Performance Enhancement of DCCP TCP-like over Long Delay Link Networks

Shahrudin Awang Nor, Suhaidi Hassan, Osman Ghazali, Mohammed M. Kadhum, and Mohd. Hasbullah Omar

***Abstract*— The performance of DCCP TCP-like degrades significantly over long delay link networks. Despite the TCP-like congestion control mechanism follows the TCP SACK, the performance is really affected by the congestion window growth algorithms as employed by Jacobson based TCP variants. In this paper, all the experiments are done using Network Simulator ns-2, and we manipulated the congestion window size drop during congestion avoidance phase to enhance the performance of DCCP TCP-like over long delay link networks. Instead of halving the current congestion window when congestion events are detected, the reduction of current congestion window drop has been shown to improve the DCCP TCP-like throughput with minimal drop packet percentage.**

***Index Terms*—Congestion window, DCCP, TCP-like**

I. Introduction

Transport Control Protocol (TCP) [1] has known to be a reliable transport protocol with congestion control for delivering data traffic. Moreover, TCP can deliver the best-effort services for error-intolerant and delay-tolerant data such as web, email, file transport, etc. All that features of TCP make it suitable for the delivery of important, mission critical, and error-free data which requires a reliable data connection.

On the other hand, TCP is not suitable to send multimedia data such as audio and video which request time-sensitive and error-tolerant transmission. For multimedia data transmission, UDP is a suitable transport protocol and has been the favorite choice for decades among Internet users because it is a simple transport protocol and can comply with the transmission requirements. However, the extensive use of UDP can endanger and collapse the network because UDP is greedy protocol, which means that it will send data as much as it can without congestion control, and it is not friendly to other congestion controlled protocol such as TCP. One of the solutions regarding this is the introduction of Datagram Congestion Control Protocol (DCCP) [2] which is an unreliable with congestion control transport protocol.

In addition to a concern about congestion collapse, there is a concern about `fairness' for best-effort traffic. Because TCP "backs off" during congestion, a large number of TCP connections can share a single, congested link in such a way that bandwidth is shared reasonably equitably among similarly situated flows. The equitable sharing of bandwidth among flows depends on the fact that all flows are running compatible congestion control algorithms. For TCP, this means congestion control algorithms conformant with the current TCP specification.

In this paper, we are enhancing the performance of DCCP TCP-like when delivering multimedia data traffic over long delay link networks through the reduction of current congestion window drop. For long delay link network, the throughput of TCP-like behaves unsmoothly during the congestion avoidance phase. The solution introduced here is to reduce the current congestion window drop when congestion event is detected. And as a result, we managed to minimize the obvious zigzag like into smoother throughput with minimal jitter.

This paper is organized as follows: This introductory section is followed by Section 2 of related works done by other researchers. Section 3 describes the congestion window used in TCP and DCCP TCP-like in dealing with congestion control mechanism. In Section 4, we describe the experimental setup and performance metrics. The results and analysis are included in Section 5, and finally Section 6 concludes the findings.

II. Related Works

There are many researches done regarding congestion window in TCP. Since TCP-like congestion control for DCCP follows the congestion control mechanism of TCP SACK, all the researches on TCP, particularly on TCP SACK are relevant to TCP-like. TCP-like is also utilizing the congestion window which can grow or shrink depends on the condition of the network. In normal case, the current congestion window will be halved, i.e. it will be dropped 50% from the current value during congestion avoidance phase when a congestion event via packet loss is detected. During congestion avoidance phase, packet loss is detected through three duplicate ACKs or ECN marked packets.

M.S. Abdalla et al. [3] proposed an enhanced SACK (ESACK) to adjust congestion window size for enhancing TCP in low earth orbit (LEO) satellite networks. Their mechanism tracks losses in two consecutive windows and accordingly takes more protective actions. It saves connection throughput from aggressive congestion window reduction when there is no-congestion loss, and at the same time it takes appropriate actions when there is a high probability of network congestion. They claimed that their new proposed mechanism provides better throughput with fairness compared to the conventional SACK.

