International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

Performance Analysis of Utility Based Congestion Control for Delay Tolerant Networks

B. Saivenugopal & Dr.K.S.Yuvaraj
#1PG Student, Dept. of MCA, St. Ann's College of Engineering & Technology, Chirala.
#2Assistant Professor, Dept. of MCA, St. Ann's College of Engineering & Technology, Chirala.

Abstract

During the presence of buffers in the middle of network systems, every congestion task prompts buffer queuing and along these lines to a maximizing end-to-end delay. On the event of delay security applications, a huge delay won't be adequate and a solution to appropriately handle congestion tasks while keeping up a low end-to-end delay is required. Delay based congestion techniques are a reasonable solution as they focus to restrict the accomplished end-to end delay. They don't perform well when sharing the data bandwidth with congestion control methodologies not controlled by delay limitations. Our objective is to fill this hole, proposing a novel congestion control technique for delay constrained network over best exertion packet distributed systems. The proposed technique can keep up a limited queuing delay while contending with other delay based streams, and keep away from starvation while contending with loss based streams. The work receive the price based distributed component as congestion control, yet the work present a novel non linear mapping between the accomplished delay and the price capacity, and the work join both delay and loss data into a own price term based on packet interarrival methodologies. The work investigation for our novel technique and the project demonstrate its performance in the reproduction results about completed in the NS3 system. Practice results about exhibit that the proposed technique is ready to: accomplish great intra-convention decency properties, control productively the efficiently to-end delay, lastly, and protect the spill out of starvation when different streams cause the queuing delay to develop unreasonably.

Keywords: Delay-sensitive communication, congestion control, network utility maximization.

I. INTRODUCTION

Today's, numerous Internet applications goal is to work not just at increasing their throughput, yet additionally at meeting difficult delay requirements in the transmission of information streams[8]. Video conferencing applications are a better example of such delay private administrations, where an over the top playback delay with the sound/video stream can radically influence the quality of a web call [9]. On-line video gaming and system remote control are different cases of utilizations that require low latency interchanges [7]. At the point when network joins are congested these applications need to alter their sending rate with the end goal that the accomplished one-way delay is kept low and limited, while protecting fairness with different streams [4]. This decreases to a limited resource allocation issue, which must be unraveled in a completely distributed way because of versatility issues. Congestion control methodologies can be viewed as conveyed techniques to take care of ideal system resource allocation issues [2]. These methodologies are generally classified in primal or double methodologies, in view of the comprehending technique adopted. From a more practical perspective, the primal and double congestion control methodologies generally compare, however not precisely, to loss based and delay-based methodologies [1]. Loss based controllers are broadly conveyed over the web (e.g., TCP) and utilize congestion tasks activated by packet losses to perform rate adjustment [3]. However this class of controllers does not consider any sort of delay estimation, for example, the One-Way Delay (OWD) or the Round Trip Time (RTT). Consequently, there is no control on the latency that the packets may understand on their route and extensive delays can be knowledgeable about the instance of long buffers in the internal network hubs [5]. Then again, delay-based congestion control methodologies can defeat the expanding delay issue by recognizing congestion tasks from OWD calculations. Delay based congestion control methodologies are in this way reasonable for low delay applications since they can keep a low communication delay by adjusting the sending rate to the development of the delay [6]. In any case, they for the most part experience the ill effects of poor performance when offering the system to loss based controllers. Loss based congestion control methodologies dependably fill the buffers of the inward system nodes before activating congestion tasks. In this way, any delay based stream sharing the same bottleneck may encounter a too huge queuing delay also; rapidly achieve starvation (i.e., a sending rate near zero). There is the requirement for a congestion control that could empower low delay communication at whatever point possible and that is highest against the presence of loss based.

