

Improving Network Efficiency by Selecting and Modifying Congestion Control Constraints

Saleem Ullah, Faisal Shahzad, Shahzada Khurram, Waheed Anwer

Department of Computer Science & IT, The Islamia University of Bahawalpur, Bahawalpur, Pakistan
Email: saleemullah@iub.edu.pk, faisalsd@gmail.com, khurram@iub.edu.pk, waheed@iub.edu.pk

Received 8 December 2013; revised 23 January 2014; accepted 20 February 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Congestion in wired networks not only causes severe information loss but also degrades overall network performance. To cope with the issue of network efficiency, in this paper we have proposed and investigated an efficient mechanism for congestion control by the selection of appropriate congestion window size and proactive congestion avoidance, which improves system overall performance and efficiency. The main objective of this work is to choose the accurate size of congestion window based on available link bandwidth and round trip time (*RTT*) in cross and grid topologies, instead of choosing number of hops (Previous researches), we have achieved significant improvement in the overall performance of the network. General simulation results under distinctive congestion scenarios are presented to illuminate the distinguished performance of the proposed mechanism.

Keywords

Congestion Avoidance, Congestion Control, Congestion Window (cwnd), Network Performance

1. Introduction

TCP plays a very vital role in today's internet communication. The increase in user terminals and other applications requires networks to perform smoothly, and this network requires transport layer to provide reliable data transfer. Therefore congestion is a big challenge in today's increasing internet traffic. However, we cannot determine the optimal TCP performance due to the following main problems:

- 1) Contention between sharing terminals.
- 2) Hidden terminal problems.

How to cite this paper: Ullah, S., Shahzad, F., Khurram, S. and Anwer, W. (2014) Improving Network Efficiency by Selecting and Modifying Congestion Control Constraints. *Communications and Network*, 6, 105-111.
<http://dx.doi.org/10.4236/cn.2014.62013>

- 3) Packet loss at the MAC layer.
- 4) Path disconnection in case of mobility.
- 5) Reordering.
- 6) Exponential retransmission back-off at TCP layer.

To address these challenges, a simple yet effective scheme is proposed in this paper. As its name implies, the scheme contains relevant mechanisms for avoiding congestion intelligently, detecting congestion timely and eliminating congestion reactively. Control operations are performed by choosing a new congestion window on the measured channel bandwidth and *RTT*. While previous work was performed by selecting number of hops between sender and receiver [1]-[5], which raise many issues that we will discuss in later sections. Rest of the paper is categorized as; Section 2 will focus some previous work done in this area and some drawbacks at some level, Section 3 will describe our proposed mechanism, Section 4 will based on simulation results, conclusion and future work.

2. Agenda for Potential Development

The application requirements and the limitations of congestion control circumscribe a framework for potential improvements. We identify five distinct cases of interest, which have been discussed in the recent literature.

1) Additive Increase leads naturally to congestion, which, in turn, degrades throughput for two reasons: a) Routers need time to recover even from a transitory congestive collapse and sources need time to detect and retransmit missing packets [6]; b) Congestion control is triggered upon congestion; the window is adjusted backwards and the timeout is extended, which in turn degrades the protocol's capability to detect and exploit error free conditions and bandwidth availability, respectively [6] [7].

2) Additive Increase is not efficient when the network dynamics encompass rapid changes of bandwidth availability. For example, when short flows that cause congestion complete their task, bandwidth becomes available. Similarly, when a handoff is completed in a cellular network, the entire channel's bandwidth becomes available. A more rapid response is then clearly indicated [8].

3) Multiplicative decreases causes transmission gaps that hurt the performance of real-time applications that experience jitter and degraded good put. Furthermore, multiplicative decrease with a factor of $\frac{1}{2}$ or a window adjustment to 2 packets characterizes a rather conservative strategy [9].

4) Error detection lacks an appropriate classification module that would permit a responsive strategy, oriented by the nature of potential errors. That is, when errors appear to be transient due to short-lived flows or random network interference, congestion control mechanisms (*i.e.* timeout extension and multiplicative window adjustment) are triggered unduly. The insufficient error detection/classification may also lead to unfair bandwidth allocation in mixed networks. By default, flows that experience network errors do not balance their bandwidth loss with a more aggressive recovery although such behavior could be justified: flows that experienced no losses have occupied extra bandwidth at the router temporarily, when the network errors forced some senders to back off. This situation is discussed as an open issue in [10]; we demonstrate the validity of this argument, based on experimental results, in Section 5.