Manuscript received April 10, 2012; revised May 15, 2012.

S. A. Nor, S. Hassan, O. Ghazali, M. M. Kadhum, and Mohd. H. Omar are with the Universiti Utara Malaysia (e-mail: shah@uum.edu.my, suhaidi@uum.edu.my, osman@uum.edu.my, kadhum@uum.edu.my, mhomar@uum.edu.my). All of the authors are the member of the InterNetWorks Research Lab, School of Computing, College of Arts and Sciences, Universiti Utara Malaysia.

Congestion window growth algorithm that is based on logarithmic growth utilizing information gained via bandwidth estimation was done by Joel Sing and Ben Soh [4]. Their approach only requires minor modifications at the sender and receiver, hence it does not require any form of support from the underlying network infrastructure and can function correctly with network level encryption in place.

There are other researches for congestion window for TCP, including window distribution [5], congestion window validation over satellite paths [6], congestion window controller [7], and congestion window for TCP Westwood [8],[9].

Congestion window size is not limited to the size recommended by TCP standard. There is also a solution for highspeed TCP utilizing large congestion window as described in RFC 3649 [10]. HighSpeed TCP is a modification to TCP's congestion control mechanism for use with TCP connections with large congestion windows because the congestion control mechanisms of the current Standard TCP constrains the congestion windows that can be achieved by TCP in realistic environments.

III. Congestion Window in TCP and DCCP TCP-like

Congestion window in TCP represents a buffer of packet that can be sent into the network. It is one of the key components in TCP's congestion control [11]. In TCP, congestion window [12] controls the number of packets a TCP flow may have in the network at any time. However, long periods when the sender is idle or application-limited can lead to the invalidation of the congestion window, in that the congestion window no longer reflects current information about the state of the network.

In addition, congestion window is a parameter in TCP where it buffers the packets in the network. As TCP-like [13] is a congestion control mechanism for DCCP which follows TCP SACK congestion control, the utilization of congestion window in TCP-like is also for the purpose of controlling congestion in the network. The congestion control mechanism in TCP-like is about the same as TCP standard,. The congestion event is detected through time-out, receiving three duplicate acknowledgements or marked packets by the sender during congestion avoidance phase.

DCCP has two congestion control mechanisms; TCP-like and TFRC. TCP-like follows the same congestion control mechanism like standard TCP but with some modifications. Congestion control mechanism in TCP and TCP-like consists of two phases, i.e. slowstart and congestion avoidance. In slowstart phase, congestion window size starts from one packet, then increase exponentially for every RTT until it reaches the threshold value. From here onwards it enters congestion avoidance phase by increasing by one for every RTT until the detection of congestion event indicated by packet loss by the sender through congestion event time-out, three duplicate ACKs or marked packet. If congestion event is detected through time-out, the process starts all over again with starting congestion window size of one, but some recommends more, i.e. two, three or four [14]. On the other hand, if congestion event is detected through receiving of 3 ACKs or marked packets, the congestion window size will be halved and the process will continue in the same congestion avoidance phase. During this congestion avoidance phase, it is utilizing Additive-Increase Multiplicative-Decrease (AIMD) where congestion window size increases linearly, i.e. by one for every RTT.

IV. Experimental Setup and Performance metrics

Congestion window in TCP represents a buffer of packet that can be sent into the network. It is one of the key components in TCP's congestion control [11]. In TCP, congestion window [12] controls the number of packets a TCP flow may have in the network at any time. However, long periods when the sender is idle or application-limited can lead to the invalidation of the congestion window, in that the congestion window no longer reflects current information about the state of the network.

A. Simulation Environment

The experiments have been carried out by means of simulation with the simulation topology as shown in Fig. 1. The network simulation topology uses classic dumbbell topology. Dumbbell topology is a very common topology that has been used in many TCP network simulations.

