Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2021 SCPE. Volume 22, Issues 2, pp. 193–201, DOI 10.12694:/scpe.v22i2.1893

RESEARCH ON TCP PERFORMANCE MODEL AND TRANSPORT AGENT ARCHITECTURE IN BROADBAND WIRELESS NETWORK

LINTAO LI[*] PARV SHARMA[†] MEHDI GHEISARI[‡] AND AMIT SHARMA[§]

Abstract. The problems of Internet stability, heterogeneity, fairness of bandwidth sharing among streams, efficiency of use and congestion control have been solved in this article. This paper proposes an improved scheme of TCP proxy acknowledgement based on Automatic Repeat Request (ARQ), which improves throughput, reduces delay and saves uplink bandwidth of wireless link, and is more suitable for future asymmetric networks. The substantial improvement is observed during the experimentation as processing efficiency of protocol. The observed results revealed that overall processing time for each packet is approximately equals to one fourth of the transfer control protocol and the reduction of 59% is also observed in the utility of resources. The protocol also incorporates various simple techniques for the recovery of loss to improve the throughput in noisy wireless conditions. The results show that the adoption of the average diversity combining technology is helpful to improve the throughput and effective factor performance, and can reduce the requirement of radio link protocol (RLP) maximum retransmission times. As nearly 90% of uplink acknowledgement frames are filtered, the uplink bandwidth utilization rate is significantly improved. Decomposing large data frames into small data frames is also helpful to improve system performance.

Key words: Broadband Wireless Network; TCP performance; Transport agent; Bandwidth utilization.

AMS subject classifications. 68M14

1. Introduction.

Wireless communication is one of the most active and rapidly developing fields in the field of communication today, and it is also one of the fields of science and technology that will have a great impact on human life and social development in the 21st century. TCP, as a reliable connection-oriented transport layer protocol, provides high-reliability data communication between two hosts. Combined with the widely used IP network, it provides a reliable transport layer on the unreliable IP layer and forms a complete TCP/IP protocol family [1]. Because TCP protocol was originally designed for wired networks and fixed hosts, its performance in wired networks is ideal, and it can control and reduce congestion by its error detection and error recovery mechanism. However, its application in the growing and huge wireless network shows its drawbacks. Therefore, how to ensure the TCP performance in wireless networks and how to improve the TCP protocol to better serve the wireless networks have become a hot spot in wireless network research in recent years.

The mission of mobile computing approach is to provide unavoidable axis two different forms of information like data, video and voice [2]. Mobile computing devices and its integration with existing internetwork system that consists stationary hosts has been playing vital role in drinking best from current state of art approaches and making them one-step closer to the mission realization [3]. On the other hand there are many challenges that has been rising with the continue usage of technology in combination with wireless network Technology. These challenges are rising because of portable mobile devices requirement and the necessity of other important characteristics of wireless networks [4].

In mobile computing, the probability is considered as one of the most important element. With advancement in technology, the physical parameters such as size and weight are shrinking continuously and therefore the power capabilities and mobile computing reducing along with the constrained capacity of battery [5]. In order to maintain the efficient performance through mobile applications there is a strong requirement for minimizing

[*] XinXiang University, Henan Xinxiang, 453000, China (lintaoli12@outlook.com).
[†] Department of Computer Science, Jaypee University of Information Technology, Solan, India (parvsharma560@gmail.com).
[‡] Research Scientist, Islamic Azad University, Iran (mehdi.gheisari61@gmail.com).
[§] Department of Computer Science and Engineering, Chitkara University, Punjab, India (sharma.amit@chitkara.edu.in).

the overall processing load from portable mobile systems [6].

The prime objective behind this study is to develop transfer protocol for computing limited environment of wireless networks. This article presents lightweight architecture as mobile transfer control protocol (MTCP), annoying the functionality of TCP among the mobile and stationary host. The process reduces the overall processing load in in mobile host with less usage of wireless medium. The communication among the mobile and the base station is carried out through a single link in order to enhance the performance [7]. Therefore, the design protocol refers as link layer protocol. For enhancing the performance, many of the functions are simplified or neglected in wireless medium off connection, which leads to the design of a lean protocol called in a machine [8]. Additional the implementation also reveals that the protocol provides offload processing through mobile devices.

