An End-to-End Loss Discrimination Scheme for Multimedia Transmission over Wireless IP Networks

Hai-tao Zhao[1], Yu-ning Dong[1,2,*], and Yang Li[1]

[1] College of Communication and Information Engineering,
Nanjing University of Posts and Telecommunications, 210003 Nanjing, China
zhaohtmail@gmail.com, mm_liyang@126.com
[2] Jiangsu Provincial Key Lab of Computer Information Processing Technology,
Suzhou University, 215006 Suzhou, China
Tel.: +86-25-83492402; Fax: +86-25-83492420
dongyn@njupt.edu.cn

Abstract. As the rapid growth of wireless IP networks, wireless IP access networks have a lot of potential applications in a variety of fields in civilian and military environments. Many of these applications, such as realtime audio/video streaming, will require some form of end-to-end QoS assurance. In this paper, an algorithm WMPLD (Wireless Multimedia Packet Loss Discrimination) is proposed for multimedia transmission control over wired-wireless hybrid IP networks. The relationship between packet length and packet loss rate in the Gilbert wireless error model is investigated. Furthermore, the algorithm can detect the nature of packet losses by sending large and small packets alternately, and control the sending rate of nodes. In addition, by means of updating factor *K*, this algorithm can adapt to the changes of network states quickly. Simulation results show that, compared to previous algorithms, WMPLD algorithm can improve the networks throughput as well as reduce the congestion loss rate in various situations.

Keywords: Wireless IP networks, Congestion control, Wireless loss, Quality of Service.

1 Introduction

With the rapid development of the wireless networks technology, many applications of wireless service are based on heterogeneous wired-wireless hybrid IP networks. TCP considers all the packet losses as the symbol of network congestions, which causes the transmission rate overly reduced [1]. Moreover, the transmission rate does not really reach the capacity limit of the network, which causes the waste of network bandwidth.

To improve the performance of TCP in wireless networks, lots of works focusing on differentiating packet loss types (random channel error or congestion losses) have been reported. The TFRC-ASN [2] algorithm which modifies TFRC [1] by attaching extra sequence number to packet in the router, could distinguish the type of packet

* Corresponding author.

J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 718–728, 2010.

loss at the cost of software modification of routers. A congestion control algorithm based on ECN [3] judges by marking the IP packets and regards all the packet losses as congestion losses in the case of successive packet losses. The Biaz [4] algorithm and its modified version SPLD [5] use packet inter-arrival time to differentiate losses. The simulation shows that it works very well in a network where the last hop is a wireless and bottleneck link. However, they only evaluated their algorithm when a single flow was using the network in isolation. This algorithm may mistake random channel error as congestions when many flows compete in the bottleneck of wireless link. The Spike [6] algorithm defines two thresholds of statistical ROTT (Relative One-way Trip Time) to judge the types of packet losses. When the network state changes frequently, Spike performs badly. The ZigZag [7] algorithm measures one-way transmission time from the source to the destination and the number of the continuous lost packets to discriminate the cause of packet loss. Through simulation and laboratory implementation, they show that ZigZag improves performance in wired as well as wireless networks. However, when the quality of the wireless channel gets worse, ZigZag can not differentiate the cause of each packet loss accurately.

Based on AVTC [8], a video transmission control scheme was proposed in [10]. This scheme could distinguish the packet loss types, but sometimes it may not work stably. In order to better tackle the multimedia stream transmission control problem over hybrid wired-wireless IP networks more efficiently, we propose a new transmission control algorithm WMPLD (Wireless Multimedia Packet Loss Discrimination). The WMPLD algorithm distinguishes packet loss types on end-to-end basis without changing the intermediate nodes. By recording packet loss history and introducing an update factor, the algorithm adaptively tracks the network states. Simulation results show that the proposed algorithm outperforms previously reported end-to-end packet loss distinguishing algorithms.

2 Theoretical Foundation of the WMPLD Algorithm

In wireless IP environment, there are mainly two types of packet losses: One is caused by the random bit errors of the wireless channel, and the other is due to network congestions. Lee [11] proposed a novel method based on loss history to improve the performance of TCP over wireless networks. We modify the method and adapt it to our algorithm. The sender sends large and small packets alternately, and the receiver end estimates the number of large and small packets. When there are losses, the receiver end calculates the congestion loss rate and feeds it back to the sender. With the rate control mechanism of WMPLD, the sender will decrease the source's sending rate. Thus, (1) If this high loss rate is due to network congestions of the wired link, the loss will ease considerably within several RTT time; (2) If however, the loss rate doesn't show any obvious easement within this time interval, one may most probably regard present packet losses as most probably due to the bit errors (or other problems) of the wireless link.

