# Comparative Study of Congestion Control Techniques in High Speed Network

Sahrul Hilmi Ibrahim, Yuki Matsuzawa, Noor M. Asraf, Salahuddin Ali, Siti Aufa, Jalil Md Desa
Advanced Communication Network Laboratory, TM R&D Sdn Bhd,
Cyberjaya, Malaysia

Sevia M. Idrus, Nadiatulhuda Zulkifli
Photonic Technology Lab,
University Teknologi, Malaysia

***Abstract*— Due to enhancement of broadband infrastructure, many multimedia applications such as streaming media, IPTV, video conference, online gaming and video surveillance are emerging. These video streaming generally require high bandwidth but are not responding to network congestion. And most of them prefer timeliness to reliability. TCP seems not suitable to real time applications because it rather focuses on ensuring data transmission. Currently most of the applications are using UDP, but UDP is lacking of congestion protocol and no guarantee of packet delivery. DCCP is a new transport protocol being standardized by IETF that provides unreliable congestion controlled flows of data packets. In this paper, we compare the behavior of congestion control of these transport protocol by manipulating the queue size, link capacity and packet delay. Network Simulator NS-2 was used to evaluate the network scenarios.**

**Index Terms— Transport Protocol, TCP, DCCP, Congestion Control, NS-2**

## I. INTRODUCTION

High Definition (HD) and three dimensional (3D) type of TV and video streaming is seen to be the next milestone in the evolution of digital video storage and transmission. It will transforming TV watching into an immersive interactive experience with technology capitalizing on advances in digital TV broadcast, 3D visualisation, image processing, and efficient communication of rich interactive multimedia material has become an important target to create new revenues in the broadband arena [1],[2]. Fast growing internet media applications such as streaming media, video conferences, video surveillance and online gaming need a new requirement of network protocol. They are extremely sensitive to quality fluctuation and delay, but losing a certain number of packets would not affect the quality of service [3].

The first consideration to the transmission over IP network is the bandwidth needed to ensure an appropriate Quality of service (QoS) at the transport layer (figure 1), because at this layer the service could be sent in unicast, multicast or broadcast. Another important issue with the real time, high bandwidth media delivery over IP is congestion prevention and control because transmission of large data without suitable congestion control may reduce the throughput and increase delay for other applications which sharing the same links [4].

Table 1: Traffic flows from application to link layer

| | |
|---|---|
| Applications | HTTP, FTP, 3DVideo Traffic (MPEG/H.264) |
| Transport | TCP, UDP, SCTP, DCCP, … |
| Network | IP, ICMP, IGMP, … |
| Link | Device driver |
| Physical | Wired, Wireless Technology |

Transmission Control Protocol (TCP) [5] and User Datagram protocol (UDP) [6] are no longer suitable as the transport protocol since they present several problems when working with modern real time applications and networks. They are also does not use a standardized way to adjust for congestion. TCP has its own limitations, for example it rather focuses on ensuring data transmission. With UDP having no connection state, firewalls will often not allow traffic through which means that media applications will revert to using TCP. This is becoming a bigger problem as home users switch to broadband connections behind firewalls, utilize network address translator (NAT) and extensively use media applications. Where UDP is used it also causes the problem of higher than desired traffic on the Internet as many packets are discarded due to lack of congestion control. The Datagram Congestion Control Protocol (DCCP) [7], [8] is a new transport protocol, but it is no longer too young to be usable, since the first RFCs were published in 2006, and a stable and quite complete Linux implementation exists. But how good is the service provided to applications by this protocol? And how the congestion control works among the same sets of traffic flow?

In this paper, the behavior of congestion control of TCP and DCCP is investigated and we present results of experiment evaluations in network states varying queue size, link capacity and packet delay.

## II. TRANSPORT LAYER PROTOCOL

In computer networking, Transport Layer provides end-to-end communication services. When data transmitted, the transport layer gets data from Application layer and divides them into several data packets. In this section, fundamental knowledge of typical protocols in this layer which are TCP, UDP and DCCP are described.

