TSBCC: Time Series-Based Congestion Control Algorithm for Wireless Network

Weifeng Zhao[1, 2]
1. School of Computer Science and Technology, Tianjin University, Tianjin, China
2. School of Computer and Information Engineering, Tianjin Institute of Urban Construction, Tianjin, China
Email: zhaowf@tju.edu.cn

Xin Huang, Kai Shi
School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin, China
Email: shikai0229@163.com

Lei Zhang *
School of Computer Science and Technology, Tianjin University, Tianjin, China
Email: lzhang@tju.edu.cn

***Abstract*—To tackle the major problem of traditional TCP in wireless network, an end-to-end congestion control algorithm (TSBCC) is proposed based on time series. The algorithm creates time series consisting of the time interval of ACK, calculates the next ACK arrival time by the GARCH modal, analyzes the types of data packet loss, and sets congestion window size using different strategies according to different types of data packets loss. Simulation results show that the proposed mechanism can effectively recognize the reason of packet loss, and improve the throughput of the network.**

***Index Terms*—wireless network; congestion control; time series; GARCH**

I. Introduction

The Transmission Control Protocol (TCP) is one of the most important methods of end-to-end congestion control in the Internet [1]. About 90% of the Internet data are transmitted by TCP protocol. So TCP plays a crucial role on the performance of the network. TCP relies on the Additive Increase and Multiplicative Decrease (AIMD) strategy to adjust the sender congestion window, and it uses packet loss as an indication of network congestion. As a result, the traditional TCP performs well in low bandwidth wired networks with negligible link error rates. But, with the explosive growth of network applications and users, the network becomes more and more congested. Especially, with the popularity of wireless network, it becomes possible for people to download real-time steaming media data by using laptops or other mobile devices via Wireless Local Area Networks (WLAN). However, the performance of traditional TCP is not satisfactory in wireless network.

The traditional TCP has a poor performance in wireless network mainly because it cannot distinguish packet loss of dynamic wireless connection from packet loss due to the congestion. Traditional TCP reduces the congestion window based on packet loss. This method performs well in wired network which has a low packet error rate. However, in the wireless network with high packet error rate, traditional TCP cannot effectively distinguish the reason of packet loss. This will make the TCP sender reduce the congestion window mistakenly when there is packet loss due to the unstable wireless link. Thus, high performance of wireless network cannot be achieved. At the same time, the network bandwidth is not fully used.

In the past decades, researchers have been trying to improve the performance of TCP in wireless network, and many solutions have been proposed to deal with the problems above [2, 17]. On the one hand, according to the location where the algorithm is implemented, these solutions can be divided into three categories: sender-centric, intermediate-nodes-centric, and receiver-centric. On the other hand, according to the mechanisms of the solutions, these solutions can be divided into: optimizing existing congestion control algorithm [5] and new theory to redesign congestion control algorithm [6].

Most of the improved congestion control algorithms are implemented at the sender, TCP Westwood [8], which is a typical end-to-end congestion control protocol for wireless network. In this protocol, the sender calculates the available bandwidth based on the arrival rate of ACK, and accordingly adjusts the congestion window size. Westwood improves the performance of traditional TCP in the wireless network. The implementation does not need the help of either the receiver or the intermediate node. But it has a disadvantage over the traditional TCP, since the calculated available bandwidth may be larger than the real available bandwidth. TCP Vegas [9] estimates the congestion degree based on the relationship between the real-time RTT and two thresholds, and uses

* Corresponding author.

doi:10.4304/jnw.8.5.1058-1064

the available bandwidth to adjust congestion window. The advantage of TCP Vegas is that the congestion control mechanism only depends on the change of RTT. However, the conservative buffer-occupying strategy makes it unsuitable to coexist with traditional TCP protocols. In Ref. [10], the author proposed a cross-layer PI rate control algorithm at TCP senders. This algorithm calculates the maximum transmission rate based on the feedback queue length of the bottleneck router to make the queue length stable at a target queue length. But it needs modify the protocol of the sender and all the intermediate nodes, and requires bottleneck node sending timing information to the sender. This breaks the end-to-end semantics of TCP, and leads to the new protocol not widely used in the existing network.

