Router-Assisted TCP-Friendly Traffic Control for Layered Multicast

J. Y. Son[1], K. R. Kang[2], D. Lee[2], S. H. Kang[2], Y. H. Lee[2], and D. W. Han[1]

[1] Electronics and Telecommunications Research Institute
Gajeong-dong, Yusong, Daejon, 305-350, Korea
{jyson,dwhan}@etri.re.kr

[2] School of Engineering, Information and Communications University
58-4 Hwaam-dong, Yusong, Daejon, 305-732, Korea
{korykang,dlee,kang,yhlee}@icu.ac.kr

Abstract. In this paper, we propose an efficient TCP-friendly traffic control scheme for layered multicast with router assistance. The proposed scheme is based on Network-based Layered Multicast (NLM), which dynamically adjusts the traffic on each link at a router to ensure the high quality data reception at the receivers as much as possible the network allows. The proposed scheme enhances TCP-friendliness of NLM by improving the traffic control granularity. The performance results show that the proposed scheme yields better performance compared with the original NLM and RLC, an end-to-end traffic control scheme for layered multicast.

1 Introduction

With the advancement of computer and network technology, multi-party multimedia applications such as video conferencing and video on demand have become of great interest. However, still the network bandwidth and the computing capability of the receivers are various and the efficient adaptation of a given network condition is inevitable to support heterogeneous receivers. The *layered multicast* approaches, such as Receiver-driven Layered Multicast (RLM)[1] and Layered Video Multicast with Retransmission (LVMR)[3], have been widely recognized as an efficient mechanism to handle the receiver heterogeneity.

In the layered multicast, video data are encoded into multiple layers: *base layer* and *enhancement layers*. The sender transmits the encoded data over separate multicast groups, and each receiver determines how many layers to subscribe depending on its capability or desired level of quality of video. However, RLM and LVMR shows poor inter-session fairness [2] and they can harm TCP traffic which is the dominant one of the Internet. The requirement that a rate control mechanism should work similar to TCP is called *TCP-friendliness*.

To enhance TCP-friendliness in layered multicast, RLC[7] and FLID-DL[6] have been proposed. However, they are an end to end approach and require at least round trip time between the sender and the farthest receiver. To reduce the time delay in rate adaptation, network-based layered multicast approaches have

H.-K. Kahng (Ed.): ICOIN 2003, LNCS 2662, pp. 202–211, 2003.

been introduced [5, 8, 9]. The network based approaches allow fast adaptation to network traffic changes over time since they determine the number of layers at the router or exploit the information provided by the router. However the granularity of traffic change is much coarser than TCP and TCP traffic may suffer instability.

In this paper, we propose an efficient TCP-friendly traffic control scheme for layered multicast with router assistance. The proposed mechanism exploits the previous work, *Network-based layered multicast (NLM)* [5] and enhances the traffic control granularity to improve the TCP-friendliness. Like NLM, Time-To-Live (TTL) threshold is used in determining whether or not to forward packets and Type-of-Service (TOS) bits of the IP header is used to distinguish the traffic under our control scheme. To enhance the traffic control granularity, we consider both the total number of outgoing sessions and the queue occupation ratio of video traffic at the same time. We simulated and evaluated the proposed scheme using ns-2. The performance evaluation results show that the proposed scheme yields better TCP-friendliness compared with the original NLM and RLC[7].

The remainder of the paper is organized as follows: Section 2 describes the existing TCP-friendliness schemes for layered multicast. In Section 3, we describe design considerations and the details of the proposed scheme and then, in Section 4, we present the simulation results and analysis. Finally, Section 5 offers the conclusion.

2 Related Works

To support TCP-friendliness in a layered multicast, Xue Li [2] proposed layer-based congestion sensitivity rate control. It showed that the basic rate control scheme in the layered approach do not handle inter-session fairness well when there are multiple video sessions competing for bandwidth and that the basic layer adaptation scheme can bring unfairness to the competing TCP traffic as TCP is more sensitive to congestion. To achieve better inter-session fairness, they used equation-based TCP-friendly rate control to provide the fairness with TCP traffic. This scheme guarantees bounded fairness with respect to TCP using equations. However, it is hard to measure the accurate round trip time in a real network environment.