The implementation of our protocol depends on model of split connection where the connection among the mobile and base station is divided into two connections [9]. One connection is considered as the connection among fixed host and base station whereas other connection is considered as the connection among base station and mobile host. The first connection is termed as wide connection and other connection is considered as wireless connection [10]. This division is not a new idea and it was already used by many previous research work. The reason behind opting this split connection strategy as this approach is efficiently providing the proxy style architecture for enabling the development of efficient protocol for wireless segment [11].

In the present work for reducing the overall communication overhead, our work incorporates different efficient simple approaches to address heavy losses in wireless communication. The wireless links are prone to various losses due to many factors like interference, user mobility, noise and channel fading. It has been observed from the literature that the transfer control protocol performs poor in wireless networks as it considers all of the losses occurs due to the congestion problems among network [12, 13]. In response to each loss, the transfer control protocol initialized various steps to provide transmission rate to the congestion issues. However, this transfer of transmission read for the leads to the poor performance of network, which exhibits as non-congestive losses. Many researches address these issues and various studies has been proposed to address the effect of non-congestive relevant losses on the performance of transfer control protocol in wireless or similar type of high loss links [14]. The base of our implementation is model of split connection, which addresses these issues by separating the recovery of losses over wireless link from wide link. The performance of the proposed mobile transfer control protocol is observed and its ability is evaluated in terms of offload processing and its throughput is measured for poor loss conditions [15].

2. Literature Review. The smooth development of mobile communication network also plays a powerful role in promoting WLAN. WLAN is a network established in a certain local area by using wireless communication technology, and it is the product of the combination of computer network and wireless communication technology. It uses wireless multiple access channel as transmission medium, provides the function of traditional wired LAN, and enables users to truly access broadband network at anytime, anywhere and at will. The frequency band used by WLAN can be 2.4 GHz and 5.8 GHz, and the data rate can be up to 54 Mbps. When the high-band LAN is running, the communication quality is good without mutual interference, which can ensure the communication safety and meet the service quality requirements [16].

TCP is designed based on wired channel. At this time, network congestion is the main reason for packet loss. After the sender finds packet loss through multiple acknowledgments or retransmission timeout timer, it immediately starts congestion control mechanism, reduces congestion control window and extends retransmission timeout timer, so as to reduce the load pressure on the network and reduce packet error caused by network congestion. When TCP runs on the wireless channel with high packet error rate, TCP will also start the congestion control mechanism when packet error is caused by channel error, which leads to the unnecessary reduction of the throughput of end-to-end TCP connection. At the same time, the "exponential regression" algorithm of retransmission timeout timer timing length further prolongs the time of packet error recovery. The main disadvantage of link layer protocol is that TCP layer and data link layer compete for retransmission. Wong have improved the link layer protocol in the literature, which effectively reduces the problem of competitive retransmission between TCP layer and data link layer and improves TCP throughput. Another method to improve the performance of TCP over wireless channels is to use split connection protocol, which divides the TCP connection between the sender and the receiver into two segments with the wireless base station as the boundary, one spanning the wired channel and the other spanning the wireless channel [17, 18]. However, the shortcomings of split connection protocol are also obvious: first, the retransmission timer timeout caused by high packet error rate in wireless channel will lead to the stop of TCP connection packet transmission on wired channel. Second, the TCP connection is divided into two sections at the base station, which correspondingly increases the software overhead when the packet is transmitted through the base station. Third, when the packet arrives at the base station and the receiver has not received it, the acknowledgement information of the corresponding packet has arrived at the sender, which destroys the semantics of TCP end-to-end connection [19, 20].

A third method to improve the performance of TCP on wireless channel is monitoring protocol. The specific method is to introduce monitoring module to the base station at the boundary between wired channel and wireless channel, monitor the packets in TCP connection passing through the base station bidirectional, buffer the packets that have not been confirmed by the receiving end, and retransmit the packets by using the buffered packets once the error packets are found. The focus of the above research on TCP wins is the throughput rate of TCP connection, which is the same as the performance analysis of TCP connection on TDMA wireless link. When TCP connection runs on CDMA wireless link with limited interference, it is necessary to introduce new performance evaluation index to reflect the performance of TCP connection on CDMA wireless link [21, 22]. The innovation of this paper: The ARQ-based TCP performance improvement schemes under two wireless architectures are proposed, which are base station-mobile terminal architecture and access point-fixed terminal architecture (AP-CPE architecture).