For all the experiments, the simulations consist of a DCCP TCP-like and a standard TCP senders. At the receiver's side, there are DCCP TCP-like and TCP receivers. All the senders and receivers are connected to the routers through 100 Mbps links with 1 ms propagation delay.

In our simulation environment, we have simulated DCCP as a competing protocol to TCP, so that we can see how the other protocol such as DCCP behaves when they coexist with TCP. The utilization of bandwidth by these two competing protocols is set into a scenario so that a DCCP sender will fully utilize the 2 Mbps bandwidth with the sending rate of 2 Mbps CBR traffic. The CBR packet size used is 500 bytes. In this case, TCP sender sends the file transfer data using FTP application, and here we can see the friendliness of DCCP protocol towards TCP. Unlike DCCP, where the transmission bit rate can be set by the application like CBR, the maximum bit rate occupied by FTP application on TCP will be calculated by the transport protocol itself based on the link bandwidth provided, packet size, propagation delay, etc. From the simulation results, we will see how congestion window size drop affects the performance of DCCP TCP-like.

The network topology used in our simulation includes two interconnected routers, R1 and R2 with queue size of 20 packets. For the router to router connection, a long delay bottleneck link is set to have a bandwidth of 2 Mbps with 300 ms propagation delay. This long delay bottleneck link can be used as an emulation of satellite or wireless links with a fixed forward link delay of 300 ms and fixed return link delay of 300 ms. This assumption is reasonable based on Henderson and Katz [15] for the satellite link. There is also research done by other researchers that used this assumption for a long delay link [16]. In addition, we considered that the bottleneck link has enough bandwidth allocation for the data transfer to flow from the sender to the receiver. For simplicity, instead of using other types of queue management such as Random Early Detection (RED), the type of queue management used in this link is Drop Tail, which implements First-In First-Out

(FIFO). The network simulator *ns-2* [17] with DCCP module [18] installed is chosen for the simulation.

In all the simulations, we use SACK TCP because it is the same congestion control mechanism used by DCCP CCID-2 TCP-like. As a future plan, we are looking forward to implementing congestion window drop reduction in DCCP CCID-2 TCP-like if the result is convincing.

The throughput is measured between Router 1 and Router 2 where the TCP-like and TCP flows compete with each other on the long delay link. The TCP connection is monitored while it coexists with DCCP connection.

The simulation time is set to 1000 seconds because this period is long enough to get the picture of the overall performance within this time. In all the simulation experiments, the FTP application using TCP is started first, i.e. at time 0.5 seconds, whereas the CBR application for DCCP TCP-like is started at time 10 seconds. We assume that 10 seconds is enough to allow the TCP data flow to utilize the bandwidth without any contention with another flow, so that we can see the effect on throughput of having other flows joining the bottleneck link after that.

The calculations for the average throughput, packet drop percentage, average delay and average jitter are measured from the simulation time at 200s to 980s for more precise average value. These are done to avoid the data collected during the times for start-up and tear-down connections, and for the transport protocols to adjust for the optimum throughput.

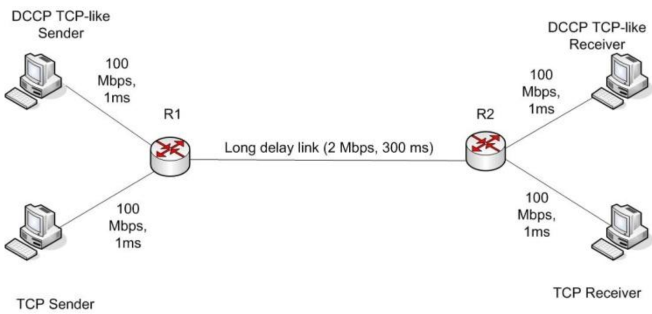

Fig. 1. Simulation topology.

B. Performance Metrics

There are four performance metrics used in this simulated experiment. There are throughput, packet loss, average delay and jitter.

