Performance Analysis of Datagram Congestion Control Protocol (DCCP)

Iffat Sharmin Chowdhury, Jutheka Lahiry, Kazi Chandrima Rahman, *Member, IACSIT,* and Syed Faisal Hasan

***Abstract*—Applications like streaming audio, Internet telephony and multi-player online games prefer timeliness in packet delivery to reliability. TCP's reliability through packet retransmission and abrupt rate control features are unsuitable for these applications. As a result, these applications prefer UDP as the transport layer protocol. UDP does not have any congestion control mechanism which is vital for the overall stability of the Internet. For this reason, a new transport layer protocol--Datagram Congestion Control Protocol (DCCP) has been introduced by the Internet Engineering Task Force (IETF). DCCP is suitable for these applications because of its exclusive characteristics. It can be useful for those applications which need a session and congestion control unlike UDP and do not need reliability or retransmission like TCP. However, since DCCP is a new protocol, its performance for these applications has to be analyzed thoroughly before it emerges as a de facto transport protocol for these applications. This paper describes the basic principle of DCCP, its congestion control mechanism and measures the performance of DCCP. The results show that DCCP provides better performance for those applications that suffers the tradeoff between delay and in-order delivery.**

***Index Terms*—CCID2, CCID3, congestion control, DCCP, IETF, streaming multimedia.**

I. Introduction

Now-a-days, almost every website includes streaming multimedia applications. So, it is really essential to send audio video files in time. Generally, Internet uses Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) for sending application data. TCP ensures ordered packet delivery and reliability. So, when a packet is lost, TCP retransmits it and all the following packets have to wait and TCP considers that network is congested. Then TCP sender reduces its sending rate and that rate may not meet the requirements of streaming multimedia. So, TCP is not suitable for streaming multimedia applications. On the other hand, if we use UDP for those applications, then it may be quite impossible to recover from congested network because UDP has no congestion control mechanism. For this reason, a new transport layer protocol, DCCP (Datagram Congestion Control Protocol), is proposed by IETF [1]. It is a message oriented transport layer protocol. It provides reliable connection setup, congestion control and feature negotiation. It is useful for those applications where timing constraints exists in delivery of data but does not require reliable ordered delivery. DCCP does not provide congestion control at the application layer. It has built in congestion control mechanism. Two congestion control mechanisms of DCCP are TCP-like (Congestion Control IDentifier 2) and TCP-friendly (Congestion Control IDentifier 3). This is useful for those applications where a steady rate of data transmission is required rather than reliable in order delivery of packets. Some experiments have been done to measure the performance of DCCP's congestion control mechanism [2]. Those experiments evaluate the performance of TCP, UDP, and DCCP. In this paper, the performance analysis of DCCP's two alternative congestion control mechanism is illustrated as DCCP is completely a new protocol. It is still under research whether DCCP can be used for real time applications practically. The objective of the work is to measure the performance of TCP and DCCP at various environments and to show whether the performance of DCCP is better or not.

From the experiments given later, it can be ensured that there are no abrupt changes in bit rate of CCID 3. So, CCID 3 can be used for those applications that needs smooth rate.

The rest of the paper is organized in the following ways-- Section I introduces DCCP, section II depicts some related works, section III illustrates background study, section IV and V explain performance evaluation goal and testing consecutively, section VI demonstrate contribution and future works.

II. Related Works

Floyd et al. [13] initially proposed and introduced the definition of TCP-friendly flows.

In an experimental study, Timothy Sohn and Eiman-Zolfaghari [14] reported an initial implementation and experimentation of Datagram Control Protocol (DCP) and its equation-based congestion control mechanism to show its TCP-friendliness behaviors. Horia Vlad Balan, Lars Eggert, Saverio Niccolini and Marcus Brunner [15] evaluated the voice quality that Internet telephony calls achieve over prototype implementations of basic DCCP and several DCCP variants, under different network conditions

Manuscript received March 31, 2011; revised September 22, 2011.

Iffat Sharmin Chowdhury is with the Department of Computer Science and Engineering, University of Dhaka, Bangladesh. (e-mail: sharmin.iffat@gmail.com).

Juthika Lahiry is with the Department of Computer Science and Engineering, University of Dhaka, Bangladesh. (e-mail: jutheka@gmail.com).