3. Research Methods.

3.1. Proposed Algorithm. ARQ technology in data link layer transforms unreliable physical links with errors into reliable data logical links, and TCP connection will not start slowly due to data frame loss on wireless links, which greatly improves TCP performance, but still has shortcomings. Because the bit error rate (BER) of mobile computing environment is very high, and the probability of continuous errors is very high, it may cause multiple retransmissions of the same data packet on the wireless link, and finally cause the source TCP to start congestion control because of waiting for acknowledgement timeout, first retransmit unacknowledged packets, and reduce the congestion window to reduce the sending rate. Then activate congestion control mechanism, including time-out clock exponential regression and reducing slow start threshold. Finally, the congestion avoidance stage is entered to ensure that the congestion is relieved. This error recovery mechanism of TCP will lead to the degradation of protocol performance, including the decrease of throughput and the increase of delay [23].

3.2. Overall Framework. The realization of TCP proxy validation algorithm in this system includes two parts: base station part and mobile terminal part. In the base station part, a TCP proxy confirmation module is added to the partition/merge/reassembly/disassembly module of RLC entity in fully reliable mode. This module analyzes the IP packets submitted by the upper layer to the partition/merge module for processing and then sent by the upper layer and the IP packets submitted by the reassembly/disassembly module to the upper layer, and decides whether to generate an IP packet containing TCP confirmation segments for the upper layer. The mobile station part adds a filtering module before the segmentation and recombination module of RLC entity in fully reliable mode. According to the filtering algorithm, which TCP segments can be filtered [24-27].

The proposed design is a slight variation in transfer control protocol which reduce the latency is depicted in Figure 3.1. In the proposed protocol instead of beginning the transfer of state during mobile has completed handing off, our work initiates the transfer of state when the process of handoff is expected. In the proposed approach the exchange among the old base station and new base station is carried out through the unicast data transfer process. The process of data transfer depends on the lower layers for providing the idea about mobile station that the hand off is pending and information about the candidate address of base station [28, 29].

Once the base station receives notification the process of state transfer is initiated. The time until mobile station starts handoff, higher amount of data have already been processed as depicted in Figure 3.1. Once the handoff is completed the rest of the data and its state of transfer control protocol is for the transferred for accurately mirroring the old state. From the experimentation it is observed that the latency can be reduced more specifically for the situations where the transfer of data is large and data requires to transfer over slow

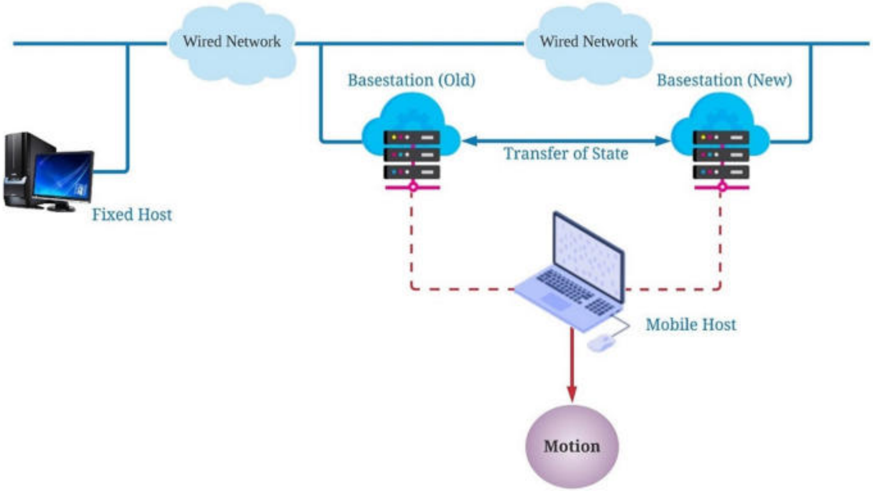

Fig. 3.1. *State transfer process between base stations where mobile is handoff*

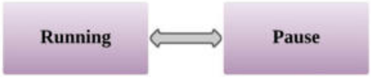

Fig. 3.2. *Proxy state transition diagram of terminal*

links.