At the intermediate nodes, congestion control algorithms improve network performance mainly by the management of queue length. In Ref. [11], a PI RED queue controller was proposed to improve the performance of RED algorithm. Comparing with the RED controller, PI RED not only improves the response speed, but also has better robustness on the queue length adjustment. In Ref. [6], the author proposed a PI rate control algorithm. Routers calculate the maximum data sending rate by PI controller, and send this maximum value to all the TCP senders which have packet flow through this router. This approach can make the router queue length stable at the target buffer, reduce congestion packet loss, and improve the anti-burst robustness of the network. But it requires each router monitoring the real-time information of all traffic flowing through the router, and calculating the sending rate of each sender according to the real-time queue length at the router. So it consumes the router resources seriously. It also needs to modify the cross-layer protocol, and breaks the TCP end-to-end semantics.

Instead of adopting traditional sender-centric congestion control, the Reception Control Protocol (RCP)[12] applies a receiver-centric control mechanism at the mobile receiver. Compared with TCP, RCP has the advantages of better loss recovery, congestion control, and power management. However, the authors did not provide a method that can directly make full use of this mechanism. Ref. [13] and Ref. [14] proposed a receiver-assisted TCP and a mobile host-centric TCP respectively. They make congestion control decisions based on the information retrieved from receiver or mobile host. However, these algorithms need to modify protocols of mobile host, which is not feasible for mobile hosts at some time.

In this paper, we proposed an end-to-end congestion control algorithm (TSBCC) based on time series to tackle the problem of traditional TCP in wireless network. According to the time intervals of ACK received by sender, the new algorithm forecasts the next ACK by GARCH model, compares the predicted results with actual results, and then takes different actions to adjust the congestion window. Compared with the traditional TCP, the new algorithm does not adjust the congestion window based on the packet loss information. So it can reduce the impact on the network performance caused by wireless packet loss. The simulation results show that, compared with the traditional congestion control algorithms, the proposed algorithm can improve the network utilization and throughput.

II. PROPOSED MODEL

A. Time Series Model

Time series is a set of numerical values which are arranged in chronological order. Time series analysis is to model the time series with curve fitting, parameter estimation with statistics analysis, and forecast the development trend according to the model. Time series analysis method is a quantitative method of dynamic data processing. It can reflect the dynamic changes of things, such as the trend changes, random variation. Time series analysis method is widely used in social life. The basic principle include: 1) the continuity of the development of things, using the historical data to predict the development trend of things. 2) The randomness of the development of things, things may be affected by random factors in the development.

Time series modeling has following steps: 1) sampling dynamic data of monitored system by measurement according to the time sequence. 2) Analyzing the obtained data, making the correlation diagram, calculating the autocorrelation function. 3) Selecting the stochastic model to curve fitting.

Time series analysis is mainly used to: 1) System description: According to the time series obtained by measurement, the system is depicted by the method of curve fitting. 2) Systems analysis: understanding the mechanism of time series by analysis. 3) Forecast: forecasting the next value of the time series by fitting the time series. 4) Decision and control: According to the result of forecast and real measurement, adjusting the input to maintain the system ideal value.

B. RTT Model

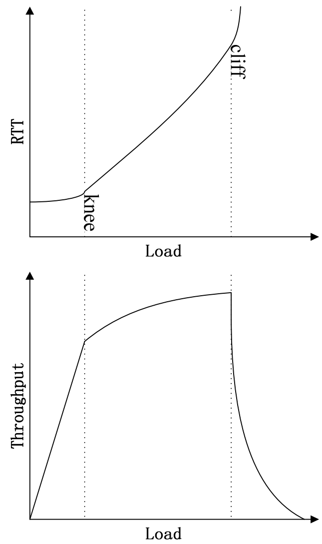

Figure 1. Relationship of Load, Throughput, and RTT

From the view of transport layer, the networking congestion status can be reflected by the relations of load, throughput and RTT, which are shown in Fig. 1 [15]. There are two points: knee point and cliff point. The network throughput and RTT have significant changes at the two points. The knee point means there is a queuing in the router, and the cliff point means that congestion occurs. So we can use RTT as the signal of network congestion, and the goal of congestion control is to maintain RTT between knee point and cliff point.

