

Dynamic Flow Control using Packet Loss Rate and Round Trip Time

HO-KEUN SONG*, HA-SUNG KOO*

* Department of Computer and Information Science, Hanseo University, Korea

Abstract: - This paper proposes a dynamic end-to-end flow control algorithm that is more effective than previous methods considering either the Rate of Packet Loss (RPL) or Round Trip Time (RTT). According to our quantitative analysis, the proposed method performs 1.6 to 6 times better than the previous method in terms of the RPL. At the same time, the total number of transmitted packets is increased, which indicates that the proposed method can provide greater bandwidth capacity than previous methods.

Key-Words: *network flow control, packet loss, round trip time, bandwidth control,*

1 Introduction

A More and more applications and services need to transmit multimedia data over increasingly congested networks. One such application is the video teleconferencing system in which video data needs to be streamed to multiple recipients over continuously changing networks. This type of video teleconferencing system must be able to quickly adapt to changing network status. However, the existing Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) technologies do not provide sufficient bandwidth capacities for this system. To improve the video quality for end users, various end-to-end flow control methods have been proposed [1]-[5]. An end-to-end flow control method estimates the possible use of bandwidth in the current network status and then obtains the transmission rate that it controls.

One such method uses the Rate of Packet Loss (RPL) to determine the status of the current network[3]. RPL is a reliable measure for revealing congested network status, but the RPL method is problematic in that it increases network traffic by raising the transmission rate until packet loss is generated.

Another method makes use of the Round Trip Time (RTT) [5] to adjust the transmission rate of the network. RTT refers to the time it takes for a packet to move, round trip, between the sender and the recipient in the video teleconferencing system. The RTT can quickly and easily measure the current network status before packet loss occurs. However, when packet loss is caused by a poorly performing recipient system or is due to an overload of network traffic, the RTT method cannot cope with the rapid loss of packets.

This paper proposes a new dynamic flow control algorithm for teleconferencing systems. In our method, we categorize the network status into four distinct states: UNLOADED, LOADED, LIGHTCONGESTED, and CONGESTED. When the RPL is under normal conditions, the network status is defined, based on its average RTT, as being in one of the first three states. The average RTT is calculated by applying the smoothing factor to the currently reported RTT. However, if the RPL exceeds a predetermined critical level in the network, the network will be deemed to be in a fourth state.

Depending on the state of the network, we estimate possible uses of bandwidth and obtain suitable transmission rates of the network. For this purpose, two parameters for controlling the transmission rate, Frames Per Second (FPS) and Compression Quality of Frame (CQF), are introduced. FPS can either be increased or decreased according to the network status in certain ratio. CQF is classified into four cases : Quality_Maximum, Quality_High, Quality_Medium and Quality_Low. Each CQF case corresponds, respectively, to one of the four network states. For example, in a CONGESTED state, where packet loss is expected, FPS is reduced by given ratio and CQF is defined as Quality_Low.

In the experiments, we measured the performance of a video teleconferencing system to determine the values of some critical parameters and then selected the optimal threshold values. Then, we compared our method to other algorithms. The results confirm that our proposed method shows improved performance in the rate of packet loss and in the total number of transmitted packets over a given period of time.

2 Proposed Dynamic Flow Control Algorithm

2.1 Measuring Network Status

A pseudo-code for measuring network status whenever the RR packet of the RTCP arrives at the sender is shown in Fig. 1. As shown in lines (12) through (20) of Fig. 1, our algorithm categorizes the network status into four different states: UNLOADED, LOADED, LIGHTCONGESTED, and CONGESTED. To check the network status, both RPL and RTT are needed.

```

(1) // Packet loss-rate measurement
    loss_Rate = (1 - (Packet_cnt / Maximum_cnt)) * 100;

(2) // smoothed average RTT measurement
    Avg_RTT = (1 - 0.75) * Avg_RTT + 0.75 * Cur_RTT;

(3) if((Old_RTT / Cur_RTT) > 1.0)
(4) {
(5)     Loaded_Ratio += 0.025;
    Congested_Ratio = 0.5;
(6) }
(7) else
(8) {
(9)     Congested_Ratio -= 0.025;
    Loaded_Ratio = 0.2;
(10) }

(11) if(Cur_RTT > Max_RTT)
(12) {
(13)     Max_RTT = Cur_RTT;

(14) Congested_RTT = Congested_Ratio * Max_RTT;
(15) Loaded_RTT = Loaded_Ratio * Max_RTT;
(16) }

(17) if(loss_Rate >= 0 && loss_Rate < 4)
(18) {
(19)     if(Avg_RTT >= Loaded_RTT) // Unloaded state
(20)         NetworkState(UNLOADED);
(21)     else if(Avg_RTT < Loaded_RTT && Avg_RTT >= Congested_RTT) // Loaded state
(22)         NetworkState(LOADED);
(23)     else if(Avg_RTT < Congested_RTT) // Light congested state
(24)         NetworkState(LIGHTCONGESTED);
(25) }
(26) else if(loss_Rate >= 4 && loss_Rate < 100)
(27) {
(28)     NetworkState(CONGESTED); // Congested state
(29) }

