International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.8 (2016), pp.257-270
http://dx.doi.org/10.14257/ijmue.2016.11.8.27

A Congestion Control Mechanism for Streaming Media Transmission Over Wireless Environment

Hancui Zhang [a], Shuyu Chen [a*], Qianyun Wang [a] and Jun Liu [b]

[a] *College of Software Engineering, Chongqing University, Chongqing, China*
[b] *College of Computer Science, Chongqing University, Chongqing, China*
zhc_813@126.com, netmobilab@cqu.edu.cn, dark0932@126.com, liujuncqcs@163.com

Abstract

TFRC (TCP-Friendly Rate Control Protocol) has been recognized as a relatively good mechanism for TCP friendly congestion control on cable network. However, TFRC is sensitive to the network status and cannot actually distinguish congestion losses and non-congestion losses well in wireless networks. The high error rates and large delay of wireless networks result in significant performance degradation and poor quality of the streaming media transmission service in transmission. To solve this problem, this paper proposes an Explicit Congestion Notification TCP (ECN-TCP) based TCP Friendly Wireless streaming Media Rate Control (TF-WMRC) congestion control mechanism. This mechanism improves TFRC and conjunctions with ECN and RED (Random Early Detection) methods, differentiating correctly congestion losses and wireless error losses and effectively alleviating the throughput degradation caused by wireless links. Performance evaluation of network efficiency, TCP friendliness and streaming media stability in NS2 simulation platform has shown that it is a good congestion control mechanism for streaming media transmission in wireless environment.

***Keywords**: TFRC, ECN, RED, wireless, streaming media transmission*

1. Introduction

The rapid development of 3G, 4G, WLAN has resulted in significant growth in streaming media, such as online audio and video services, Internet telephony, video conferencing and distance learning. The quality of these applications is more sensitive to packets delay and delay jitter than packets loss. Also the bakeoff and retransmission mechanisms in TCP can lead to undesirable end-to-end delays, violating the timeliness requirement of streaming media. Therefore, most of them use UDP as its transport layer protocol. And in order to keep the health of the Internet, the TCP Friendly Rate Control (TFRC), a popular rate control scheme over wired networks has been introduced, which is available to avoid congestion collapse and be fair to TCP flows.

However, in wireless networks, TFRC suffers significant performance degradation. Since the original design of TFRC is not according to the characteristic and demand of streaming media transmission, it merely regards the congestion packet as the only signal of congestion control, rather predict the possibility of congestion from the change of delay jitter. Whereas the wireless network usually has the characteristics of high error rates, low bandwidth, large delay and high mobile switching frequency, which makes the reliability of the wireless links relatively poor. So when TFRC is applied to the wireless network, the lack of distinguishing congestion losses and wireless error losses makes the over-adjustment of transmission rate and the throughput degradation, and also affects the quality of streaming media transmission service.

This paper proposes the ECN-TCP based TF-WMRC mechanism, where we first improve the TFRC mechanism by adding the threshold, the minimum rate and timer

ISSN: 1975-0080 IJMUE

mechanisms to make it have the congestion warning mechanism to determine whether the network is congestion by delay jitter. Then use the ECN in conjunction with the RED method to differentiate congestion loss and wireless error loss. Our simulation results show that, such mechanism can not only maintains higher TCP friendliness [1], but also improves the non-TCP flow's throughput.

The remainder of this paper is organized as follows. Section 2 describes the related work, including several previous enhancement of TFRC over wireless network. In Section 3, we give a brief introduction to the basic mechanism of TFRC. And then we do some work to improve it. In Section 4, we propose an ECN-TCP based TF-WMRC congestion mechanism. Design the ECN-TCP based TF-WMRC congestion control system in Section 5. And we evaluate the performance of our enhancement based on NS-2 simulation experiments in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

To optimize the performance of streaming media transmission congestion control mechanism over the wireless network, some schemes have been proposed.

Since the TFRC uses packet loss rate as the signal of network congestion, so how to predict the network congestion and adjust the sending rate in time is crucial to guarantee the streaming media real-time transmission. Qiang L *et al*. [2] provided a new algorithm of TFRC by using the jitter as a warning congestion signal. By estimating the packet jitter and detecting the hidden network congestion, the network transmission efficiency has been improved.

Zhou B *et al*. [3] proposed the TFRC Veno mechanism. It replaced the Reno equation in TFRC with the Veno equation and determined the packet loss by estimating the number of packets on the connection N over the bottleneck link. The equation N=X*(R-Rmin) is available to do this, where X is the sending rate of sender, R is the round-trip time and Rmin is the minimum of the measured round-trip time collected so far. According to the threshold α, if N≧α when a packet loss occur, the packet loss is regarded as network congestion loss, the sender need to reduce the sending rate, if not, the packet loss is wireless error loss, the sender does not need to adjust the sending rate, and usually α is 3.