With receiver-driven layered control (RLC), Vicisano et al. [7] developed a scheme in which the receivers join or leave a layer based on their measured loss rates. Using specially flagged packets, the sender indicates synchronization points at which receivers might join or leave a specific layer. With RLC, the sender divides its data into layers and sends them on different multicast sessions. To test resource availability, the sender periodically generates a short burst of packets followed by an equally long relaxation period in which no packets are sent. The data rate of the flow is doubled during the burst. After receiving a packet burst, the receivers can join a higher layer if the burst is lossless; otherwise they remain at their current subscription level. The receivers might leave a layer at any time if losses are measured.

The Fair Layered Increase/Decrease with Dynamic Layering (FLID-DL) proposed in [6] enhances the RLC by using dynamic layering. Dynamic layering reduces the leave latency when dropping a layer, since a receiver has to periodically join additional layers to maintain a non-decreasing rate.

Some network based adaptation algorithms have also been proposed. Bhattacharyya et al.[8] introduced a useful reduction technique at the router using dependencies between video frames, which they refer to as the Group-Of-Picture (GOP)-level discard technique. They also demonstrated that network-based adaptation could yield significant performance gains for multicast video distribution. As another approach, Gopalakrishnan [9] proposed a hybrid scheme of layered and network driven adaptations, which is called Receiver-driven Layered Multicast with Priorities (RMLP). It is based on RLM [1] and a two-priority dropping scheme at the router. They demonstrated that their scheme improves the stability and intra-session fairness over those of RLM. However, as presented in [5], it shows low TCP-friendliness.

3 Proposed Scheme

3.1 Design Considerations

Our design goal is to develop a TCP-friendly layered multicast scheme allowing fine-tunable traffic control dynamically adapting to transient network traffic changes. It is desirable to allow moderate traffic control granularity since too large traffic control granularity may incur highly frequent traffic changes. To the contrary, too small granularity may result in poor TCP-friendliness because TCP sessions adapt their traffic too aggressively. For fine-tunable traffic control, we consider following two factors.

Number of Video Sessions Sharing a Bottleneck Link. Network based layered multicast approaches consider video sessions as a whole and not separately at a given link. As the number of sessions increases, does the number of video sessions that could be affected by the control scheme. This incurs coarse traffic granularity of the traffic control and, as a result, the reception quality of the receivers is destabilized. To avoid this problem, the traffic control scheme should moderate the sensitivity of layer add/drop criteria according to the number of sessions.

Bandwidth Occupation Ratio of Video Traffic. In the layered encoding[1], as the level of layer increases, the data rate does exponentially. Thus, the number of video sessions is not enough for fine grained traffic control. With the same number of sessions, the larger number of layers is allowed for a session, the more video traffic in a link. Dropping a higher layer will incur more traffic change than a lower layer. Therefore, the larger is the bandwidth occupation ratio of video traffic, the more conservatively should the control algorithm work.

3.2 NLM

NLM [5] is designed to deliver layered video to a heterogeneous set of receivers using the network-wide traffic information. Like other layered multicast schemes, it assumes that the source encodes the video signal into multiple discrete layers and each layer is transmitted on a separate multicast group. In NLM, the source also assigns a proper Time-to-Live (TTL) value and Type-of-Service (TOS) bits to each packet. The TTL value is used as a mark to show which layer the flow belongs to and the TOS bits are used to differentiate the packets under NLM scheme. A receiver subscribes to as many layers as its link bandwidth permits and the initial membership may last until the end of the flow. The receiver only reproduces the original data using the received data, and the quality of data is determined by the number of data layers received.