Ref. [16] and Ref. [18] have show that RTT obeys approximate normal distribution through simulation and experimentation.

C. ARCH Model

The ARCH (Autoregressive Conditional Heterosked-asticity) model uses the autoregressive form to describe the variance change based on the current information. Under the condition of known information, a noise obeys normal distribution, the mean of the normal distribution is zero, the variance changes with time, and the variance is the linear combination of square of the sampled noise value. It can be expressed as:

$$y_t = f(t, y_{t-1}, y_{t-2}, \cdots) + \varepsilon_t . \tag{1}$$

$$\varepsilon_t = \sigma_t e_t . \tag{2}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 . \tag{3}$$

where $f(t, y_{t-1}, y_{t-2}, \cdots)$ is the autoregressive model of $\{y_t\}$ and $e_t \sim$ iid N(0,1). That is a normal distribution with expectation of 0 and the variance of 1. Where $\omega > 0$, $\alpha_i \geq 0$, $\sum_{i=1}^{p} \alpha_i < 1$. If the disturbance of the conditional heteroscedasticity does not has autocorrela-tion, $\sum_{i=1}^{p} \alpha_i = 0$ and $\varepsilon_t = \sigma^2 = \alpha_0$. However, for most of p, the question of ARCH is that non-restricted estimator often disobey the condition of $\alpha_i \geq 0$.

D. GARCH Model

The GARCH (Generalized Autoregressive Conditional Heteroscedasticity) makes a further model to the variance of error on the basis of ARCH model. It includes the p order autocorrelation of Heteroscedasticity function. GARCH model can be expressed as:

$$y_t = f(t, y_{t-1}, y_{t-2}, \cdots) + \varepsilon_t . \tag{4}$$

$$\varepsilon_t = \sigma_t e_t . \tag{5}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 . \tag{6}$$

where, p is the order of the ARCH moving average, and q is the number of order of autocorrelation. $p > 0$, $\beta_i \geq 0$, $1 \leq i \leq p$. In order to make the conditional variance of GARCH(q, p) has a clear definition, all the coefficient of the ARCH(∞) model $\sigma_t^2 = \theta + \theta(L) u_t^2$ must be positive.

III. Algorithm

A. Network Bandwidth Prediction

In order to improve the performance of wireless networks, it is necessary to predict the available bandwidth of the wireless network, and adjust the size of the congestion window according to the forecast results. The method which is used to forecast available bandwidth in our new algorithm is summarized as follows:

Firstly, when the data is sent to the receiver, the sender can receive ACK continuously. The sender can calculate a time series which is composed by the interval of ACK. In this process, we need to pay attention to two questions: 1) when we sample ACK, we only record the ACK which packet is transmitted successfully without retransmission. We can decrease the error caused by the data packet retransmission. 2) The more number of sampling time series, more accurate the parameter will be. However, the more time it takes to sample ACK, the lower sensitivity of the window regulator will be. Therefore, according to the experimental results, we take thirteen as the number of time series.

Then, we predict the next data packet delay by the GARCH model:

$$r_t = \mu + \sum_{i=1}^{n} a_i r_{t-i} + \rho_t . \tag{7}$$

$$\rho_t = \gamma_t + \sigma_t . \tag{8}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{n} a_i r_{t-i} + \sum_{j=1}^{n} \beta_j \sigma_{t-j}^2 . \tag{9}$$

where r_t is prediction delay, σ_t^2 is the delay variance, and ρ_t is the residual error.

Finally, we can get the available bandwidth of the wireless network according to the formula:

$$Bandwidth = packet\ \ size / forcast\ \ delay . \tag{10}$$

Considering the implementation of the new protocol, we use two arrays (ACK_time[N] and ACK_Space[N]) to monitor and record the change of the arrival time and interval of ACKs respectively. In this paper, the number of samples N is 13. The left of the array represents the historical data, and the right of the array represents the new sampled data, as shown in Fig. 2.