3.2.1. Time processing process. The TCP proxy validation module of base station can be divided into several different processes according to its functions: time processing process, pre-transmission analysis process, segmentation merging or recombination de-merging process, reception analysis process and proxy validation process. The base station corresponds to the state transition diagram of the proxy acknowledgement control of a TCP connection, and the proxy corresponding to each TCP connection has two states: Running and Pause is depicted in Figure 3.2.

When a new connection entry is established, the initial state is set according to whether the notification window is 0. If the notification window is 0, the initial state is Pause, otherwise, the initial state is running. In the Pause state, once an uplink TCP segment indicating that the receiving window is not 0 or a downlink TCP segment with a data length not 0 is received, it is transferred to the Running state. If no upstream and downstream data is received for T_BASE_PAUSE, delete this connection in the trace connection table entry. In the Running state, if no upstream and downstream data is received for a continuous time of T_BASE_NODATA_TIMEOUT, delete this connection in the trace connection table entry; And if that uplink TCP message segment indicate that the receiving window is 0 is receive, shifting to the Pause state. Upon receiving other uplink TCP segments, generate uplink TCP acknowledgement segments according to the strategy of generating TCP acknowledgement segments by the base station agent. The time constants used above are defined as follows:

T_BASE_PAUSE: The longest time that a connection in the base station stays in the Pause state. In the specific implementation, it is set to 60 seconds.

T_BASE_NODATA_TIMEOUT: A connection in the base station lasts for no uplink and downlink data at most. In the specific implementation, it is set to 1800 seconds.

3.2.2. Analysis process before sending. The base station analyzes the frame submitted by the upper layer that has not been segmented. If TCP is not encapsulated in the frame. Packet, the address of the frame and the flag of whether it is TCP packet are saved in the related structure of the interface with the split and merge module, and sent to the designated buffer. If it is a TCP packet, proceed to the next step.

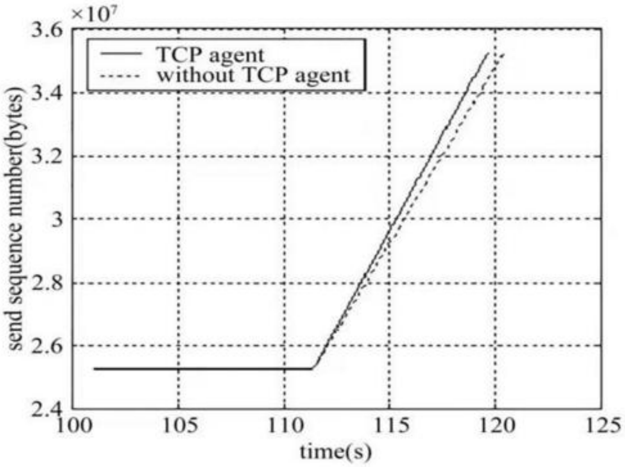

FIG. 4.1. *Comparison of transmission sequence numbers with single connection transmission rate of 150PDU/ time slot and zero frame error*

If the TCP packet encapsulated in the frame has been confirmed by the base station agent, the unnecessary retransmission TCP data segment is encapsulated in the frame, which can be filtered without sending. In the specific implementation, the sending sequence number and the data length of the TCP segment encapsulated in the frame are read, and if the sum of the sending sequence number and the data length is less than or equal to the maximum confirmed sequence number of the connection entry in the connection tracking table, the downlink data frame is filtered and not submitted to the split and merge module for sending.

3.2.3. Agent confirmation process. The ARQ module will inform the proxy confirmation module of the window information after moving a certain sending window. When the confirmation module judges that the IP packet containing the complete TCP segment arrives at the receiving end, it will move the IP packet recording node from the IP packet recording node linked list of the corresponding terminal. If it is in the Running state, it will generate the TCP proxy confirmation segment.

