Asian Journal of Information Technology 15 (20): 3986-3990, 2016
ISSN: 1682-3915

Effective Video Streaming Using Reset-Controlled DCCP along with Data Quality Analysis

[1]B. Chellaprabha and [2]RMD. Sundaram
[1]CSE Department, SNS College of Engineering, Coimbatore, India
[2]Wipro Technologies, Bengaluru, India

Abstract: The Datagram Congestion Control Protocol (DCCP) is a message-oriented transport layer protocol. Multimedia applications including streaming audio, internet telephony and video conferencing has to reach the end user on time and hence prefers timeliness in packet delivery as compared to reliability. Packet retransmission or packet control as provided by Transmission Control Protocol (TCP) is not going to be of much use in these kinds of applications. Hence DCCP is introduced which has better results in terms of performance as compared to other mechanisms. The algorithm is little faulty and hence in this study we have tried to address the limitations of DCCP through modified reset algorithms and also provide a mechanism to verify the content quality at the receiving end especially video without using any reference mechanisms.

Key words: Datagram Congestion Control Protocol, Transport layer protocol, video streaming, no reference mechanisms, linitations

INTRODUCTION

A multimedia application is an application which routines a group of multiple media sources e.g. text, images, sound/audio, video graphics and animation. DCCP is suitable for applications with timing restrictions on the transfer of data. Such applications consist of internet telephony, streaming media, multiplayer online games etc. The main piece of these applications is that old communications quickly become stale so that receiving new communications is favored to resending missing messages. Currently such applications have often either established for TCP or used User Datagram Protocol (UDP) and applied their individual congestion governor mechanisms or have no control on the congestion at all (Rahman *et al*., 2012).

DCCP implements bidirectional, unicast networks of congestion-controlled, unpredictable datagrams. Though it provides reliable handshakes for connection setup and tear down, it has unreliable flows of datagrams. Forward error correction, semi-reliability and multiple streams are layered on top of DCCP as needed while the basic functionality was included in DCCP. Different multimedia applications require different forms of congestion control and DCCP allows the multimedia applications to choose from a set of different control mechanisms. User Datagram Protocol (UDP) which is an alternative to TCP lacks congestion control. Moreover high-bandwidth UDP applications are expected to handle congestion control themselves which is also a difficult task (Ye *et al*., 2014).

There are two congestion control algorithms that are available in DCCP including, DCCP-TCP-like and DCCP-TFRC and the better option among them is decided by the application. The former is known as Congestion Control Identification 2 (CCID 2) and the latter as CCID 3 (Azad *et al*., 2009). In this study we propose a new algorithm for congestion control with respect to both DCCP-TCP-like and DCCP-TFRC. The algorithm depends on the time between back-to-back reset () calls or the frequency of reset () calls and hence the proposed mechanisms is called as Reset Controlled DCCP-TCP-like (RC-DCCP-TCP-like) and Reset Controlled DCCP-TFRC (RC-DCCP-TFRC) respectively. In order to measure the data delivery, we also discuss about an objective no-reference video quality metric. This metric tells us about the effect of degradations added by video broadcast over mixed IP networks.

Video streaming requirements: DCCP has got a wide variety of applications and one among them is streaming of video data as shown in Fig. 1. Since the users are very sensitive to both quality and timeliness, retransmission of the data will not help in such applications. The protocol designed is hence minimal in terms of both functionality and mechanism. The algorithm Is also expected to be almost same without violating TCP affability. In other words it should have all the features of modern TCP congestion control, including Explicit Congestion Notification (ECN), selective acknowle

Corresponding Author: B.Chellaprabha, CSE Department, SNS College of Engineering, Coimbatore, India