Then he modified TFRC protocol based on Veno's state differentiator in [4]. This Veno's state differentiator uses congestion detection scheme in TCP Vegas to distinguish non-congestion and congestion losses. Likewise it uses N as a signal of network to determine whether it is in congestive state or not. The formula of N is $N = R_{act} * (RTT - RTT_{min})$. Where R_{act} is the actual sending rate, RTT is the smoothed round-trip time, and RTT_{min} is the minimum round-trip time. The value of R_{act} is $R_{act} = cwnd/RTT$. And the value of expected sending rate R_{exp} is $R_{exp} = cwnd/RTT_{min}$. Where the cwnd is the current congestion window size. The difference of the R_{act} and R_{exp} is $R_{dif} = R_{exp} - R_{act}$. As the RTT>$RTT_{min}$, there is bottleneck link where the packets accumulate. Then compared with threshold β, if N≥β, the network is in congestive state and the packet loss is congestion loss, otherwise it is in non-congestive and the packet loss is non-congestion loss. Here the β is 1.

Yang X *et al*. [5] improved the TFRC-Veno, it proposed a novel end-to-end congestion control algorithm TCP Eveno. The TCP Eveno introduced the "jitter factor" div= Rmin /R (0<div≦1) into the Veno equation to improve the congestion status differentiation mechanism and congestion control mechanism to achieve higher network resources utilization. The simulation of Eveno algorithm had shown that it can improve throughput and link utilization in wireless network, but the quality of streaming media service still had not been improved.

Based on the real-time transport control protocol (RTCP), Tao Z *et al*. proposed the idea of an improved TFRC (I-TFRC) [6] to improve the quality of service in mobile streaming media. It added the RTP proxy to the transmission session to calculate the packet loss rate and use the average RTCP packet size to improve the packet transmission interval.

Sunghee Lee *et al*. [7] suggested two enhancements of TFRC to increase the quality of multimedia streaming service. One is a RTT-fair bandwidth estimation method, and the other is Low Loss Slow-start (LLS) mechanism. The goal of it was to reduce packet losses during slow start and increase RTT-fairness. Simulations of the new slow start mechanism showed that it can reduce the packet losses and provide RTT-fairness.

Sathiaseelan A *et al*. [8] presented the Faster Restart method to accelerate slow start after idle and long data-limited periods to better support bursty applications with TFRC. It evaluated behavior both in terms of the application performance benefit and the implications on other network traffic that share an Internet bottleneck. And the result showed that the new methods improve the performance of bursty media.

Chen M and Zakhor A [9] put forward that using one TFRC connection in wireless streaming applications can results in underutilization of the wireless bandwidth, and then proposed the use of parallel TFRC flow MUL-TFRC. MUL-TFRC would reduce the number of TFRC flow as soon as Ravg–Rmin>β *Rmin, where Ravg is average round-trip time, Rmin is the minimum round-trip time, β is a constant, usually taken as 0.2. Otherwise, an increasing would happen. But it will cost more system resources such as much more memory would be needed for opening more connections and this would be companied with much more complex procedures.

In addition, Proxy-based TFRC [15] utilized proxy node to separate the TFRC link into two sections. One is from sender to proxy node based on TFRC protocol, another is from proxy node to receiver based on RTP protocol. WM-TFRC [16] used Access Point to measure the wireless error rate and reported to sender to protect the sender get the real packet lost rate. Though both of them could distinguish the congestion loss and wireless error loss, it’s still hard to deploy.

3. TFRC Congestion Control Mechanism

TFRC protocol uses the TCP throughput formula to adjust the sending rate of the sender. The basic idea was that once packet losses occur, the receiver will calculate the packet loss rate p as a congestion signal feedback to the sender. According to the returned information, the sender calculates the RTT. The optimal transmission rate was calculated with the following equation:

$$X\_Bps = \frac{s}{R\times\sqrt{\frac{2bp}{s}}+t\_{RTO}\times 3\sqrt{\frac{3bp}{8}}\times p\times(1+32\times p^2)} \tag{1}$$

Where X_Bps is the average sending rate in bytes/sec, s is the packet size in bytes, R is the round-trip time in second, p is the packet loss rate, $0 < p < 1$, t_RTO is the retransmitted timeout value[10] in second, b is the number of packets that are acknowledged by a received ACK.

From above we can see that TFRC is dependent on the loss event p and RTT, which can’t absorb the network fluctuation well and lacks of early warning mechanism of congestion. Also, it adjusts the sending rate only based on the feedback of receiver. Excessive regulation will affect the quality of streaming media service. So we do the following improvements to enhance the TFRC mechanism.

3.1. Stability Improvement Strategies

Let R_i be the i_{th} packet round-trip time, and let R_{i-1} be the $i-1_{th}$ packet round-trip time, then the difference ΔR_i calculated as follows:

$$\Delta R_i = |R_i - R_{i-1}| \tag{2}$$

Using EWMA (Exponentially Weighted Moving Average) method [11] to estimate smoothing of ΔR_i:

$$\mathrm{EWMA}(t) = \lambda Y(t) + (1-\lambda)EWMA(t-1) \tag{3}$$

Where $\mathrm{EWMA}(t)$ is the estimated value of time t, $Y(t)$ is the measurement value before time t, $\lambda(0 < \lambda < 1)$ is EWMA history weight coefficient, the more close to 1, the more higher weight of the past measurement value.