### A. TCP

TCP is meant for highly reliable end-to-end protocol. Web browser is one of the best examples of TCP applications. To establish a connection, TCP uses a "three-way handshake". Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections. Once the passive open is established, a client may initiate an active open. Figure 1 shows the three-way handshake. Firstly, client send SYN packet and server reply as sending SYN+ACK packet. By replying ACK, connections between client and server are established, and terminals can send data packet. It goes the similar to the disconnection [9].

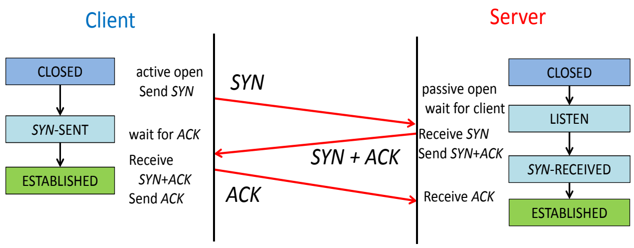



Fig 1. TCP Three-way handshake

When sending data, TCP control the number of transmission data by maintaining *congestion window* to avoid congestion. When receiver received data packet, every packet reception, has to send ACK packet to sender. If sender could not receive ACK for certain period of time, it will retransmit same data packet. Using these controls, TCP provides packets reliability. The advantage and the drawback of TCP are as below.

*Advantage*

・Guarantee packets delivery

・Friendly to other protocol or other session packet.

*Drawback*

・High latency due to some process

### B. DCCP [10], [11, [12]

DCCP is a massage oriented transport layer protocol that implements reliable connection setup, tear down, and congestion control. It is used for applications that have strict timing constraints on the delivery of data. It also provides a congestion control mechanism at user's choice but without data retransmission. In DCCP, there are choices of congestion control mechanisms which are made via Congestion Control identifiers (CCIDs). CCID2 and CCID3 are the mature identifiers and already implemented in Linux OS.

#### 1) *CCID2* [13]

CCID2 provides a TCP-like congestion control mechanism that describes Additive Increase Multiple Decrease (AMID). This mechanism has the following features: [5]

a) Sender maintains a congestion window and sends packets until that window is full.

b) One ACK per 2 packets by default.

c) ACK declares exactly which packets were received.

d) Dropped packets and ECN (Explicit Congestion Notification) indicate congestion.

e) Response to congestion is to halves the congestion window.

f) ACK contain the sequence numbers of all received packets within some window related to selective ACK (SACK)

#### 2) *CCID3* [14], [15]

CCID3 or TCP-Friendly Rate Control (TFRC) is an equation-based and rate-controlled congestion control mechanism. TFRC is designed to be reasonably fair when competing for bandwidth with TCP-like flows. TFRC congestion control in DCCP's CCID3 uses a different approach. Instead of a congestion window, a TFRC sender uses a sending rate. The receiver sends feedback to the sender roughly once per round trip time (RTT) reporting the loss event rate. The sender uses this loss event rate to determine its sending rate. If no feedback is received for several round-trip times, the sender halves its rate.

### C. *Congestion Control Difference in TCP and DCCP*

The main difference between this two is that DCCP packet is datagram, and TCP packet is segment. So, DCCP does not have to retransmit. Other differences are as below.

・DCCP uses ACK as a detection of congestion, TCP uses ACK as a prompting retransmission.

・DCCP does congestion control for not only data packet also ACK by using ACK ratio control.

・Terminal can send ACK packet when process of header using DCCP. Using TCP, terminal can't send ACK packet, until process all data.

When establish and disconnect end-to-end connection, DCCP uses hand shake, as same as TCP. However when data packet are sent, DCCP and TCP are different. There are lots of work has been done on comparing the performance of streaming video over DCCP with TCP and UDP showing promising results [16], - [21].

## III. SIMULATION SETUP

In this section, the performance of TCP, UDP and DCCP are compared by referring some papers related, with modification to the simulation architecture. The network simulation topology used is classic dumb-bell which is a very common topology that has been used in many network simulations. The network simulator NS-2 version 2.35 was used in this simulation. The DCCP patch in this version is based on the patch written by Nils-Erik Mattsson for NS-2 version 2.26 [22], [23], [24]. The default parameters are set as follows. All the senders and receivers are connected as stated in figure 2 and table 3 as the default values. The connection to the routers is through 10 Mbps links with 2ms propagation delay.