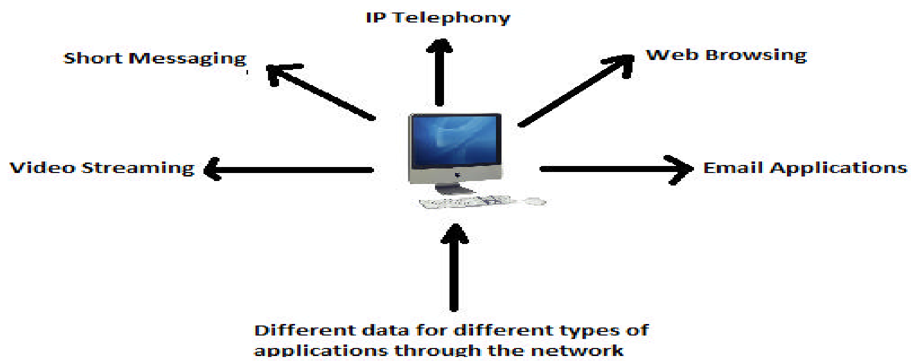

Fig.1: Multiple application running on a single device

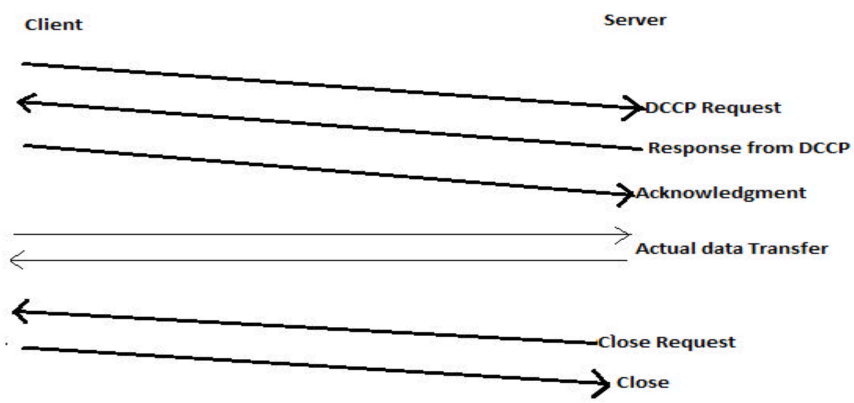

Fig. 2: DCCP handshake mechanism

dgements, verification of acknowledgement and so on . It should also make sure that the DCCP provides applications with simple API's as similar to UDP protocol.

DCCP and proposed modifications: DCCP is a message oriented TCP and is a unicast, connection oriented protocol allowing data flow in both the directions. Connection between the source and destination has 3 way handshakes as shown in Fig. 2.

DCCP protocol: DCCP packets contain a standard header and equivalent fields in case of data and the return packets. The header here contains spaces for source and destination port numbers followed by data offset to indicate the start of the data and CCVal which indicates if CCID is needed. There is also a type field to indicate the DCCP packet type of which 9 are currently defined and the rest reserved. The options field which is present in the protocol helps in feature negotiations, handshake mechanisms and in congestion control mechanisms.

- Header including Source and destination port numbers
- Type of DCCP packet
- Options field
- Application

DCCP working mechanism: Data flows in both directions and a connection setup is required before that. The connection is established by handshake procedure where the client sends a DCCP request to a server which accepts the connection and sends a DCCP response packet.

Data gets transferred once the connection is established and accredited with Acknowledgement packets. When the data transfer gets completed the server sends a close request request which is acknowledged by a close packet from the client. For different congestion control mechanisms, CCID is specified. CCID 0 is reserved; CCID 1 is for sender-based congestion control; CCID 2 suggests TCP-like congestion control mechanism; CCID 3 is a TCP friendly rate control

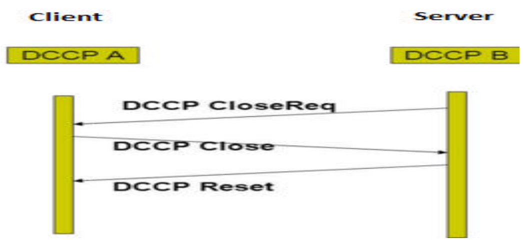

Fig. 3: DCCP connection reset mechanism

algorithm and finally CCID 4 proposes TCP friendly rate control for small packets.

Modified DCCP working mechanism: New congestion control algorithms can be added or existing algorithms can be modified to DCCP to be more suitable for a specific application. In this study, few selected DCCP factors of the reset () function are considered for dynamic changes to progress the DCCP performance with respect to Wireless Sensor Network (WSN) applications as shown in Fig. 3.

The client and server raise the reset () function multiple times repeatedly in all their states of operation during a typical communication process in DCCP. Due to multiple reasons including an error in communication, during handshake failure of feature negotiation or erroneous size of feature option, the reset () function call occurs. Hence it is found that the properties are underutilized or drained unreasonably. We propose to modify some selective parameters during each reset () call thereby avoiding the traditional method of resetting to default values.

We track the time elapsed between the successive reset () function calls in order to assess the level of congestion present. The level of congestion is less if the lapsed time with respect to the current call and its direct previous call is higher than that of the former reset () call and its direct predecessor. If a reduced congestion is sensed, in order to exploit the available link, the transmission rate is increased with respect to the accumulated average reset interval. This is applicable only on reduced congestion but when there is increased level of congestion, the proposed mechanism will behave like a normal DCCP algorithm by means of setting all the parameters to the predefined values.