old ← → new

12	11	10	9	...	3	2	1	0

Figure 2. Data Array

All the data of ACK_time[N] and ACK_Space[N] are initialized to be 0. The change of array can be executed by left-shifting the element and setting the right-end element of the array. When a new data is sampled, we do the following operations to update the data of the two arrays:

```
For (i = 12; i > 0;i++)
    ACK_time[i] = ACK_time[i-1];
ACK_time[0] = now
    // now is the arrival time of the new ACK
For (i = 12; i > 0;i++)
    ACK_Space [i] = ACK_Space[i-1];
ACK_Space[0] = ACK_time[1]- ACK_time[0];
```

Then, we predict the delay of the next data packet based on the data of array ACK_space[N] by GARCH model:

```
For(i = t-13;i <= t;i++)
    P[t] = w + a[i]*ACK_space[t-i] + b[i]*p[t-j];
For(i = t-13;i <= t;i++)
    ACK_space[i] = u + ACK_space[t-i] + p[t];
```

Where a[i] and b[i] is a random series respectively, $\sum_{i=1}^{13} a_i = 1$ and $\sum_{i=1}^{13} b_i = 1$.

B. Packet Loss Treatment Strategy

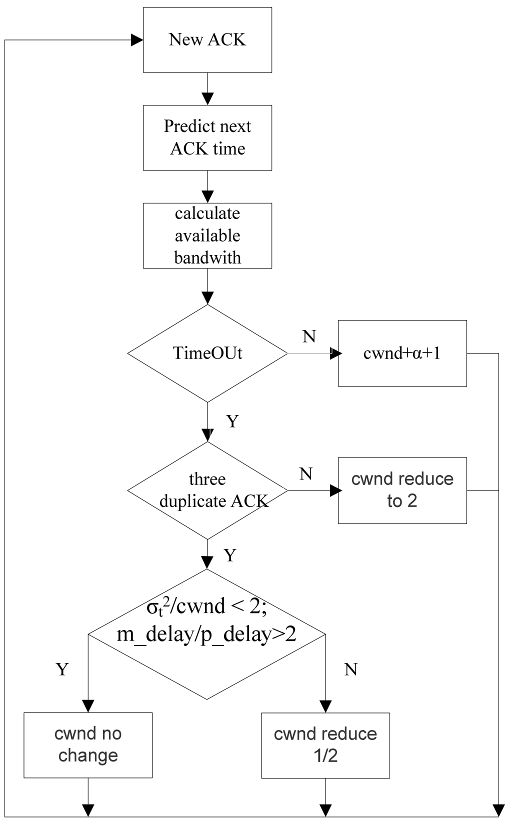

Figure 3. Flow Chart

Traditional TCP protocol uses packet loss as the signal of network congestion, and adjusts congestion window size by AIMD strategies. In this paper, when the sender gets a packet loss signal, the new algorithm takes different tactics to adjust congestion window size according to the relationships of timeout, duplicated ACK, and the forecast bandwidth. The principles to adjust the congestion window size are as the following principles:

1) When the sender receives three duplicate ACKs, the new algorithm does not halve the congestion window size. It takes different actions based on the bandwidth mean variance: If the ratio of mean variance to congestion window is less than 2 and the last measured ACK delay is as twice as predicted ACK delay, the lost data packet is retransmitted without adjusting the congestion window size. Otherwise, after retransmitting the lost packet, the congestion window will be halved.

2) When the sender gets timeout signal, it retransmits the lost packet, reduces the congestion window size to 2, and then restarts slow start.

3) When there is no packet loss signal, congestion window size will be set as follow:

$$cwnd = cwnd + (ACK\_space[i]-ACK\_space[i-1])/space[i-1] + 1;$$

The main flow chart of the algorithm is shown in Fig.3.

C. Two Phases of Congestion Control

Similar to the traditional TCP, the congestion window adjustment tactics of the new time series-based algorithm is divided into two phases: Slow-Start and Congestion Avoidance. But the implementation of the congestion avoidance phase is different from traditional TCP.

Slow start phase: When the connection is established, the congestion window size is initialized to be 2. And then, the traditional TCP mechanism to adjust the congestion window size exponentially is adopted. When the data is transmitted, the newest sequential 13 ACKs are recorded. When the congestion window size is larger than the slow start threshold, the algorithm turns into congestion avoidance phase.

Congestion avoidance phase: The next delay is calculated according to the time series by the GARCH model, and the size of the congestion window is set based on predicted results of bandwidth. During this process, if there is timeout or three duplicate ACKs, it will be dealt with on the methods discussed above.

