International Journal of Hybrid Information Technology
Vol.7, No.5 (2014), pp.225-234
http://dx.doi.org/10.14257/ijhit.2014.7.5.21

Study on Westwood Algorithm Optimization

Junyou Huang and Huanling Li

Sichuan Information Technology College
scgyhjy@gmail.com

Abstract

In the scope of wireless network, traditional TCP congestion control mechanism exhibits low bandwidth utilization and poor reliability, so a congestion control protocol adaptable to the characteristics of wireless network is worthy of investigation. An intelligent gateway model with network boundary access control features has been established based on the analysis of TCP protocol. To optimize this model, loss differentiation mechanism was added and additive increasemechanism has been improved, and finally NS-2 emulation was performed with an optimized TCP Westwood algorithm (TCPW-I). It is demonstrated that TCPW-I can distinguish packet loss causes and postpone the occurrence of TCP congestion to effectively improve the throughput of TCP in wireless network circumstances, maintaining satisfying TCP friendliness and fairness of its own.

Keywords: *Wireless network, Gateway, Westwood, Congestion control mechanisms, Additive increase*

1. Introduction

In wired environment with high quality of links, data loss is mainly due to network congestion according to TCP congestion control mechanism. When data loss is detected from the destination end, congestion control will be functioned [1]. However, there are disadvantages such as high bit error rate, large latency, asymmetric bandwidth, limited mobility and energy, short-term flows existing in wireless networks, resulting in adaption problem of TCP congestion control to wireless network environment of relatively poor link quality [2].

TCPW (TCP Westwood) algorithm was proposed to solve the random packet loss problem in wireless networks based on Reno algorithm [3]. The idea is using real-time bandwidth estimation to control TCP congestion window and set ssthresh, measuring average rate of returned ACK on TCP connection in real time through the source end, to complete the effective estimation of end-to-end bandwidth. To a certain extent, intermediate router is not necessary for TCPW algorithm that fully complies with the "end to end" design principle and eliminates the impact of random error on network bandwidth utilization [4]. But in wireless network circumstances of high bit error rate and large latency, TCPW algorithm can not distinguish network congestion loss from random error packet loss, leading to frequent calls of network congestion control and hence network bandwidth utilization efficiency would be decreased. Meanwhile, TCPW and Reno algorithm share the identical additive increase mechanism at congestion avoidance stage, so the network has been saturated in additive increase phase and about to enter network congestion state. The same increment mechanism of TCPW and Reno will enlarge congestion window rapidly, which will induce the re-congestion of network [5].

ISSN: 1738-9968 IJHIT

In view of TCP protocol limitations existing in wireless network, we have proposed an intelligent gateway model with network boundary access control function and an improved TCPW algorithm (TCPW-I) was added to the model.

2. Intelligent Gateway Internal Model

It is critical of intelligent gateway to rapidly extract and forward a large amount of real-time packet status information in system. Its internal structural model is shown in Figure 1.