Kazi Chandrima Rahman is with the Department of Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh. (e-mail: kchandrima@ yahoo.com).

Dr. Syed Faisal Hasan is with the Department of Computer Science and Engineering, University of Dhaka, Bangladesh. (e-mail: hasansf@gmail.com).

and with different codecs. Saleem Bhatti, Martin Bateman and Dimitris Miras [16] compared the performance of DCCP CCID2 relative to TCP New Re-no. They assessed overall throughput and fairness-how well these protocols might respond to each other when operating over the same end-to-end network path.

Unlike their work, we focus on the performance of CCID 2 and CCID 3 relative to TCP.

III. Background Study

There are two very common transport layer protocols named TCP and UDP. Recently, a new transport layer protocol DCCP is invented to meet various dynamic changes of network bandwidth.

A. TCP Congestion Control

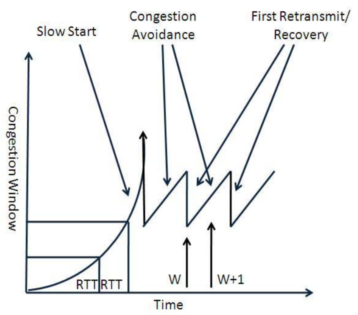

Fig. 1.TCP congestion control mechanism

Fig. 1 depicts that when a TCP connection begins, the value of congestion window (CongWin) is typically initialized to 1 MSS (RFC 3390), resulting an initial sending rate of roughly MSS/RTT. After every RTT, the sender increases its rate exponentially by doubling its value of CongWin. TCP congestion control algorithm behaves differently after a timeout event than after the receipt of triple duplicate ack. After a timeout event, CongWin is reduced to 1 MSS that is called Slow Start phase. Whereas after receiving triple duplicate acknowledgements, it only cuts it's CongWin in half and then grows linearly. The cancelling of the slow start phase after triple duplicate acknowledgements is called fast recovery.

B. Datagram Congestion Control Protocol (DCCP)

Datagram Congestion Control Protocol (DCCP) is message and connection oriented transport layer protocol. It differs from UDP, in that, it includes congestion control mechanism and it differs from TCP, in that, it does not provide guaranteed reliability.

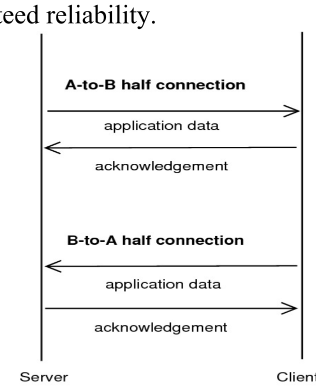

Fig. 2.DCCP's two half connections

C. B.1 The DCCP Connection

DCCP implements bidirectional connections between hosts. The connection is established between two hosts and any host can initiate the connection [3]. Data may passes from any host to another host which is depicted in Fig. 2. A DCCP connection consists of two unidirectional connections, called half-connection but this distinction is logical [4].

D. B.2 DCCP Packet Structure

The DCCP header consists of 12 to 1020 bytes and the first part of the header is the same for all packet types. After the generic header, comes the additional fields which depend on types of packets and then comes variable length optional field of options. Application data follows the header and the packet structure [4] is depicted on Fig. 3.

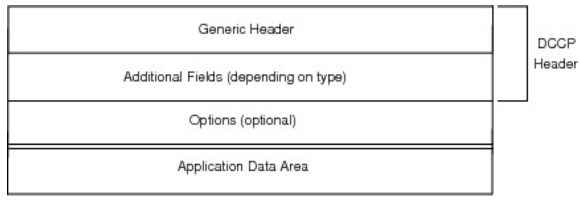

Fig. 3.DCCP packet structure

E. B.3 Unreliable Data Transfer

Each DCCP packet carries a sequence number so that losses can be detected and reported. But there is no re-transmission of lost packets and hence DCCP is an unreliable protocol.

F. B.4 DCCP Connection Management

DCCP server and client go through many states when establishing a connection between them. The steps are depicted at Fig. 4.