D. Algorithm Analysis

Time series-based congestion avoidance algorithm is based on the randomness of the wireless network packet loss:

1) When network congestion leads to packet loss, as is shown in Fig. 1, the network throughput will decrease rapidly, and the round trip delay increases sharply. The phenomena can bring a big delay variance. We can regard the lost packet as congestion packet loss since there are both packet lost and a big delay variance. We need to reduce the congestion window size to relieve the congestion.

2) When the packet loss is caused by the wireless link error, the network is not congested; both the bandwidth and RTT are relatively stable, as shown in Fig. 1. These will bring a small delay variance. The fact is that packet

loss is due to the intermittent wireless link and it is not necessary to adjust the congestion window size.

3) When the network has enough available bandwidth, our new algorithm which modifies the congestion window size according to the predicted delay can timely increase the data sending rate to the available bandwidth. And the effect of network delay fluctuations on network performance can be decreased by using delay variance. On the other hand, when the data sending rate exceeds the available bandwidth, the new algorithm also can quickly reduce the data sending rate to avoid congestion.

In summary, we can distinguish the congestion packet loss from wireless link error packet loss by using the delay time series, avoid the mistakenly reduce of congestion window size when packet loss is due to the bad wireless link quality, and make a good use of network bandwidth.

V. Performance Evaluation

We evaluated the performance of the new algorithm compared with the traditional TCP Reno [19], TCP Westwood using Qualnet simulation software.

A. Scenarios and Parameters

We use three scenarios. In each scenario, we do experiment over 300 seconds, and every simulation is run ten times with different random seeds. We use FTP as the application layer protocol. The three scenarios are shown as following:

Linear scenario: As shown in Fig. 4, seven wireless networking nodes compose a linear topology to transfer one FTP data flow. Each wireless networking node is 200 meters away to ensure that only the adjacent two nodes can transfer packet directly. The wireless data rate is 11Mbps.

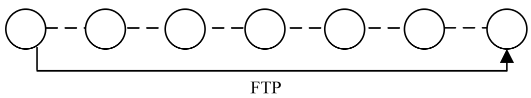

Figure 4. Linear Topology

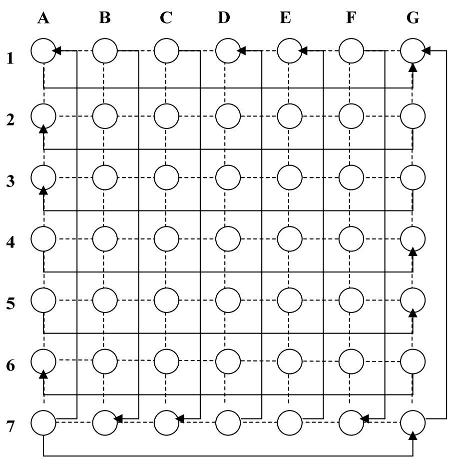

Figure 5. Grid Topology

Grid scenario: We use 49 nodes to create a grid topology shown in Fig. 5. In this topology, we created two, four, six, and eight FTP data flows which are in different directions. The wireless networking data rate is fixed at 11Mbps at MAC layer.

Dumbbell scenario: WLAN is an important topology to access internet. In dumbbell topology, shown as Fig. 6, five senders Si are connected with five receivers through router and Access Point (AP). Every connection transfers one long FTP flow. The senders are connected to the router by a wired link with a bandwidth of 100Mbps and several delays (10ms、15ms、20ms、25ms、30ms). The receivers communicate with AP through 54Mbps channel. The router is connected to AP by a wired link with a bandwidth of 1Gbps and delay of 50ms.

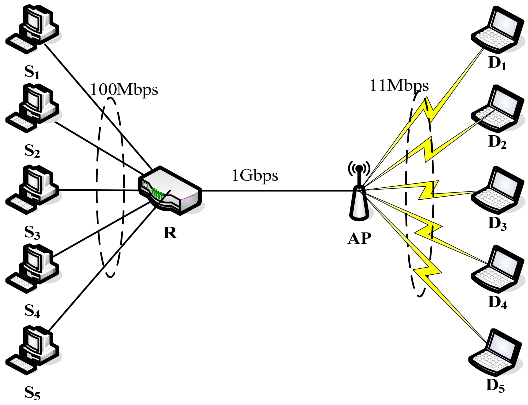

Figure 6. Dumbbell Topology

B. Effect of Packet Error Probalilities

In the Linear scenario, we test the throughput under different packet error probability. The result is shown in Fig. 7.

