

# An Efficient Distributed Congestion Control Protocol with Stability and Fairness

**Vikram .A**

Assistant Professor,

Department of Computer Science and Engineering,  
Saranathan College of Engineering,  
Panjappur, Tiruchirappalli, Tamil Nadu- 620 012.  
[vikram.aug1984@gmail.com](mailto:vikram.aug1984@gmail.com)

## Abstract:

Recent research efforts to design better Internet transport protocols combined with scalable Active Queue Management (AQM) have led to significant advances in congestion control. One of the hottest topics in this area is the design of discrete congestion control algorithms that are asymptotically stable under heterogeneous feedback delay and whose control equations do not explicitly depend on the RTTs of end-flows. In this paper, we first prove that single-link congestion control methods with a stable *radial* Jacobian remain stable under arbitrary feedback delay (including heterogeneous directional delays) and that the stability condition of such methods does *not* involve any of the delays. We then extend this result to generic networks with *fixed consistent* bottleneck assignments and max-min network feedback. To demonstrate the practicality of the obtained result, we change the original controller in Kelly's work [15] to become robust under random feedback delay and fixed constants of the control equation. We call the resulting framework *Max-min Kelly Control* (MKC) and show that it offers smooth sending rate, exponential convergence to efficiency, and fast convergence to fairness, all of which make it appealing for future high-speed networks

**Keywords:** AQM, MKC, EMKC, High Speed Networks.

## I. Introduction

Over the last fifteen years, Internet congestion control has evolved from binary-feedback methods of AIMD/TCP [2], [33] to the more exciting developments based on optimization theory [22], [23], game theory [11], [19], and control theory [10], [11], [24], [26].

It is widely recognized that TCP's congestion control in its current shape is inadequate for very high-speed networks and fluctuation-sensitive real time multimedia. Thus, a significant research effort is currently under way (e.g., [5], [6], [9], [12], [15], [16], [19], [29], [32]) to better understand the desirable properties of congestion control and develop new algorithms that can be deployed in future AQM (Active Queue Management) networks.

One of the most important factors in the design of congestion control is its *asymptotic stability*, which is the capability of the protocol to avoid oscillations in the steady-state and properly respond to external perturbations caused by the arrival/departure of flows, variation in feedback, and other transient effects. Stability proofs for distributed congestion control become progressively more complicated as feedback delays are taken into account, which is especially true for the case of *heterogeneous* delays where each user  $i$  receives its network feedback delayed by a random amount of time  $D_i$ .

Many existing papers (e.g., [4], [10], [11], [12], [17], [18], [19], [23]) model all users with homogeneous delay  $D_i =$

$D$  and do not take into account the fact that end-users in real networks are rarely (if ever) synchronized. Several recent studies [20], [24], [27] successfully deal with heterogeneous delays; however, they model  $D_i$  as a deterministic metric and require that end-flows (and sometimes routers) dynamically adapt their equations based on feedback delays, which potentially leads to RTT-unfairness, increased overhead, and other side-effects (such as probabilistic stability).

In this paper, we set our goal to build a discrete congestion control system that maintains both stability and fairness under heterogeneously delayed feedback, allows users to use fixed parameters of the control equation, and admits a low-overhead implementation inside routers. We solve this problem by showing that any single-link max-min fair system with a stable radial Jacobian remains asymptotically stable under arbitrary directional delays, extend this result to multi-link networks under fixed bottleneck assignments, and apply it to the original controller proposed by Kelly *et al.* [15].

We call the result of these efforts *Max-min Kelly Control* (MKC) and demonstrate that its stability and fairness do not depend on any parameters of the network (such as delay, path length, or the routing matrix of end-users). We also show that with a proper choice of AQM feedback, MKC converges to efficiency exponentially fast, exhibits stability and fairness under *random* delays, converges to fairness almost as quickly as AIMD, and does not require routers to estimate any parameters of individual flows.

By isolating bottlenecks along each path and responding only to the most-congested resource, the MKC framework allows for very simple stability proofs, which we hope will lead to a better understanding of Kelly's framework in the systems community and eventually result in an actual implementation of these methods in real networks.

