal congestion window in XCP). On the other hand, SIRENS captures a snapshot of the path status on a per-hop basis and enables the receivers to freely make use of this information and to perform more precise and flexible congestion control. We exploited this advantage for optimization of TCP Limited Slow-Start in [12].

2.2. Architecture

The key concept of SIRENS is that a receiver can find out the overall path status by assembling all the cumulative notifications that indicate the status in the single hop, and a sender can share the path status using feedback from the receiver. In this framework, all that routers have to do is write specified information into the header of the packets to which the routers are corresponding.

SIRENS is an in-band notification scheme, and the SIRENS header is added to every data packet. The SIRENS header is put between IP header and that of a transport protocol. The receiver notification transfers specified information from a router to a receiver. The sender notification enables a sender to share the information with a receiver. Note that the sender notification can also be piggybacked with an ACK packet in the ACK-based transport protocols such as

TCP.

In SIRENS, req_probe field in the SIRENS header defines the kind of collected information. In the current implementation, we define three profiles for req_probe: LINK, LOSS, and QUEUE. Each profile further defines two kinds of information as follows.

- LINK: link bandwidth and available bandwidth
- LOSS: packet loss rate and link error rate
- QUEUE: queue size and link delay

Figure 1 shows the basic behavior of SIRENS for a LINK profile. The sender sets req_ttl field in the SIRENS header cyclically ranging from the same value as TTL in the IP header to 1 in decreasing order in every data packet. At the routers supporting SIRENS, req_ttl in the header is compared with TTL in the IP header, and if these values are the same, the router writes information specified by req_probe into data field. Note that this request packet is only processed at this router along the path. The receiver, if necessary, writes the collected information together into the data field like a stack in the SIRENS header (the sender notification), and returns the sender notification to the sender.

The advantages of SIRENS are as follows.

- **Scalability:** Each router is not required to keep any per-flow state for the framework. In addition, feedback processing at each router is simple. All that each router needs to do is write measured or pre-configured QoS semantics into feedback packets. Interpretation of QoS semantics and its application to congestion control is imposed on the end-hosts.

- **Robustness:** Feedback packet drops are less effective because of the per-packet feedback frequency. In addition, the framework has the potential to cope with flooding-based DoS attacks.

- **Flexibility:** The framework is flexibly applicable to a broad class of transport protocols because it is easy to define new req_probe(res_probe), and in addition the interpretation of QoS semantics are individually defined at end-hosts.

Note that SIRENS is just a notification scheme and does not define how to make effective use of the information. As a typical application of SIRENS, we showed parameter optimization of TCP Limited Slow-Start [2], called *optimized Limited Slow-Start* [12]. In Section 3, as another typical application, we study multi-rate multicast congestion control with explicit router feedback.

3. Explicit Multi-rate Multicast

In this section, we first show the basic mechanism of multi-rate multicast and show the traditional issues of TCP-friendliness and low responsiveness. Then we show EMcast, multi-rate multicast congestion control with SIRENS. We also show a set of simulation results and implementation status.

3.1. Multi-rate Multicast

Multi-rate multicast [9, 10] is an attractive solution to cope with heterogeneous receivers. Several different source rates are pre-defined and each receiver can receive at a rate suitable for its environment. In multi-rate multicast, congestion control is generally done in a receiver-driven fashion to avoid feedback implosion. Each receiver independently estimates network conditions and tries to optimize the subscription level.

Multi-rate multicast is realized in practice through layered multicast [11] or simulcast [1]. The basic idea of layered multicast is to divide the source stream into a hierarchy of exclusive additive layers and then transmit each layer with a different multicast address. Receivers then decide how many groups (layers) to join using IGMP. On the other hand, simulcast is a method where the sender chooses two or more encodings and transmits two or more different streams containing the same information with a different multicast address. Receivers then decide which group to join using IGMP in the same way as in layered multicast. In this paper, we focus on simulcast because of its simplicity. It is well known that the congestion control schemes proposed for each model can be easily shared.

In a network with a high-speed bottleneck link ranging from several hundred Mbps to 1 Gbps, traditional congestion control schemes such as trial-based adaptation [11] or estimation-based adaptation [8] do not work well. Trial-based adaptation traditionally suffers from the issues of TCP-friendliness and low responsiveness. We apply SIRENS to address these issues. In EMcast, each receiver is informed of the available bandwidth along the path from a sender to the receiver and independently determines optimal subscription level. This router feedback enables EMcast receivers to quickly grasp the remaining bandwidth (join the higher-bandwidth multicast group) when the available bandwidth suddenly increases due to sudden increase of the link bandwidth or the termination of competing flows. In addition, EMcast receivers decrease the receiving rate (join the lower-bandwidth multicast group) in accordance with TCP's steady-state throughput model. This congestion avoidance policy enables EMcast receivers to achieve TCP-friendliness.