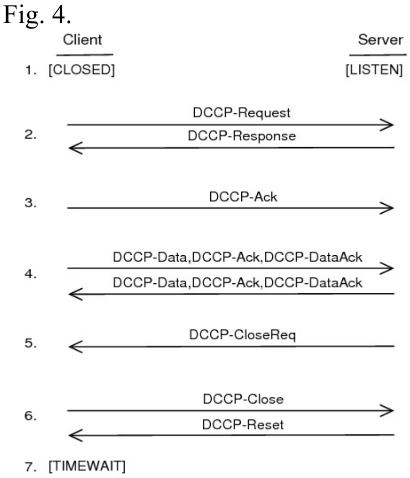

Fig. 4.DCCP's connection management

- Client is in closed state and server is in listening state.
- Client sends DCCP request, which specifies server and client ports, and server sends DCCP response to specific client, which means the willingness of server to exchange messages.
- Client sends DCCP acknowledgement to server to inform that DCCP response is received.
- Server and client then exchange DCCP-Data, DCCP-Ack and DCCP-DataAck packets, which includes piggybacked acknowledgement.
- Server sends DCCP-CloseReq to client for re-questing to close the connection.

• Client acknowledges the request by sending DCCP-Close packet. Server then sends DCCPReset packet and clears its connection state.

• Client receives DCCP-Reset packet and holds the time wait state for two maximum segment lifetimes to allow on transit packets to clear the network.

G. DCCP Congestion Control

DCCP implements congestion control and the user of the applications can make a choice of congestion control mechanisms. The two hosts agreed on the congestion control mechanism during the initiation of the connection. One byte congestion control identifier called CCID, defines the mechanisms. Among various Congestion Control IDentifier, CCID 2 and CCID 3 are well de-fined.

H. C.1 CCID 2

CCID 2 is TCP like congestion control mechanism. It is perfect for those applications which can adapt to the changes of congestion control window and which need as much bandwidth as possible in the network. CCID 2 uses TCP like congestion control mechanism [5]. There are some particular features of CCID2 connection-

• Duplicate acknowledgement indicates some loss of data packet.

• The sender has timeout option, which is handled like TCP's retransmission timeout. The sender calculates round trip time for a window at most once and uses TCP's algorithm for maintaining the round trip time.

After a congestion event occurs, CCID 2 reduces its congestion window (cwnd). Every congestion event consists of explicitly indicated that is ECN marked or via duplicate acknowledgements. For this case, cwnd is halved.

I. C.2 CCID 3

CCID 3 is TCP friendly rate control mechanism [6]. It provides TCP friendly rate by reducing the changeable characteristics of TCP or TCP like congestion control. The sender maintains its sending rate by observing the loss event send by the receiver and goes through a constant sending rate [7].

CCID 3 uses TCP friendly rate control mechanism for congestion control. The DCCP sender calculates its transmission rate based on the following equation-

$$T = \frac{s}{R\sqrt{\frac{2bp}{3}} + t_{RTO}\left(3\sqrt{\frac{3bp}{8}}\right)p(1+32p^2)}$$

- T = transmission rate in bytes/second
- s = packet size in bytes
- R = round trip time in seconds
- b = number of packets acknowledged by a single TCP acknowledgement
- p = loss event rate
- tRTO = TCP retransmission time out value in seconds

This results a fair smooth transmission rate which is required for real time applications.

IV. Performance Evaluation Goal

In this paper, we have compared the performance of TCP and CCID2 and CCID3 of DCCP based on throughput. We have varied loss rate and delay. The performance is compared by sending fixed size packet because the audio/video streaming applications sent fixed size packets.

V. Testing

For these experiments, we had to compile the Linux kernel [8] and we used some tools like-iperf [9], GnuPlot [10] and a network emulator [11]-Netem. The SI unit for magnetic field strength *H* is A/m. However, if you wish to use units of T, either refers to magnetic flux density *B* or magnetic field strength symbolized as $\mu_0 H$. Use the center dot to separate compound units, e.g., "A·m^2."

A. Experimental Setup

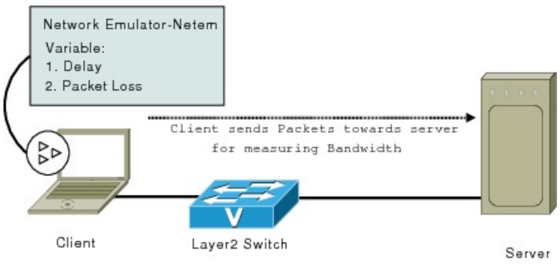

Fig. 5.Testbed Configuration

The setup of two machines is illustrated in Fig. 5. One machine acts as DCCP server and other machine acts as DCCP client. Client machine is used to emulate network changes. The server machine acts as a sink. These network conditions are applied on the interface of the client machine using Netem [12] functionality.