Past this conjunction challenge, new congestion control methodologies is i) to give great between convention performance while going up against existing controllers, for example, TCP; ii) to for the most part act at the endpoints instead of at the internal system hubs (alterations of the internal system hubs are especially troublesome); iii) to be strong to noisy estimation of system parameters (i.e., spread delay). Numerous congestion control methodologies have all things considered been proposed before, however to the best of our learning there is no delay based congestion control methodologies ready to well perform within the sight of loss based streams furthermore, fulfilling in the meantime the three principle usage challenges recorded previously. In this work, the work focus to fill this hole by proposing another conveyed Delay-Constrained Congestion Control (DCCC) methodologies that can adjust the sending rate to both loss what's more, delay-based congestion task and to overcome the previously mentioned issues. A definitive objective is to save the low end-to-end delay limitation that is forced by the application, while contending with other delay based controlled streams, and in the meantime, maintain a strategic distance from starvation while contending with loss based streams. In more subtle elements, the work consider a situation where clients send delay-touchy information over a packet distributed system, The system is made out of an arrangement of connections furthermore, hubs, with the connections being shared among various clients who set up uni cast communication between two endpoints of the system. The proposed controller measures the experienced OWD and the interarrival time of the fetching packets at the recipient hub, and alters the rate as needs be all together to maximize the general utility of the system streams. The key instinct is that the interarrival time of the packets is related to the two loss and queueing delay varieties. Subsequently, by utilizing this metric, the controller can work in both delay-based and loss based conditions. The capacity to obey starvation while going up against loss based streams, and still ensure a limited experienced delay is made conceivable by the utilization of a non-straight mapping between the accomplished OWD also, the punishment congestion signal utilized by the rate update condition. The DCCC methodologies has been actualized in the NS3 network test system and has been tried under various topologies and working conditions. Practical results about demonstrate the capacity of the proposed methodologies to keep limited the value of the accomplished OWD, to gain a decent intra-convention fairness and to maintain a strategic distance from starvation while contending with loss based streams, for example, TCP. Note that the proposed methodology depends on the OWD measure, which is significant just in the instance of synchronized endpoints.

II. DELAY-CONSTRAINED CONGESTION CONTROL ALGORITHM

In this segment, the work describes our DCCC methodology, appearing how it defeats the fundamental limitations of the current controllers. The rate update condition that the work considers for our methodologies is the following:

$$x_r = k_r x_r (U^{'}(x_r) - u_r(e_r) - e_r - \pi_r) x_r \ldots (1)$$

(Eq.1) the work now describes the distinctive terms of Eq. (1) in detail. The parameter k_r tunes the update speed of ratex_r. The primary term in the sections is the derivate of the utility function U_r $(\cdot)$. The term V_r $(\cdot)$ is the delay penalty process that maps the OWD into a penalty. So also to the loss value definition in Eq.1, the work composes the delay penalty as:

$$u_r(e_r) = \beta(\frac{e_r - T_r}{RTT_r})\ (e_r - T_r) = \beta\ (\frac{e_r - T_r}{e_r + e_r})(\ e_r - T_r)\ \ldots\ldots\ (2)$$

(In the above equation), where RTT_r is the round trip time of client r, e_r and e_r^b are the forward and reverse experienced delays of client r individually, β is a scaling element and T_r is the delay threshold of client r. The delay edge is identified with delay that the framework experiences at the equilibrium. The estimation value of V_r (e_r) is equivalent to zero if $e_r - T_r < 0$ and equivalent to β $(e_r - T_r)/\ RTT_r)$ or else. The standardization of the cost by RTT_r is motivated by rate fairness enhancements and stability conditions. Note that this work isn't a linear function of the experienced delay, e_r, but instead a monotonically expanding function of it. The derivative of the experienced delay,e_r does not update the equilibrium of the framework, since the time derivative at the equilibrium point will be zero by definition. Be that as it may, it enhances the controller performance during the homeless people, since it gives data about the variety rate of the feedback variable. The last term in Eq. (1), π_r, takes into account the experienced losses, following Eq. (2), so that the mechanism can work in both delay and loss based situations. On account of no losses (π_r= 0), our controller carries on as a delay based controller. The experienced delay at the equilibrium is evaluated by setting the time derivatives to zero in Eq. (1):

$$e_r = \frac{RTT_r}{\beta} U_r(x_r) + T_r = RTT_r \frac{h}{\beta x_r} + T_r \ldots\ldots.\ (3)$$

$$e_r = \frac{e_r^b h/(\beta x_r) + T_r}{1 - h/(\beta x_r)} \ldots\ldots\ldots\ldots\ldots\ldots\ (4)$$