4. Research Results. In all simulations, the transmission rate is 50Mbps. First, simulate the case of single user and single TCP connection, and then simulate the case of multi-user and multi-TCP connection, and each case will get 18 sets of data. For each connection running 10M FTP download service, the simulation shows that there is not much difference in the total download time when using TCP proxy to confirm with zero frame error and when not using TCP proxy to confirm, that is to say, their average throughput rates are not much different. Figure 4.1 is a simulation graph of ”transmission segment number-time” when the transmission rate is 150PDU/ time slot and the frame error rate is 0%. The solid line indicates that TCP proxy confirmation is started, and the dashed line indicates that TCP proxy confirmation is not started.

Because the sending sequence number in TCP protocol is in bytes, and the sending sequence number at each moment minus the starting sequence number is the total number of bytes sent in this period, so this value can be used to calculate the average throughput. The two oblique lines in Figure 4.1 are very close, which shows that the download time for downloading the same business volume is very close in both cases. However, under the condition of FTP download with the same traffic, the average throughput rate of the curve with TCP proxy acknowledgement is still higher than that without TCP proxy acknowledgement. We can use data to illustrate that the difference between the ordinate of the starting point and the end point of the curve in a period of time t is the number of bytes sent in this period of time, so B/t is the average throughput rate in this period of time. The average throughput calculated from the simulation data also illustrates this point. The average throughput without TCP proxy acknowledgement is 8.5 Mbps, and the average throughput after TCP proxy acknowledgement is 9.25 Mbps, which is 8.82% higher than the former. Under the other two transmission rates with zero frame error, the simulation data show that the download time difference is not great when TCP proxy acknowledgement is not used and when TCP proxy acknowledgement is used, but the total time spent

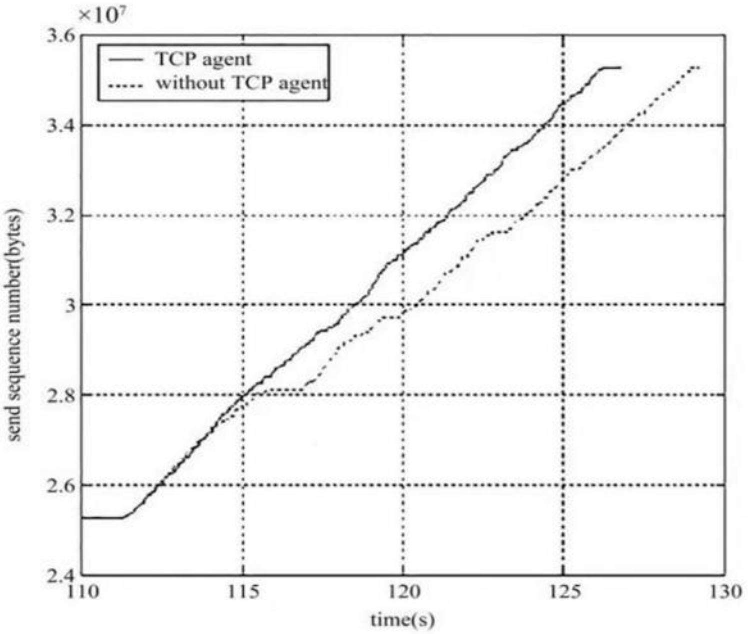

Fig. 4.2. *Comparison of transmission sequence numbers when single connection transmission rate is 150 PDU/ time slot and frame error rate is 10%*

decreases with the increase of transmission rate.

Increase frame error by 10% and load FTP service. Figure 4.2 shows the comparison of simulation curves of "sending segment number-time" when the sending rate is still 150PDU/ time slot. The solid line is the curve of increasing the sequence number of TCP sending segments with time after TCP proxy confirmation is started, and the dashed line is the curve of increasing the sequence number of TCP sending segments with time when TCP proxy confirmation is not started. According to the simulation data, the average throughput under two conditions can be obtained, in which the average throughput without TCP proxy acknowledgement is 4.4 Mbps, and the average throughput after TCP proxy acknowledgement is 5.073Mbps, which is 15.3% higher than the former. When the frame error rate increases to 20%, the average throughput rate increases even more, reaching about 45%.

As can be seen from the above figure 4.2, the total time spent after increasing frame errors increases because retransmission occurs, but the number of retransmissions decreases after TCP proxy confirmation is started. The figure 4.2 shows that when TCP proxy confirmation is not started, the increase of transmission sequence number is not completely linear, but some small "steps" appear, and each "step" means that retransmission has occurred. The retransmission occurred here is not a timeout retransmission, but a fast retransmission. The above conclusion can be easily drawn by comparing with the corresponding time in the plug window graph.