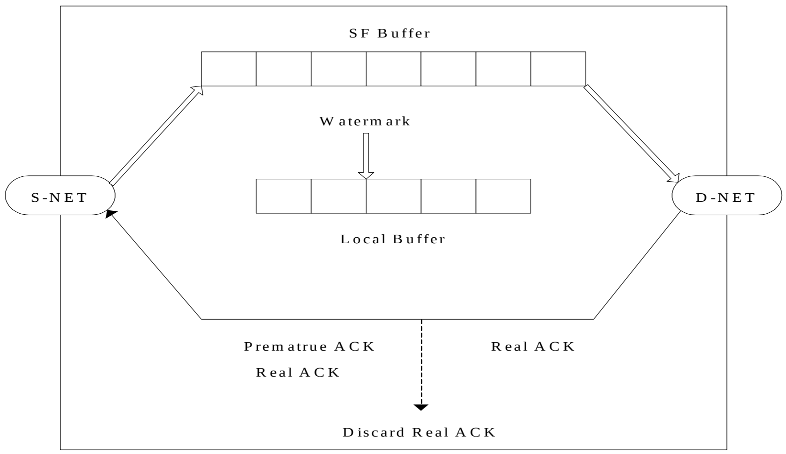

Figure 1. Internal Structural Model of Gateway

The whole network of this model was divided into source end (S-Net) and destination end (D-Net). Data packets and ACK (Acknowledgement) are mainly processed in internal gateway. When a packet is received from S-Net and forwarded to the target host in D-Net, the packet backup will be stored in the Local-buffer in response to the retransmission, and a Pre ACK will be sent to the source host instead of the response from destination host to the source host. The Pre ACK received by source host will trigger the sending of new data packet, which improves the transmission rate at source end. When a packet reaches the D-Net host, a Real ACK will be generated. Intelligent gateway decides whether to delete the backup of packet in the Local-buffer according to the returned Real ACK.

3. Improved tcpw-i Algorithm

3.1. Packet Loss Differentiation Mechanism

In order to distinguish between network congestion loss and wireless random error loss, two values of Expected and Actual are continuously measured at the sending end in TCPW-I algorithm.

$$Expected = cwnd / RTT_{\min} \quad (1)$$

$$Actual = cwnd / RTT_{c} \quad (2)$$

The cwnd in Equation (1) and (2) represents for the current congestion window; RTTmin is the minimum RTT measured; RTTc is the smoothed RTT.

$$RTT_c = sumRTT / cntRTT \quad (3)$$

After each ACK is received, TCPW-I will record RTT value in real time, and add it to the sumRTT, and meanwhile cntRTT is added by 1. Diff is given by

$$Diff = Expected - Actual \quad (4)$$

When RTT>RTTmin, it is indicated that a certain amount (N) of backlog packets exist in the bottleneck link cache. And equation (3) and (4) will be integrated to be equation (5)

$$RTT_c = RTT_{\min} + N / Actual \quad (5)$$

As can be seen, the additional latency of RTTc compared to RTTmin is caused by the packets backlog in cache.

$$N = Actual \times (RTT_c - RTT_{\min}) = Diff \times RTT_{\min} \quad (6)$$

TCPW-I algorithm considers the N as an indicator for judging whether the current network connection is in congestion phase. A threshold β with a reasonable value of 3 [6] is introduced in this algorithm, and the current network connection state will be distinguished by comparing the size of N and β . If N $< \beta$, operation is normal and all packet loss is considered to be caused by random errors in network. However, if N $\geq \beta$, it is at congestion state and all packet loss could be caused by network congestion. Therefore, TCPW-I algorithm can distinguish the reasons of link packet loss.

3.2. Congestion Control Mechanism

3.2.1. Slow Start: When cwnd<ssthresh, the improved algorithm and TCPW algorithm grow exponentially as Reno in slow start phase and maintain friendly relations with Reno.

3.2.2. Plus Growth: However, when cwnd>ssthresh, TCP connection converts into additive increase phase, and TCPW-I algorithm distinguishes the current connecting stage based on the amount of packets backlog - N.

If N$<\beta$, the sequence cwnd=cwnd+1/cwnd will be conducted at the arrival of each new ACK, which is consistent with TCPW algorithm.

If N$\geq \beta$, cwnd=cwnd+1/cwnd will be run at the arrival of every two new ACKs.

The throughput of TCP connection can be obtained from cwnd/RTT. If there are no backlog packets in queue cache, the increase of cwnd will not increase RTT but improve the throughput of connection [7]. When packets backlog exists in queue cache, an increasing cwnd will improve the amount of packets backlog, resulting in the increase of RTT, and the throughput of connection will not be improved. As a result, the increasing rate of cwnd will be lowered and cwnd will be a large number for a long time, postponing the appearance of network congestion.

3.2.3. Multiplicative Decrease: When three duplicate ACKs are received at sending end, indicating that random error or network congestion occurs in the link, TCPW-I is improved compared with TCPW algorithm: if N$<\beta$, packet loss may be caused by the random error in the link, and ssthresh = cwnd, which is different with TCPW algorithm. And if N$\geq \beta$, the packet transmission loss may be caused by network congestion, and cwnd=(BE*RTTmin) / PackerSize, which is consistent with TCPW algorithm.