Then, let delay jitter smoothing estimated value G_i:

$$G_i = q * \Delta R_i + (1-q) * G_{i-1} \tag{4}$$

Where q is the weight constant, exhibiting certain ability to absorb some instantaneous abrupt change. With smaller q, the stability is stronger.

3.2. Early Warning Mechanism Improvement Strategies

When the network status is good, G_i is close to 0. The value of G_i rose as the network fluctuation occurs. Once it increases to a threshold, we believe that the network has the potential to generate congestion problems, then you need to adjust the sending rate to avoid the occurrence of congestion. The threshold value is not fixed, according to the RTT. The improved part of TFRC workflow as Figure 1:

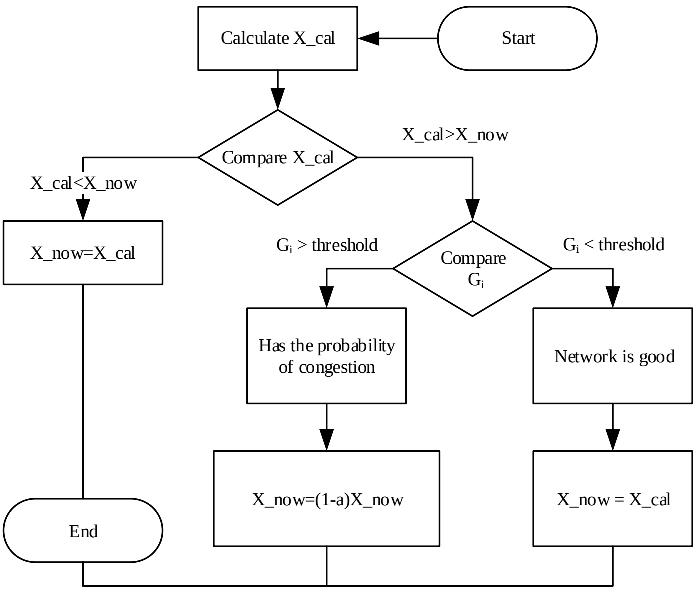


Figure 1. The improved Part of TFRC Workflow Chart

3.3. The Minimum Transmission Rate Strategies

Characteristics of streaming media transmission requires the minimum transmission rate to ensure no interruption, crash phenomenon. Therefore, this paper proposes the minimum transmission rate strategies which can guarantee the quality of streaming media service and would not lose the TCP friendliness.

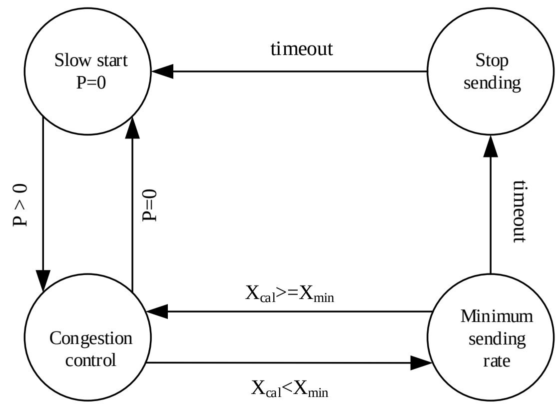


Figure 2. The Improved TFRC Work State Transition Diagram

Sender calculate the minimum rate of normal play of the streaming media X_{min} by the quality of streaming media files. As in Figure 2, once network congestion occurs, TFRC works, and if $X_{cal} < X_{min}$, then enter the minimum rate state. Then, the timer starts and sets the timeout. During the period, if the network congestion has eased and $X_{cal} > X_{min}$, then leave the minimum rate state, considering a X_{cal} as the sending rate. If not, enter the stop state. Timer is used to prevent from long time interrupt. Enter the slow start state as soon as the timeout.

4. ECN-TCP Based TF-WMRC Mechanism

Although the TFRC has been improved, it still regarded the packet loss as the signal of network congestion, cannot accurately distinguish congestion loss and wireless error loss in the wireless network, causing the sending rate calculation not accurate. Here, we use the ECN and RED method to join TFRC to differentiate the congestion loss and wireless error loss efficiently. Where ECN is the way to feedback early congestion to the sender and the RED in the router is used to detect the average length of queue to calculate the probability of ECN marked packet.

4.1. Explicit Congestion Notification Method

The basic method of ECN [13] is when a router occurs congestion in the early stage, no data will be discarded. What's more, the packet were marked as far as possible, reducing the network delay, and the sender can be informed of congestion with the feedback.