MATERIALS AND METHODS

Video transmission and capture method: DCCP has been intended especially to back real time traffic including streaming media that has more timing limitations when compared to reliability. It lets to implement and to relate strategies revised to the transport of multimedia contents. One of such application includes Video on Demand (VoD) architecture which consists of a server, a mixer and a client, open up real potentials for the optimization and evaluation of intricate plans in a real context of flow agreement. Most of the work in this area tried to integrate new modules of real content streaming and Peak Signal-to-Noise Ratio (PSNR) calculation for estimating the video quality (Linck *et al*., 2006).

Reduced packet loss leads to improved video quality and hence if there is a mechanism to provide a feedback on the video quality received, it helps in improvising the data transmission and reception. Hence we propose to use a video capture mechanism that can record the raw frames from the end point buffer and have it ready for analysis.

The Microsoft DirectShow Application Programming Interface (API) is used in our work which is a media-streaming architecture for Microsoft Windows. Using DirectShow, the applications can perform high-quality video capture as shown in Fig. 4. Multimedia streams contain huge data and needs to be processed also quickly. Moreover data comes in different formats as well. Direct show is designed to streamline the task of making digital media applications on the Windows platform, by detaching applications from the complications of data passages, hardware variances and organization (Zhang and Li, 2008). The AVI Mux filter in Fig. 4. takes the video stream from the capture point and bundles it into an AVI stream. The File Writer block will then write the output stream to file for further analysis.

Video quality measures: Mean Squared Error (MSE) and PSNR are the error metrics generally used to find the image/video frame quality. These metrics requires the reference frames to estimate the quality and also is a poor indicator of subjective quality. We may not be able to provide the reference frames in most of the cases during video streaming applications and it would be an unnecessary overhead of comparing with the reference images as well. These metrics are criticized mainly for not in sync with the perceived quality measurement.

In this study we propose to use a new algorithm for measuring the video quality at the receiving end and which can estimate the quality without using any reference mechanisms. We also make sure that the algorithm is computational and memory efficient. The different kinds of artifacts that can be observed on the received packets could be of blocking issues in general. So we devise an algorithm that can detect the presence of blockiness and noise in the data received. We also measure the timeliness on which the frames get received and the details associated with it including frames per second, time difference between successive frames etc.

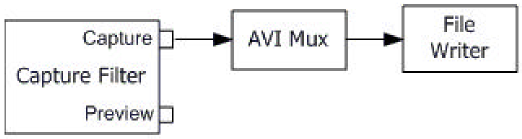

Fig. 4: Capturing video to an AVI file

which helps in finding the proposed DCCP algorithm effectiveness. Amplitude and Phase information of the received signal is generally used to estimate the blockiness in MPEG2 coded video accurately without using any reference signal (Tan and Ghanbari, 2002). But computing these values would be not efficient in terms of time and memory and hence we propose a simpler method of detecting the blockiness present in the signal. We convert the color frame in to YUV format and take only the luma component for further processing. The absolute variance between horizontally neighboring pixels is calculated. If the image is represented by Im (x,y), then this computation is given by,

$$A(x,y) = |\, Im(x,y)-Im(x+1,\, y)| \tag{1}$$

where:
$0 <= x <=$ Width of the image and
$0 <= y <=$ Height of the image.

We then find the edges present in the image using Roberts cross operator (Sharifi *et al*., 2002) which accomplishes a simple, fast to compute and 2-D spatial gradient measurement of an image. Generally we expect the pixel values to be like their neighboring pixels (Vasudevan *et al*., 2010). The surfaces turn only slowly and there will be only few changes around the edges. So as defined in Eq. 1 if the variance is too high and if it does not fall under the edge category as per Roberts cross operator then we count it as pixel with some issue and continue the same process for the whole image. We also divide the Robert cross operator based edge detected image in to small 8×8 pixels and then find the difference between measures of the edges at boundary pixels across x and y directions as Imx and Imy respectively. The overall boundary block is then found using Eq. 2

$$Block = \max (Imx, Imy) \tag{2}$$

From these two computations we set the threshold according to the category of images and mean opinion score from subjects. Based on the value computed we will then be able to tell the absence or presence of blockiness distortion. In other words if the same block gets repeated or corrupted due to DCCP issue we will be able to find it out and can even extend to use it as feedback to the algorithm to improve the quality on the future frames. We also measure the time the frame has arrived and the interval between frames along with the number of frames per second in order to measure the efficiency of the proposed algorithm (Table 1).