It's known that in the wireless IP environment, the erroneous packet loss rates are correlative with the packet sizes, and the larger a packet is, the higher its loss probability will be [11]. On the other hand, the congestion packet losses in wired networks (drop-tail router) are generally independent of the packet size [12]. (Assume the buffer management used in routers is based on packet) Based on the observations above, one may send large and small packets alternately, and estimate current main cause of packet losses from their loss patterns. To do this, we define the following variables (at a time interval):

l^S : The number of lost small packets;

l^S_{cong} : The number of lost small packets due to congestions;

l^S_{rand} : The number of lost small packets due to bit errors;

l^L : The number of lost large packets;

l^L_{cong} : The number of lost large packets due to congestions;

l^L_{rand} : The number of lost large packets due to bit errors.

As discussed above, we know that the number of lost small packets is equal to the number of lost large packets due to congestions ($l^S_{cong} = l^L_{cong}$). We suppose that the ratio of large packet to small packet is α. The relationship between packet length and packet loss rate over the Gilbert wireless error model is positive relation. Under certain wireless channel conditions, we have $\alpha = l^L_{rand} \div l^S_{rand}$. Thus, we have equations below:

$$l^S = l^S_{cong} + l^S_{rand} \quad . \tag{1}$$

$$l^L = l^L_{cong} + l^L_{rand} = l^S_{cong} + \alpha \times l^S_{rand} \quad . \tag{2}$$

At the receiver end, one can calculate the values over a time interval. Thus, if α is known, one will get other values, such as $l^S_{cong}, l^S_{rand}, l^L_{cong}, l^L_{rand}$ by solving equations (1) and (2), namely, the congestion loss rate and random loss rate of large and small packets respectively. From these figures one can then acquire current congestion level of the wired networks and fading condition of the wireless link. The algorithm we proposed is based on this fundamental scheme. The behaviors of senders and receivers are introduced in Section 3.

3 Implementation Aspects of WMPLD

3.1 Senders Behavior

The algorithm of WMPLD will decrease the sender's sending rate when high packet loss rates are reported, expecting the loss rates to drop quickly (assuming the multimedia stream shares the network bandwidth with other TCP-friendly traffic). We propose the following sending strategy of the WMPLD algorithm at the sender end.

At the receiver end, if there are loss of large packets, one updates loss queue, namely, $l^L + 1$; otherwise, $l^S + 1$, and simultaneously calculates the current congestion level value of networks $P_{reduce} = (l^L_{cong} + l^S_{cong})/(l^L + l^S)$. When P_{reduce} is greater than a fixed threshold which is selected based on the network topology changes, we judge that current packet loss type is congestion loss and record the loss packets; otherwise current packet loss type is judged as random loss and we do not record the loss packets. Every certain time, the receiver end feeds back the loss rate (the ratio of the recorded loss packets to the total received packets) and round trip time to the sender.

The sender receives the packet of feedback and calculates the rate with the loss rate and round trip time based on TFRC algorithm. The calculated rate may not suit for the current networks state due to change of round trip time. If the calculated rate is more 1.25 times than the current send rate within this time interval, the packet loss rate rises to a high level and the sender controls rate based on MIMD (a, b) [9] congestion control algorithm. Otherwise the rate control mechanism of AVTC will decrease the source's sending rate.

3.2 Receivers Behavior

A loss queue of 32 packets is used in WMPLD packets. If the length of queue is too long, the response to the change of the network state will be slow, although the sending rate can be kept relatively stable. On the other hand, if the queue is too short, the sending rate may become fluctuating. Further experiments show that, most of the time, the packets congestion losses rates are relatively small and the queue cannot update quickly. In order to solve this problem, we introduce an update factor *K* to the WMPLD algorithm. When *K* continuous packets are received, we virtually insert one lost packet into queue, whose type is opposite of the current receiving packet. Inserting large and small packets alternately can reduce the response time to the change of network states, since the WMPLD algorithm can make correct judgment of current network states.

