Performance of constant quality video applications using the DCCP transport protocol

Performance of Constant Quality Video Applications using the DCCP Transport Protocol*

J. Van Velthoven, K. Spaey and C. Blondia,
University of Antwerp, IBBT, Middelheimlaan 1, B-2020 Antwerpen, Belgium

Abstract

In this paper we study the influence the DCCP transport protocol has on the end-user quality of video applications. We consider an application that aims to provide a video of constant quality to the user and introduce a model for such a source. The end-user quality of the application is evaluated in a variety of simulation scenarios.

1 Introduction

The Datagram Congestion Control Protocol (DCCP) [2] is a transport protocol that combines unreliable flows with built-in congestion control. It uses acknowledgement mechanisms and Explicit Congestion Notification (ECN) [3] to discover packet loss and congestion. We focus on the congestion control mechanism CCID-3 [1], which is based on the equation-based TCP-Friendly Rate Control (TFRC), appropriate for applications requiring a smooth throughput over time. We will study the influence the use of DCCP/CCID-3 has on the quality of video applications. For more details about the work we present here, refer to [4].

2 Simulation scenario

Consider a DSL-access network. The downstream buffer capacity in the DSL node is set to 120 packets, the downstream (upstream) delay in the access network is 20 ms (3 ms). The bottleneck link is a symmetrical 12 Mbps link. Because of the high capacity links within the rest of the network, the delay introduced there is negligible compared to the delays introduced in the access network.

We will evaluate the quality of the video application based on a method described by Verscheure et al. [6]. To exclude the influence dropped packets have on the end-user quality, we set the actual buffer size such that no loss occurs due to buffer overflow and implement the downstream buffer by setting the maximum threshold for the RED protocol to 120 packets. If this threshold is exceeded, newly arriving packets are marked, resulting in the same reaction of DCCP/CCID-3 that packet loss would have caused. The minimum RED threshold is set to 50 packets.

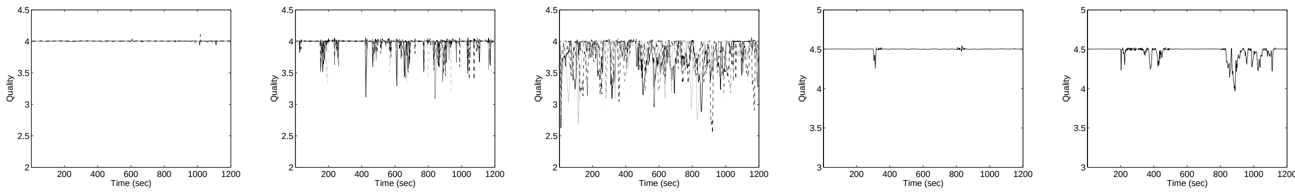
A typical video will contain easy as well as difficult scenes. To be able to capture this behavior, we set up a model with K different scene types and use a K-state continuous time Markov model to represent the scenes and the scene transitions of the video flow. To this extent we need to generalize the quality model of Verscheure for video sources with multiple scenes. The end-user quality can then be given as a function of the bit rate R and the scene type k by $Q(R,k) = Q_0 - \chi_{Q,k}\left(R/\chi\right)^{-1/\xi}$.[1] The main goal of the application is to provide a video flow of a certain *constant* quality to the end-user. To achieve this goal a higher bit rate is required for the difficult scenes compared to the rate needed during an easy scene. In our model, every scene will therefore correspond to a certain sending rate R_k proportional to the complexity of the scene k.

We use a Gaussian distribution to obtain the different rates corresponding to the scene types.[2] We assume that there is no correlation between the scene duration and the complexity of a scene. To limit the number of variables needed, we also assume that the transition rates from a certain scene to another scene are independent of both scenes. First, we need to fix the target quality q that we want to deliver to the end-user. Thus, for every scene type k, we want to determine an associated rate R_k such that $Q(k)$ equals q for every $k = 1, \ldots, K$. Video traffic will then be sent into the network at the minimum of R_k and the allowed sending rate determined by TFRC. Based on the target quality q and the extension of the model of Verscheure, we determine the minimum and maximum sending rate corresponding to the easiest and the most difficult scenes as $R_{min} = 125\left(\frac{5.3-q}{0.025}\right)^{-1.11}$ and $R_{max} = 125\left(\frac{5.3-q}{0.045}\right)^{-1.11}$.

*This work was carried out within the framework of the IWT (Flemish Institute for the Promotion of Scientific and Technological Research in the Industry) project CHAMP.