1) Throughput

Throughput is the total amount of data transferred from one source node to destination node during a specified time in a unit of mega bits per second, kilo bits per seconds and etc. Equation (1) is used to measure the throughput of the simulation.

$$Throughput = \frac{TransferSize}{TransferTime} \qquad (1)$$

2) Packet Loss

Packet loss is the difference of the total number of packets received at the receiver and the total number of packets sent at source. Packet loss is measured using equation (2).

$$PacketLoss = SentPacket - ReceivedPacket \qquad (2)$$

3) Average delay

Delay is time taken by packet to travel from source to destination. The delay includes the sum of application's processing delay, propagation delay, queuing delay, etc. Average delay is calculated as given by equation (3), i.e. by summing up all the delays of all packets and divides them by the total number of packets.

$$AverageDelay = \frac{\sum_{i=1}^{n} DelayOfPacket}{n} \qquad (3)$$

4) Jitter

Jitter is a variation of delay. The performance of delay sensitive applications such as audio or video streaming is much affected by the value of jitter. Equation (4) gives the method used to calculate the jitter.

$$\begin{aligned} \Delta arrival_n &= |Arrival_n - Arrival_{n-1}| \\ Jitter_n &= |\Delta arrival_n - \Delta arrival_{n-1}| \end{aligned} \qquad (4)$$

Where, *n* is the current packet.

V. Results and Analysis

In all the simulations, it is bearing in mind that all the data traffics have to go through a bottleneck link with 2 Mbps bandwidth and 300 ms propagation delay for long delay link. This bottleneck link is the link that connects the two routers in the simulation topology. So it is the link that limits the sending rate of the application data between these two routers. The reduction of congestion window size drop of 25% and 5% are done for TCP-like congestion control mechanisms.

The results presented here are given in the Table I which shows the average throughput, packet drop, delay and jitter for DCCP TCP-like and TCP flows.

A. Congestion window size drop of 50% for TCP-like

Same like TCP, the congestion window size in DCCP TCP-like is halved whenever there is a congestion event detected during congestion avoidance phase. Fig. 2 shows that the throughput of TCP-like is like zigzag when it enters congestion avoidance phase at time around 170 seconds until end of the simulation time.

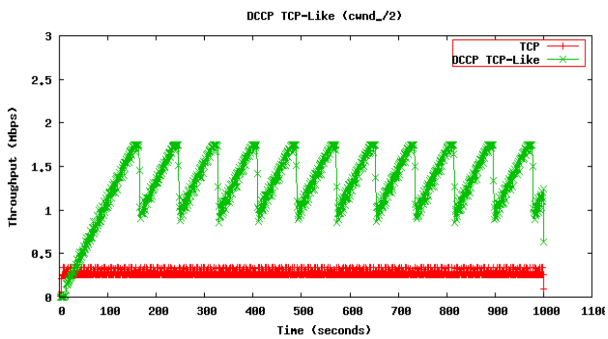

Fig. 2. Congestion window size drop of 50%.

C. Congestion Window Size Drop of 25% for TCP-Like

As in Fig. 3, the throughput is improved compared to (5.1). There is better throughput and jitter for TCP-like flow with acceptable packet loss.

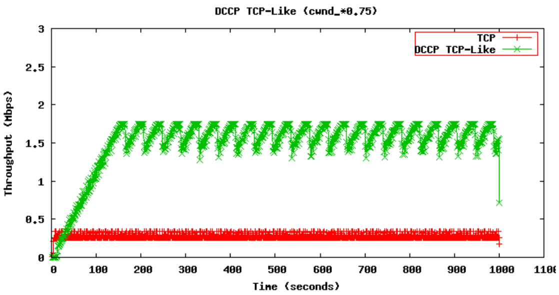

Fig. 3. Congestion window size drop of 25%.

D.Congestion Window Size Drop of 5% for TCP-Like

Fig. 4 depicts the result that shows how the throughput and jitter are improved a lot for TCP-like flow when the drop of TCP-like's congestion window size is reduced by 5%.