B. Performance Evaluation

We are going to show the results and to analyze whether our goal has been satisfied or not. We will also discuss the various behaviors of the transport layer protocols in different environment.

C. B.1 Time vs Bit Rate

The transmission rate of TCP, CCID 2 and CCID 3 at time interval 1 second and total transmission time 10 seconds are shown in the following graph.

Fig. 6 shows the bit rate in Mbps to y-axis corresponding to the time in seconds to x-axis. The graph shows us that the bit rate of TCP and CCID 2 which is TCP like congestion control are high and the bit rate of CCID 3 which is TCP friendly rate control is low and a bit smooth.

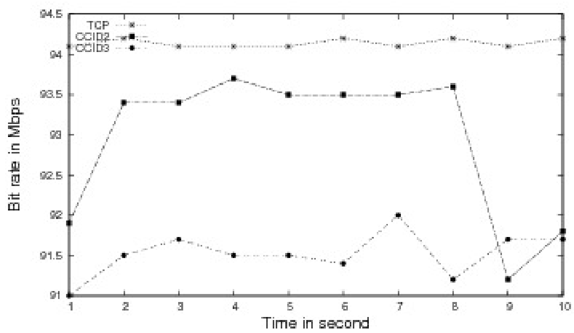

Fig. 6.Time vs. Bit rate of TCP, CCID 2 and CCID 3

In the graph, we observe that at interval 1-2 and 8-9 to the time axis, in CCID 2, there are sudden changes in bit rate. But there is no abrupt change in CCID 3. As there is no loss of data between sender and receiver, there is no sharp rise and fall on TCP throughput. In our experimental setup, the link bandwidth was 100 Mbps, so TCP throughput can't exceed this range.

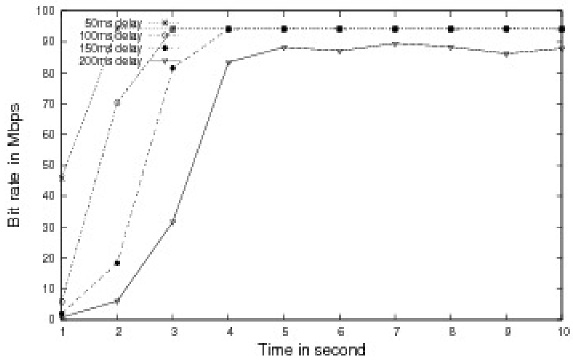

Fig. 7.Time vs Bit rate of TCP in varying delay

D. B.2 Delay Variant

Fig. 7 shows TCP's behavior in varying delay. The X-axis shows the time in seconds and Y-axis shows the bit rate in Mbps. The graph shows the bit rate with delay 50ms, 100ms, 150ms and 200ms.

With low delay, TCP's transmission rate is very much high and it looks like a straight line. That means, there are no changes in bit rate as there was no congestion event or loss event in the environment at that moment. From Fig. 7, we see that with the increasing of delay, the bit rate decreases. With delay 200 ms, the bit rate is lower than with delay 150 ms.