In conclusion, TCPW-I can distinguish the cause of packet loss by comparing the size of N with β. When entering into additive increase phase, the state of connection can be classified with the comparison of the size of N and β . The procedure of TCPW-I algorithm is shown in Figure 2.

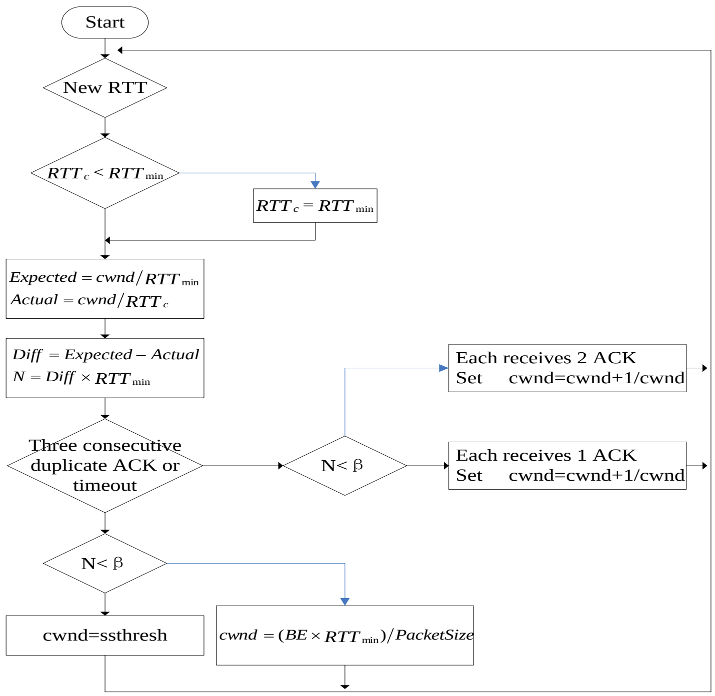

Figure 2. TCPW-I Algorithm Procedure

4. Emulation Test

Linux congestion control module was successfully operated in NS-2 with NS-2 simulation software combined with Linux TCP Agent of NS-2 TCP-Linux that can fulfill ACK processing, RTT sampling, SACK processing, rapid recovery, transmission latency *etc*., in Linux [8-10].

4.1. Algorithm Simulation in the Presence of only Random Errors

The simulation topology is shown in Figure 3 [11].

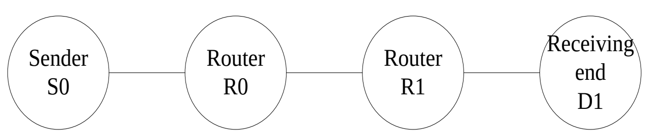

Figure 3. Simulation Topology of Single Link and Single Data Stream

All the link bandwidths were supposed to be 10 Mb, and link latency of 10 Ms. FTP data stream following TCP protocol was transmitted from S0 at the first second, passing by R0 and R1, to D1 for 100s. The random error percentage was assumed to be 10% of the link between R1 and D1 for the simulation of radio link. Data of S0 were sent according to TCPW and TCPW-I algorithm separately, and the routers adopted Drop Tail queue management algorithms. The simulation results are shown in Figure 4.