(Eq.3 & Eq.4) This implies that the delay cost v_r (e_r) in our DCCC methodology never forces the sending rate to be lower than h/β. The principle advantages of our penalty function can be compressed as follows: (a) the non-linearity of the penalty function ensures the flows from starvation while competing with loss based functions. Fig. 2 describes the state of the penalty function of Eq. (3) for various values of the propagation delay, when the regressive delay, e_r^b is thought to be equivalent to the restricted propagation delay in the forward direction. The estimation of the penalty soaks to β for huge values of the experienced delay, which is the regular situation that the work encounter while contending with loss based flows. As an outcome the experienced delay can never force the sending rate to minimize to a value lower than h/β along these lines preventing starvation. (b) The non-linearity of

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

the penalty work alleviates unfairness issues caused by heterogeneous propagation delays among the clients. Since our control methodology employments the aggregate experienced one-way delay rather than the queuing delay, it might prompt unfairness when a bottleneck interface is shared among clients with various propagation delays. However the non-linear mapping of the delay serves to reduce this issue when the accessible limit is low. This can without much of a stretch be understood by looking at the state of the penalty work in Fig. 1.

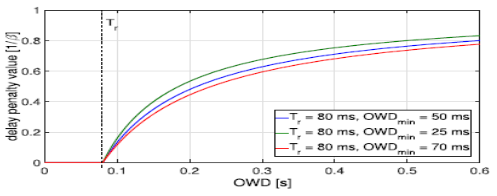

Figure 1: Delay penalty as a function of the experienced delay for different values of propagation delays, measured in units of β.

Since the penalty value has a tendency to immerse for large delays, i.e., low accessible capacity, it implies that clients with various propagation delays will have comparable penalty values in this situation, what's more, as a consequence similar sending rates.

III. IMPLEMENTATION AND SIMULATIONS

The work now proposes simulations results to the proposed DCCC algorithm. Initially, the work clarifies how the controller performance is influenced by every parameter and how to proficiently set them. At that point the work provide the simulation outcomes about where the work analyze the basic behavior of our methodology, the intra-convention fairness, the TCP coexistence and the work furnish a comparison with other similar congestion control algorithms. To analyze the performance of our controller, the work carried out examinations utilizing the NS3 network simulator stage. The work have tested the controller in various system topologies furthermore, situations with a specific end goal to demonstrate that the algorithm is capable to work in loss based and additionally delay-based situations. Specifically, the work thinks about a solitary connection topology, Topology 1 (see Fig. 2) in our simulations.

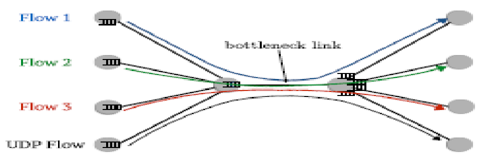

Fig. 2. Network Topology 1.

The initial one is the classic dumbbell topology, where a few clients share a similar one of a kind bottleneck link. The second topology is the purported parking lot topology, with two bottleneck joins, where two clients utilize just a single of these joins while a third client, with a longer information path, utilizes both congested connections. Every one of the connections associating the endpoints to the bottleneck is set to a high association speed, e.g. 100 Mbps. The work concentrate on low/medium estimations of the bottleneck capacity since this is the run of the typical, and most critical, situation for real time applications, The work analyze the performance of the DCCC algorithm under various measurements, for example, throughput, self-perpetrated delay and fairness. The work also compared at the algorithm with other delay based congestion controls, to be specific: the Network Assisted Dynamic Adaptation (NADA) congestion control implementation, the Google congestion control (GCC) algorithm and the Low Extra Delay Background Traffic (LEDBAT) Technique.