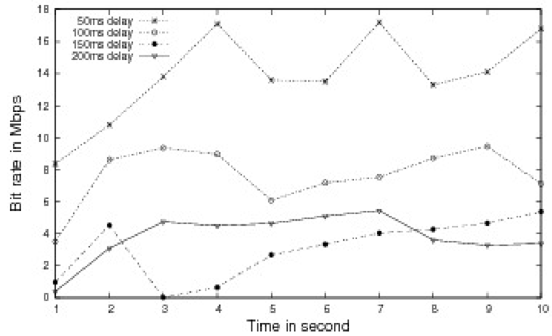

Fig. 8.Time vs Bit rate of CCID 2 in varying delay

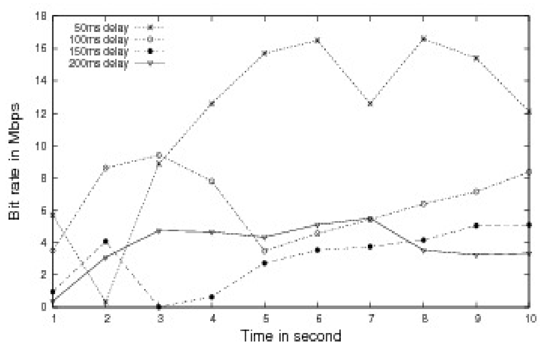

Fig. 9.Time vs Bit rate of CCID 3 in varying delay

Fig 8. and Fig. 9 shows CCID 2's and CCID 3's behavior in varying delay consecutively. Again the X-axis shows the time in seconds and Y-axis shows the bit rate in Mbps. The graphs show the bit rate at delay 50ms, 100ms, 150ms and 200ms. There is a subtle difference between the graph for TCP and the graphs for CCID 2 and CCID 3. CCID 2 and CCID 3's bit rate are very much low with delay 200ms.

E. B.3 Loss Variant

Fig. 10 shows TCP's behavior in varying loss rate. The X-axis shows the time in seconds and Y-axis shows the bit rate in Mbps. The graph shows the throughput of TCP at loss rate 0%, 1%, 3%, 5%, 10% and 15%. When graph goes down, we can say, at that moment, a loss has occurred and TCP sender reduces its transmission rate. When maximum loss occurs, sharpness can be seen on the graph.

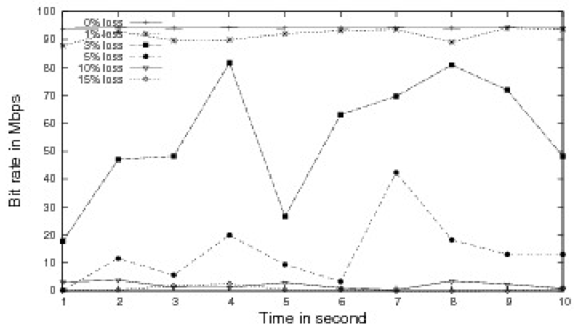

Fig. 10.Time vs Bit rate of TCP in varying loss rate

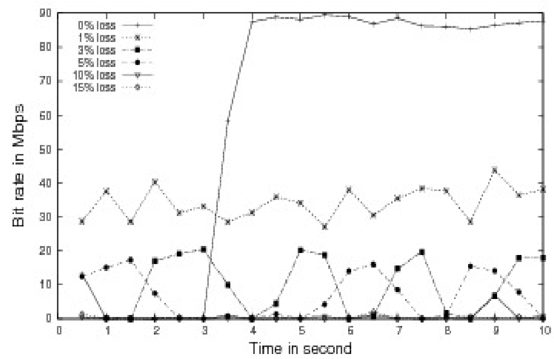

Fig. 11.Time vs Bit rate of CCID 2 in varying loss rate

Fig. 11 shows CCID 2's behavior in varying loss rate.

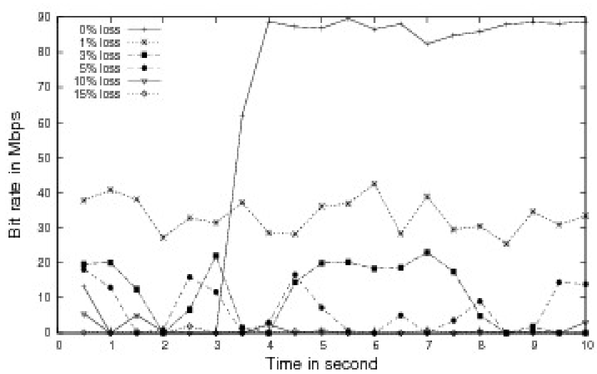

Fig. 12.Time vs Bit rate of CCID 3 in varying loss rate

Fig. 12 shows CCID 3's behavior in varying loss rate. Again the X-axis shows the time in seconds and Y-axis shows the bit rate in Mbps. The graph shows the bit rate at loss rate 0%, 1%, 3%, 5%, 10% and 15%. Sharp fall of the graph is less than the previous graph of TCP. But if we compare the graph for CCID 2 and CCID 3, we can see that CCID3 falls smoothly than CCID 2.

F. B.4 Behavior of CCID 2 and CCID 3

If we integrate the graph of CCID 2 and CCID 3, then we find the graph like Fig. 13.