5) Source-based decisions on the transmission rate, based on the pace of the acknowledgements, necessarily incorporate the potentially asymmetric characteristics (e.g. ack delays and/or losses) of the reverse path [11]. Hence, the sender's transmission rate does not always reflect the capacity of the forward path. This situation has a direct impact on efficiency since available bandwidth remains unexploited. Several proposals have been presented to tackle the problems of TCP over networks. Most of these proposals rely on some form of local retransmission at the wired boarder, and do not deal (either directly or indirectly) with real-time application constraints, see [12] for a detailed description). Some recent protocols restrict the modifications at the transport level. TCP-Freeze [13] distinguishes handoffs from congestion through the use of the Advertised Window. WTCP [14] implements a rate-based congestion control replacing entirely the ACK-clocking mechanism. TCP-Probing [15] grafts a probing cycle and an Immediate Recovery Strategy into standard TCP, in order to control effectively the throughput/overhead tradeoff. Although TCP-Probing deals effectively with both throughput and energy performance in heterogeneous networks, due to its probing mechanism, it may not satisfy the requirements of current service.

It is possible to have node- and link-level congestion occurring at the same time in networks. Both of them have direct impacts on network performance and application objective. Thus, the onset of congestion must be

predicted in advance or detected in time so that it can be relieved thereafter.

3. Proposed Mechanism

3.1. Appropriate Congestion Window Selection

TCP performance is also affected by choosing larger $cwnd$ size, which increases the probability of packet contention and packet losses from excessive collisions [16]. Therefore a new $cwnd$ size is chosen based on the available current channel bandwidth and RTT in cross as well as in grid topologies.

From the figure we can see that bandwidth measurement is performed by TCP sender on receiving *Acks* from receiver. Let's suppose that $C_{E(k-1)}$ is the bandwidth at any $(k-1)th$ -*Ack* segment, then the available current bandwidth would be BW

$$BW_{curr} = \left[\left(C_{E(k-1)} \times RTT \times \lambda_k \right) / RTT + T_k \right] \quad (1)$$

Here λ is the total bytes transmitted successively, which acknowledges between $(k-1)th$ and kth *Acks*. From Equation (1) we can estimate exponentially weighting by using β then we can control the current and smoothed measure bandwidth as,

$$BW_{E(k)} = C_{E(k-1)} \times \beta + BW_{curr} (1 - \beta) \quad (2)$$

Using Equation (2) we can also control the ratio between newly and smoothed measured bandwidth. While Equation (3) gives us maximum $cwnd$ size when RTT and packet size are given.

$$cwnd_{max} = RTT \times BW / \lambda \quad (3)$$

$$\text{Packet_loss} = MSS \times 1.5 / RTT \times \rho^{65} \quad (4)$$

As mentioned before, we have to measure the bandwidth of the multi-hop networks and RTT to determine the appropriate maximum $cwnd$ size. Before applying maximum $cwnd$ to the TCP layer some time will be consumed to measure and RTT . In beginning of transmitting TCP segments, $cwnd$ size is not limited, which does not matter, because the maximum $cwnd$ size can be determine before TCP segments are sent into the system.

3.2. Congestion Avoidance

TCP uses a network congestion avoidance algorithm that includes various aspects of an additive increase/multiplicative decrease (AIMD) scheme, with other schemes such as slow-start in order to achieve congestion avoidance. The TCP congestion avoidance algorithm is the primary basis for congestion control in the Internet.

$$\begin{aligned} &\text{if } cwnd(t) > ssthresh(t) \\ &cwnd(t+t') = cwnd(t) + 1/cwnd(t) \end{aligned}$$

where, $ssthresh(t)$ is a threshold value at which TCP changes its phase from slow start phase to congestion avoidance phase. When packet loss is detected by retransmission timeout expiration, $cwnd(t)$ and $ssthresh(t)$ are updated as,

$$cwnd(t) = 1 \text{ and } ssthresh(t) = cwnd(t)/2$$

On the other hand, when TCP detects packet loss by a fast retransmit algorithm, it changes $cwnd(t)$ and $ssthresh(t)$ as;

$$cwnd(t) = ssthresh(t) \text{ and } ssthresh(t) = cwnd(t)/2$$

TCP Reno then enters a fast recovery phase if the packet loss is found by the fast retransmit algorithm. In this phase, the window size is increased by one packet when a duplicate Ack packet is received. On the other hand, $cwnd$ is restored to $ssthresh(t)$ when the non-duplicate Ack packet corresponding to the retransmitted packet is received.