Coexistence of TCP

The work now thinks about the performance of the DCCC when it competes against TCP flows. The work again utilizes a solitary connection topology, with a power of 2.5 Mbps and a propagation delay of 50 ms. Three flows share at the same time the connection: an unresponsive UDP flow with a consistent sending rate of 500 kbps, a flow running the DCCC algorithm and a HSTCP flow (in the supplementary material the work give test cases TCP New Reno and TCP Westwood). The delay edge of the DCCC algorithm has been set to 100 ms. The work run simulations for various droptail buffer sizes running from 30 to 180 packets (comparing generally to 100 ms what's more, 600 ms of most extreme queueing respectively). The simulation outcomes about Fig. 3 demonstrate the normal rate at equilibrium for the DCCC and TCP algorithms. The work can see that the level of fairness against TCP based upon the buffer size. This dependency is caused by the delay based piece of the congestion algorithm, since the buffer size has an effect on the accomplished delay and along these lines on the rate at the equilibrium. Within the sight of little buffers, our methodologies achieves a higher rate at equilibrium than TCP one. This is because of the way that the loss based piece of DCCC is more aggressive than the TCP congestion control. On the other hand, on account of expansive buffer size, the DCCC method endures against TCP. Nonetheless, due to the non-linearity of the DCCC penalty procedure, the DCCC flow is secured, and it never starves, achieving the normal lower bound rate of h/β (h/β = 200 kbps in the simulations).

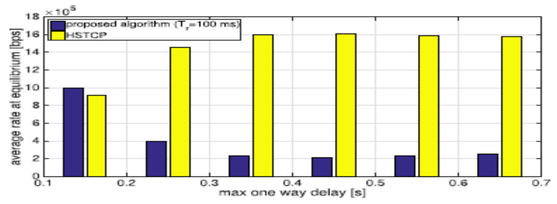

Fig.3: Average rate at equilibrium of our algorithm and HSTCP when competing for a bottleneck for different drop tail buffer size.

By adjusting the value of h, the work can tune the ensured rate that is come to an expansive delay conditions and in this manner the work can limit performance degradation while competing TCP. All in all, when sharing the bottleneck link with TCP, the DCCC algorithm can't ensure TCP fairness, however despite everything it contrasts positively and regard to other delay based techniques proposed in the past literature, which are not ready to ensure a lower bound on the delay based sending rate.

Comparison with Other Congestion Control Algorithms

The work now leads a few experiments to think about our algorithm with other delay based congestion controllers. The work focus on the behavior of the algorithms when they operate in delay based mode and not with respect to how they perform in lossy situations. The work thinks about two hopeful algorithms of the IETF RMCAT (RTP Media Congestion Avoidance Systems) Working Group: NADA and the GCC. The work thinks about a solitary connection topology. With a fluctuating channel capacity and propagation delay of 25 ms.

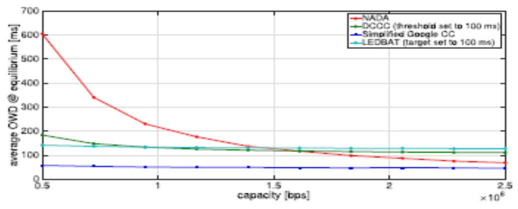

Fig.4: Average OWD at equilibrium for the NADA, the simplified version of the GCC and the DCCC algorithms, in a single link topology.

In Fig.4, the work can see a correlation of the normal self-inflicted delay at equilibrium for NADA, GCC, LEDBAT and the DCCC algorithm. As the work can be watched NADA, LEDBAT and DCCC demonstrate an alternate self-inflicted delay versus bottleneck capacity behavior. Nothing is the one that demonstrates the most noteworthy delay variation while LEDBAT is the most stable of the three, while DCCC demonstrates a middle of the road behavior. An expansive self-inflicted delay variation implies that the stream is to a great degree flexible and can achieve a moderately high rate notwithstanding when a competing flow influences the experienced delay to increment to greatly huge values. Then again, a little variation implies that the delay at equilibrium is relatively free of the sending rate; however the flow may starve if another flow in the network powers the lining queuing to increment over this value. The kind of behavior that is best depends upon the type of information that the congestion control needs to deal with. LEDBAT is ideal for low priority foundation traffic, while a behavior like NADA is best if the objective is to remove starvation within the sight of a remotely imposed high delay.