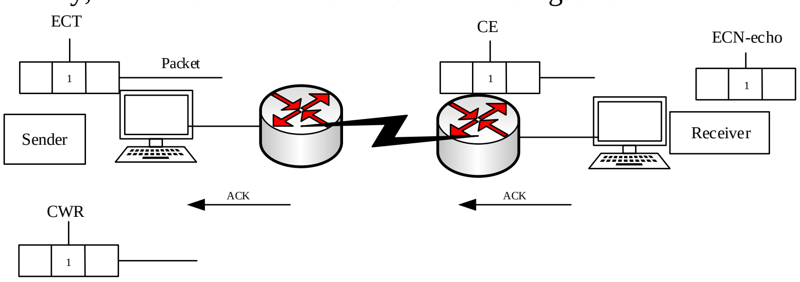


Figure 3. The Basic Scheme of Explicit Congestion Notification (ECN)

Figure 3 shows the basic scheme of ECN. First, the sender set the ECT (ECN-Capable Transport) to indicate whether it support ECN function. 1 means support, 0 means not. Then the router set CE (Congestion Experience) to inform receiver whether it is congestion. 1 means congestion, 0 means no. The receiver will always send ECE (ECN-Echo) to sender while it receive an ECN packet with CE, until sender set the CWR (Congestion Window Reduced) equals 1.

4.2. Random Early Detection Method

In addition, another important factor affecting the TCP throughput is the buffer management algorithm used in node routers in the network. Currently using Drop Tail [12] algorithm, the queue tail packet will be discarded when routing buffer overflow. Which will cause the TCP global synchronization, reducing or increasing the sending rate at the same time, and as a result the network state will change back and forth between have not been used effectively and congestion disaster.

RED is more fair than Drop Tail, it is equal to a plurality of data flow, judging the discarded packets in a way that is dependent with data flow bandwidth and transmission time [14]. And the traditional mechanism of RED flow is in Figure.4. Where q_{avg} is the average queue length, $q_{MinThres}$ is the minimum threshold of queue, $q_{MaxThres}$ is the maximum threshold.

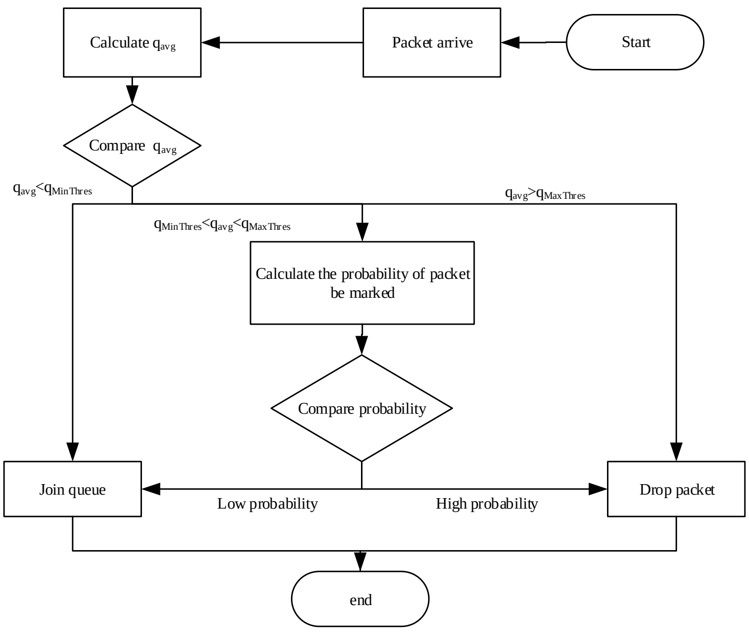


Figure 4. The Flow Chart of RED Algorithm

When $q_{avg} < q_{MinThres}$ all the packets will not be discarded, and when $q_{MinThres} < q_{avg} < q_{MaxThres}$ the relationship between discarding probability and q_{avg} was linear, and the packet will be discarded as soon as $q_{avg} > q_{MaxThres}$.

However, when the queue is not full, that is $q_{MinThres} < q_{avg} < q_{MaxThres}$, the router might mark the packet with ECN and inform the sender that the congestion loss occurred, resulting in overestimate the degree of network congestion and over-adjustment the transmission rate.

In our proposal, q_{avg} is introduced as an early congestion signal that calculated by the current length of queue $q_{current}$ and the queue length before the packet arrived:

$$q_{avg} = (1-\omega) \times q_{avg} + \omega \times q_{current} \tag{5}$$

Where ω is the weight factor, and let the value be 0.1 to avoid bursty traffic.

Then compare q_{avg} with q_{thres} to calculate the probability of ECN marked packet P_M[16]:

$$P_M = \begin{cases} 0, q_{avg} \le q_{thres} \\ 1, q_{avg} > q_{thres} \end{cases} \quad (6)$$

Set the q_{thres}values 0.9B, where B is the router buffer size. $P_M = 1$ as soon as the $q_{avg} > q_{thres}$, or else $P_M = 0$. And when the sender received the ECN-echo ACK, it can judge whether the congestion loss.

When $q_{avg} < q_{MinThres}$ all the packets will not be discarded, and when $q_{MinThres} < q_{avg} < q_{MaxThres}$a the relationship between discarding probability and q_{avg} was linear, and the packet will be discarded as soon as $q_{avg} > q_{MaxThres}$.