On the other hand, in estimation-based adaptation, estimation schemes such as packet-pair suffer from the link layer effects and significantly degrade the estimation accuracy especially in high-speed networks. This inaccurate estimation results in burst packet drops and may lead to seri-

```
// For each arriving explicit notification:
/* Add Process */
if ( $t - t_0 \geq$ Interval && $r_i < R$ ) {
    while ( $T_{i+1} < T \times C_3$ ) {
        if ( $i <$ HighestLevel ) {
            $i++$;
            $t_0 = t$;
        }
    }
}
/* Drop Process */
if ( $t - t_0 \geq$ Interval && $r_i \geq R$ ) {
    if ( $i >$ LowestLevel ) {
        $i--$;
        $t_d = t$;
    }
}
```

Figure 2. EMcast Congestion Control Algorithm

ous performance degradation.

3.2. EMcast Congestion Control

In EMcast, each receiver is informed of the available bandwidth by the LINK profile in SIRENS along the path from a sender to the receiver and independently determines the optimal subscription level when there is enough available bandwidth for a higher subscription level. The information collected at a receiver is used for receiver-driven rate adaptation and is not returned to a sender.

The system model of EMcast is as follows. Explicit notification is preliminarily implemented on RTP (a control channel) apart from data streams (data channels). A single RTP control channel is created for each set of data streams (a session) containing the same information and all the receivers in the session are required to also join this control channel in addition. The senders must be located close enough to share the RTP control channel. A base channel (the channel with the lowest rate) is responsible for the RTP control channel. In this model, all the channels need to share a bottleneck link.

Figure 2 shows the EMcast congestion control algorithm. Here, t, T, T_i, and r_i respectively denote the current time, available bandwidth, rate of the i-th level, and measured packet loss rate. R denotes the packet loss rate calculated using the following basic model that approximates TCP's steady-state throughput [3]:

$$T_i = 1.3 \times \text{MTU} / (\text{RTT} \times \text{sqrt(R)})$$

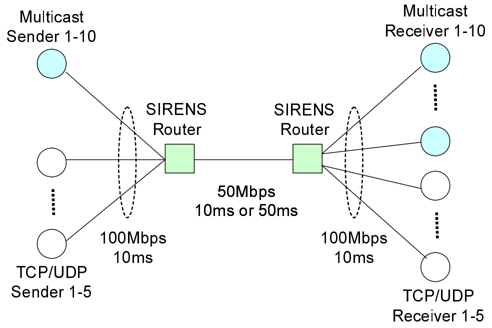

Figure 3. Simulation Topology

C_3 is constant and is set to 0.8 in our simulations. We set the value of Interval to 0.2 s.

Note that the trigger for decreasing the subscription level is packet drops, rather than the detected decrease in the available bandwidth, so that a receiver can achieve TCP-friendliness. Also note that a receiver determines the subscription level based on available bandwidth, instead of the TCP steady-state throughput, in order to quickly grasp the remaining bandwidth without being limited by the TCP steady-state throughput when TCP flows cannot immediately reach their fair share . This congestion control mechanism enables receivers to immediately optimize their subscription level when network conditions change while achieving TCP-friendliness at the same time.

3.3. Simulation

We preliminarily evaluated the performance of explicit multi-rate multicast using the network simulator $ns2$ [15]. Figure 3 shows the simulation topology. The sending rate of channel i is set to $100 \times 2^{i-1}$ Kbps, and the maximum channel rate is 51.2 Mbps ($i = 10$). We set the shared bottleneck's bandwidth to 50 Mbps. The queue size of each link is 100 packets, and we adopted the RED queue management scheme (min = 50 packets).

Figures 4 and 5 show the throughput of EMcast (with ten channels and one receiver) and five competing TCP flows when we set the end-to-end delay to 30 ms and 70 ms, respectively. These results show that EMcast can gradually decreases the subscription level to fairly share the bottleneck link with TCP flows.