CONCLUSION

In this work, the work have designed and analyzed a novel hybrid delay based congestion control algorithm, in particular the DCCC algorithm. The proposed methodology is capable to a) keep up a limited delay communication if the system conditions permits it; b) avoid starvation while competing against loss based flows. Presenting a price measure based on the interarrival time of the packets, the work can give a controller that consequently acts as delay based protocol; in view of the genuine event that triggers the congestions. Besides, as a result of the non linear mapping between the experienced delay and the delay based congestion signal, the DCCC methodology keeps away from starvation while contending against loss based flows. The non-linearity mapping too mitigates unfairness issues when the information paths of the clients have various propagation delays. At long last, by utilizing the total experienced delay rather than the real queuing delay, the work stay away from estimation issues and injustice issues due to latecomer flows. The capacity to accomplish a bounded delay at the equilibrium and the adaptability of having the capacity to not starve against loss based flows makes the DCCC methodology an appropriate congestion control algorithm to be utilized for delay sensitive network system applications, e.g. video conferencing.

REFERENCES

[1] L.Budzisz, R.Stanojevic, A.Scholte,F.Baker, andR.Shorten, "On the fair coexistence of loss- and delay-based TCP,"IEEE/ACM Transactions on Networking, vol.19, No.6,pp.1811-1824,December 2011.

[2] Gaetana Carlucci, Luca De Cicco, Stefan Holmer,and Saverio Mascolo, "Congestion Control for web real-time communication," IEEE/ACM Transactions on Networking, vol.25, no.5 October 2017.

[3] David X wei, Cheng Jin , Steven H.low and Sanjay hedge, "FastTCP, Motivation, Arctecture,Algorithms,Performance," IEEE/ACM Transactions on Networking, vol. 14, no.6, December 2006.

[4] E. Altman, T. Bas¸ar, T. Jimenez, and N. Shimkin, "Competitive routing in networks with polynomial costs," IEEE Transactions on Automatic Control, vol. 47(1), pp. 92–96, January 2002.

[5] Yueping Zhang, Seong-Ryong Kang and

Dmitri Loguinov, "Delay-independent stabilityand performance of Distributed congestion control," IEEE/ACM Transactions on Communications, vol. 15, no.5 October 2007.

[6] David Ros and Michael Welzl, "Assessing LEDBAT's delay impact," IEEE Communications Letters, vol.17, no.5, May 2013.

[7] Junzhou Luo, Jiahui Jin and Fengshan, "Standardization of low-lattency TCP with explicit congestion notification: A survey," IEEE/Internet computing, 2017.

[8] Mustafa Ozmen, M.Cenk Gursoy, "Secure transmission of delay-sencitive data over wireless fading channels," IEEE Transaction on information Forensics and security, vol.12, no.9, September 2017.

[9] R. Gibbens and P. Key, “Distributed control and resource marking using best-effort routers,” IEEE Network, pp. 54–59, May 2001.

[10] E. Altman and T. Bas¸ar, “Multi-user rate-based flow control,” IEEE Transactions on Communications, vol. 46(7), pp. 940–949, july 1998.

[11] F. Kelly, A. Maulloo, and D. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability,” Journal of the Operational Research Society, vol. 49, pp. 237–252, 1998.

ABOUT AUTHORS:

BUDATI SAIVENUGOPAL is currently pursuing his MCA in MCA Department ,St.Ann’s college of engineering & technology,chirala A.P He received his B.Sc computer Science Degree in N.N.S Vidya Degree College,Chirala, AP.

Dr.K.S.YUVARAJ Ph.D in Computer Science, Specialization Advanced Networking and Data Mining currently working as an Associate Professor in Department of Computer Science Engineering, St’Ann’s College of Engineering & Technology, Chirala, AP.