Our initial thrust in this direction includes ns2 simulations of MKC, which show that finite time-averaging of flow rates inside each router coupled with a naive implementation of end-user functions leads to undesirable transient oscillations, which become more pronounced when directional delays  $D_i$  and  $D_i$  to/from each router increase. We overcome this drawback with simple changes at each end-user and confirm that the theoretically predicted monotonic convergence of MKC is achievable in real networks, even when the routers do not know the exact combined rate of end-flows at any time instant  $n$ . We also show that our algorithms inside the router incur low overhead (which is less than that in XCP [12] or RED [7]) and require only one addition per arriving packet and two variables per router queue.

## II. Background

### A. Delay-Dependent Congestion Control

Recently, a large amount of theoretical and experimental work has been conducted on designing robust congestion controls. One direction is to model the network from an optimization or game-theoretic point of view [11], [17], [18], [19], [23]. The original work by Kelly *et al.* [14], [15] offers an economic interpretation of the resource-user model, in which the entire system achieves its optimal performance by maximizing the individual utility of each end-user. To implement this model in a decentralized network, Kelly *et al.* describe two algorithms (*primal* and *dual*) and prove their global stability in the absence of feedback delay.

However, if feedback delay is present in the control loop, stability analysis of Kelly controls is non-trivial and currently forms an active research area [4], [10], [20], [24], [27], [29]. Recall that in Kelly's framework [15], [24], each user  $i \in [1, M]$  is given a unique route  $r_i$  that consists of one or more network resources (routers). Feedback delays in the network are heterogeneous and directional. The forward and backward delays between user  $i$  and resource  $j$  are denoted by  $D_{ij}$  and  $D_{ji}$ , respectively. Thus, the round-trip delay of user  $i$  is the summation of its forward and backward delays with respect to any router  $j \in r_i$ :  $D_i = D_{ij} + D_{ji}$ .

### B. Delay-Independent Congestion Control

To the best of our knowledge, the first delay-independent stability condition is due to Vinnicombe, who proposes and examines the following continuous fluid model of a network with sources operating TCP-like algorithms [28]:

$$\dot{x}_i(t) = \frac{x_i(t-D_i)}{D_i} (\alpha_i(t) - (\alpha_i(t) + \beta_i(t))\eta_i(t)) \quad (1)$$

where  $\alpha_i(t) = a(x_i(t)D_i)^n$ ,  $\beta_i(t) = b(x_i(t)D_i)^m$ ,  $a, b, m, n$  are constants,  $\eta_i(t)$  is the network feedback and link price  $p_j(t) = (y_j(t)/C_j)^B$  is an approximation of packet loss at link  $j$  of capacity  $C_j$  and buffer size  $B$ .

## III. Classic Kelly Control

In this we discuss intuitive examples that explain the cryptic formulas in the previous section and demonstrate in simulation how delays affect stability of Kelly controls (1). We then show that the original Kelly control [15], or any mechanism that relies on the *sum* of feedback functions from individual routers, exhibits a tradeoff between linear convergence to efficiency and persistent stationary packet loss.

### A. Delayed Stability Example

The following example illustrates stability problems of (1) when feedback delays are large. We assume a single-source, single-link configuration and utilize a congestion indication function that computes the estimated packet loss using instantaneous arrival rates:

$$p(n) = \frac{x(n) - C}{x(n)} \quad (2)$$

Where  $C$  is the link capacity and  $x(n)$  is the flow rate at discrete step  $n$ . We note that the price function  $p(n)$  in the original Kelly control is nonnegative; this results in slow linear AIMD-like probing for link capacity until the slowest link in the path is fully utilized, which is generally considered too slow for high-speed networks. Thus, under AQM feedback assumed throughout this paper, we allow *negative* values in which signals the flow to increase its sending rate when  $x(n) < C$ . We show that the negative component of packet-loss improves convergence to efficiency from linear to exponential.

Applying  $p(n)$  to Kelly control yields a linear end-flow equation:

$$x(n) = x(n-1) + kw - k(x(n-D) - C) \quad (3)$$

Next, assume a particular set of parameters:  $K = 1/2$ ,  $\omega = 10$  mb/s, and  $C = 1,000$  mb/s. We have that the system is stable if and only if delay  $D$  is less than four time units. As illustrated in Figure 1(a), delay  $D = 1$  keeps the system stable and monotonically convergent to its stationary point.