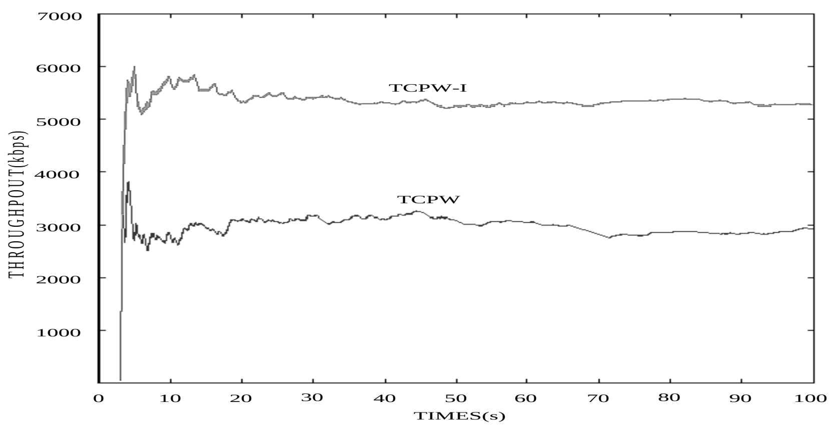

Figure 4. Instantaneous Throughputs of TCPW and TCPW-I with Evolution in the Presence of Only Random Errors

There is no bottleneck link in single link and single data flow simulation, the network throughput is dependent on the random error rate of link [12]. The results showed that the instantaneous average throughput of TCPW-I is approximately twice that of TCPW. The reason is attributed to the specific packet loss causes distinguishing mechanism of TCPW-I. The sending window at source end has not been reduced in the situation that there are only random errors, and thus the throughput performance is better than that of TCPW algorithm. The average throughput of network in Figure 5 is fitted on the basis of Figure 4 with the random error rate between R0 and R1 being 0.1%, 0.5%, 1% and 5% respectively and other parameters consistent. It is demonstrated that the average throughput of both algorithms decreases and the curve becomes steeper with the increase of link random error rate, indicating that in the single link of which network throughput is crucially dependent on random error rate, TCPW-I can distinguish between reasons for data loss, and its throughput is more advantageous than that of TCPW algorithm as link error rate increases.

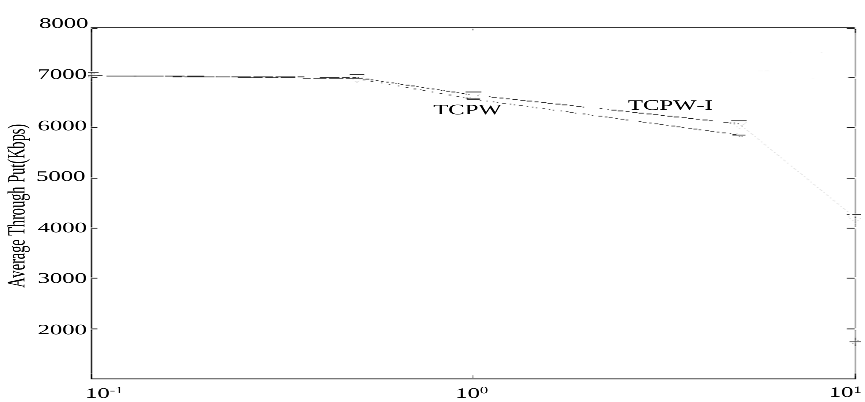

Figure 5. Average Throughput Versus Link Random Error Rate of TCPW and TCPW-I in the Presence of Only Random Errors

4.2. Algorithm Simulation in the Presence of Network Congestion and Random Errors

The link bandwidth between S0 - R0 and R1 - S1 was assumed to be 10 Mb, and link latency of 10 Ms. The link bandwidth between R0 and R1 was 1 Mb, and link latency of 40 Ms. The experimental procedure is the same as that of Ⅲ(1). The simulation results are presented in Figure 6.