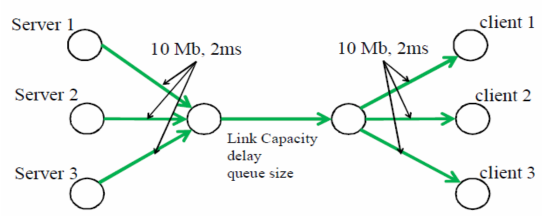



Fig. 2. Simulation Topology

Table 3. Default Parameters

| | |
|---|---|
| TCP maximum window size | 20 |
| Queue Size | 20 |
| Link Capacity | 10 Mb |
| Delay | 2 ms |
| Packet size | 500 byte |

In this simulation environment, the performance of those protocols area measured and compared with changing value - the bottleneck's link capacity, the queue size and the packet delay value. To simulate video streaming, Constant Bitrate (CBR) is used in this simulation. Table 4 shows the traffic time for each client.

Table 4. Start / Stop time traffic flows

| | Client1 | Client2 | Client 3 |
|---|---|---|---|
| Start time | 0 s | 5 s | 10 s |
| Stop time | 50 s | 50 s | 50 s |

## IV. RESULTS AND ANALYSIS

### A. *TCP*

Fig.3 shows the result using default value as mentioned in Table.1. This figure shows, Client 1 flow start first seems to have advantage over the bandwidth. In this simulation we did not set client priority.

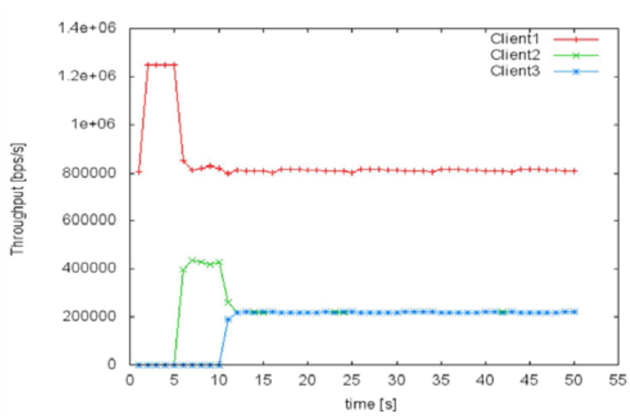



Fig. 3. TCP default value

When the link queue size is increased from 20 to 100, all clients shared the bandwidth fairly as show in Fig. 4. Bigger queue can accommodate packet, so all client's packet can stack in queue and transmitted fairly.

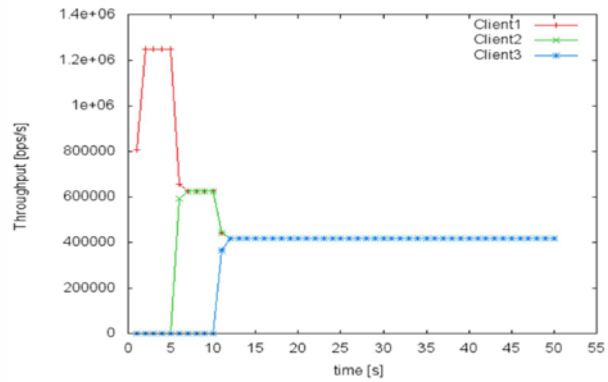



Fig. 4. TCP Queue Size = 100

Figure 5 is a result when the TCP link capacity increased from 10Mb to 20 Mb. It seems that the network can accommodate more packet, and all client's throughput are increased compared to previous results, but still client1 has advantage compared to other clients who start late.