5. The Design of the ECN-TCP Based TF-WMRC Congestion System

5.1. The Sender

Each I intervals, sender update TCP-Friendly sending rate X_k based on the packet loss rate P_k and the time stamp information from feedback information. And use the stamp information to calculate the round-trip time R_k. Then use the P_k and R_k to get X_k.

$$X_k = \frac{s}{R_k\sqrt{\frac{2bp_k}{s}}+3t\_RTO\sqrt{\frac{3bp_k}{8}}\times p_k\times(1+32{p_k}^2)} \quad (7)$$

5.2. The Receiver

Receiver generates a RTCP receiver report periodically in each T interval and sends it to the sender. The design of the receiver is as Figure. 5.

Once receiver receives a packet, n_pkt is increased by 1, and if the packet is marked by ECN, then n_pkt and n_M are increased by 1. Get the t_s from send report and generate receiver report, including n_pkt, n_M, t_s and t_d to the sender. Where n_pkt is the total amount of packets, n_M is the number of ECN marked packets, t_s is the time stamp and t_d is the report generation time.

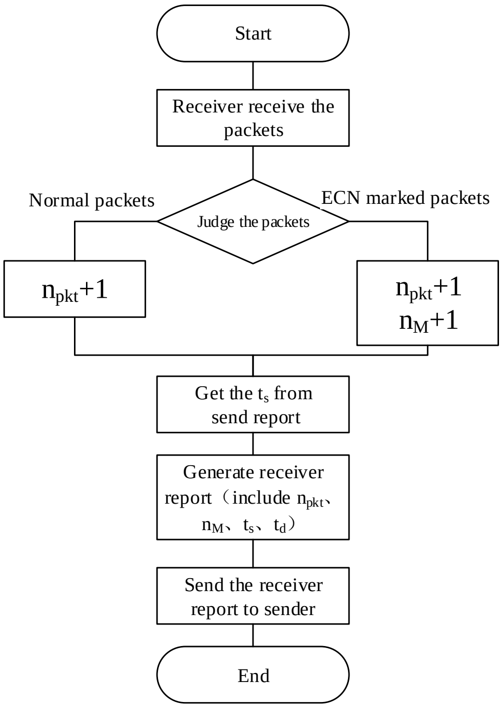


Figure 5. The Work Flow Chart of the Receiver of TF-WMRC

6. Performance Evaluation

In this section, we present our simulation results to validate our proposed ECN-TCP based TF-WMRC congestion control mechanism. We use NS-2 network test and simulation platform. And different aspects of performance such as network efficiency, TCP-friendliness, streaming media playback smoothness have been studied. The topology of our experiment on NS-2 is shown in Figure 6.

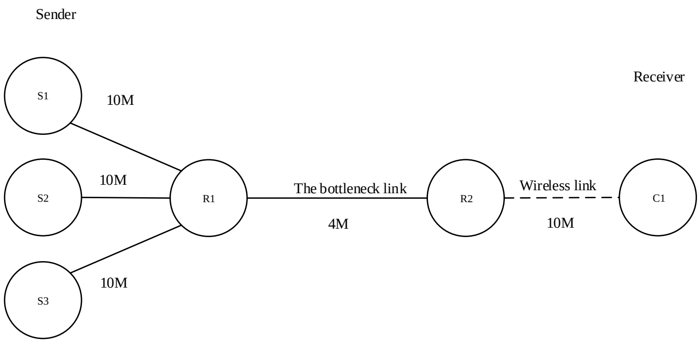


Figure 6. The Experimental Network Topology

The left side is sender, right is receiver. R1 and R2 represent two router support ECN and RED method, the link between them is the bottleneck link. Sender S1 run the TF-WMRC, S2 run the TFRC and S3 is the TCP flow in order to competition for resources.

6.1. Network Efficiency

In Figure 6, S1~C1 and S2~C1 establish TF-WMRC flow and TFRC flow, S3~C1 run TCP flow. The initial wireless error rate is set to 1% and increase gradually. Two experiments use the same network configuration and the test time is 100 seconds.

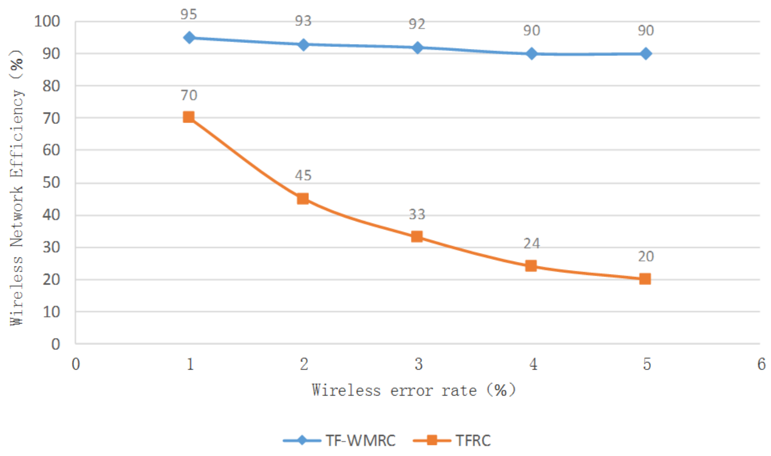


Figure 7. The Relationship of Wireless Bit Error Rate and Wireless Link Utilization

As shown in Figure 7, with the increase of wireless error rate, the network efficiency of TF-WMRC basically maintained above 90%. While TFRC is declining due to the lack of distinguish congestion loss and wireless error loss. And when the wireless error rate is increased to 5%, the network efficiency of TFRC is only 20%.