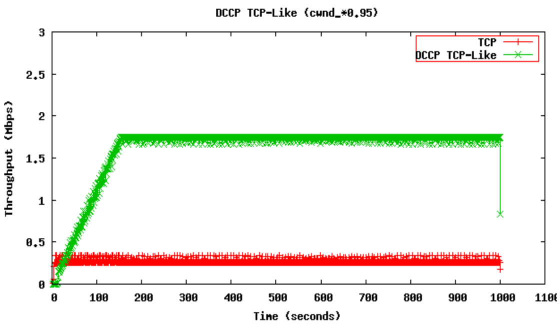

Fig. 4. Congestion window size drop of 5%.

TABLE I: AVERAGE THROUGHPUT, PACKET DROP, AVERAGE DELAY AND AVERAGE JITTER

Congestion window size drop	Average throughput (kbps)	Packet Drop (%)	Average Delay (ms)	Average Jitter (ms)
50% TCP-like	1396.65	0.002400	309.940460	1.285047
50% TCP	282.46	0	305.768980	0.000086
25% TCP-like	1607.06	0.003994	313.252196	1.277461
25% TCP	281.33	0	307.715231	0.000146
5% TCP-like	1737.36	0.016820	330.106516	1.273508
5% TCP	272.74	0	316.923449	0.000226

VI. CONCLUSION

In this paper, we have presented a performance enhancement of DCCP TCP-like when delivering data over long delay link network. Our approach is to consider the reduction of congestion window drop when there is any congestion event detected by the sender through three duplicated acknowledgements during congestion avoidance phase.

For normal case during congestion avoidance phase, when any congestion event is detected, the congestion window will be halved from the current congestion window size. This causes the delay in getting maximum throughput when the congestion window size keep increase additively until the next congestion event detected.

Our results show that through the reduction of congestion window size drop, the throughput and jitter become better with acceptable packet loss rate. Instead of halving the congestion window when congestion even detected, the congestion window is dropped into higher value, i.e. congestion window size drop of 25% or 5% from the current congestion window size.

There is also shown in this research that when the performance of TCP-like is improved, it still maintain its friendliness with TCP when sharing the same bottleneck link.

As future work, this concept is feasible to apply to DCCP TCP-like mechanism for the transmission of multimedia data over long delay link networks to improve the performance in term of throughput and jitter where we can tolerate with a little bit higher packet loss. A bit higher of packet drop is considerable because DCCP is unreliable transport protocol and there is no significant effect when transmitting multimedia data such as audio or video.

REFERENCES

[1] ISI-USC, "Transmission Control Protocol," Internet Engineering Task Force, 1981.

[2] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," Internet Engineering Task Force, March 2006.

[3] M. S. Abdalla, A. M. Abdelrahman, and B. M. Ali, "Optimisation of congestion window for TCP adaptation to LEO satellite networks," *Electronics Letters*, vol. 40, pp. 634 - 636, 13 May 2004.

[4] J. Sing and B. Soh, "Improving Congestion Window Growth in Large Bandwidth Delay Product Networks," in *Networks, 2007. ICON 2007. 15th IEEE International Conference on*, pp. 382 -387, 19-21 2007.

[5] A. Misra and T. J. Ott, "The window distribution of idealized TCP congestion avoidance with variable packet loss," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1564 -1572 vol.3, 21-25 1999.

[6] M. I. Biswas and G. Fairhurst, "An Investigation of TCP Congestion Window Validation over Satellite Paths," in *Advanced Satellite Mobile Systems, 2008. ASMS 2008. 4th*, pp. 37 -42, 26-28 2008.

[7] C. Sall, E. El Alaoui, S. Doubabi, and E. Warraki, "A robust controller for the TCP congestion window," in *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pp. 137 -142, 6-9 2008.

[8] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo, "TCP Westwood: congestion window control using bandwidth estimation," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 3, pp. 1698 -1702 vol.3 2001.