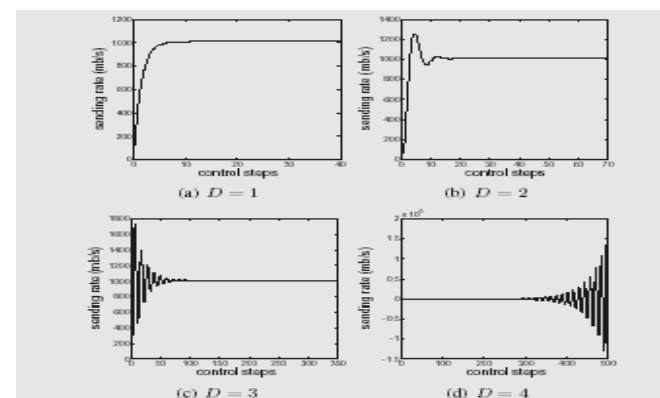


Figure 1: Stability of Kelly control under different feedback delays ( $k=1/2, \omega=10$  mb/s, and  $C = 1,000$  mb/s)

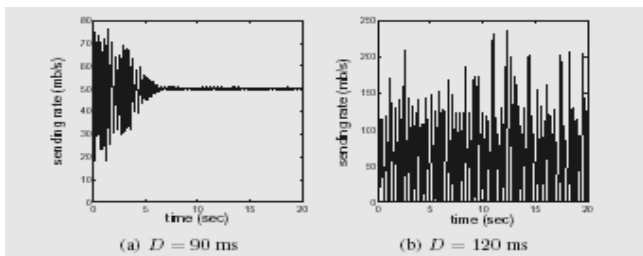


Figure 2: Simulation results of the classic Kelly control under different delays ( $k=1/2$ ,  $\omega=10$  mb/s, and  $C=1,000$  mb/s)

Under larger delays  $D = 2$  and  $D = 3$  in Figures 1(b) and (c), the flow exhibits progressively increasing oscillations before entering the steady state. Eventually, as soon as  $D$  becomes equal to four time units, the system diverges as shown in Figure 1(d).

Using the same parameter  $\cdot$  and reducing  $\omega$  to 20 kb/s, we examine ns2 simulations, in which a single flow passes through a link of capacity 50 mb/s. We run the flow in two network configurations with the round-trip delay equal to 90 ms and 120 ms, respectively. As seen in Figure 2, the first flow reaches its steady state after decaying oscillations, while the second flow exhibits no convergence and periodically overshoots capacity  $C$  by 200%. Since Kelly controls are unstable unless condition is satisfied, a natural strategy to maintain stability is for each end-user  $i$  to adaptively adjust its gain parameter  $K_i \sim 1/D_i$ . However, this method depends on reliable estimation of round-trip delays  $D_i$  and leads to unfairness between the flows with different RTTs.

## B. Stationary Rate Allocation

In this the price function should allow negative values, such that the convergence speed of Kelly control is improved from linear to exponential. However, we show next that this modification presents a problem in the stationary resource allocation. Consider a network of  $M$  resources and  $N$  homogeneous users (i.e., with the same parameters  $K$  and  $\omega$ ). Further assume that resource  $j$  has capacity  $C_j$ , user  $i$  utilizes route  $r_i$  of length  $M_i$  (i.e.,  $M_i = |r_i|$ ), and packet-loss  $\eta_i(n)$  fed back to user  $i$  is the aggregate feedback from all resources in path  $r_i$ . We further assume that there is no redundancy in the network (i.e., each user sends its packets through at least one resource and all resources are utilized by at least one user). Thus, we can define routing matrix  $A_{N \times M}$  such that  $A_{ij} = 1$  if user  $i$  passes through resource  $j$  (i.e.,  $j \in r_i$ ) and  $A_{ij} = 0$  otherwise. Further denote the  $j$ -th column of  $A$  by vector  $V_j$ . Clearly,  $V_j$  identifies the set  $s_j$  of flows passing through router  $j$ .