From the simulation result in Fig. 7, we can find that the new algorithm provides higher throughput than TCP Reno and TCP Westwood. This is because using RTT as a congestion indicator can overcome bad effect caused by the sporadic loss in wireless network to the congestion window size.

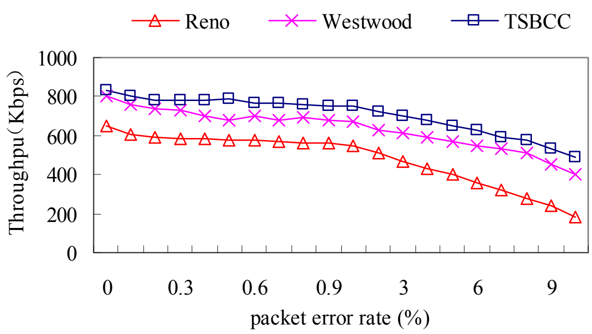

Figure 7. Effect of loss rate to throughput

C. Effect of Wireless Hops to Throughput

In this simulation, we study the impact of the number of hops to network performance. We use the linear scenario, and simulate the throughput from three hops to six hops. The result is as Fig. 8.

The x-axis of Fig. 8 shows the number of hops. The results show that, with all the protocols, the throughput descends rapidly with the hops increasing from three to five. But the throughput becomes stable when the hops exceed five. This phenomenon is caused by the interference of wireless nodes. When the number of hop is more than five, the interference becomes stable. But our new algorithm gets a better throughput than other protocols even under the interference environment. From the Fig.8, we can see that the throughput improves 3 percent compared with Westwood on average.

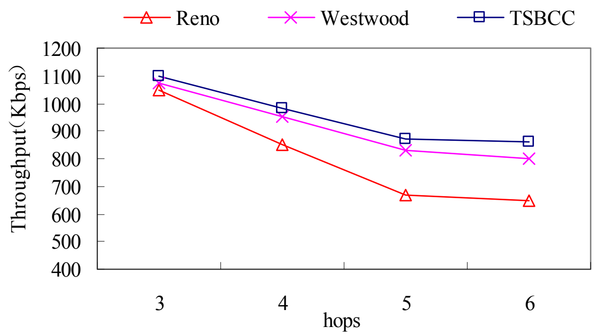

Figure 8. Effect of wireless hops to throughput

D. Effects of Grid Traffic Flows

In this simulation, we study the impact of cross traffic to network performance. We select different FTP traffics in the grid scenario to create simulation traffic, for example: (4, D), (3,5,C,E), (2,4,6,B,D,F), (1,3,5,7,A,C, E,G). The total throughput of network is shown in Fig. 9.

Under each combination, data flow is cross and bidirectional (up and down). This design is to test the multiple data flow's fairness and throughput.

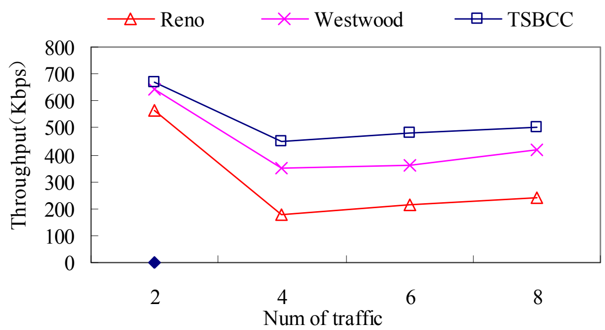

Figure 9. Effect of grid traffic to throughput

The results show that the new protocol achieves higher throughput than the other two. With all the protocols, the throughput descends rapidly with the number of traffic flow from two to four. But the throughput becomes stable when the number of traffic flows exceeds four. This is because the link competition is more intensive when there are four traffic flows. But with the unceasingly increase of traffic flow, the competition becomes stable.