Following factors are considered in choosing *K:*

(1) *K* is an odd number; insert large and small packets alternately.
(2) $K > 1/PER_B$; the loss queue of bad state will not be influenced by *K*.
(3) *K* is not too big; the queue can adapt the variety of state in time.

As discussed above, *K* is a function of PER_B and the time *T* of keeping.

$K = f(PER_B, T)$

Subject to:

$$K \in [K_{\min}, K_{\max}] \text{ (}K\text{ is an odd number)} . \quad (3)$$

$$K_{\min} = \max[1/PER_B, 3] . \quad (4)$$

$$K_{\max} = aT * \frac{N}{W} . \quad (5)$$

$$a = \min[3W/NT, \pi_G/\pi_B] . \quad (6)$$

The terms PER_B and T represent packet loss rate at bad states and duration of state. a describes weight with K in duration of state. N denotes the number of transmitted packets per second and W represents the length of loss queue. π_G, π_B characterizes the ratio of good states or bad states to total time. In experiment, we can calculate the range K of by equation (3) and select a value of K. Because of $PER_B > 1/3$ in experiment, from equation (4), we consider 3 as a minimum value of K and 32 packets of the loss queue are all updated. In simulations, the loss queue size is 32(W=32), the number of transmitted packets per second is 25(N=25), $a = 2.3$ and duration of state is 10(T=10). We can calculate the value of K over parameters above by equations (5) and (6), and the result is 18. We select the odd numbers of the scope 3, 5, 9, 17 and ∞ which do not adopt updating factor as the value of K and simulate in wireless last hop (WLH). We set bandwidth of wired shared link 260Kbps. The simulation results are shown in Table 1. The table shows the throughput (TP) and congestion packet loss rate (CER) for various K values. RER denote wireless loss rate, PER denote total loss rate in the table. From Table I we can see that the WMPLD algorithm has higher throughput and lower packet loss rate when is K 5. $K = 5$ is used in the following.

Table 1. Relation of updating factor *K* and WMPLD algorithm performance

K	TP(bps)	TP (%)	RER	CER	PER
3	246.226k	94.70%	4.94%	0.37%	5.30%
5	248.276k	95.49%	4.40%	0.19%	4.59%
9	238.436k	91.71%	6.23%	0.65%	6.88%
17	237.608k	91.39%	7.52%	0.91%	8.43%
∞	243.328k	93.59%	4.70%	1.54%	6.24%

4 Simulation Results

To achieve these goals, we evaluate these algorithms via simulation using ns2 in this section. We study the performance and differentiation accuracy of the WMPLD under two main wireless network topologies: networks with wireless last-hop links and networks with wireless backbones. The wireless last-hop topology corresponds to cellular networks or satellite Direct-TV system, and the wireless backbone topology corresponds to high-bandwidth backbones or wireless LAN networks such as 802.11.

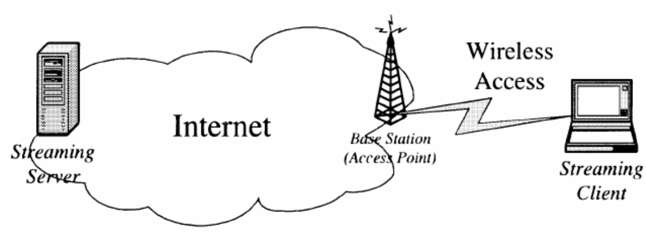

Fig. 1. A typical wireless multimedia streaming application scenario

We then study the WMPLD under various scenarios of competing traffic where multiple flows use the same WMPLD. Fig.1 shows a typical wireless multimedia streaming application scenario. The data stream is transmitted in wired-wireless networks with the wireless link as the last hop where each wireless link has a relatively constant bandwidth.

4.1 Wireless Last Hop

In the wireless last hop (WLH) topology (Fig.2), the last link to the receiver is a wireless link (dotted lines) with the bandwidth and delay parameter settings similar to those used in [7]. The other parts of the network are wired links. N traffic streams share a common wired link. If two or more streams competing for bandwidth at the common link, the bottleneck is the wired shared link. If there is only one traffic flow in the network, the wireless link is the bottleneck.