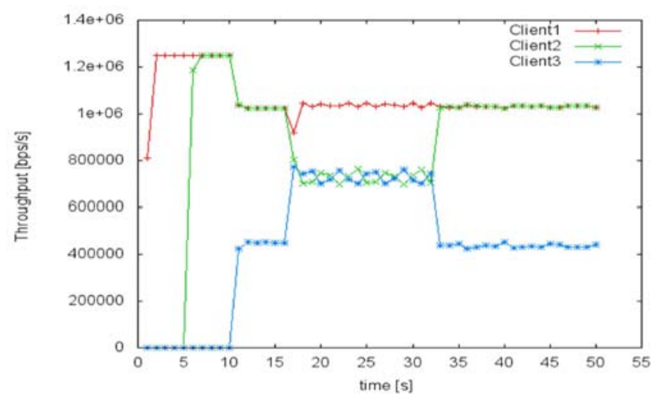



Fig. 5. TCP Link Capacity = 20Mb

Default value for Link delay is set at 2ms. Following results was the behavior when we increased the delay value to 10ms, and 50ms. Fig.6 and Fig 7 shows the result of link delay = 10ms, 50ms respectively. Due to delay, throughput decrease, however clients could share bandwidth with each other.

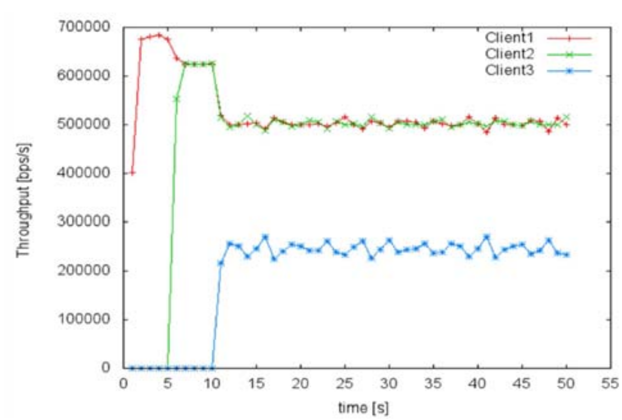



Fig. 6. TCP Link delay = 10ms

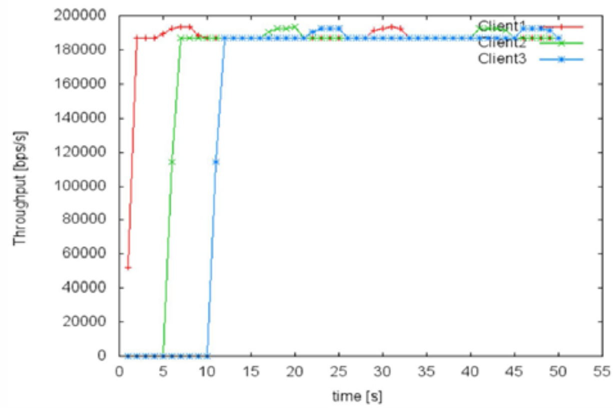



Fig. 7. TCP Link delay = 50ms

*B. DCCP – CCID2*

Figure 8 shows a result of DCCP – CCID2 using default value in table 3. Compare to TCP, CCID2 could have fair share with each other, but until 25 second we can see a fluctuation among clients. This big fluctuation may have influence of quality of service or TCP-Like congestion control mechanism.

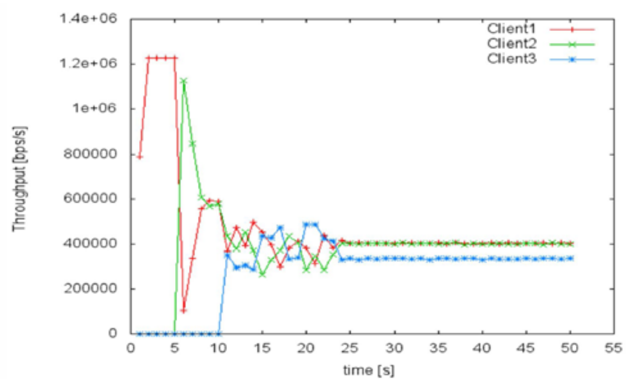



Fig. 8. DCCP-CCID2 using default value

When we increase the queue size, the fluctuation among clients was bigger and took longer time to get stable. However, all clients still consume bigger bandwidth. From this result (figure 9) when the queue size is big, RTT may longer in NS2.