E. fairness

In the dumbbell scenario, we firstly test the total throughput of five flows under different packet error probability. The total throughput is the sum of the five throughputs. The result is shown at Fig. 10. And then, we test the fairness of five flows at some packet error probability. We use the formula (11) to calculate the fairness [20]. The result is shown at Fig. 11.

$$\text{Fairness}=(\sum_{i=1}^{n} x_i)^2 / (n\sum_{i=1}^{n} x_i^2) \qquad (11)$$

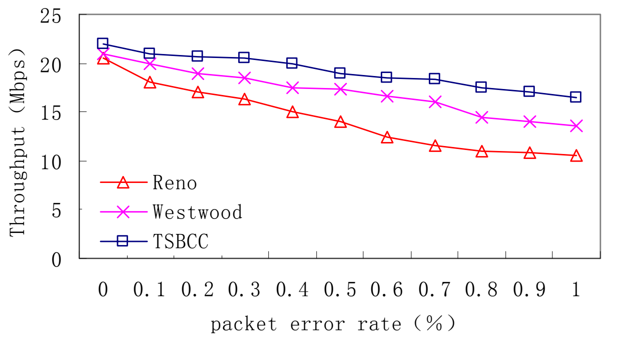

Figure 10. Effect of wireless error rate to throughput

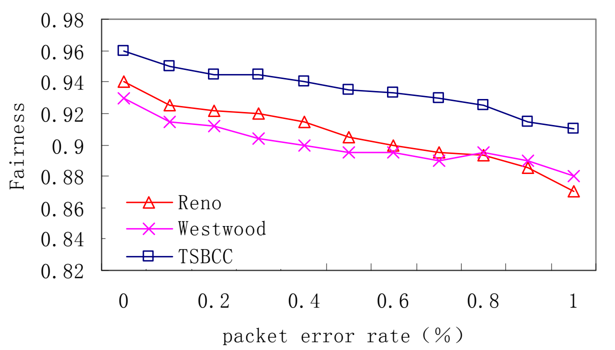

Figure 11. Effect of wireless error rate to fairness

From Fig. 10, we can find that the throughput of proposed algorithm is better than the other two protocols; it is 32 percent higher than TCP Reno on average. And the improved proportion increases with the increase of packet error rate.

The Fig. 11 shows that the fairness of our new algorithm is better than other algorithms. So we can find that the new protocol has a better performance under the dumbbell scenario.

VI. CONCLUSION

In this paper, we proposed a new TCP congestion control algorithm--TSBCC to deal with the problem of poor performance of traditional TCP in the wireless network. Comparing with the conventional TCP, time series-based method makes the new protocol more robust to sporadic packet loss due to intermittent wireless channel. The algorithm predicts the arrival time of the next ACK based on the time series of ACK interval by GARCH model, compares the predicted result with the real result, analyses the reason of the packet loss, and takes different tactics to adjust the congestion window size effectively.

The paper introduces the principle and implementation of time series-based congestion control algorithm. The simulation results show that, compared with TCP Reno and TCP Westwood, the new protocol can decrease the effect of wireless packet loss, and improve the throughput of wireless network. At the same time, the new protocol maintains a good fairness.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 61072063, No. 61202379.

REFERENCES

[1] S. Floyd, K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Transaction on Networking, 1999, 7(4), pp. 458-472.

[2] S. Bhatti, M. Bateman, "Transport Protocol Throughput Fairness", Journal of Networks, 2009, 4(9), pp. 881-894.

[3] S. Hu, H. Shu, T. Lin, "Analysis on Stability of a Network Based on RED Scheme", Journal of Networks, 2011, 6(4), pp. 654-661.

[4] G. Gao, W. Xiao, Z. Liu, W. Wan, "An Application of the Modification of Slow Start Algorithm in Campus Network", Journal of Networks, 2011, 6(11), pp. 1549-1556,.

[5] C. PARSA, J. J. GARCIA-LUNA-ACEVES, "Differentiating congestion vs. random loss: a method for improving TCP performance over wireless links," Proceedings of the 2000 IEEE Wireless Communications & Networking, 2000, pp. 90-93.