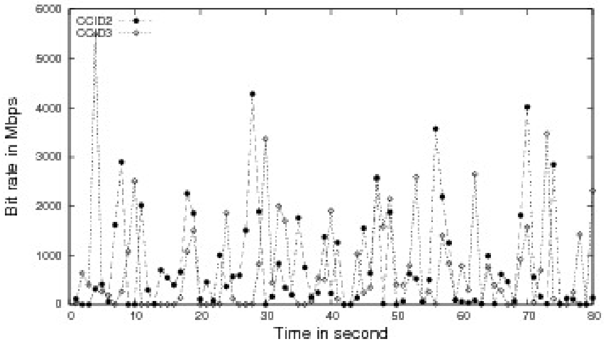

Fig. 13.Time vs Bit rate of CCID 2 and CCID 3 on 10% loss

The bit rate of the graph shows the characteristics of CCID 2 and CCID 3 with 10% loss. From this graph, we can see that the graph of CCID 2 is sharper than CCID 3 and CCID 3 goes smoothly.

The graphs which we have presented have been taken by emulating network. In the same environment and in the same network conditions, the bit rate may change and the graph behavior may change too. Even in some environment, CCID 2 changes smoothly than CCID 3. But most of the time, CCID 3 will be a bit smooth in bit rate than CCID 2. For these reasons, DCCP is till now experimental. To fix the characteristics of this protocol, many researches are going on DCCP.

VI. Contribution and Future Works

In this paper, we have compared the protocol behavior of DCCP (Datagram Congestion Control Protocol) with mostly used protocol TCP. We have also compared the characteristics of Congestion Control Identifier 2 (CCID 2) and Congestion Control IDentifier 3 (CCID 3).

- From our experiments, we observe that with low delay, throughput of TCP is high and throughput of CCID 2 is close to TCP. Though the throughput of CCID 3 is not so good, its rate of changes is much smoother than CCID 2.
- From experimental graphs, we can ensure that there are no abrupt changes in bit rate of CCID 3. So, CCID 3 can be used for those applications that needs smooth rate.
- In varying loss rate, we found that TCP and CCID 2 take advantages of the available band-width in an environment. On the other hand, CCID 3 does not use as much bandwidth as possible because it tries to minimize abrupt changes of bandwidth.

From all the experimental results it is clear that CCID 3 maintains a fair rate. It can be stated that CCID 3 can be a better choice for real time applications. DCCP can be used for those applications that suffer the trade-off between delay and in order delivery.

In future, we will further study, how to satisfy the dynamic requirements for multimedia applications. We will also try to transfer live audio/video files using congestion control mechanism CCID 3 and evaluate DCCP to long range wireless links. Finally, because DCCP is directed towards those applications which currently use UDP without any form of end-to-end congestion control, an area of interest would be to implement a layer of reliability on top of the DCCP layer.

Acknowledgement

We are grateful to chairperson, Professor Dr. Suraiya Pervin, Department of Computer science and Engineering, University of Dhaka, for her support and help and for giving us the opportunity to carry out the project work by providing us with the necessary re-sources and materials.