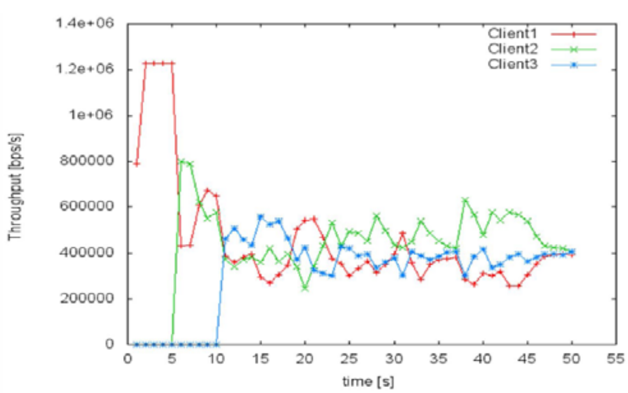



Fig. 9. DCCP-CCID2 Queue Size = 100

Figure 10, is the result of CCID2 when the link capacity was increased to 20Mb. We can see the throughput was increased and still clients have a fairly shared bandwidth.

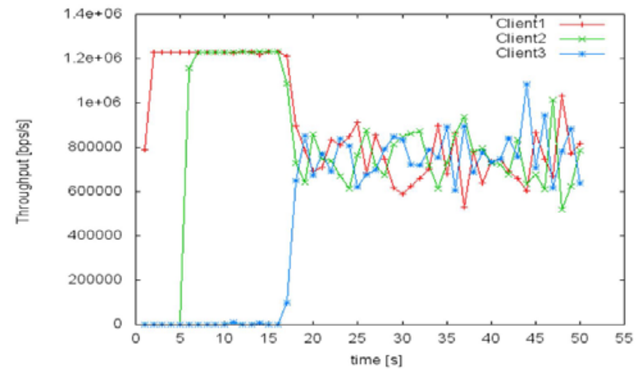



Fig. 10. DCCP-CCID2 Link Capacity = 20Mb

Figure 11 and 12 are the result when the link delay was set to 10ms and 50ms. From the graph, we can see a big fluctuation, and the fluctuation velocity get low. This could because of CCID2 control traffic by prediction and ACK packets. But due to long a delay, sender cannot get ACK information rapidly. And causing congestion window to increase until congestion happened. Sometimes sender couldn't get ACK packets within the certain times, and congestion window get decreases.

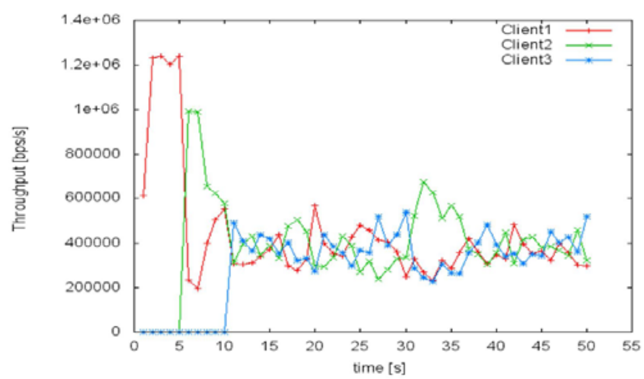



Fig. 11. DCCP-CCID2 Delay = 10ms

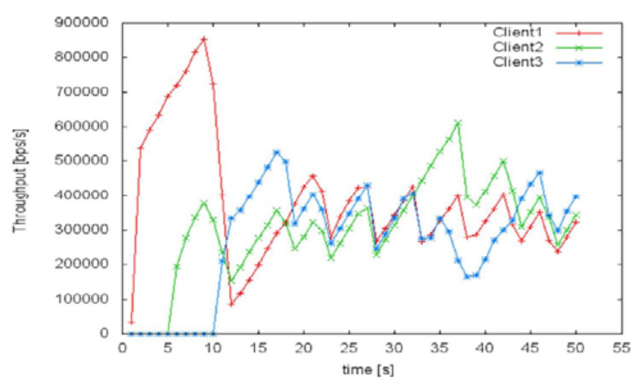



Fig. 12. DCCP-CCID2 Delay = 50ms

### *C. DCCP-CCID3*

The result of using default value for CCID3 is shown in Figure 13. Compare to CCID2's result in figure 8, fluctuation in CCID3 is smaller, and the throughput line seems smoothly. CCID3 known to be more suitable for application such as VoIP and Video streaming.