[6] Y. HONG, OLIVER W. W. YANG, "Design of adaptive PI rate controller for best-effort traffic in the internet based on phase margin," IEEE Transactions on Parallel and Distributed Systems, 2007, 18(4), pp. 550-561.

[7] J. Wang, J. Wen, J. Zhang, and Y. Han, "TCP-FIT: An improved TCP congestion control algorithm and its performance," IEEE INFOCOM, 2011, pp. 2894-2902.

[8] S. MASCOLO, C. CASETTI, M. GERLA, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," Proceeding of 7th ACM Annu. Int. Conf. Mobile Comput. Networking. Rome, Italy: 2001, pp. 287-297.

[9] L. Brakmo, S. O'Malley, L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," SIGCOMM Proceedings of the conference on Communications architectures, protocols and applications. 1994, pp. 24-35.

[10] Y. HONG, OLIVER W. W. YANG, "End-to-end PI congestion controller for best-effort traffic: single source case" , electronics letters, 2003, 39(5), pp. 480-482.

[11] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," Proceeding of the 2001 IEEE INFOCOM, 2001, pp. 1726-1734.

[12] H. Hsieh, K. Kim, Y. Zhu, and R. Sivakumar, "A receiver-centric transport protocol for mobile hosts with heterogeneous wireless Interfaces," In: Proceedings of ACM MOBICOM, San Diego, CA, USA: 2003, pp. 1~15.

[13] K. SHI, Y. SHU, OLIVER YANG, and J. Luo. "Receiver assistant congestion control in high speed and lossy networks," Proceedings of the 2009 IEEE-NPSS Real Time Conference, 2009, pp. 91-95.

[14] Y. SHU, W. GE, N. JIANG, Y. Kang, and J. Luo, "Mobile-host-centric transport protocol for EAST experiment," IEEE Transactions on Nuclear Science, 2008, 55(1), pp. 209－216.

[15] Z. Wang, J. Crowcroft, "A New Congestion Control Scheme: Slow Start and Search (tri-s)," ACM SIGCOMM Computer Communication Review, 1991, 21 (1), pp. 32—43.

[16] D. Gao, Y. Shu, L. Yu, M. Y. Sanadidi, and M. Gerla, "TCP SPC: Statistic Process Control for Enhanced Transport over Wireless Links," IEEE GLOBECOM 2008. [doi:10.1109/GLOCOM.2008.ECP.1041].

[17] W. Zhao, D. Gao, Y. An, Y. Shu, M. Y. Sanadidi, "Transmission Control Protocol Based on Statistic Process Control", International Journal of Advancements in Computing Technology, 2013, 5(5), pp. 1206-1214.

[18] W. Zhao, C. Liu, Y. Shu, and M. Ma, "A SPC-based adaptive RTO algorithm," wireless communications, Networking and Mobile Computing (WiCOM), 2011, [doi: 10.1109/wicom.2011.6040444].

[19] M. Allman, V. Paxson, W. Stevens, RFC 2581 TCP congestion control, Internet Engineering Task Force, 1999.

[20] R. Jain, D. Chiu, W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report, TR-301．1984.

Weifeng Zhao received his B.S degree in 2002 form School of Computer Science and Technology, Shandong University, his M.S degree in 2007 form School of School of Computer Science and Technology, Tianjin University, and work hard at D.S degree at School of Computer Science and Technology, Tianjin University from 2008. His research interests include computer networks and wireless communications.

Xin Huang received his B.S degree in 2011 form School of Computer and Communication Engineering, Tianjin University of Technology. His research interest is wireless communications.

Kai Shi received his D.S degree from School of Computer Science and Technology, Tianjin University in 2010. He is a associate professor in School of Computer and Communication Engineering, Tianjin University of Technology. His research interests include computer networks and wireless communications.

Lei Zhang (S'03–M'08) received her Ph.D. degree in Computer Science from Auburn University (Auburn, AL, USA) in 2008. She worked as an assistant professor from 2008-2011 in the Computer Science Dept. at Frostburg State University (Frostburg, MD, USA). She is now an assistant professor in the School of Computer Science and Technology at Tianjin University (Tianjin, P. R. China). Her research interests include computer networks, wireless communications, distributed algorithms and network security.