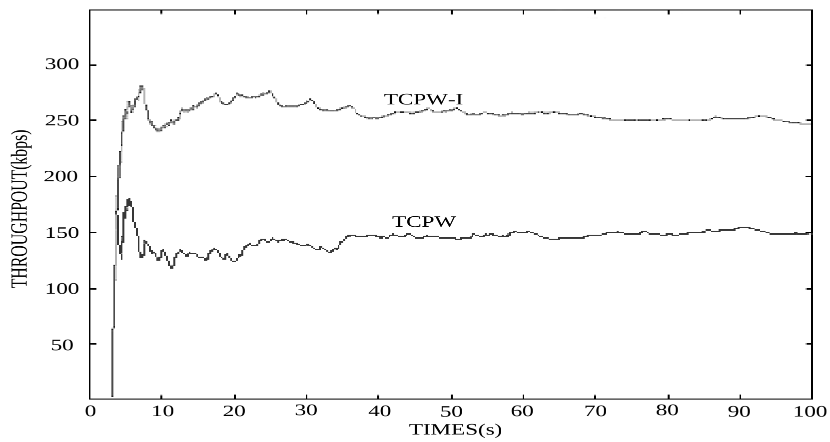

Figure 6. Instantaneous Throughputs of TCPW and TCPW-I with Evolution in the Presence of Network Congestion and Random Errors

The instantaneous average throughput of TCPW and TCPW-I algorithm fluctuates at approximate 130 and 250kbps respectively, which means that the throughput of improved TCPW-I algorithm is superior to that of TCPW algorithm in the presence of random error rates and network congestion. The average throughput of network in Figure 7 is fitted based on Figure 6 with the random error rate between R0 and R1 being 0.1%, 0.5%, 1% and 5% respectively and other parameters constant. It could be concluded that the average throughput of both algorithms reduces with the increment of link random error rate but the throughput of TCPW-I has been higher than that of TCPW algorithm. In the presence of congestion packet loss and random error packet loss, TCPW-I could improve loss differentiation mechanism and additive increase mechanism through its elevated throughput.

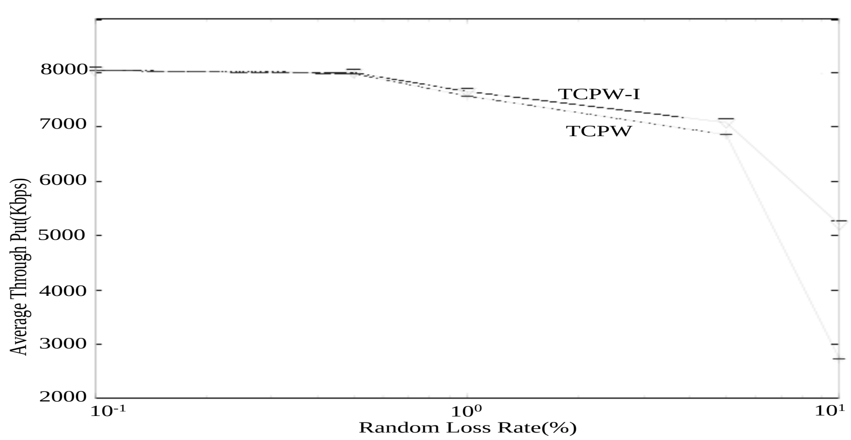

Figure 7. Average Throughputs Versus Link Random Error Rate of TCPW and TCPW-I in the Presence of Network Congestion and Random Errors

4.3. Hybrid Network Simulation with the Existence of Non-response Stream

Throughput the experiment, as shown in Figure 8, the link bandwidth between S0, S1 and R0; and between AP and D1 is 100 Mb and link latency of 2 Ms. Link bandwidth between AP (Access Point) linking wireless and wired access and R0 is of 3 Mb and the link latency is 10 Ms. AP and D0 were connected with an 802.11b of wireless link of a maximum transmission rate of 11 Mbps. In addition, S0 sent FTP data stream to D0 using TCP protocol, and S1 sent UDP data stream to D1 using CBR (Constants Bit Rate).

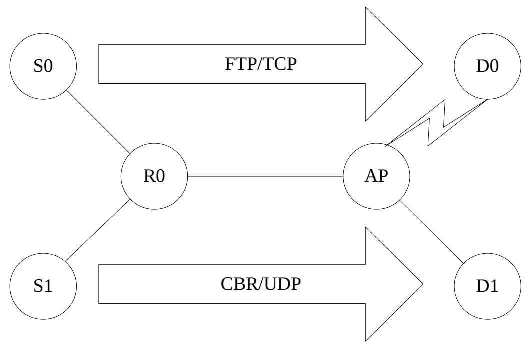

Figure 8. Topology of Wireless - Wired Hybrid Network in the Presence of Non-Response Stream

UDP stream was employed as non-responsive background data stream competing with TCP flows for bandwidth, to examine algorithm throughput under wireless - wired hybrid network environment. TCPW and TCPW-I at sender S0 were adopted and tested respectively. S0 continued sending TCP data stream to D0 during second 0-30, and S1 continued sending UDP data stream with transmission rate of 1 Mbps to D1 during second 0-10 and 20-30 separately. The simulation results are shown in Figure 9.