[9] M. Tekala, "TCP Westwood with Limited Congestion Window," in *Advanced Computer Control, 2009. ICACC '09. International Conference on*, pp. 687 -692, 22-24 2009.

[10] S. Floyd, "HighSpeed TCP for Large Congestion Windows," in *RFC 3649*, December 2003.

[11] S. Floyd, "Congestion Control Principles," in *RFC 2914*, September 2000.

[12] M. Handley, J. Padhye, and S. Floyd, "TCP Congestion Window Validation," in *RFC 2861*, June 2000.

[13] S. Floyd and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control," Internet Engineering Task Force, March 2006.

[14] M. Allman, S. Floyd, and C. Partridge, "Increasing TCP's Initial Window," in *RFC 3390*, October 2002.

[15] T. R. Henderson and R. H. Katz, "Transport Protocols for Internet-Compatible Satellite Networks," vol. 17, pp. 326-344 1999.

[16] A. Sathiaseelan and G. Fairhurst, "Use of quickstart for improving the performance of TFRC-SP over satellite networks," in *Proc. International Workshop on Satellite and Space Communications*, pp. 46-50, 14--15 Sept. 2006.

[17] *The VINT Project. The Network Simulator - ns-2.* [Online]. http://www.isi.edu/nsnam/ns/, retrieved on 20 January 2011.

Dr. S. A. Nor is a lecturer at Universiti Utara Malaysia and received his B.Eng degree with honours in Electronic Engineering from Universiti Sains Malaysia, and MSc degree in Computer and Information Networks from University of Essex, United Kingdom. He received his PhD degree in Computer Networks from Universiti Utara Malaysia in 2012. His research interest includes Computer Network and Security, Communication and Network Performance, Network Transport Protocol and Computer Organization.

Dr. S. Hassan, SMIEEE, is an associate professor and InterNetWorks Research Lab chairman at Universiti Utara Malaysia. He received his BSc degree in Computer Science from Binghamton University, New York (USA) and his MS degree in Information Science (with concentration in Telecommunications and Networks) from the University of Pittsburgh, Pennsylvania (USA). He received his PhD degree in Computing (specializing in Networks Performance Engineering) from the University of Leeds in the United Kingdom. His research interest in Computer Networks are Protocol Engineering, Network Quality of Service (QoS), Internet Architecture and Technologies, Network Performances, Network Traffic Engineering, Network Security, Wireless and Mobile Networking, Local Area Networking Telecommunications/Network Management.

Dr. O. Ghazali is a senior lecturer at UniversitiUtara Malaysia and he received his Bachelor of Information Technology, Master Science of Information Technology and PhD of Information Technology (Computer Network) from Universiti Utara Malaysia in 1994, 1996 and 2008 respectively. He is actively pursuing research and supervising postgraduate students in the area of distributed computing and computer networks. His research interest are Ad-hoc Network, Cloud Computing, Internet Worms, Layered Multicast, Network Performances, Network Traffic Engineering, Packet Error Correction, Wireless and Mobile Network and Video Streaming.

Dr. M. M. Kadhum, MIEEE, is a visiting lecturer at Universiti Utara Malaysia. He received his PhD degree in Computer Sciences from University Utara Malaysia in 2010. He is doing his post doctorate in Traffic Engineering and Bandwidth Provisioning at Queens University in Canada. His research Interest includes Network Optimization and Performance, Traffic Engineering, Congestion Control, Wireless and Mobile Ad Hoc Networks, Cloud Computing, and Network Modeling.

Dr. M. H. Omar, MIEEE, is a lecturer at Universiti Utara Malaysia. He received his Bachelor degree in Electronic, Telecommunication and Computer Engineering from University of Bradford, United Kingdom and Master degree by research from Universiti Utara Malaysia. He received his PhD degree in Computer Networks from Universiti Utara Malaysia. His research interest are Cognitive Radio, Signal Detection, Dynamic Spectrum Access (DSA), Network and Computer Security, Wireless Communication and Network Performance.