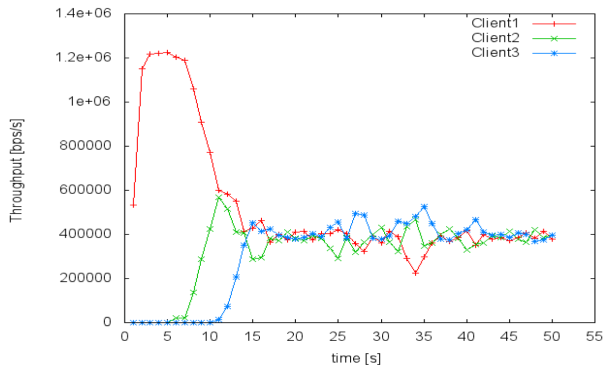



Fig. 13. DCCP-CCID 3 using default value

Figure 14 is the result when the queue size was increased to 100. From this result, when queue size is big, fluctuation is small and it took longer time for new coming packet to stabilize the throughput.

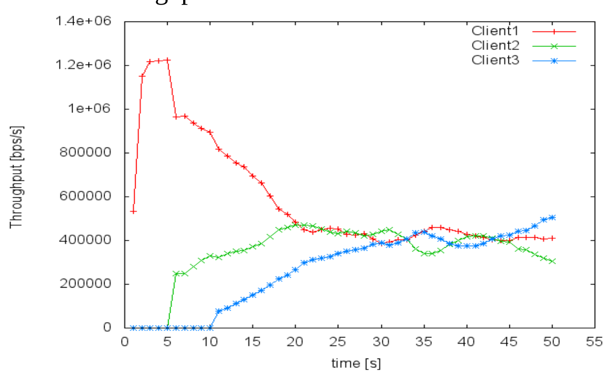



Fig. 14. DCCP-CCID 3 Queue Size = 100

When the link capacity increased to 20Mb, we can see from the graph (Figure 15) that throughput increased and the performance are not that fluctuated. This behavior shows the CCID3 (TFRC) uses a smooth rate adjustment.

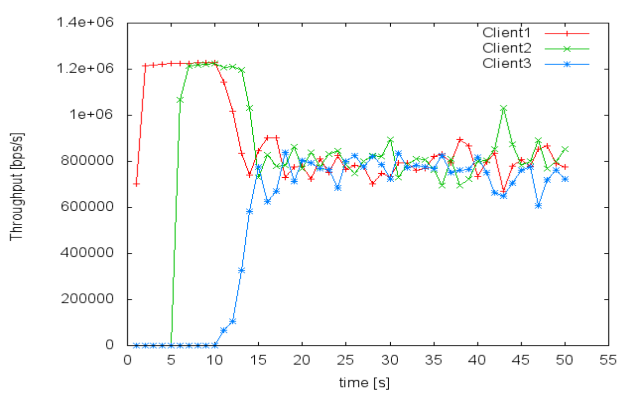
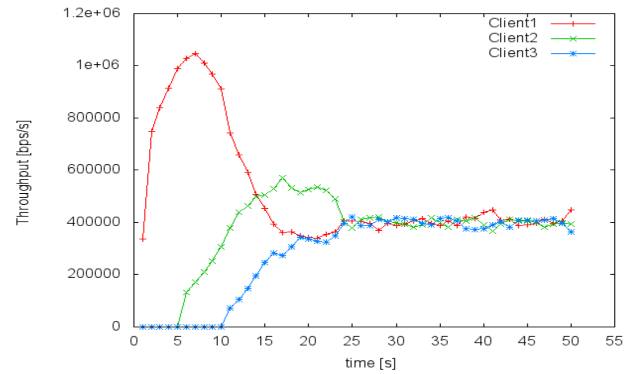



Fig. 15. DCCP-CCID 3 Link Capacity = 20Mb

On the delay test for the CCID 3 shows significant difference compare to other protocol. Figure 16 shows the effect when we increased the delay to 10ms. Here the graph still shows similarity to the default graph (figure 13). Only the throughput need time to be stabilized. But, from figure 17, we can see that the CCID3 cannot transmit packet when the delay equal to 50ms. CCID3 control traffic by received ACK. However, within certain time if sender cannot receive ACK packets, CCID3 decrease the traffic to 1 packet. So, high delay has bad impact on CCID3.