We have seen that TCP connection starts up in slow start mode and exponentially increase the $cwnd$ until it cross the $ssthresh(t)$. Once $cwnd(t)$ is greater than $ssthresh(t)$, TCP moves to congestion avoidance phase. In this

mode the primary objective is to maintain high throughput without causing congestion. If TCP detects any segment loss, this is a strong indication that network is congested, so as a corrective action TCP reduces its data flow rate by reducing $cwnd(t)$ after that it again goes back to slow start phase.

Having some deeper look on working mechanism, we choose $ssthresh(t) = 65535$ bytes and $cwnd(t) = 512$ bytes. Since $cwnd(t) < ssthresh(t)$ TCP state is slow start. TCP $cwnd(t)$ grows from 512 bytes to 64947 bytes also it has been assumed that no segment loss during slow start phase. During slow start $cwnd(t)$ is incremented by one segment on every successful acknowledgment from the receiver. When $cwnd(t) = 65459 + 512 > 65535$, at this point TCP changes its state to congestion avoidance phase, and the primary objective in this phase is to get the higher throughput. If during this phase any three duplicate Acks received the TCP moves to the fast retransmit without waiting for the retransmit timer out and then after it moves to the fast recovery phase.

3.3. Congestion Control

We have modified the TCP default behavior of three-duplicate Acks to two-duplicate Acks, as it has been seen that 87% of the packets loss after two duplicate Acks, so why not change the default behavior? This decreases time delay. The proposed algorithm is as:

Algorithm: Congestion Control

```

On receiving two-duplicate Acks or if a timeout occur do the following;
  IS_congestion=0;
  If((max_rate > min_rate) && (curr_rate - min_rate < max_rate - curr_rate)) {
    IS_congestion=1;
  }
  If(cwnd > thresh)
  IS_congestion=1;

  If(IS_congestion)
  {
    Slowstart();
  }
  else
    fast_recovery();
  }

```

4. Simulation Results and Performance Analysis

To verify our proposed mechanism we use NS-2 (2.34) simulator, model used in our experiment is consists of 50 nodes in a chain wise topology. We set link bandwidth to 1Gbps in order to analyze maximum $cwnd$ size. Simulated scheme is investigated by increasing number of nodes, and we compared the performance results with TNR.

If we analyze $cwnd$ with saw-tooth effect in [Figure 1](#), it can be seen clearly that the proposed mechanism is more effective and consistent then TNR, as number of users increases the $cwnd$ loses its stability as well as performance, but our suggested mechanism has consistent effect in terms of stability as well as performance, which increase system overall performance.

Packet loss in networks affects the overall performance of the system. Packet loss is due to signal degradation, congestion in channel and many other common factors. After sending data in bulk we observe that proposed method has very few ratio of packet loss (0.2%) as compared to TNR. It is also clear that $cwnd$ in this case is bigger then the proposed but on the other hand decrease in $cwnd$ size is also clear which may lead to higher delay and decreasing the network throughput. Therefore proposed mechanism has better results than the TNR ([Figure 2](#)).

We have modified the default behavior of TCP duplicate acknowledgement from two to three Acks, because it has been observe that 87% of the packets loss after two duplicate Acks, so why not modify this to two instead of three. Which decrease delay and increase throughput as shown in [Figure 3](#).