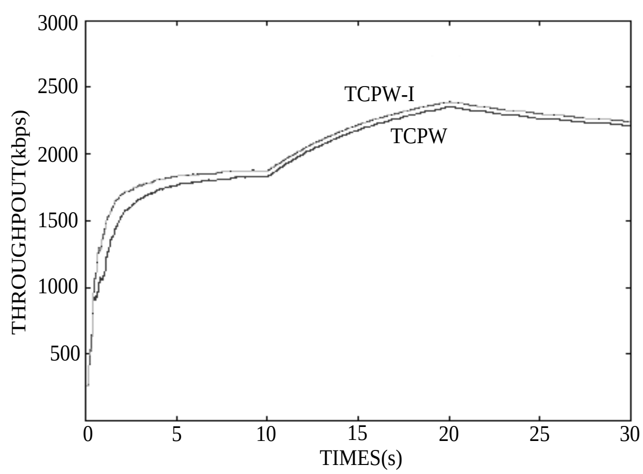

Figure 9. Instantaneous Throughputs of TCPW and TCPW-I with Evolution in the Presence of Non-Response Stream

At second 10 and 20, non-response stream emerged in bottleneck link of network to compete for bandwidth as a result of UDP flow. Both algorithms adjusted their throughputs promptly but the improved algorithm is always superior to TCPW algorithm in throughput performance.

Response the rate test, the experimental topology is shown in Figure 8 and parameters were set consistent with the previous experiments. The test was conducted twice, each for 10s. S0 continued sending TCP and improved data flow to D0 respectively, and S1 sent UDP stream with transmission rate of 1 Mbps and duration of 0.5 s to D1 at an interval of 0.5 s. It is indicated by the simulation results that both algorithms reacted rapidly and promptly the moment UDP streams flowed in and out (every 0.5 s). The judgment of TCPW-I on link bandwidth, especially residual bandwidth, is more accurate and rapid than TCPW.

5. Friendliness Simulation

The topology could be built on the basis of Figure 8 through minor correction that AP is changed into router R1 which links to D1 with wired connection [13-14]. A bottleneck link between R0 and R1 with a bandwidth of 1 Mb and latency of 40 Ms was installed, and the bandwidth of remaining links with latency of 10 Ms is commonly 10 Mb. Drop Tail algorithm was applied. Since second 1, S0 sent TCP stream employing TCPW and improved algorithm to D0, and S1 sent TCP stream using reno algorithm to D1 for 400 seconds. The results are shown in Figure 10.

Through the comparison of instantaneous throughputs, TCPW-I lowered the increasing rate of window in additive escalation phase, thus reducing the aggressiveness of the algorithm and improving friendliness.