[1]

Fig. 16. DCCP-CCID 3 Delay = 10ms

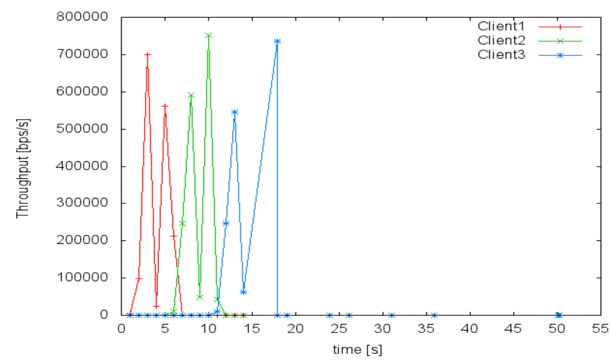



Fig. 17. DCCP-CCID 3 Delay = 50ms

## V. CONCLUSION

Many research papers discussed on the DCCP effect over TCP and UDP. In this paper, an experimental of TCP, DCCP-CCID2 and DCCP-CCID3 behavior over three servers-clients on the same network is presented. It is shown that certain time of delay can give a bad impact to the TCP and DCCP transmission. However, DCCP-CCID2 (TCP-Like) still can react to the situation compare to DCCP-CCID3 (TFRC). In term of capability to have a fairly bandwidth tolerance among other transmission, DCCP shows a better result compare to TCP where the later packet in the network normally will not have chances to gain better throughput. Among all, DCCP-CCID3 (TFRC) shows the best traffic flows as they have a smooth adaptation to make the total transmission less fluctuate and suitable for Video streaming and VoIP.

## ACKNOWLEDGEMENT

The authors would like to express their heartiest gratitude to Telekom Malaysia Berhad for their kind support and generous funding for the projects that partially used for the work presented in this paper.

## REFERENCES

[1] A. M. Tekalp, E. Kurutepe and M. R. Civanlar, “3DTV over IP”, IEEE Signal Processing, November 2007

[2] Tasos Dagiuklas, “3D Media over Future Internet: Current Status and Future Research Directions”, IJCSI International Journal of Computer Science Issues, Vol. 9, January 2012

[3] Jiawei Chen, “Congestion Control Mechanisms of Transport Protocols”, ACM/IEEE Conference on Supercomputing, April 2009.

[4] Michael Schier, Michael Welzl: "Using DCCP: Issues and Improvements", accepted for publication, IEEE ICNP 2012, 30 October - 2 November 2012, Austin, Texas, USA.

[5] Jon Postel, “Transmission Control Protocol,” RFC 793, Sep, 1981.

[6] Jon Postel, “User Datagram Protocol,” RFC 768, Aug, 1980.

[7] E.Kohler, M.Handley, S.Floyd, “Datagram Congestion Control Protocol (DCCP),” RFC 4340, March 2006.

[8] S. Floyd, M. Handley, E. Kohler, Problem Statement for the Datagram Congestion Control Protocol (DCCP), IETF RFC 4336, March 2006.

[9] M. Allman, V. Paxson, W. Stevens TCP Congestion Control, IETF RFC5681, September 2009.

[10] E. Kohler, M. Handley, and S. Floyd, “Designing DCCP: Congestion control without reliability,” in ACM SIGCOMM, vol. 36, Pisa, Italy, Sep. 2006

[11] L. M. de Sales, “Developing multimedia applications with DCCP congestion control,” Linux Magazine, pp. 56–62, Aug. 2008.

[12] S. Floyd, E. Kohler, Profile for DCCP Congestion Control ID 4: the Small-packet Variant of TFRC Congestion Control, IETF RFC 5622, August 2009.