Figures 6 and 7 show the throughput of EMcast (with ten channels and ten receivers) and a single or five competing TCP flow(s). We set the end-to-end delay to 30 ms. The throughput in these simulations denotes the total throughput of all the ongoing multicast groups in a session. This definition causes the throughput fluctuation in EMcast before TCP's throughput increases. However, almost all the re-

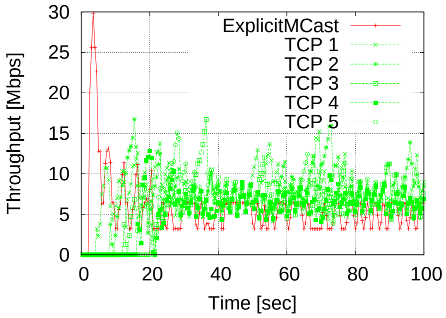

Figure 4. EMcast with 5 TCP Flows (Single Receiver, Delay = 30 ms)

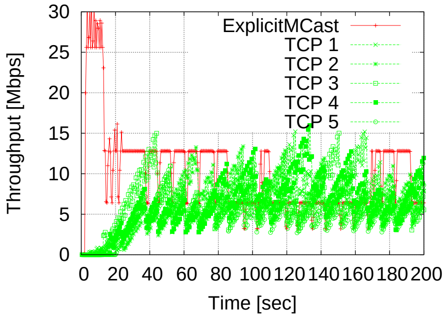

Figure 5. EMcast with 5 TCP Flows (Single Receiver, Delay = 70 ms)

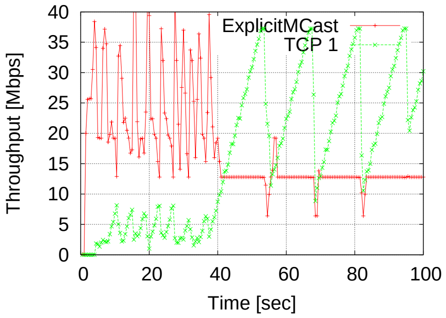

Figure 6. EMcast with a TCP Flow (10 Receivers)

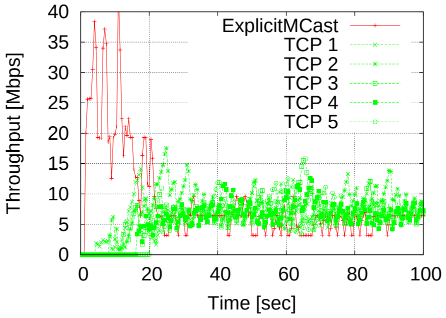

Figure 7. EMcast with 5 TCP Flows (10 Receivers)

ceivers eventually join the same multicast group to be in the steady state. These results show that EMcast can approximately achieve intra-session fairness even when competing TCP flows.

Figures 8 and 9 show the throughput of RLM and EMcast (with ten channels and a single receiver) when CBR cross traffic with the sending rate of 45 Mbps is suddenly generated from 50 seconds to 80 seconds. Such sudden change of the available bandwidth would be typical in the mobile environments. We set the end-to-end delay to 30 ms. RLM takes about 50 seconds to reach optimal subscription level. In addition, due to this slow response to the congestion, the RLM receiver suffers from the significant packet drops for about 10 seconds. Note that we configured RLM parameters to be more sensitive to the congestion and to be more aggressive to join the higher layer in this simulation (e.g. Join-timer = 3.0, Detection-timer = 4.0, Loss threshold = 0.1). On the other hand, EMcast can adapt its subscription level quickly in response to network conditions. At the beginning of the session and after the termination of cross CBR traffic, a receiver can quickly grasp the remaining bandwidth using SIRENS information for the available bandwidth.

3.4. Implementation

We are now working on implementation of EMcast receiver-driven congestion control based on Iperf (version 2.0.2) [13], a well-known unicast / multicast performance measurement tool.

Figure 10 shows the local experimentation environment for EMcast in our laboratory. The SIRENS-capable multicast router (implemented with *mrouted*) with the network-processor-based high-performance network emulator sets the bottleneck link bandwidth. We can also generate competing TCP / UDP flows between multicast servers and multicast receivers.

We use the standard Iperf client program at three multicast servers with different sending rates (e.g. 10Mbps, 100Mbps, and 500Mbps). We modify the Iperf server program for multicast receivers. We add the EMcast congestion control mechanism shown in Figure 2 to the Iperf server program. We configure the SIRENS-capable multicast router to write the available bandwidth into the SIRENS header in the Iperf packets using LINK profile. In this preliminary implementation, the sending rate of each multicast group is manually preset at each receiver.