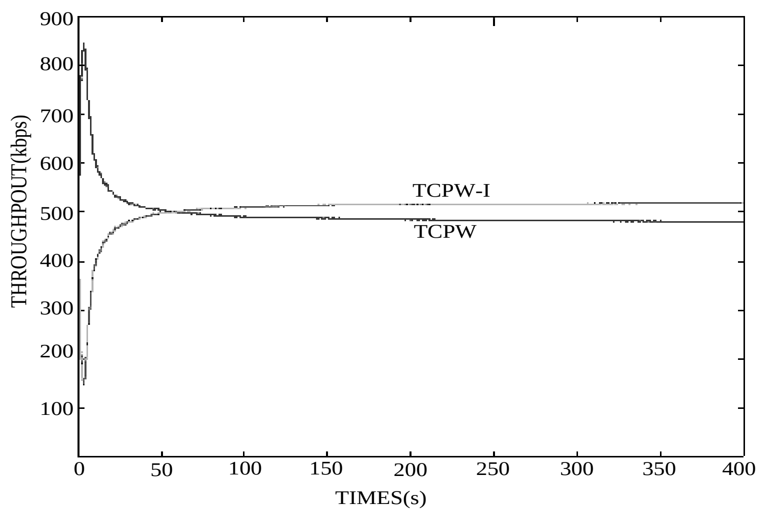

Figure 10. Friendliness Simulation Result

6. Fairness Simulation

Link bandwidth between 10 senders used in this test and router R0 is commonly 10 Mb with latency of 1 Ms. Link bandwidth between R0 and R1 is 10 Mb with link latency of 10 Ms, and the link bandwidth between R1 and 10 receiving ends D0-D10 is 10 Mb with latency of 1 Ms. 10 TCPW-I, TCPW and reno data streams were transmitted through the bottleneck link between R0 and R1 for simulation to test the fairness of each algorithm. Average bandwidth of each data stream is presented in Figure 11. All data streams of experiment lie in the abscissa, and vertical coordinate represents the average throughput derived by each data stream. Fairness values of these three kinds of algorithms were calculated according to the fairness index of Jain [8-9] to be TCPW-I fairness index of 0.989, reno of 0.994 and TCPW of 0.972, indicating that the same type of data flows should be fair among themselves as long as the adopted algorithm converges, without the consideration of global synchronization caused by Drop Tail algorithm.

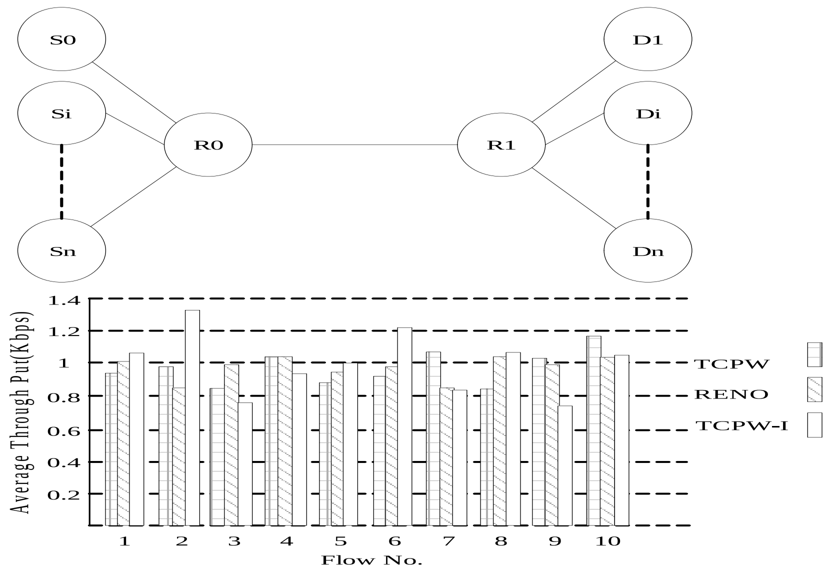

Figure 11. Topology and Experimental Results of Fairness Simulation

7. Conclusion

The intelligent gateway model with access control function has shown a high performance, constructed in heterogeneous network boundary, to address the problem that TCP protocol transmission performance is lowered due to large latency and high bit error rate etc. With the addition of TCPW-I algorithm into the intelligent gateway model, network effective throughput improve and bandwidth utilization were improved and the network transmission performance was ameliorated theoretically. The applications of intelligent gateway from theory to real environment have been promoted to a certain extent as well. But the bandwidth estimation mechanism of TCPW-I algorithm has not essentially changed with respect to westwood algorithm, so the further investigation and improvements of algorithm bandwidth estimation mechanism could be our next research focus aiming at more accurate bandwidth estimation. As to simulation, TCPW-I algorithm can be embedded into the commonly used Windows system, and thus a more complex network model in the presence of controllable routing variations will be built to test the effectiveness of TCPW-I algorithm in actual complex network.