6.2. TCP Friendliness

Here we use the ration between TCP and non-TCP flow throughput to measure the TCP friendliness. The formula is:

$$F_r = \frac{T_{NOTTCP}}{T_{TCP}} \tag{8}$$

Where T_{NOTTCP} is the throughput of non-TCP flow, and $T_{TCP}$is the TCP flow. Usually one protocol has a better TCP friendliness as the value of F_r is between 0.7 and 1.5.

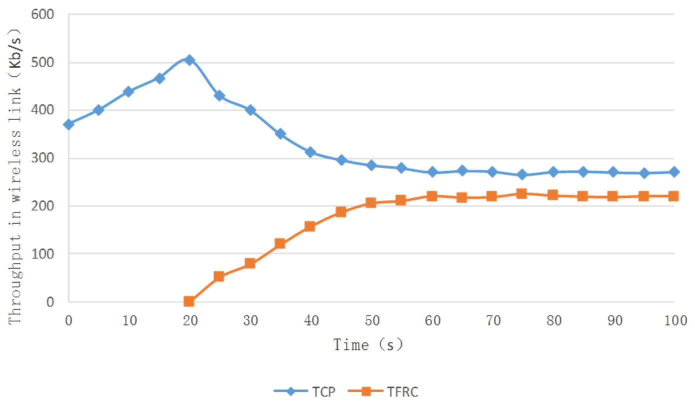


Figure 8. The Change of the Wireless Link Throughput Under the TFRC

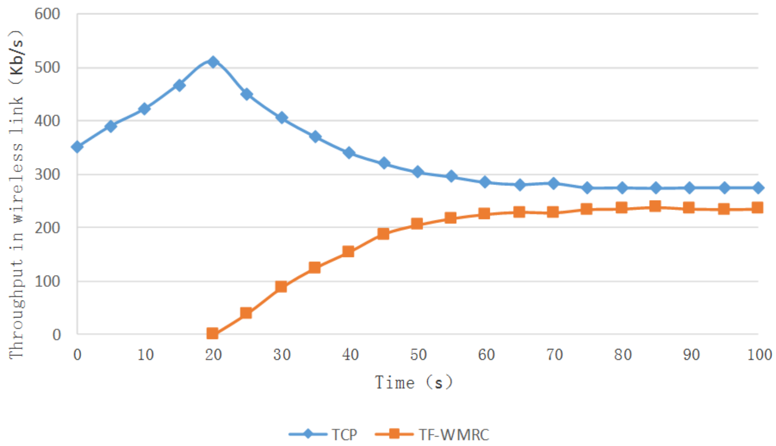


Figure 9. The Change of the Wireless Link Throughput Under the TF-WMRC

In Figure. 8, we compare S2~C1 TFRC flow with the S3~C1 TCP flow, and start TCP flow and TFRC flow in 0s and 20s. The total test time is 100s. Likewise, we compare S1~C1 TF-WMRC flow with S3~C1 TCP flow in Figure. 9. And the TCP flow and TF-WMRC flow start in 0s and 20s, and the test time is 100s.

From Figure.8 and Figure.9, we can see that when the TFRC and TF-WMRC start in 20s, the TCP flow throughput descends immediately. And then gradually turns to be stable in 50s. This trends indicate that both of our mechanism TF-WMRC and TFRC have a good characteristics of maintain TCP friendliness.

Then we get the throughput of TCP and non-TCP flow under TFRC and TF-WMRC in Table 1. In it, we can see that in the testing environment of TF-WMRC and TCP flow, it has 311.38 kb/s throughput of TCP flow, 194.44 kb/s throughput of Non-TCP flow on average. And in the testing of TFRC and TCP flow, it has 299.13kb/s throughput of TCP flow and 186.5kb/s throughput of Non-TCP flow. From the comparison between TF-WMRC and TFRC, it shows that our TF-WMRC mechanism has the higher throughput than TFRC in the same testing network environment.

Table 1. Average Throughput of TCP and Non-TCP Flow under TFRC and TF-WMRC

Algorithm	Average throughput of TCP flow(kb/s)	Average throughput of Non-TCP flow(kb/s)	Total throughput (kb/s)
TF-WMRC	311.38	194.44	505.82
TFRC	299.13	186.50	485.63

We show the TCP friendliness F_r of TF-WMRC and TFRC in Figure.10.