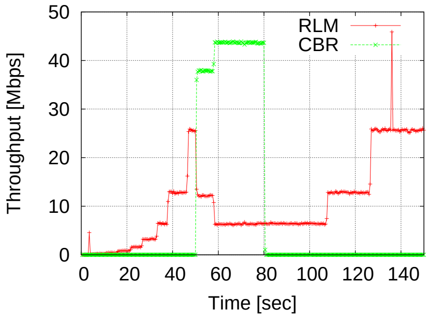

Figure 8. RLM with a CBR Flow (Single Receiver)

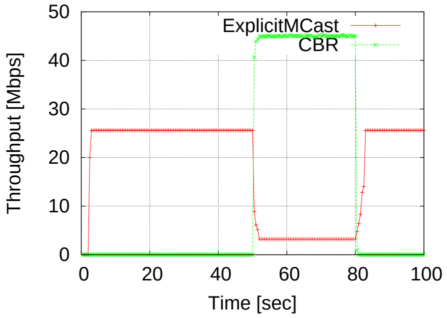

Figure 9. EMcast with a CBR Flow (Single Receiver)

4. Conclusions

In this paper, we studied congestion control for multi-rate multicast using SIRENS, a scalable, robust, and flexible explicit notification framework that can provide various kinds of fine-grained information from within the network. We have proposed EMcast that can achieve both TCP-friendliness and high responsiveness by exploiting the features of SIRENS. We evaluated the basic performance of EMcast by a set of simulations and made clear its advantage over traditional multi-rate multicast congestion control. We also described the current status of implementation on Iperf.

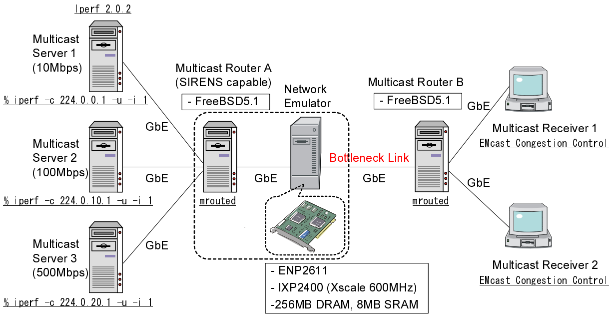
```
% iperf -c 224.0.0.1 -u -i 1
% iperf -c 224.0.10.1 -u -i 1
% iperf -c 224.0.20.1 -u -i 1
```

Figure 10. Experimentation Environment for EMcast

References

[1] S. Y. Cheung, M. H. Ammar, and X. Li. On the use of destination set grouping to improve fairness in multicast video distribution. *Proc. IEEE INFOCOM '96*, Mar. 1996.

[2] S. Floyd. Limited Slow-Start for TCP with Large Congestion Windows. *RFC 3742*, Mar. 2004.

[3] S. Floyd and K. Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, Aug. 1999.

[4] A. Jain, S. Floyd, M. Allman, and P. Sarolahti. Quick-Start for TCP and IP. *Internet Draft, draft-ietf-tsvwg-quickstart-00.txt*, May 2005.

[5] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay Product Networks. *Proc. ACM SIGCOMM '02*, Aug. 2002.

[6] R. Krishnan, J. Sterbenz, W. M. Eddy, C. Partridge, and M. Allman. Explicit Transport Error Notification (ETEN) for Error-prone Wireless and Satellite Networks. *Computer Networks*, 46.

[7] S. S. Kunniyur. AntiECN Marking: A Marking Scheme for High Bandwidth Delay Connections. *Proc. ICC '03*, May 2003.

[8] A. Legout and E. W. Biersack. PLM: Fast convergence for cumulative layered multicast transmission schemes. *Proc. of ACM SIGMETRICS 2000*, June 2000.

[9] J. Liu, B. Li, and Y.-Q. Zhang. Adaptive video multicast over the internet. *IEEE MultiMedia*, 10(1):22–33, 2003.

[10] A. Matrawy and I. Lambadaris. A survey of congestion control schemes for multicast video applications. *IEEE Communications Surveys and Tutorials*, 5(2), 2003.

[11] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. *Proc. ACM SIGCOMM '96*, Aug. 1996.

[12] K. Nakauchi and K. Kobayashi. Studying Congestion Control with Explicit Router Feedback Using Hardware-based Network Emulator. *Proc. PFLDnet '05*, Feb. 2005.

[13] NLANR/DAST. Iperf version 2.0.2 (2005). *http://dast.nlanr.net/Projects/Iperf/*.

[14] K. K. Ramakrishnan, S. Floyd, and D. L. Black. The Addition of Explicit Congestion Notification (ECN) to IP. *RFC3168*, Sept. 2001.

[15] UCB/LBNL/VINT. Network Simulator – ns (version 2) (1999). *http://www-mash.cs.berkeley.edu/ns/*.

[16] M. Welzl. PTP: Better Feedback for Adaptive Distributed Multimedia Applications on the Internet. *Proc. IEEE IPCCC 2000*, Feb. 2000.

[17] Y. Xia. One More Bit Is Enough. *Proc. ACM SIGCOMM '05*, Aug. 2005.