Figure 4.3 is a comparison diagram of congestion window size when the transmission rate is 150PDU/ time slot and the frame error rate is 10%. The solid line is when TCP proxy confirmation is not started, and the dashed line is when TCP proxy confirmation is started. It can be clearly seen from figure 4.3 that a slightly downward fold line appears around 1m56s of the solid line, which just corresponds to the first "step" of the solid line. This slightly downward broken line indicates that a retransmission has occurred. If you continue to enlarge Figure 4.3, you can see more retransmission places, which correspond to Figure 4.1 one by one.

TCP Protocols are mainly congestion avoidance and congestion control protocols. The above simulation only shows the superiority of TCP proxy confirmation technology in congestion avoidance stage. In order to reflect the performance of TCP in the congestion control stage, we should also simulate the processing ability of TCP when congestion occurs. In the simulation, if the size of the buffer at the receiving end is reduced, the buffer at the receiving end will be saturated soon when the frame error rate is large, and the message segment

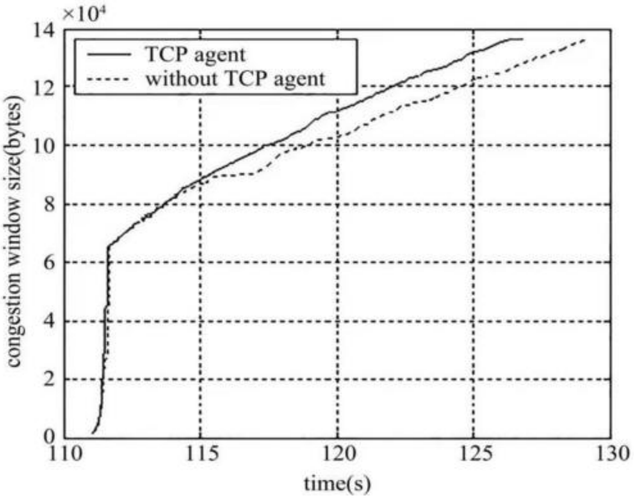

Fig. 4.3. *Comparison of congestion window size when single connection transmission rate is 150 PDU/ time slot and frame error rate is 10%*

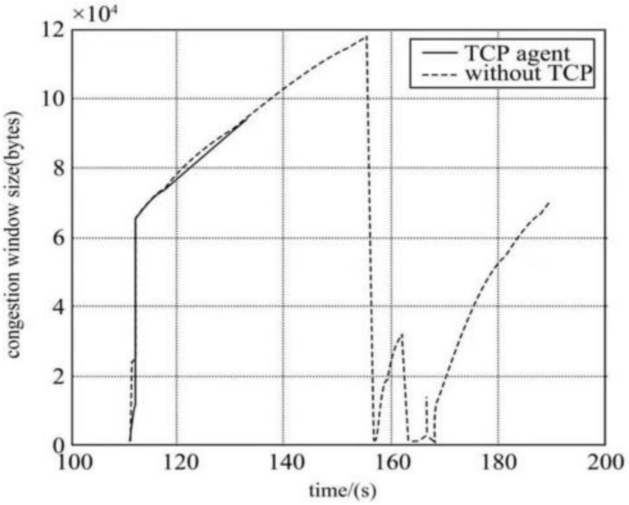

Fig. 4.4. *Comparison diagram of single connection congestion control stage*

sent later will be discarded. Once the sending end does not receive the acknowledgement after reaching the timeout, it will be considered that the link is congested, thus entering the slow start and congestion control stage.

In Figure 4.4, congestion is simulated. The simulation condition is that the transmission rate is 150PDU/ time slot and the frame error rate is 10%. In figure 4.4, the solid line is the simulation curve when TCP proxy confirmation is started, and the dashed line is when TCP proxy confirmation is not started.