```

Fig. 1 Proposed Network Status Measurement Algorithm

First of all, as shown in line (1), the rate of packet loss, *loss_Rate*, is calculated by applying the number of packets currently reported back as well as the maximum number of packets. In line (2), the average RTT, *Ave_RTT*, is estimated by applying the smoothing factor, 0.75, to the currently reported RTT, *Cur_RTT*. Since RTCP feedback information is normally transmitted approximately every 5 seconds, more weight on *Cur_RTT* than previously on *Ave_RTT* is needed to promptly reflect changes in network status. In lines (3) through (7) we introduce two important parameters of bandwidth ratio, *Load_Ratio* and *Congested_Ratio*. The initial values of the ratios, which were selected based on experimental data, are 0.2 and 0.5, respectively. To adjust the values based on the current network status, we compare *Cur_RTT* with *Old_RTT*, where the former is currently measured RTT and the latter is previously measured RTT. If *Cur_RTT* is greater than

Old_RTT, then the current network status has worsened. Therefore, *Congested_Ratio* is reduced by a predetermined unit ratio, 0.025, and *Loaded_Ratio* is assigned a value of 0.2. Otherwise, *Congested_Ratio* is assigned a value of 0.5 and *Loaded_Ratio* is increased by the unit ratio. Lines (8) through (11) are the code for maintaining RTT values within a certain range to reduce packet delay. First, lines (8) and (9) continuously update the maximum RTT value, *Max_RTT*, from all previous RTT values. Then, line (10) and (11) set the two critical values, *Loaded_RTT* and *Congested_RTT*, applying the above two bandwidth ratios to the *Max_RTT*. These two values have significant influence on the performance of a video teleconferencing system. As the *Loaded_Ratio* value increases, the transmission rate increases until the *Ave_RTT* value reaches the *Loaded_RTT* value and more network bandwidth is used. On the contrary, as the difference between *Loaded_RTT* and *Congested_RTT* increases, *Ave_RTT* is maintained within the two critical values and thus the transmission rate is more stable. In lines (12) through (20), the network status is determined based on the rate of packet loss and the average RTT. If the *loss_Rate* has a value ranging from 0% to 4%, the network status falls into one of three states, UNLOADED, LOADED, or LIGHTCONGESTED, according to *Ave_RTT*, *Loaded_RTT*, and *Congested_RTT*, respectively. On the other hand, if *loss_Rate* exceeds a 4% range value and the packet loss was caused by a poorly performing recipient system or by network traffic overload, then the network is deemed to be in a CONGESTED state.

2.2 Adjustment of Transmission Rate

Depending on the status of the network, we estimate the most suitable transmission rate for the network. For this purpose, two parameters, FPS (Frames Per Second) and CQF (Compression Quality of Frame), are introduced to control the transmission rate, as shown in Fig. 2. In line (1), *Last_time* indicates the elapsed time since the previous transmission rate was determined. Therefore, a new transmission rate will be determined after a period of time equal to the *ave_RTT* has passed. If the network is in a LOADED state, CQF is set to *QUALITY_HIGH* and a new *threshold_fps* is calculated without changing the *Cur_fps*, as in lines (2) and (3). The *threshold_fps* comes under the influence of the *Cur_fps*. For a CONGESTED network state where packet loss is expected, *Cur_fps* is reduced geometrically at a ratio of 0.65 and CQF is defined as *QUALITY_LOW*, as in