## IV. Max-min Kelly Control

We start our discussion with the following observations. First, we notice that in the classic Kelly control (1), the enduser decides its current rate  $x_i(n)$  based on the most recent rate  $x_i(n-1)$  and delayed feedback  $\eta_i(n-D_{ij})$ . Since the latter carries information about  $x_i(n-D_i)$ , which was in effect *RTT time units earlier*, the controller in (1) has no reason to involve  $x_i(n-1)$  in its control loop. Thus, the sender quickly becomes unstable as the discrepancy between  $x_i(n-1)$  and  $x_i(n-D_i)$  increases. One natural remedy to this problem is to retard the reference rate to become  $x_i(n-D_i)$  instead of  $x_i(n-1)$  and allow the feedback to accurately reflect network conditions with respect to the first term of (1).

Second, to avoid unfairness between flows, one must fix the control parameters of all end-users and establish a uniform set of equations that govern the system. Thus, we create a new notation in which  $K_i \omega_i = \alpha$ ,  $K_i = \beta$  and discretize Kelly control as following:

$$x_i(n) = x_i(n-D_i) + \alpha - \beta \eta_i(n) x_i(n-D_i) \quad (4)$$

where  $\eta_i(n)$  is the congestion indication function of user  $i$ .

Next, to overcome the problems of proportional fairness described in the previous section and utilize negative network feedback, we combine with max-min fairness under which the routers only feed back the packet loss of the *most-congested* resource instead of the combined packet loss of all links in the path:

$$\eta_i(n) = \max_{j \in r_i} p_j(\eta-D_{ij}) \quad (5)$$

where  $p_j(\cdot)$  is the congestion indication function of individual routers that depends only on the aggregate arrival rate  $y_j(n)$  of end-users. We call the resulting controller *Max-min Kelly Control (MKC)* and emphasize that flows congested by the same bottleneck receive the *same* feedback and behave independently of the flows congested by the other links. Therefore, in the rest of this paper, we study the single-bottleneck case since each MKC flow is always congested by only *one* router.

## V. Exponential MKC

Consider a particular packet-loss function  $p(n)$

$$p(n) = \frac{\sum_{u=1}^N x_u(n-D_u) - C}{\sum_{u=1}^N x_u(n-D_u)} \quad (6)$$

where we again assume a network with a single link of capacity  $C$  and  $N$  users. This is a rather standard packet-loss function with the exception that we allow it to become negative when the link is under-utilized and achieves exponential convergence to efficiency, which explains the *Exponential MKC (EMKC)*.

### A. Packet Loss

EMKC converges to the combined stationary point  $X^* = C + N\alpha/\beta$  which is above capacity  $C$ . This leads to constant (albeit usually small) packet loss in the steady state. However, the advantage of this framework is that EMKC does not oscillate or react to individual packet losses, but instead adjusts its rate in response to a *gradual* increase in  $p(n)$ . Thus, a small amount of FEC can provide a smooth channel to fluctuation-sensitive applications such as video telephony and various types of real-time streaming. Besides being a stable framework, EMKC is also expected to work well in wireless networks where congestion-unrelated losses will not cause sudden reductions in the flow rates.

Also notice that EMKC's steady-state packet loss  $p^* = N\alpha/(\beta + N\alpha)$  increases linearly with the number of competing flows, which causes problems in scalability to a large number of flows. However, it still outperforms AIMD, whose increase in packet loss is quadratic as a function of  $N$  [21]. Furthermore, if the network provider keeps  $N = \Theta(C)$ , EMKC achieves *constant* packet loss in addition to exponential convergence to fairness.

Finally, observe that if the router is able to count the number of flows, zero packet loss can be obtained by adding a constant  $\Delta = N\alpha/(\beta C)$  to the congestion indication function [3]. However, this method is impractical, since it needs non-scalable

estimation of the number of flows  $N$  inside each router. Hence, it is desirable for the router to adaptively tune  $p(n)$  so that the system is free from packet loss. One such method is AVQ (Adaptive Virtual Queue) proposed in [17], [20]. We leave the analysis of this approach under heterogeneous delays and further improvements of EMKC for future work.