The contrast in the figure 4.4 is very obvious. When TCP proxy confirmation is not started, the congestion window of about 2 m 37 s suddenly decreases to 1, which indicates that the sender detects timeout and TCP starts slowly. Moreover, it can be observed that the second slow start is entered because of timeout before the first slow start is over. Once the TCP starts slowly, it will degrade the performance of TCP a lot. After TCP proxy confirmation is started, the download time is short, and the congestion control phase is avoided. Because when TCP proxy acknowledgement is not started, if the burst frame error is high, the downlink data will be lost. Although ARQ is used for retransmission in the link layer, it is still possible to cause TCP retransmission over time, and the uplink TCP acknowledgement segment will also be lost on the link, which increases the possibility of retransmission over time. After TCP proxy confirmation is started, it depends on ARQ confirmation information, so there is no problem that a large number of TCP confirmation frames are lost, and because ARQ confirmation efficiency is very high, it makes The validation efficiency of TCP is also greatly improved. This greatly improves the throughput of the link, so that FTP download with the same traffic can be completed quickly.

Simulation data show that when the frame error increases further, the performance of TCP deteriorates further when TCP proxy acknowledgement is not started, and the congestion window when TCP proxy acknowledgement is started always grows linearly, and the time taken is much less than that when TCP proxy acknowledgement is not started.

5. Conclusion. Under the condition of a single TCP connection, two stages of congestion avoidance and congestion control are simulated. The number of retransmissions when TCP proxy confirmation is started in congestion avoidance stage is less than that when TCP proxy confirmation is not started. Congestion control stage completely avoids the occurrence of congestion after TCP proxy confirmation is started, but when TCP proxy confirmation is not started, it will enter slow start for many times. The average throughput rate and average round-trip delay are improved in different degrees in the two stages. Under the condition of multiple TCP connections, the average round-trip time is greatly improved, and the average throughput of links is also improved. At the same time, the phenomenon of bandwidth preemption is found, which makes unfair among TCP connections. Due to filtering nearly 90% of uplink acknowledgement frames, the uplink bandwidth utilization rate is greatly improved. The performance of TCP has been improved, and it is observed that the worse the condition will be, the better its performance.

REFERENCES

[1] Bouattane, O., Youssfi, M., Vassilis, K., and Papakostas, G., *Generic distributed polymorphic learning model for a community of heterogeneous cyber physical social robots in MAS Environment and GPU Architecture*, International Conference on Intelligent Systems and Computer Vision (ISCV), 2020.

[2] Tsaoussidis, V., and Matta, I., *Open issues on TCP for mobile computing*, Wireless Communications and Mobile Computing, 2(1), 3-20, 2002.

[3] Ahmed, F., Pradhan, S. K., Islam, N., and Debnath, S. K., *Performance Evaluation of TCP over Mobile Ad hoc Networks*, arXiv preprint arXiv:1002.2189, 2010.

[4] Baccarelli, E., Cordeschi, N., Mei, A., Panella, M., Shojafar, M., and Stefa, J., *Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study*, IEEE Network, 30(2), 54-61, 2016.

[5] Shojafar, M., Cordeschi, N., Abawajy, J. H., and Baccarelli, E., *Adaptive energy-efficient qos-aware scheduling algorithm for tcp/ip mobile cloud*, In 2015 IEEE Globecom Workshops (GC Wkshps), 1-6, 2015.

[6] Han, B., Qian, F., Hao, S., and Ji, L., *An anatomy of mobile web performance over multipath TCP*, In Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, 1-7, 2015.

[7] Kim, M., Ko, S. W., and Kim, S. L., *Enhancing TCP end-to-end performance in millimeter-wave communications*, In 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 1-5, 2017.

[8] Mohammad, A. S., and Potrus, M. Y., *A Method for Compensation of TCP Throughput Degrading During Movement Of Mobile Node*, ZANCO Journal of Pure and Applied Sciences, 27(6), 59-68, 2016.

[9] Safa, H., Artail, H., and Tabet, D., *A cluster-based trust-aware routing protocol for mobile ad hoc networks*, Wireless Networks, 16(4), 969-984, 2010.

[10] Luo, C., Yu, F. R., Ji, H., and Leung, V. C., *Cross-layer design for TCP performance improvement in cognitive radio networks*, IEEE Transactions on Vehicular Technology, 59(5), 2485-2495, 2010.