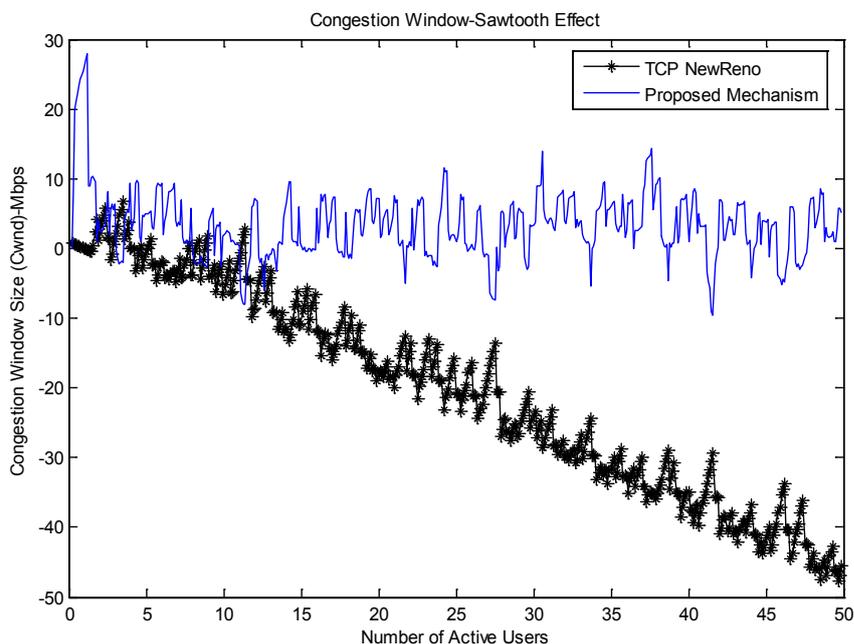


Figure 1. Effect of congestion window ($cwnd$).

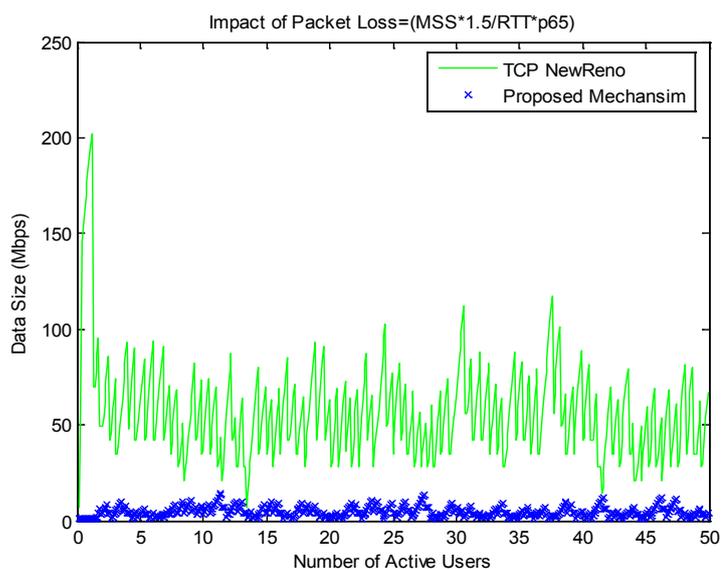


Figure 2. Packet loss comparison.

5. Conclusion and Future Work

In this work, we propose a scheme for congestion control, congestion avoidance by selecting appropriate $cwnd$ size which detects and alleviates congestion in networks. The main objective is to provide high transmission quality for the data traffic under conditions of congestion. The scheme comprises three main mechanisms. Firstly, it attempts to suppress the source traffic from event area by carefully selecting a set of representative nodes to be data sources. Secondly, the onset of congestion is indicated in a timely way by jointly checking buffer occupancy and link utilization. Lastly, the network attempts to alleviate congestion in the traffic hotspot by either resource control or traffic control, which is dependent on the specific congestion condition. The ns-2 based simulation has confirmed the advantages of proposed scheme and demonstrated significant performance improvements over existing schemes.

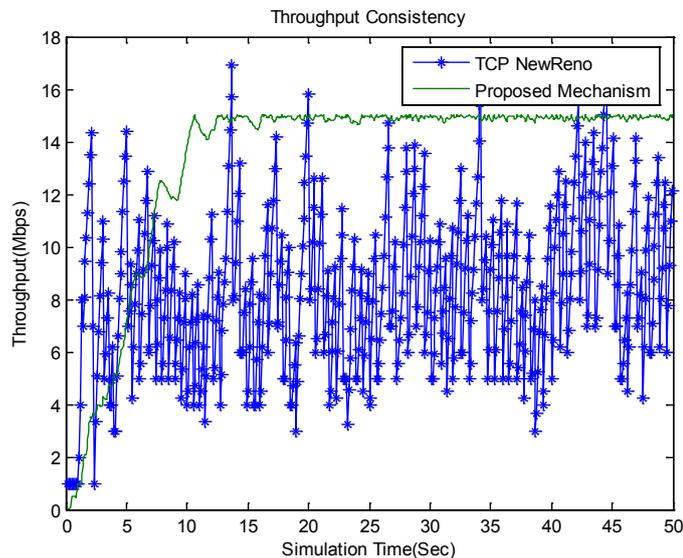


Figure 3. Flow consistency behavior (throughput analysis).