NLM employs a traffic controller including a *filter* and a *measurer* in a router. The *filter* is located in front of the queue and controls the amount of output packets of the link. It checks if the packet is qualified to forward. The *measurer* measures the average queue length, which is used as traffic metric. Based on a queue length, NLM categorizes the level of current link traffic status into three levels: *unloaded*, *loaded* and *congested*. Each status level has a corresponding tuple of highest and lowest threshold, and change direction. The direction indicates whether the threshold is to be increased or decreased. The value of the direction is -1 for a decrement and +1 for an increment. The set of these tuples is called *Guide*. It exists per output link interface and the direction of the changes of the router is determined by taking a minimum value among the directions of each outgoing interface. Using this traffic state information and the Guide, a traffic controller requests the neighboring controller for a given link to change the TTL threshold. The neighbor controller modifies the corresponding TTL as requested. By changing the value of the TTL threshold to reflect the traffic state, the traffic controller achieves traffic control. It can moderate the traffic by dropping the data of less significant layers, resulting in a change in the number of data layers transmitted through the network interface. These steps are repeated periodically.

3.3 Session-Based Traffic Adaptation Algorithm

As mentioned in Section 3.1, we aim to design a fast adaptive and TCP-friendly traffic control scheme for layered multicast. To support fast adaptation to the network traffic change, the proposed scheme assumes that the source and the router act as specified in NLM[5]: the source encodes the video stream into multiple layers and transmits data packets, assigning a proper TTL value according to its layer; The value of TOS bits is the same as NLM; The router drops the multicast packets whose TTL exceeds the TTL threshold. For improving the TCP-friendliness of NLM, we propose a session-based traffic adaptation algorithm which considers the number of the sessions in a link and bandwidth occupation ratio of each type of traffic.

The algorithm assumes that the traffic in a link consists of NLM traffic and TCP traffic. NLM traffic is identified by the TOS bits of the IP header.

Table 1. TTL threshold adaptation parameters

Parameters	Description
$ttl\_threshold_i(x)$	TTL threshold value for outgoing interface i at time interval x
max_r	maximum change ratio of the TTL threshold
$delta_i(x)$	threshold change ratio for outgoing interface i at time interval x
$q_t(x)$	average queue size of total traffic at time interval x
$q_v(x)$	average queue size of NLM traffic at time interval x
$n(x)$	the number of NLM sessions on a link at time interval x
$direction_i(x)$	threshold change direction for outgoing interface i at time interval x

Periodically, the router measures the bandwidth occupation ratio of NLM traffic over the total traffic and the number of NLM sessions in a out-going link. We assume a NLM session as a one-to-many data distribution, and a set of flows of the same source address and different destination addresses is identified as a single session. Using the bandwidth occupation ratio and the number of NLM sessions, the algorithm determines the amount of the TTL threshold change as described in equation (1).

Table 1 presents the parameters used for describing our algorithm. To update the change unit of the TTL threshold adaptively with an upper bound and lower bound, the TTL threshold value for an outgoing interface i at time interval x, $ttl\_threshold_i(x)$ is dynamically changed by the ratio $delta_i(x)$.

$$\begin{aligned} &ttl\_threshold_i(x) = ttl\_threshold_i(x-1) + delta_i(x), \\ &delta_i(x) = direction_i(x) \times \tfrac{max\_r}{n(x)} \times \tfrac{q_t(x)}{q_v(x)} \end{aligned} \tag{1}$$

The value of the $direction_i(x)$ is one of -1, 0, and $+1$. The condition to determine $direction_i(x)$ is the same as that of NLM. The *Guide* is configured by the network administrator and, $direction_i(x)$ is determined according to the traffic status to which the expected average queue length of time interval x corresponds. Parameter max_r is set to 50, which is the value at which one layer can be dropped or added. The max_r is divided by the number of NLM sessions to moderate the amount of traffic change incurred by the TTL threshold change. The factor $q_t(x)/q_v(x)$ has the same effect, in other words, the larger amount of NLM traffic results in the smaller threshold change. As a result, the total amount of traffic change can be controlled in a fine-grained manner.