[11] Pinto, J., Martins, R., and Sousa, J. B., *Towards a REST-style architecture for networked vehicles and sensors*, In 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 745-750, 2010.

[12] Hossain, E., Chow, G., Leung, V. C., McLeod, R. D., Mišić, J., Wong, V. W., and Yang, O., *Vehicular telematics over heterogeneous wireless networks: A survey*, Computer Communications, 33(7), 775-793, 2010.

[13] Afanasyev, A., Tilley, N., Reiher, P., and Kleinrock, L., *Host-to-host congestion control for TCP*, IEEE Communications surveys & tutorials, 12(3), 304-342, 2010.

[14] Liu, K., and Lee, J. Y., *On improving TCP performance over mobile data networks*, IEEE transactions on mobile computing, 15(10), 2522-2536, 2015.

[15] Pokhrel, S. R., Panda, M., Vu, H. L., and Mandjes, M., *TCP performance over Wi-Fi: Joint impact of buffer and channel losses*, IEEE Transactions on Mobile Computing, 15(5), 1279-1291, 2015.

[16] Putra, S. A., Trilaksono, B. R., Riyansyah, M., and Laila, D. S., *Multiagent architecture for bridge capacity measurement system using wireless sensor network and weight in motion*, IEEE Transactions on Instrumentation and Measurement, PP (99), 1-1, 2020.

[17] Zhang, W., and Liu, Y., *Research and Optimization for the Election Mechanism of Distributed Wireless Networks*, In 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 622-626, 2019.

[18] Song, Y., Liu, Z., He, X., and Jiang, H., *Research on data fusion scheme for wireless sensor networks with combined improved leach and compressed sensing*, Sensors (Basel, Switzerland), 19(21), 2019.

[19] Song, Y., and Chen, J., *Agent-based multi-usv intelligent command and control cooperative system*, Journal of Physics: Conference Series, 1813(1), 012035 (5pp), 2021.

[20] Liu, Y., Sun, Q., Sharma, A., Sharma, A., and Dhiman, G., *Line monitoring and identification based on roadmap towards edge computing*, Wireless Personal Communications, 1-24, 2021.

[21] Nasir, Y. S., and Guo, D., *Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks*, IEEE Journal on Selected Areas in Communications, PP(99), 1-1, 2019.

[22] Budampati, R. S., Kolavennu, S. N., and Foo, K. D., *Method and apparatus for providing security in wireless communication networks*, U.S. Patent 8,280,057, issued October 2, 2012.

[23] Yang, W., Dong, P., Cai, L., and Tang, W., *Loss-aware throughput estimation scheduler for multi-path tcp in heterogeneous wireless networks*, IEEE Transactions on Wireless Communications,PP(99), 1-1, 2021.

[24] Jin, Q., Guo, Q., Luo, M., Zhang, Y. M., and Cai, W., *Research on High Performance 4G Wireless VPN for Smart Factory Based on Key Technologies of 5G Network Architecture*, 2020 International Wireless Communications and Mobile Computing (IWCMC), 2020.

[25] Lit, A., Rusli, M. S., and Marsono, M. N., *Comparative performance evaluation of routing algorithm and topology size for wireless network-on-chip*, Bulletin of Electrical Engineering and Informatics, 8(4), 2019.

[26] Choi, H., and Lee, B. G., *CA TCP agent scheme based on active buffer control to support lossless handover in broadband wireless networks*, In Global Telecommunications Conference GLOBECOM'02. IEEE (Vol. 3, pp. 2493-2497), 2002.

[27] Tomar, R., Sastry, H. G., and Prateek, M., *Establishing parameters for comparative analysis of V2V communication in VANET*, 2020.

[28] Tomar, R., Prateek, M., and Sastry, H. G., *Analysis of beaconing performance in IEEE 802.11 p on vehicular ad-hoc environment*, In 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON) (pp. 692-696), 2017.

[29] Tomar, R., Kumar, S., and Awasthi, M. K. , *To Beacon or Not?: Speed Based Probabilistic Adaptive Beaconing Approach for Vehicular Ad-Hoc Networks*, In International Summit Smart City 360° (pp. 156-170). Springer, Cham, 2020.

Edited by: Pradeep Kumar Singh
Received: May 26, 2021
Accepted: Sep 20, 2021