lines (4) and (6). Then, in line (5), threshold_fps is lowered at a ratio of 0.75 to cautiously increase Cur_fps and to prevent packet loss in the following network state. In that manner, if the network state is LIGHTCONGESTED, the CQF is set to QUALITY_MEDIUM and Cur_fps is cut down at the ratio of 0.75 as in lines (7) and (8). Last of all, when network status is the UNLOADED state, the transmission rate can be efficiently adjusted by applying the TCP like two flow control mechanisms: the Slow-Start phase and Congest-Avoidance phase. In the Slow-Start phase of the TCP, the transmission rate is increased geometrically from a given unit ratio to cope with the best condition of the network.

```

#define DELTA_VALUE      0.25    // frame per second
(1) if(Avg_RTT >= Last_time)
{
    if(NetworkState == LOADED) // loaded state
    {
(2)      Threshold_fps = (1 - 0.25) * Threshold_fps + 0.25 * Cur_fps;
(3)      CompressionQuality(QUALITY_HIGH);
    }
    else if(NetworkState == CONGESTED) // congested state
    {
(4)      Cur_fps = max(Cur_fps * 0.65, MIN_FPS);
(5)      Threshold_fps = (1 - 0.25) * Threshold_fps;
(6)      CompressionQuality(QUALITY_LOW);
    }
    else if(NetworkState == LIGHTCONGESTED) // Lightcongested state
    {
(7)      Cur_fps = max(Cur_fps * 0.75, MIN_FPS);
(8)      CompressionQuality(QUALITY_MEDIUM);
    }
    else if(NetworkState == UNLOADED) // Unloaded State
    {
        if(Cur_fps < Threshold_fps) //slow-start phase
(9)          Cur_fps += DELTA_VALUE;
        else // Congestion-avoidance phase
(10)         Cur_fps += (1 / Cur_fps) * DELTA_VALUE * 0.25;
(11)         CompressionQuality(QUALITY_MAXIMUM);
    }
}

```

Fig. 2 Proposed Transmission Rate Adjustment Algorithm

However, in the Congest-Avoidance phase of the TCP, if the rate exceeds half the rate of the previous phase, then it is increased linearly in anticipation of congestion. In our proposed method, those two phases are determined by comparing Cur_fps with threshold_fps. When Cur_fps is less than threshold_fps, Cur_fps is increased by a predetermined DELTA_VALUE, as in line (9). If Cur_fps is greater than or equal to threshold_fps, as in line (10), Cur_fps is increased by a reciprocal of Cur_fps. Then in line (11), CQF is set to QUALITY_MAXIMUM in order to transmit high quality video data to the recipient.

3 Experiment and Results

In this section, we present experimental results of the proposed flow control algorithm. First, we measured the performance of a video teleconferencing system for different values of some critical parameters, and then we selected optimal threshold values. Finally, our method is compared to previous algorithms. The testing platform is a Pentium III PC running at 850MHz. The platform used a video conferencing tool [2], VIC 2.8, to measure the quantity of transmissions and packet loss on the conferencing network path. The experiment network path between Hanseo University and ThinkGate Corporation includes a total of 9 routers. To measure the actual bandwidth of the network between the two organizations, a program called PING was developed using the echo of ICMP (Internet Control Message Protocol) [6]. For our purposes, the PING's ICMP echo packets were generated every 5 seconds as in RTCP control.

3.1 Initial bandwidth ratio

In the proposed network status measurement, we introduced two parameters, Load_Ratio and Congested_Ratio, and a bandwidth control coefficient, 0.025, for bandwidth ratio control. To determine the initial values of the parameters, the system performance is checked against some values. As a result, if the Congested_Ratio has a small value and the difference of the two values decreases, our system shows a more sensitive reaction on the current bandwidth condition. Therefore, we can select the optimum values out of the given values. The selected Congested_Ratio and Load_Ratio are 0.5 and 0.2, respectively.

To make the bandwidth control more accurate, the two selected parameters are varied with some values of the bandwidth control coefficient. Then the system performance is checked once more. If the coefficient value becomes smaller, the estimated network status follows the current status more accurately. Therefore, the value of the bandwidth control coefficient is fixed as 0.025.

3.2 Quality levels of transmitted frames

In a teleconferencing system, the transmission rate can be controlled by adjusting not only the number of transmitted frames but also the quality of the frames. Therefore, we conducted an experiment on the quality level of the transmitted frames. In Intel Co.'s JPEG standard [7][8], the quality of an image is determined by a Q vector ranging from 1 to 100. The relationship between the Q vector and the quantity of transmitted

data frames with the lapse of time reveals that the difference in frame quantity corresponding to the adjacent predetermined Q vector values is about 500 bytes, and the difference of compression ratio is about 0.01. Therefore, if we need a transmission rate of 15 frames/second, then it can be cope with network bandwidth requirements ranging from 36.63 Kbytes/sec to 73.26Kbyte/sec. In addition, we can classify the CQF into four cases based on the network bandwidth condition.