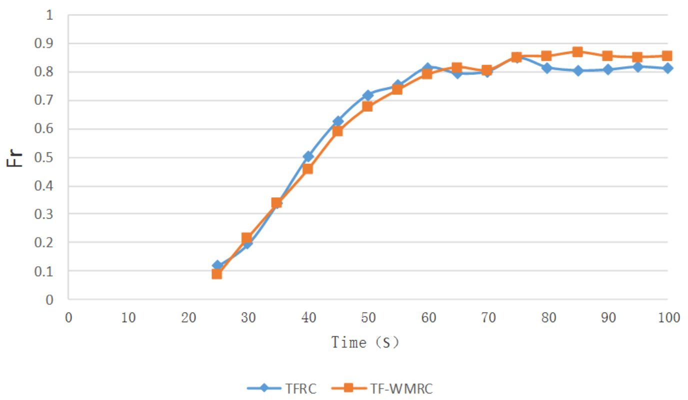


Figure 10. The Change of Fr Value under TFRC and TF-WMRC

According Figure.8 and Figure.9, the TFRC and TF-WMRC start in 20s, the competition of TCP flow and Non-TCP flow start at the same time. As shown in Figure.10, the F_r is increasing. And in 50s, it tends to be stable. After stability, the value of TF-WMRC is in 0.8~0.9, shows it has a better TCP friendliness.

6.3. Streaming Media Playback Smoothness

We build S1~C1 TFRC flow, S3 ~C1 TCP flow and set S2~C1 a burst data flow. Start TFRC and TCP in 0s, start the burst data flow in 30s and stop it in 70s, the total test time is 100s. We collect experimental data every 0.1s. The result shows in Figure. 11, in which the TFRC has a serious delay jitter as soon as the burst data flow starts in 30s and 70s, resulting in poor quality of streaming media playback.

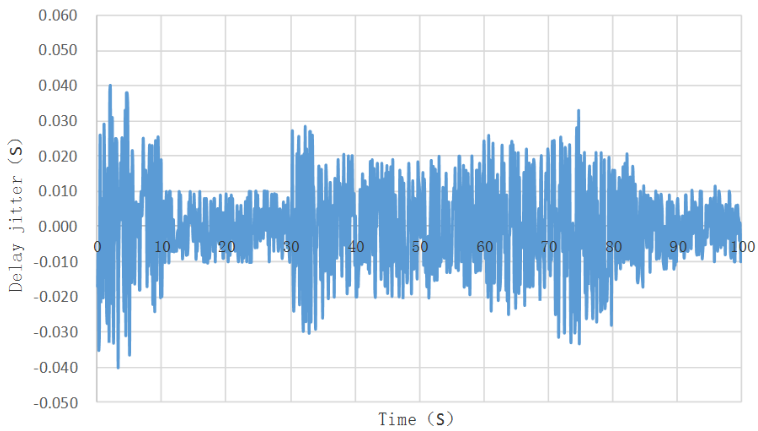


Figure 11. The Change of Delay Jitter under TFRC

As the same, we built S2~C1 TF-WMRC flow, S3~C1 TCP flow, and set S1~C1 a burst data flow. We start TF-WMRC flow and TCP flow in 0s, start the burst data flow in 30s and stop it in 70s, and the total test time is 100s. The experimental data collected in every 0.1s shows in Figure. 12, where the streaming media playback appears slight delay jitter when the burst data flow starts in 30s and 70s, maintaining a good quality of streaming media playback.

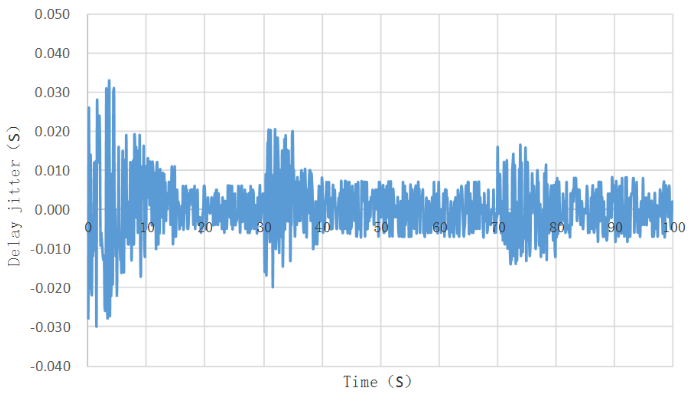


Figure 12. The Change of Delay Jitter under TF-WMRC

By contrast, we see that the improved TF-WMRC algorithm with smoothing has a better resistance to mutations impact and can remain the delay jitter in a stable and relatively small value.

7. Conclusion

Base on the research of TFRC, we improve its characteristics of stability, early warning of delay and service quality guarantee when it is applied to streaming media transmission by using EMWA method, the minimum transmission rate mechanism and timer. At the same time, according to the characteristics of higher wireless error loss in wireless network environment, we join ECN and RED method into TFRC and design an ECN-TCP based TF-WMRC congestion control system. The simulation results in NS-2 have shown the better performance.

Nevertheless, the proposed mechanism has its disadvantages. The first one is the idealistic assumptions such as this paper only consider the situation of unicast without considering the multicast. And another problem is that the implementation of ECN depends on the device, if the sender, receiver or router does not support ECN, the TF-WMRC based congestion control mechanism for streaming media transmission will be difficult to implement. So the future work we will do is to take the multicast into account and improve the TF-WMRC mechanism.

Acknowledgements

We are grateful to the editors and anonymous reviewers for their valuable comments on this paper.