References

[1] S. Floyd, M. Handley, and E. Kohler. Problem statement for Datagram Congestion Control Protocol. IETF, RFC 4336, LA, USA, 2006.
[2] Stanimir Statev and Seferin Mirtchev. Experi-mental study of datagram congestion control protocol in varied network states. Technical re-port, Technical University of Sofia, Bulgaria, 2008.
[3] lwn.net-linux info source http://lwn.net/Articles/149756/
[4] E. Kohler and M. Handly and S. Floyd. Data-gram Congestion Control Protocol (DCCP). IETF, RFC 4330, LA, USA, 2006.
[5] S. Floyd and E. Kohler. Profile for Datagram Congestion Control Protocol (DCCP), Conges-tion Control ID 2: TCP-like Congestion Con-trol. IETF, RFC 4330, LA, USA, 2006.
[6] Jitendra Padhye, Sally Floyd and Eddie Kohler. Profile for DCCP Congestion Control ID 3: TFRC Congestion Control. Internet Engineer-ing Task Force, 2002
[7] S. Floyd, E. Kohler and J. Padhye. Profile for Datagram Congestion Control Protocol (DCCP), Congestion Control ID 3: TCP-Friendly Rate Control (TFRC). IETF, RFC 4342, LA, USA, 2006.
[8] the linux kernel archives, http://www.kernel.org/
[9] Jon M.Dugan. Using Iperf. Lawrence Berkeley, National Laboratory, NANOG 43, Brooklyn, NY, 2008
[10] gnuplot, http://www.duke.edu/hpgavin/gnuplot.htm
[11] Wikipedia, http://en.wikipedia.org/wiki/Network emulation
[12] Ariane Keller and ETH Zurich. Packet Filtering and netem. tc manual, USA, 2006
[13] S. Floyd and K. Fall, Promoting the Use of End-to-End Congestion Control in the Internet, IEEE/ACM Transactions on Networking, Au-gust 1999.
[14] Timothy Sohn and Eiman Zolfaghari. Experi-mentation of the Datagram Control Protocol. California, USA, 2002.
[15] Horia Vlad Balan, Lars Eggert, Saverio Nicco-lini and Marcus Brunner, An Experimental Evaluation of Voice Quality over the Data-gram Congestion Control Protocol, Appeared in Infocom 2007.26th IEEE International conference on computer communications. IEEE, Publication date:6-12 May 2007,on pages 2009-2017
[16] Saleem Bhatti and Martin Bateman and Dimitris Miras. A Comparative Performance Evaluation of DCCP. London, UK.

Iffat Sharmin Chowdhury received the B.Sc. degree from University of Dhaka, Bangladesh and now is doing M.Sc. from University of Dhaka, Bangladesh. She is going to start research works towards the Ph.D. degree from Fall 2011 at University of Texas, San Antonio, TX.

Her research interests are in networking protocols, network measurement and distributed system.

Jutheka Lahiry graduated in Computer Science and Engineering from University of Dhaka, Bangladesh. She is a M.Sc. student of the same university and working on new transport protocols and different congestion control mechanisms .She joined in SBRC as an engineer of mobile networks and software development. Her research interests focus on the modeling and design of advanced transport protocols and end to end communication mechanisms over the Internet.

Kazi Chandrima Rahman is a Member of International Association of Engineers (IAENG). Membership # 112911. Her place and date of birth is Dhaka, Bangladesh and 27th Janurary, 1979 correspondingly. She Completed her B.Sc. (Hons) in 2000 and M.Sc in 2002, in Computer Science from University of Dhaka, Bangladesh.

She has been serving as an Assistant Professor at the Department of Computer Science and Engineering (CSE), The University of Asia Pacific. She joined in UAP on October 2009 as an Assistant Professor and on October 2005 as a Lecturer. Before, she was a Lecturer at BRAC University from September 2003 to September 2005. She also worked as a Software Developer in Grameen Solutions from May 2002 to January 2003. She has research papers on Research and Educational Network, Sensor Network, VPN and TCP Congestion Control on ICCIT, IJCIT, and IJENS-IJECS. Her research interests are on sensor network, network security, distributed and wireless network etc.

Ms. Chandrima has been selected as Potential Invitees of Emerging Research and Education Network for South Asia Region for her research work on Research and Educational Network. She is an advisor of Technical Journals Online: www.technicaljournalsonline.com and International Journal of Computer and Information Technology: www.ijcit.org and a reviewer of International Journal of Communication Networks and Distributed Systems (IJCNDS). She is a member of International Association of Computer Science and Information Technology (IACSIT). Membership # 80339174. She is a member of Alumni Executive Committee of Dhaka University, CSE Department. Her website: kchandrima.wordpress.com

Dr. Syed Faisal Hasan is currently working as an Assistant Professor at the department of computer science and engineering, University of Dhaka, Dhaka, Bangladesh. He holds a PhD in Computer Science from the University of Otago, New Zealand. His research interest is on TCP congestion control, rate control for streaming media, distributed social networking protocols and grid computing. Current he is representing Bangladesh at the Asia Pacific Advanced Network (APAN). He is also a member of Asia New Zealand foundation's young leaders' network. He is collaborating with Altruists International towards building a trust based distributed social network. He is the founding member of Internet society Dhaka Chapter. He has received Commonwealth Scholarship, Fellowship from Network StarUp Resource Center and the Internet Society.