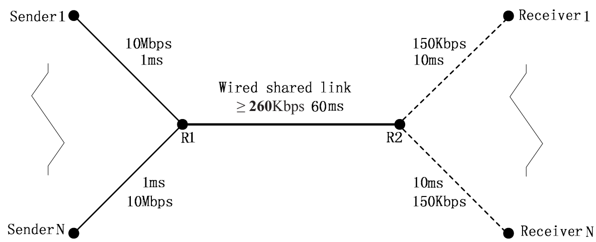

Fig. 2. A typical WLH simulation topology

The modified Gilbert wireless channel model [10] is used in our experiments to simulate the wireless loss patterns, with wireless channel bytes error rate = 0.0001, 0.0002, 0.0004 and 0.0008. The IEEE 802.11b is used at the wireless MAC/PHY layer. We compared the proposed algorithm WMPLD with TCP, TFRC, Spike, Biaz, SPLD and ZigZag algorithms in two major performance metrics: throughput and congestion packet loss rate, for one and eight flows respectively. The packet size is 762 bytes for TCP, TFRC, Spike, Biaz, SPLD and ZigZag. For a video coder that encodes at the rate of 25 frames/s, and a bit rate of 150kb/s, a frame on average would occupy 150K/25 = 6000 bits = 750 bytes. The size of 762 was chosen because it is twice 381 bytes, which is a specified packet size in the CDMA-2000 standard. The large packet of WMPLD is 1016 bytes and the small packet size of WMPLD is 508 bytes. So $\alpha = 2$ and the average packet size of WMPLD is 762 bytes. The packet loss history length is 32. Each simulation runs for 400 seconds.

4.1.1 For One Flow Only

In this case, there is only one of WMPLD, ZigZag, SPLD, Biaz, Spike, TFRC, TCP and OMNI flow running in the network from Sender1 to Receiver1. OMNI is an ideal algorithm used as a benchmark. The bandwidth of wired shared link is 260Kbps and the size of queue in R1, R2 router is 12 packets. So the wireless link between R2 and Receiver1 is the bottleneck. Because the bandwidth of wireless link is only 150Kbps ($<<$ 260Kbps), the congestion loss occurs in R2 router inevitably.

The simulation results are shown in Fig.3 and Fig.4. Fig.3 shows the throughput with bytes error rate for each type of flows. Fig.4 shows congestion packet loss rate with bytes error rate for each type of flows. From Fig.3, we see that the wireless environment becomes worse and worse with the increase of bytes error rate and the average throughput of all the flows has been decreased. TCP and TFRC had comparatively low throughput. They react to wireless losses as congestion losses, unduly reducing their sending rate. Spike and ZigZag are more conservative in that they classify some wireless losses as congestion losses. The Biaz and SPLD algorithms made few mistakes on wireless losses and were designed for this kind of topology. Because of this, they have the same slightly higher congestion loss, while Spike and ZigZag have less congestion since they misclassify more wireless losses and so reduce sending rate (see Fig. 4). WMPLD almost fully utilize the bottleneck bandwidth and misclassified no congestion losses. Throughput of WMPLD is obviously higher than other algorithms except OMNI, and its congestion packet loss rate is lower than that of others.