Fig. 1 describes the session-based traffic adaptation algorithm applying the above equation in a router. *check_status*() function is to determine the network traffic status based on the expected queue length and returns one of *unloaded*, *loaded* and *congested*. The *convert_into_direction*() function returns one of -1, 0, and 1, according to the *expected_average_queue_length*.

```
At Router: If it a˜receives threshold change request from i-th
r-router, it saves as direction_req(i)

for every network interface i at time interval x {
        it calculates the expected_average_queue_length and
        number of sessions(n), and VBR traffic ratio(qv/qt)
        in the queue;

        traffic_state = checks_status(expected_average_queue_length);
        direction = convert_into_direction(traffic_state);
        for each outgoing interface i {
                if (direction(i) < direction_req(i)
                   direction(i) = direction_req(i);
                delta(i) = direction(i) * (max_r / n) * (qv / qt);
                ttl_threshold(i) = old_ttl_threshold(i) + delta(i);
                old_ttl_threshold{i}=ttl_threshold{i};
        }
        threshold_change_request =
                        the minimum values of direction(i);
        It notifies threshold_change_request to upward routers;
}
```

Fig. 1. Traffic adaptation algorithm in a router

4 Performance Evaluation

We simulated the performance of the proposed scheme using ns-2 [11]. We also compared the proposed scheme with the original NLM and RLC. The simulation topology used in our evaluation is shown in Fig. 2.

In Fig. 2., the white-colored circles represent routers including the traffic controller. Gray-colored circles represent the source, and box-shaped nodes represent receivers. To simulate the receiver heterogeneity, the link bandwidth is selected within the range of 512 kbps and 8 Mbps.

4.1 Simulation Scenarios

We used the Variable Bit Rate (VBR) source model for NLM traffic as described in [4]. This model generates traffic over one second intervals for the base layer. In each interval, n packets are transmitted, where $n = 1$ with probability $1 - 1/P$ and $n = P \times A \; + 1 - P$ with probability $1/P$. A is the average number of packets per interval, and it is chosen to be four 4 KB packets. n packets are transmitted in a single burst, starting at uniformly distributed random time within the interval P, which represents the burst size of the traffic source, is set to 3 for modeling VBR sources. For each layer, the interval is broken into two subintervals. Each source encodes the data into four layers. The base layer is transmitted at the rate of 32 kbps, with the rate doubling for each subsequent

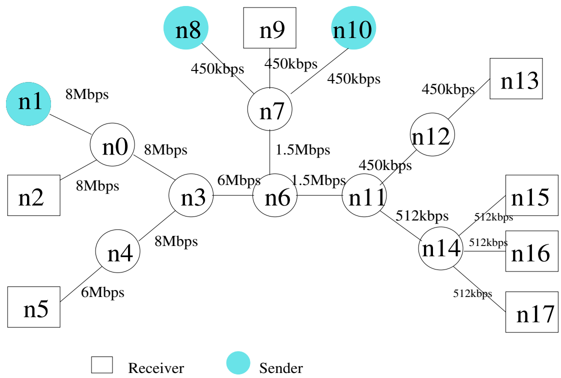

Fig. 2. Simulation network topology

Table 2. Simulation scenarios

Scenario id	Session configuration	Session description
Scenario 1	two VBR sessions (S1,S2) and two ftp sessions (T1,T2)	S1(n1):2-610, S2(n10):50-610, T1(n1,n15):30-400, T2(n2,n15):80-500
Scenario 2	three VBR sessions (S1,S2,S3) and one ftp sessions (T1)	S1(n0):2-610, S2(n10):50-610, S3(n10):120-610, T1(n1,n15):300-600
Scenario 3	one VBR session (S1) and three ftp sessions (T1,T2,T3)	S1(n0):2-610, T1(n1,n15):150-400, T2(n2,n15):250-500, T3(n10,n15):200-600

layer. FTP is used for a TCP session. At each router, the maximum queue size is set to 50.

We extend the idea of [10] to describe the TCP-friendliness in a quantitative manner by adding a time factor. TCP-friendliness at a given time interval x, $F(x)$, is defined as the ratio of the average throughput of their protocol proposed to the average throughput of TCP as follows:

$$F(x) = \frac{T_v(x)}{T_T(x)}, T_v(x) = \frac{\sum_{j=1}^{k_v(x)} T_j^v(x)}{k_v(x)}, T_T(x) = \frac{\sum_{j=1}^{k_T(x)} T_j^T(x)}{k_T(x)} \qquad (2)$$

$k_v(x)$ is the total number of the NLM sessions and $k_T(x)$ is the total number of TCP sessions at the time interval x. A TCP session is identified by the source address and the destination address. $T_1^v(x), T_2^v(x), ..., T_{k_v}^v(x)$ is the throughput of each NLM session and $T_1^T(x), T_2^T(x), ..., T_{k_T}^T(x)$ is that of each TCP session, respectively. The scheme can be said TCP-friendly if $F(x)$ is close to 1.