Table.1: Video quality analysis results

Video quality analysis				
Resolution	Packet loss	Throughput	Jitter	Quality of service
480p SD	4	0.98	0.003	3.9
	9	0.96	0.001	3.8
720p HD	12	0.94	0.008	4.0
	8	0.94	0.008	4.0
	5	0.95	0.006	4.2
1080p HD	26	0.92	0.010	4.3
	14	0.94	0.009	4.0
	18	0.92	0.010	4.3
4K UHD	20	0.90	0.012	4.5
	28	0.92	0.010	4.3

RESULTS AND DISCUSSION

DCCP was intended from the outset to provision flexible congestion control. We have devised a methodology to improve the congestion control and measured the use of the same for one video streaming application. MPEG4 gives the better presentation for video streaming applications when compared to the other types. Table 1 has the simulation readings of MPEG4 traffic observed at the transport layer. The packet loss is calculated by finding the difference between the number of data packets transmitted and received per 120 frames. Throughput refers to the total amount of data received at 1Mbps send rate. Jitter is the time variance among two consecutive delays. Delay and jitter are usually little higher in the DCCP as compared to UDP because the former checks the network condition for congestion and adjusts the transfer rate accordingly while the latter keeps sending the data without any checks. Quality of service is measured using mean opinion score and ranges between 1 (bad) to 5 (excellent)

CONCLUSION

With more and more multimedia streaming applications emerging in the market the need for timely delivery of data also increases. There are many methods to transfer the data over internet. TCP is reliable in nature but has a huge delay. UDP is better in terms of data transfer but does not provide any acknowledgement or congestion control mechanisms. DCCP is specially we designed to overcome these limitations and in our study have detailed one new method of congestion control for better transfer of data. We have also discussed about the quality measurement at the receiving end using blockiness and time measurements. Though the algorithm devised is good in terms of delivery it lacks in reliability. We have mentioned about "feedback" mechanism in the study which would be our next focus to improve the reliability of the data transfer.

REFERENCES

Azad, M.A., R. Mahmood and T. Mehmood, 2009. A comparative analysis of DCCP variants (CCID2, CCID3), TCP and UDP for MPEG4 video applications. Proceedings of the International Conference on Information and Communication Technologies ICICT09, August 15-16, 2009, IEEE, Karachi, Pakistan, ISBN: 978-1-4244-4608-7, pp: 40-45.

Linck, S., E. Mory, J. Bourgeois, E. Dedu and F. Spies, 2006. Video quality estimation of DCCP streaming over wireless networks. Proceedings of the 14th Euromicro International Conference on Parallel Distributed and Network-Based Processing (PDP'06), February 15-17, 2006, IEEE, Montbeliard, France, ISBN: 0-7695-2513-X, pp: 11-12.

Rahman, J., S. Saha and S.F. Hasan, 2012. A new congestion control algorithm for Datagram Congestion Control Protocol (DCCP) based real-time multimedia applications. Proceedings of the 7th International Conference on Electrical and Computer Engineering (ICECE), December 20-22, 2012, IEEE, Dhaka, Bangladesh, ISBN: 978-1-4673-1434-3, pp: 533-536.

Sharifi, M., M. Fathy and M.T. Mahmoudi, 2002. A classified and comparative study of edge detection algorithms. Proceeding of the International Conference on Information Technology: Coding and Computing, April 8-10, IEEE Computer Society, Washington DC. USA., pp: 117-120.

Tan, K.T. and M. Ghanbari, 2000. Blockiness detection for MPEG2-coded video. IEEE. Signal Process. Lett., 7: 213-215.

Vasudevan, S.K., T. Dharmendra, R. Sivaraman and S. Karthick, 2010. Automotive image processing technique using cannys edge detector. Int. J. Eng. Sci. Technol., 1: 2632-2644.

Ye, L., L. Mhamdi and M. Hamdi, 2014. Efficient UDP-based congestion aware transport for data center traffic. Proceedings of the IEEE 3rd International Conference on Cloud Networking (CloudNet), October 8-10, 2014, IEEE, Luxembourg, Europe, pp: 46-51.

Zhang, F. and B. Li, 2008. Direct Show based internet video on demand system. Proceedings of the International Conference on Microwave and Millimeter Wave Technology ICMMT 2008, April 21-24, 2008, IEEE, Nanjing, China, ISBN: 978-1-4244-1879-4, pp: 2077-2080.