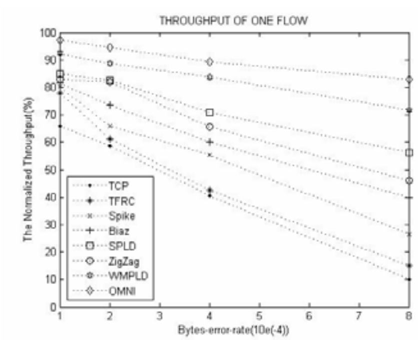

Fig. 3. Throughput of one flow

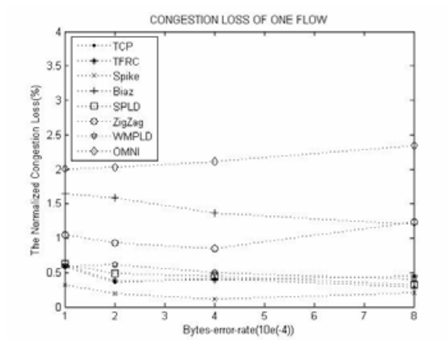

Fig. 4. Congestion packet loss rate of one flow

4.1.2 Eight Flows Competing

In this case, there are eight flows running in the network topology, such as WMPLD, Zigzag, SPLD, Biaz, Spike, TFRC, TCP and OMNI. We set the bandwidth of wired shared link 1040Kbps and the queue size in R1, R2 router is 18. Flow 1 starts at the 1st second from Sender1 to Receiver1, and Flow 2 starts at the 10th second from Sender2 to Receiver2, and Flow 3 starts at the 20th second from Sender3 to Receiver3, Flow 4 through Flow 8 are done in a similar way. So the eight streams compete for bandwidth at the common link, and congestion can happen both at the wired shared link as well as at the last wireless link. All flows stop at the 400th second. The bottleneck is now the wired shared link between R1 and R2 (routers).

The simulation results are shown in Fig.5 and Fig.6. From Fig.5 we see that the average throughput of TCP and TFRC decreases with bytes error rate. The reason is that flows will experience wireless error at wireless channels. With the increase in bytes error rate, it is likely that wireless loss will be deteriorated and unduly reduce their sending rate. Biaz and SPLD maintain high throughput regardless of the number of competing flows. However, they have high congestion loss because congestion losses at the shared bottleneck link become misclassified as wireless (see Fig. 6). The Spike and ZigZag scheme have low throughput and high congestion loss rate. Once a large ROTT is measured due to high buffer levels at both locations, it can no longer correctly gauge individual buffer levels. The high ROTT measured previously will make the scheme miss congestion loss in such case. The throughput of the WMPLD algorithm is higher than other algorithms except OMNI, but the bandwidth efficiency is also improved. As shown in Fig. 6, the congestion loss rate of the WMPLD algorithm is lower than others with the increase of bytes error rate.

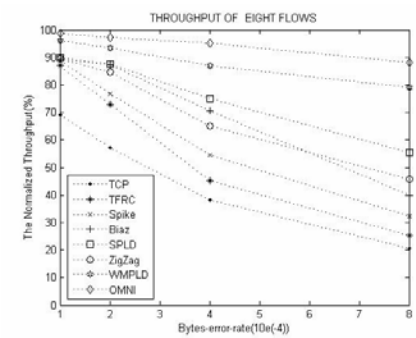

Fig. 5. Throughput of eight flows

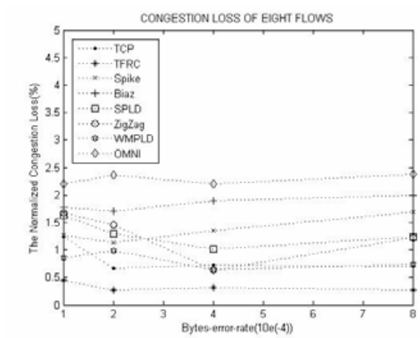

Fig. 6. Congestion packet loss rate of eight flows

4.2 Wireless Backbone

Next, we want to understand the performance of the differentiation algorithms on the wireless backbone (WB) topology, and to see how performance changes as the topology changes. As Fig.7 shows, WB topology used in simulation, where the route between R1 and R2 is a wireless link (dotted lines) with the bandwidth and delay parameter settings similar to those used in [7]. The other parts of the network are wired links. This topology simulates a scenario where LANs are connected with a high bandwidth wireless link. We see that the wireless shared link is the bottleneck if one or more streams competing for bandwidth at the common link.

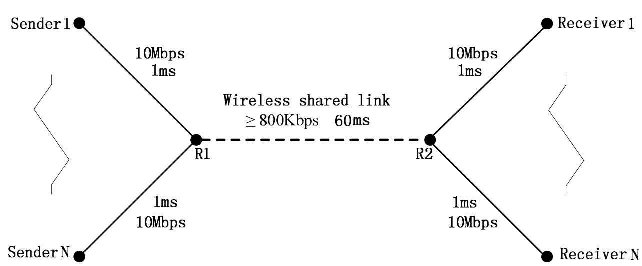

Fig. 7. Wireless backbone topology

The wireless loss pattern is the same as for WLH, with wireless channel bytes error rate = 0.00005, 0.0001, 0.0002, 0.00025 and 0.0004. We compared the proposed algorithm WMPLD with TCP, along with TFRC, ZigZag, Biaz and SPLD algorithms in two major performance metrics: throughput and congestion packet loss rate, for one and eight flows respectively. The packet size setting is similar to WLH. The packet loss history length is 32. Each simulation runs for 400 seconds.