4.2 Simulation Results

This section examines the TCP-friendliness of the proposed scheme, comparing with the original NLM and RLC. The simulation results are obtained from the receiver node n15.

Fig. 3 presents the TCP-friendliness, $F(x)$, observed in Scenario 1. The proposed scheme shows far better TCP-friendliness compared with the original NLM. The peak at the starting point of the sessions results from the fact that a VBR session starts with higher rate than FTP session. However, as the sessions go on, the proposed scheme shows better TCP-friendliness because the proposed scheme controls the number of the layers. RLC shows larger $F(x)$ than the proposed scheme, which implies the NLM traffic takes smaller portion of the bandwidth. The fluctuation of $F(x)$ is severe and it may cause the reception quality instability at the receivers.

Fig. 4 shows the TCP-friendliness, $F(x)$, of Scenario 2. The proposed scheme shows larger $F(x)$ than the original NLM while maintaining the data rate of VBR sessions doubled. As shown in Fig. 4, the variation of $F(x)$ of RLC is higher than that of Scenario 1.

In Scenario 3, the original NLM and the proposed scheme show the similar result as shown in Fig. 5. RLC still shows frequent fluctuation.

5 Conclusions

Layered multicast has been considered to be an effective mechanism for multimedia data delivery to heterogenous receivers. To incorporate TCP-friendliness and prompt reaction to network traffic changes, the approaches with router assistance have been proposed. However, they control the traffic in a coarsely grained manner and result in instability of coexisting TCP sessions. To allow more efficient traffic control and TCP-friendliness support, in this paper, we propose a router-assisted TCP-friendly layered multicast scheme. The proposed scheme shows prompt reaction to network traffic changes by exploiting the previous