[1] Inspired by [6], we take $Q_0 = 5.3$, $\chi = 125$, $\xi = 1.11$ and let $\chi_{Q,k}$ take values between 0.025 and 0.045 depending on the complexity of the video. That is, the higher the encoding complexity, the higher χ_Q.

[2] Other sending rate distributions can be plugged in straightforward.



(a) 4 DCCP flows (b) 5 DCCP flows (c) 6 DCCP flows (d) Short TCP flows (e) Long TCP flows

Figure 1. Quality of video applications using DCCP/CCID-3 on a DSL link

Number the scenes from 1 to K, 1 denoting the easiest scene and K the most difficult one. To obtain the rate R_k of scene k, partition the interval [0,1] in $K-1$ equal intervals $[x_{k-1}, x_k]$, $k = 2, \ldots, K$, and take $R_1 = R_{min}$, $R_k = \sigma\Phi^{-1}(x_k) + \mu$ and $R_K = R_{max}$, with $\mu = (R_{min} + R_{max})/2$ and $\sigma = (R_{max} - \mu)/2$ and where Φ denotes the cumulative distribution function of a Gaussian distribution $N(0,1)$. Accordingly, the values $\chi_{Q,k}$ are found such that $Q(k) = q$ for $k = 1, \ldots, K$.

3 Simulations results

Earlier results [5] showed that a flow which uses DCCP in combination with the TFRC congestion control mechanism obtains a fair share of the available bandwidth if it has to compete with other TFRC flows or with TCP flows, given that the packets sent over the different flows have an equal size. In this paper we assume that every flow uses packets of the same size.

In a first scenario we consider a situation in which there are only DCCP flows present on the link. Figures 1(a), 1(b) and 1(c) show the quality obtained when 4, 5 or 6 DCCP flows with a target quality $q = 4$ compete for bandwidth. This means that rates between 1.56 Mbps and 2.99 Mbps are generated by the video sequence. As could be expected, no problems appear when only 4 flows compete for the bandwidth on the 12 Mbps link. However, when 5 DCCP flows share the link, congestion might occur when some of these flows are in a difficult scene, requiring a high bit rate to achieve the target quality $q = 4$. When the video application at the sender's site is informed about the congestion, it decreases its sending rate, in correspondence to the allowed sending rate determined by the TFRC algorithm. Reducing its sending rate will of course result in a lower quality. Putting an extra flow on the link causes even more congestion, resulting in more and larger drops of the quality.

Next, we want to examine the effect TCP flows have on the quality of a video application using DCCP. First, we consider competition between one DCCP flow of 1200 seconds where the application has a target quality $q = 4.5$ and some short TCP flows. There are three slightly overlapping TCP flows of 20s that start around 300s and three such flows that start around 800s. After modifying the NS-2 DCCP module such that only the proper weights are taken into account when calculating the average loss interval [4], we obtain the results presented in Figure 1(d). It shows that short TCP flows have a little impact on the quality of the video application. Longer TCP flows have a greater impact in the quality, as shown in Figure 1(e). Here 1 DCCP flow is present together with 2 TCP flows between 200s and 500s and 3 TCP flows from 800s until 1100s.

From these results we can conclude the following. To let the user enjoy a constant quality video, CCID-3 is not the most suitable congestion control mechanism. It only takes into account the throughput of the flows and not the complexity of the video scenes. Because different scenes in a typical video sequence require different bit rates to achieve the same quality, an equal throughput does not translate into an equal quality. During an easy scene, the video application requires less bandwidth than it is allowed to take. At these moments, other flows can benefit by using a larger part of bandwidth. Thinking in terms of throughput fairness and quality, the video application should be able to use a larger bit rate at times corresponding to a difficult scene using the bandwidth saved up while sending at a lower rate, which is not possible in CCID-3.

References

[1] S. Floyd, E. Kohler, and J. Padhye. Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC). RFC 4342, Mar. 2006.

[2] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol (DCCP). RFC 4340, Mar. 2006.

[3] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification to IP. RFC 3168, Sept. 2001.

[4] J. Van Velthoven, K. Spaey, and C. Blondia. Performance of Constant Quality Video Applications using the DCCP Transport Protocol. *Available at http://www.pats.ua.ac.be*.

[5] J. Van Velthoven, K. Spaey, and C. Blondia. Evaluating the quality of real-time applications using the DCCP/CCID-3 transport protocol. In *Proceedings of EuroNGI Workshop on QoS and Traffic Control*, Dec. 2005.

[6] O. Verscheure, P. Frossard, and M. Hamdi. User-Oriented QoS Analysis in MPEG-2 Video Delivery. *Journal of Real-Time Imaging*, 5:305–314, Oct. 1999.