[13] IETF RFC 4341, “Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control”, March 2006

[14] S. Floyd, E. Kohler, J. Padhye, Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC), IETF RFC 4342, March 2006. E. Kohler, M. Handley, and S. Floyd, “Datagram Congestion Control Protocol (DCCP),” RFC 4340, Mar. 2006.

[15] IETF RFC 4342, “Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)”, March 2006

[16] Shigeki Takeuchi, Katsuyoshi Iida, “Performance Evaluations of DCCP for Bursty Traffic in Real-time Applications,” Applications and the Internet, Feb. 2005.

[17] Xiaoyuan GU, Kars WOLF, “Performance evaluation of DCCP: A focus on smoothness and TP-friendliness,” ANN, TELECOMMUN, 2006.

[18] Muhammad Ajmal Azad, Tahir Mehmood, “A Comparative Analysis of DCCP Variants (CCID2, CCID3), TCP and UDP for Mpeg4 Video Applications,” ICICT, 2009.

[19] Yao-Nan Lien, Yu-Chi Ding, “Can DCCP Replace UDP in Changing Network Conditions?” International Conference on Advanced Information Networking and Applications, 2011.

[20] Muhammad Ajmal Azad, Tahir Mehmood, “A Comparative Analysis of DCCP Variants (CCID2, CCID3), TCP and UDP for Mpeg4 Video Applications,” ICICT, 2009.

[21] Yao-Nan Lien, Yu-Chi Ding, “Can DCCP Replace UDP in Changing Network Conditions?” International Conference on Advanced Information Networking and Applications, 2011.

[22] N. E. Mattsson, "A DCCP module for NS-2", http:// http://lifc.univ-fcomte.fr/~dedu/ns2/dccp.pdf, May 2004.

[23] http://www.morikita.co.jp/soft/84621/

[24] http://www.isi.edu/nsnam/ns/

## AUTHORS PROFILE

**Sahrul Hilmi Ibrahim** received his Bachelor Degree of Engineering in Electronics Engineering from Saga University, Japan in 1998. He then worked as research engineer in Kyushu Matsushita Electric Co. (R&D lab) before served Telekom Malaysia R&D in 2002 as a full researcher. Obtained his Master degree in 2009 and presently he is pursuing his PhD in Electrical Engineering from Universiti Teknologi Malaysia. He is currently working as senior researcher in Advanced Communication Network Lab, TM R&D.

**Yuki Matsuzawa** is from University of Electro-Communications, Tokyo Japan. He participating the UEC Overseas Internship Program at MMU and TMRND to gain experience working in international environment.

**Noor M. Asraf** graduated with a BS in Computer Science from Western Michigan University in 2003 and obtained his MSc from University Malaya in 2011. He is currently working as a researcher at Telekom Malaysia R&D.

**Salahuddin Mohd Ali** has over sixteen years' experience in Telekom Malaysia in the technical support and operation and seven years in TM Research and Development. He begins as an engineer and progress to being the manager in support and operation.

**Siti Aufa Mohamad Rofie** received the Bachelor Degree in Computer Science (Computer Network) from Universiti Putra Malaysia, Serdang, Malaysia, in 2006. She has been working for TMR&D as Senior Assistant Researcher for almost 7 Years.

**Jalil Md. Desa** have completed my study at the University of Portsmouth, with a Bachelor Honors Degree in Electrical & Electronic Engineering, and completed his PhD in Electronic & Electrical Engineering at the University of Strathclyde and majoring in optical high-speed network design. Currently he is working in TM R&D as Head of Laboratory of Advanced Communication Network Lab.

**Sevia M. Idrus** is a member of Faculty of Electrical Engineering, Universiti Teknologi Malaysia. She received Bachelor in Electrical Engineering (1998) and Master in Engineering Management (1999) both from the Universiti Teknologi Malaysia. She obtained her PhD in 2004, in the area of optical communication system engineering from the University of Warwick, United Kingdom. She served Universiti Teknologi Malaysia since 1998 both as the academic and administrative staff.

**Nadiatulhuda Zulkifli** is a member of Faculty of Electrical Engineering, Universiti Teknologi Malaysia where she received her Bachelor in Electrical – Telecommunication Engineering in 2001. She received her Master (2003) and PhD (2009) degrees both from the University of Essex, UK.