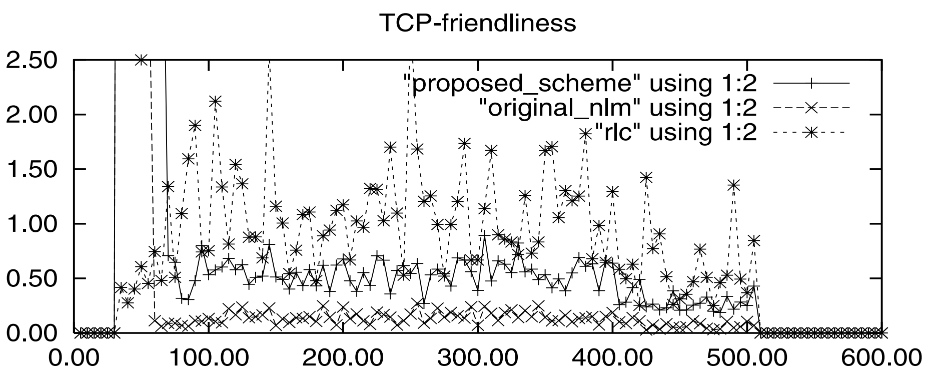

Fig. 3. Scenario 1: two FTP and two VBR sessions

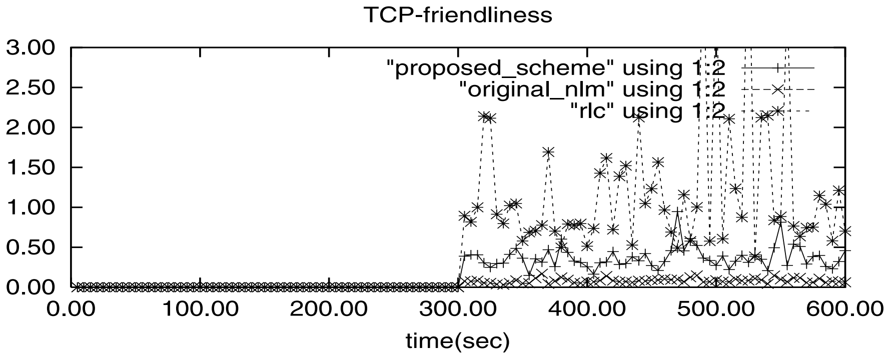

Fig. 4. Scenario 2: one FTP and three VBR sessions

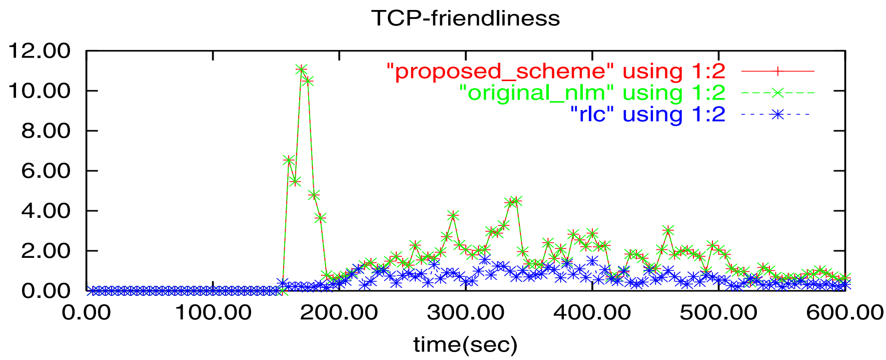

Fig. 5. Scenario 3: three FTP and one VBR sessions

work, NLM, and supports TCP-friendliness by allowing fine traffic control granularity. The performance results show that the proposed scheme provides better TCP-friendliness than the original NLM and RLC.

Acknowledgements

This work was supported in part by the National Research Laboratory Program funded by Ministry of Science and Technology, The Republic of Korea under Grant 2EG1900.

References

[1] S. McCanne, V. Jacobson, and M. Vetterli: Receiver-driven Layered Multicast. Proc. of ACM SIGCOMM96, August(1996) 117-130 202, 204

[2] X. Li, S. Paul, and M. Ammar: Layered Video Multicast with Retransmissions (LVMR): Evaluation of Hierarchicalrate Control. Proc. of IEEE Infocom'98, March/April (1998) 1062-1072 202, 203

[3] X. Li, S. Paul, and M. H. Ammar: Multi-Session Rate Control for Layered Video Multicast. Proc. of SPIE/ACM MMCN99. (1999) 202

[4] S. Bajaj, L. Breslau, and S.Shenker: Uniform versus priority dropping for layered video. Proc. of SIGCOMM98. (1998) 131-143 207

[5] Kyungran Kang, Dongman Lee, Hee Young Youn, Kilnam Chon: NLM: Network-based Multicast for Traffic Control of Heterogeneous Network. Computer Communications, Vol 24. Elsevier, Netherland (2001) 525-538 203, 204, 205

[6] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shaver.: FLID-DL: Congestion Control for Layered Multicast. Proc. of NGC 2000. (2000) 71-81 202, 204

[7] Lorenzo Vicisano, Jon Crowcroft: TCP-like Congestion Control for Layered Multicast Data Transfer. Proc. of IEEE Infocom'98. (1998) 996-1003 202, 203

[8] S. Bhattacharjee, K. L. Calvert, E. W. Zegura: Network Support for Multicast Video Distribution. Technical Document of Geogia Tech. GIT-CC-98/16. Atlanta (1998) 203, 204

[9] R. Gopalakrishnan, J. Griffioen, G. Hjalmtysson, C. Sreenan, S. Wen: A Simple Loss Differentiation Approach to Layered Multicast. Proc. of IEEE Infocom 2000. (2000) 461-469 203, 204

[10] J. Padhye, J. Kurose, D.Towseley, and R.Koodli,: A Model Based TCP-Friendly Rate Control Protocol. Proc. of NOSSDAV'99. (1999) 208

[11] The Mash Research Team. The ns network simulator. http://www-mash.cs.berkeley.edu/ns. 207