References

[1] V. Machairas, A. Tsangrassoulis and K. Axarli, “Algorithms for optimization of building design: A review”, Renewable and Sustainable Energy Reviews, vol. 31, **(2014)**, pp. 101-112.
[2] S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda and G. Piscitelli, “Performance evaluation of Westwood+ TCP congestion control”, Performance Evaluation, vol. 55, no. 1–2, **(2004)**, pp. 93-111.
[3] A. Sakib, F. Mondal and B. Luqman, “Improving TCP performance over wired–wireless networks”, Computer Networks, vol. 51, no. 13, **(2007)**, pp. 3799–3811.

[4] S. Hagag and A. El-Sayed, "Enhanced TCP Westwood congestion avoidance mechanism (TCP WestwoodNew)", International Journal of Computer Application, vol. 45, no. 5, **(2012)**, pp. 6776-9071.
[5] K. Avrachenkov, U. Ayesta, J. Doncel and P. Jacko, "Congestion control of TCP flows in Internet routers by means of index policy", Computer Networks, vol. 57, no. 17, **(2013)**, pp. 3463-3478.
[6] D. Kliazovich, F. Granelli and D. Miorandi, "Logarithmic window increase for TCP Westwood+ for improvement in high speed, long distance networks", Computer Networks, vol. 52, no. 12, **(2008)**, pp. 2395-2410.
[7] C. Callegari, S. Giordano, M. Pagano and T. Pepe, Behavior analysis of TCP Linux variants, Computer Networks, vol. 56, no. 1, **(2012)**, pp. 462-476.
[8] M. A. Alnuem, "An Extended Review of Techniques for Enhancing TCP Performance", Journal of King Saud University - Computer and Information Sciences, vol. 22, **(2010)**, pp. 45-61.
[9] G. S. Alaa, G. D. Yacine, M. S. Sidi and A. Nazim, "Measurement of TCP computational and communication energy cost in MANETs", Pervasive and Mobile Computing, vol. 7, no. 1, **(2011)**, pp. 60-77.
[10] S. Ha and I. Rhee, "Taming the elephants: New TCP slow start", Computer Networks, vol. 55, no. 9, **(2011)**, pp. 2092-2110.
[11] S. Utsumi, S. Muhammad, S. Zabir and N. Shiratori, "TCP-Cherry: A new approach for TCP congestion control over satellite IP networks", Computer Communications, vol. 31, no. 10, **(2008)**, pp. 2541-2561.
[12] C. Caini, R. Firrincieli, D. Lacamera, T. de Cola, M. Marchese, C. Marcondes, M. Y. Sanadidi and M. Gerla, "Analysis of TCP live experiments on a real GEO satellite testbed", Performance Evaluation, vol. 66, no. 6, **(2009)**, pp. 287-300.
[13] S. Lee and K. Chung, "Buffer-driven adaptive video streaming with TCP-friendliness", Computer Communications, vol. 31, no. 10, **(2008)**, pp. 2621-2630.
[14] I. Aydin, J. Iyengar, P. Conrad, C. C. Shen and P. Amer, "Evaluating TCP-friendliness in light of Concurrent Multipath Transfer", Computer Networks. vol. 56, no. 7, **(2012)**, pp. 1867-1892.
[15] B. Hagelstein, M. Abolhasan, D. Franklin and F. Safaei, "Improving fairness in IEEE 802.11 networks using MAC layer opportunistic retransmission", Computer Networks, vol. 57, no. 17, **(2013)**, pp. 3410-3427.

Author

Junyou Huang was born in Sichuan, China. After receiving his bachelor's degree in applied electronic technologies in Chengdu Institute of Information Engineering, he has been a faculty member of Department of Electronic Engineering in Sichuan Information Technology College. He received his Master's degree in Electronic and Communication Engineering in University of Electronic Science and Technology of China. He is now an associate professor with major research interests in electronic and communication technologies.