The work of this paper is supported by National Natural Science Foundation of China (Grant No.61272399) and Research Fund of the Doctoral Program of Higher Education of China (Grant No.20110191110038).

References

[1] L. Cai, X. Shen, J. Pan, and J. W. Mark, “Performance analysis of tcp-friendly aimed algorithm for multimedia applications”, IEEE Transactions on Multimedia, vol. 7, **(2005)**, pp. 339-355.
[2] L. Qiang and Z. X. Rong, “A TCP-Friendly Congestion Control Algorithm and Its Application in Multimedia Data Transmission”, vol. 29, no. 6, **(2007)**.
[3] B. Zhou, C. P. Fu and V. O. K. Li, “TFRC Veno: An enhancement of TCP friendly rate control over wired/wireless networks”, Proceedings of the 15th IEEE International Conference on Network Protocols, Valencia, Spain, **(2007)**.
[4] B. Zhou, C. P. Fu and C. T. Lau, “An enhancement of TFRC over wireless networks”, IEEE Wireless Communications and Networking Conference, WCNC, Kowloon, China, **(2007)**.
[5] X. Yang, Y. Zhou and J. Jiang, “Eveno algorithm: Research on TCP Veno enhanced congestion control algorithm”, Measurement, Information and Control (ICMIC), Harbin, China, **(2013)**.

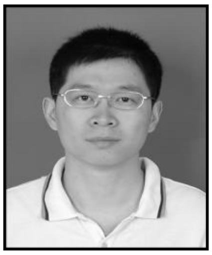
[6] Z. Tao, H. G. Ping and Y. F. Hua, "An improved TFRC mechanism of mobile streaming media based on RTCP", Proceedings International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), Changchun, China, **(2011)**.
[7] L. Sunghee, "Enhanced TFRC to improve the quality of multimedia streaming service", International Conference on ICT Convergence, **(2012)**, pp. 373-378.
[8] A. Sathiaseelan and G. Fairhurst, "TCP-Friendly Rate Control (TFRC) for bursty media flows", Computer Communications, vol. 34, no. 15, **(2011)**, pp. 1836-1847.
[9] M. Chen and A. Zakhor, "Multiple TFRC connections based rate control for wireless networks", Multimedia, IEEE Transactions on, vol. 8, no. 5, **(2006)**, pp. 1045-1062.
[10] M. Allman, V. Paxson and J. Chu, "Computing TCP's retransmission timer", RFC2988, **(2000)**.
[11] A. Ingolfsson and E. Sachs, "Stability and sensitivity of an EWMA controller", University of Alberta School of Business Research Paper, (2013-174), **(1992)**.
[12] R. Shorten, C. King and F. Wirth, "Modelling TCP congestion control dynamics in drop-tail environments", Automatica, vol. 43, no. 3, **(2007)**, pp. 441-449.
[13] K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP", RFC, vol. 2481, **(1999)**.
[14] S. Misra, B. J. Oommen, S. Yanamandra and M. S. Obaidat, "Random Early Detection for Congestion Avoidance in Wired Networks: A Discretized Pursuit Learning-Automata-Like Solution", **(2010)**.
[15] H. Hao, Z. Xiaoqing, W. Yao, P. Rong, Z. Jiang and B. Flavio, "Proxy-based multi-stream scalable video adaptation over wireless networks using subjective quality and rate models", IEEE Transactions on Multimedia, vol. 15, no. 7, **(2013)**, pp. 1638-1652.
[16] J. Y. Pyun, Y. Kim, K. H. Jang, J. A. Park and S. J. Ko, "Wireless Measurement Based Resource Allocation for QoS Provisioning over IEEE 802.11 Wireless LAN", IEEE Trans. On Consumer Electronics, vol. 49, **(2003)**, pp. 1103-1127.
[17] H. Jiawei, W. Jianxin and Y. Jin, "Improved TFRC Protocol Based on ECN Step Marking", Communications and Mobile Computing (CMC), vol. 36, **(2010)**.

Authors

Hancui Zhang, She received her B.S. degree in Chongqing University, P. R. China, at 2012. Currently she is a Ph.D. candidate in College of software engineering, at Chongqing University. Her current interests include streaming media flow control, high-speed communication networks, distributed system, Linux Kernel and cloud computing.

ShuYu Chen, He received his Ph.D. degree in Chongqing University, P. R. China, at 2001. Currently, he is a professor of College of Software Engineering at Chongqing University. His research interests include embedded Linux system, distributed systems, cloud computing, *etc.* He has published over 120 journal and conference papers in related research areas during recent years.

Qianyun Wang, She received her B.S. degree in Chongqing University, P. R. China, at 2013. Currently she is a Master Degree Candidate in College of Software engineering, at Chongqing University. Her current interests include high-speed communication networks, distributed systems, Linux Kernel.



Jun Liu, He received his B.S. degree in Southwest University, P. R. China, at 2001, and M.S. degree in Chongqing University, P. R. China, at 2009. Currently he is a Ph.D. candidate in College of Computer Science, at Chongqing University. His current interests include flash memory, information security, Linux Kernel and big data analytics.