3.3 Comparison with the previous methods

In this section, we compare the result of our proposed dynamic flow control algorithm with those of the previous methods, the RPL method [3] and the RTT method [5]. As mentioned, the RPL method is problematic in that it increases the network traffic by linearly raising the transmission rate until packet loss is generated, and then geometrically decreasing the rate. As a result, the transmission rate control repeats up-and-down and has nothing to do with the current network status. On the contrary, the RTT method can adjust the transmission rate until packet loss occurs. However, the RTT method cannot cope with the rapid packet loss because it disregards the RPL in determining network status. In proposed method using RTT as well as RPL, current network status is determined in detail (e.g. three defined states) using preceding, current and average RTT when the RPL is less than 4%. And it makes bandwidth control precise before serious packet loss occurs. If the RPL exceeds 4%, the network is deemed to be in a fourth state, i.e., CONGESTED state. Depending on the status of network, we estimate the most suitable transmission rate with not only FPS but also CQF. The FPS and the CQF values are predetermined via experiments.

Finally, Table 1 shows a quantitative analysis of those methods. According to the results, the proposed method performs 1.6 to 6 times better than the previous method in terms of the RPL. At the same time, the total number of transmitted packets is increased. This indicates that the proposed method can provide greater bandwidth capacity than any of the other methods.

Table 1. Comparison of Packet loss

method	Proposed	RTT	RPL
Total packets	13915	12072	11986
Loss packets	123	179	632
Ratio(%)	0.833	1.482	5.277

4 Conclusion

This paper proposed a new dynamic flow control algorithm for teleconferencing systems that has advantages over previous methods that consider only the RPL or RTT. The RPL is a good measure for revealing congested networks, and the RTT method can adjust the transmission rate until packet loss occurs. In the proposed method, when the RPL is under normal conditions, current network status falls into one of three defined states by using the averaged RTT. Furthermore initial bandwidth ratios are adjusted by comparing current RTT with previous RTT. And this allows precise bandwidth control before serious packet loss occurs. However, if the RPL exceeds the critical level of condition, the network is deemed to be in a fourth state. Depending on the network status, we estimate the possible use of the bandwidth and obtain the most suitable transmission rate of the network with not only FPS but also CQF. The FPS and the CQF values are predetermined via experiments. In the experiment, we measured the performance of a video teleconferencing system to determine the different values of some critical parameters, e.g. initial bandwidth ratios and the quality levels of frames. Then optimized threshold values are selected. Compared to flow control algorithms that use either RTT or RPL alone, our method is proven to show improved RPL and an increased total number of transmitted packets over a given period of time. According to our quantitative analysis, the proposed method performs 1.6 to 6 times better than the previous method. Future studies should consider more diverse network environments or conditions.

References:

- [1] Steven McCanne, Van Jacobson.: vic: a flexible framework for packet video. ACM Multimedia, Nov. (1995)
- [2] V. Jacobson.: Congestion Avoidance and Control. Proceedings of ACM SIGCOMM '88, Stanford, Aug. (1988)
- [3] Ingo Busse, Bernd Deffner, Henning schulzrinne.: Dynamic QoS Control of Multimedia Application based on RTP. <http://www.fokus.gmd.de/step/acontrol/ac.html>. May (1995)
- [4] S.K.Nah, D.H.Ko, J.S.Ahn, S.B.Kim.: Comparison of two dynamic flow control algorithms for teleconferencing system. SoftExpo 97, Dec. (1997) 381-390

- [5] K.D.Hwan, S.K.Nah, J.S.Ahn.: Network bandwidth estimation using RTCP for teleconferencing system. Proceedings of KISS autumn conference, Vol. 24, No. 3, Oct. (1997) 283 - 286,
- [6] J. Postel.: Internet Control Message Protocol - DAPRA Internet Program Protocol Specification. RFC792, Sep. (1981)
- [7] L. Berc, W. Fenner, R. Frederick, S. McCanne.: RTP Payload Format for JPEG-compressed Video. RFC 2035, Oct. (1996)
- [8]<http://developer.intel.com/software/products/perflib/ijl/>