4.2.1 For One Flow Only

The main difference between the performances of the algorithms when the flows operate in isolation on the two topologies is the difference in bottleneck bandwidth. There is only one of WMPLD, ZigZag, SPLD, Biaz, TFRC, TCP and OMNI flow running in the network from Sender1 to Receiver1. The bandwidth of wireless shared link is 800Kbps and the size of queue in R1, R2 router is 12 packets. The wireless shared link between R1 and R2 is the bottleneck.

Fig.8 shows the throughput vs. byte error rate for each type of flows. Fig.9 shows congestion packet loss rate vs. byte error rate for each type of flows. From Fig.8 and Fig.9, we see that TCP and TFRC have a much lower usage of the available bandwidth when it is 800 kb/s. The lower usage is due to the larger operating window size that comes with the higher bandwidth delay product, making the speed of the linear increase much slower than the speed of the multiplicative decrease caused by the high wireless loss. The performance of the WMPLD on the WB topology is, for the most part, similar to the WLH topology. However, ZigZag has a much lower throughput and higher congestion loss rate due to the larger window size at higher rates. Unlike the Biaz and SPLD algorithms, which also have a relatively lower throughput do not have any direct correlation with the buffer level. For the same reason as TCP and TFRC, it cannot recover the normal window size as quickly at the higher rate. The WMPLD algorithm has higher throughput and lower congestion packet loss rate than that of others.

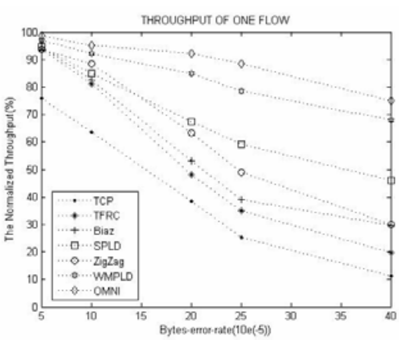

Fig. 8. Throughput of one flow

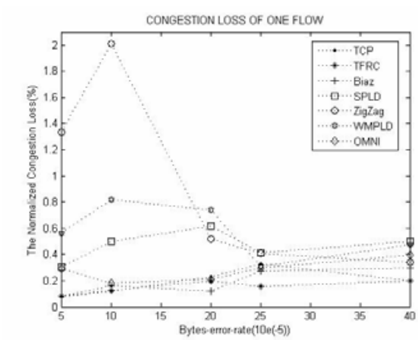

Fig. 9. Congestion packet loss rate of one flow

4.2.2 Eight Flows Competing

In this case, there are eight flows running in the network topology. We set the bandwidth of wireless shared link to 1600Kbps and the queue size in R1, R2 router is 18. Flow 1 starts at the 1st second from Sender1 to Receiver1, and Flow 2 starts at the

10th second from Sender2 to Receiver2, and Flow 3 starts at the 20th second from Sender3 to Receiver3, Flow 4 through Flow 8 are done in a similar way. All flows stop at the 400th second. The bottleneck is also the wireless shared link between R1 and R2 (routers).

The simulation results are shown in Fig.10 and Fig.11. As Fig.10 shows, one can see that TCP and TFRC reach throughput close to Biaz and maintain relatively low congestion loss. ZigZag has similar performance to SPLD. ZigZag is not able to return to the steady-state congestion window size quickly. However, ZigZag is able to fully use the available bandwidth when there are eight flows. The reason for this is partly the desynchronization effect of wireless errors, and partly lower average rate per flow. In this case, the throughput of the WMPLD algorithm is also higher than other algorithms except OMNI. From Fig.11 one can see that the congestion loss rate of the WMPLD algorithm is kept at a lower level. So WMPLD outperforms other algorithms in both throughput and congestion packet loss control. From these results, we conclude that WMPLD is the best performer under competition in the WB topology with consistently high throughput and low congestion loss rate.