In the future, we hope to extend the proposed scheme to take into account some dynamic behaviors of the network, such as node mobility and link failure. We have also planned to extensively investigate performance on sensor based networks.

Acknowledgements

This research work was partially supported by NSF of china Grant No.61003247. The assistance of Prof. Xiaofeng Liao in this work is also gratefully acknowledged. The authors also would like to thanks the anonymous reviewers and the editors for insightful comments and suggestions.

References

- [1] Harb, S.M. and Mcnair, J. (2008) Analytical Study of the Expected Number of Hops in Wireless Ad Hoc Network. *WASA '08 Proceedings of the Third International Conference on Wireless Algorithms, Systems, and Applications*.
- [2] Miller, L.E. (2001) Distribution of Link Distances in a Wireless Network. *NIST Journal of Research* (Spring).
- [3] Huang, P.-K., Lin, X.J. and Wang, C.-C. (2011) A Low Complexity Congestion Control and Scheduling Algorithm for Multi-Hop Wireless Networks with Order-Optimal Per Flow Delay. *IEEE/ACM Transactions on Networking*, **14**.
- [4] Zawodniok, M. and Jangannathan, S. (2007) Predictive Congestion Control Protocol for Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, **06**, 3955-3963.
- [5] Jiang, S.M., Zuo, Q. and Wei, G. (2009) Decoupling Congestion Control from TOP for Multi-Hop Wireless Networks: Semi TCP. *CHANTS 2009, ACM Transactions*.
- [6] Fang, W.-W., Chen, J.-M., Shu, L., Chu, T.-S. and Qian, D.-P. (2009) Congestion Avoidance, Detection and Alleviation in Wireless Sensor Networks. *Journal of Zhejiang University SCIENCE C*, **11**.
- [7] Raghunathan, V. and Kumar, P.R. (2006) A Counterexample in Congestion Control of Wireless Networks. *Elsevier Science*.
- [8] O'Neill, D.C. (2003) Adaptive Congestion Control for Wireless Networks Using TCP. *IEEE Proceeding-2003*.
- [9] Popa, L., Riaciu, C. and Stoica, I. (2006) Reducing Congestion Effects in wireless Networks by Multipath Routing. *IEEE Proceeding 2006*, 96-105.
- [10] Tiwari, B., and Chandavarkar (2010) Congestion Adaptive Routing in Wireless Mesh Networks. 2010 *Seventh International Conference on Wireless and Optical Communications Networks (WOCN)*, Colombo, 6-8 September 2010, 1-5.
- [11] Armaghani, F.R., Jamuar, S.S., Khatun, S. and Rasid, M.F.A. (2011) Performance Analysis of TCP with Delayed Acknowledgments in Multi-Hop Ad-Hoc Networks. *Wireless Personal Communications*, **56**, 791-811.
- [12] Curran, K., Woods, D., McDermot, N. and Bradley, C. (2003) The Effects of Badly Behaved Routers on INTERNET congestion. *International Journal of Network Management*, **13**, 83-94. <http://dx.doi.org/10.1002/nem.464>

- [13] Hac, A. and Ma, Y.J. (1998) A Rate-Based Congestion Control Scheme for ABR Service in ATM Networks. *International Journal of Network Management*, **8**, 292-317.
[http://dx.doi.org/10.1002/\(SICI\)1099-1190\(199809/10\)8:5<292::AID-NEM286>3.0.CO;2-Y](http://dx.doi.org/10.1002/(SICI)1099-1190(199809/10)8:5<292::AID-NEM286>3.0.CO;2-Y)
- [14] Li, J., Zhang, S.Y., Li, C.L. and Yan, J.R. (2010) Composite Light Weight Traffic Classification System for Network Management. *International Journal of Network Management*, **20**, 85-105.
- [15] Schulze, H. and Mochalski, K. Internet Study2008/2009, IPOQUE Report.
http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009
- [16] Zhang, Y., Leonard, D. and Loguinov, D. Jetmax: Scalable Max-Min Congestion Control for High-Speed Heterogeneous Networks. *Proceedings of IEEE INFOCOM 2005*, 2006, 1-13.