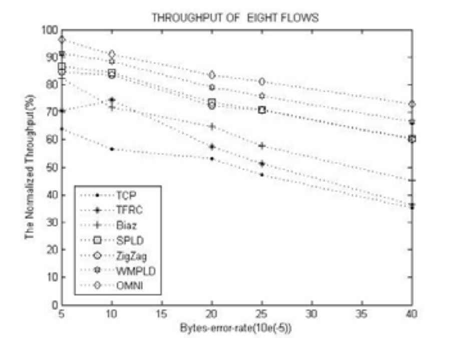

Fig. 10. Throughput of eight flows

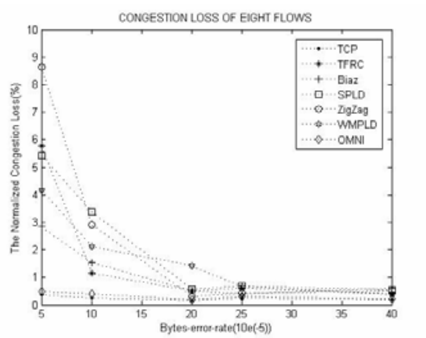

Fig. 11. Congestion packet loss rate of eight flows

5 Conclusion

In this paper, we have presented a novel algorithm WMPLD for multimedia streaming over wireless-wired IP networks. NS2 simulation results have shown that the proposed algorithm is effective in increasing packet delivery ratio and reducing packet loss rate. Simulation results also show that the proposed algorithm has better performance in the aspects of throughput and congestion packet loss control compared with previous algorithms in various situations.

The end to end solutions to the packet loss type discrimination depend on the measurement accuracy. The congestion loss is a global problem while the wireless loss is a special problem in wireless links. As the limited available information in upper layers, the control mechanisms of lower layers (for instance MAC layer) can be considered. So the across layer design is our direction of further studies.

References

1. Floyd, S., Handley, M., Widmer, J.P.: Equation-based congestion control for unicast applications. Computer Communication Review 30(4), 43–56 (2000)
2. Lee, H., Choi, C.H.: A loss discrimination scheme for TFRC in last hop wireless networks. In: 2007 IEEE Wireless Communications and Networking Conference, Kowloon, China, March 2007, pp. 3084–3088 (2007)
3. Lee, H.J., Byun, H.J., Lim, J.T.: TCP-friendly congestion control for streaming real-time applications over wireless networks. IET Communications 2(1), 159–163 (2008)
4. Biaz, S., Vaidya, N.H.: Discriminating congestion losses from wireless losses using inter-arrival times at the receiver. In: Proceedings of Application-Specific Systems and Software Engineering and Technology (ASSET 1999), Richardson, TX, March 1999, pp. 10–17 (1999)
5. Nguyen, N., Yang, E.H.: End-to-end loss discrimination for improved throughput performance in heterogeneous networks. In: 2006 3rd IEEE Consumer Communications and Networking Conference (CCNC 2006), Las Vegas, NV, January 2006, pp. 538–542 (2006)
6. Tobe, Y., Tamura, Y., Molano, A., Ghosh, S., Tokuda, H.: Achieving moderate fairness for UDP flows by path-status classification. In: 25th Annual IEEE Conference on Computer Network (LCN 2000), Tampa, FL, November 2000, pp. 252–261 (2000)
7. Song, C., Pamela, C.: Cosman, End-to-End Differentiation of Congestion and Wireless Losses. IEEE/ACM Transactions on Networking 11(5), 703–717 (2003)
8. Zhou, Z.W., Dong, Y.N.: Research on a transmission control algorithm of the real-time video stream AVTC over IP network. Computer Research and Development 41(5), 812–820 (2004) (in Chinese)
9. Sally, F., Mark, H., Jitendra, P.: A Comparison of Equation-Based and AIMD Congestion Control (May 2000), http://www.icir.org/tfrc/aimd.pdf
10. Tao, Z.Z.: Research on the QoS Control of the Real-Time Video Streaming Over Wireless Mobile Networks, M.S. Thesis, Nanjing University of Posts and Telecommunications, Nanjing, China (2006) (in Chinese)
11. Lee, C.L., Liu, C.F., Chen, Y.C.: On the use of loss history for performance improvement of TCP over wireless networks. IEICE Transactions on Communications E85-B(11), 2457–2467 (2002)
12. Floyd, S., Fall, K.: Router mechanisms to support end-to-end congestion control, LBL Technical Report (February 2004), ftp://ftp.ee.lbl.gob/papers/collapse.ps