### B. Convergence to Fairness

We next investigate the convergence rate of EMKC to fairness. To better understand how many steps EMKC requires to reach a certain level of max-min fairness, we utilize a simple metric that we call  $\epsilon$ -fairness. For a given small positive constant  $\epsilon$ , a rate allocation  $(x_1; x_2; \dots; x_N)$  is  $\epsilon$ -fair, if:

$$f = \frac{\min_{i=1}^N x_i}{\max_{i=1}^N x_i} \geq 1 - \epsilon \quad (7)$$

Generally,  $\epsilon$ -fairness assesses max-min fairness by measuring the worst-case ratio between the rates of any pair of flows. Given the definition in (57), we have the following result.

## VI. Packet Format of MKC

We next examine how to implement scalable AQM functions inside routers to provide proper feedback to MKC flows. This is a non-trivial design issue since the ideal packet loss in relies on the sum of *instantaneous* rates  $x_i(n)$ , which are never known to the router. In such cases, a common approach is to approximate model with some time-average function computed inside the router. However, as mentioned in the introduction, this does not directly lead to an oscillation-free framework since directional delays of real networks introduce various inconsistencies in the feedback loop and mislead the router to produce incorrect estimates of  $X(n) = \sum_i x_i(n)$ .

We provide a detailed description of various AQM implementation issues and simulate EMKC in ns2 under heterogeneous including time varying and the feedback delays.

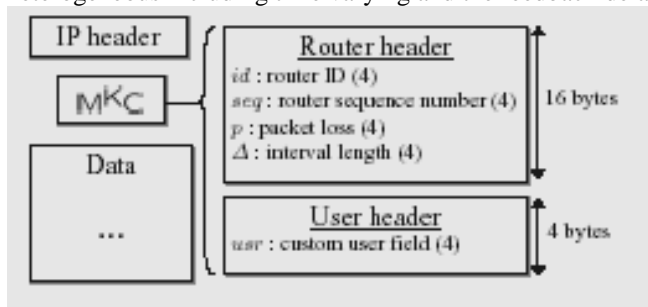


Figure 3: Packet Format of MKC

### A. Packet Header

The MKC packet header consists of two parts – a 16-byte *router* header and a 4-byte *user* header. The router header encapsulates information that is necessary for the router to generate precise AQM feedback and subsequently for the end-user to adjust its sending rate. The *id* field is a unique label that identifies the router that generated the feedback (e.g., its IP address). This field is used by the flows to detect changes in bottlenecks, in which case they wait for an extra RTT before responding to congestion signals of the new router. The *seq* field is a local variable incremented by the router each time it produces a new value of packet loss  $p$ . Finally, the  $\Delta$  field carries the length of the averaging interval used by the router in its computation of feedback.

The *usr* field is necessary for end-flows to determine the rate  $x_i(n-D_i)$  that was in effect RTT time units earlier. The simplest way to implement this functionality is to inject the value of  $x_i(n)$  into each outgoing packet and then ask the receiver to return this field in its acknowledgments.

### B. The Router

Recall that MKC decouples the operations of users and routers, allowing for a scalable decentralized implementation. The major task of the router is to generate its AQM feedback and insert it in the headers of all passing packets. However, notice that the router never knows the exact combined rate of incoming flows. Thus, to approximate the ideal computation of packet loss, the router conducts its calculation of  $p(n)$  on a discrete time scale of  $\phi$  time units. For each packet arriving within the *current* interval  $\Delta$ , the router inserts in the packet header the feedback information computed during the *previous* interval  $\Delta$ . As a consequence, the feedback is retarded by  $\Delta$  time units inside the router in addition to any backward directional delays  $D_i$ . Since MKC is robust to feedback delay, this extra  $\phi$  time units does not affect stability of the system. We provide more implementation details below.

During interval  $\Delta$ , the router keeps a local variable  $S$ , which tracks the total amount of data that has arrived  $I$  to the queue since the beginning of the interval. Specifically, for each incoming packet  $k$  from flow  $i$ , the router increments  $S$  by the size of the packet:  $S = S + si(k)$ . In addition, the router examines whether its locally recorded estimate  $\sim p$  of packet loss (which was calculated in the previous interval  $\Delta$ ) is larger than the one carried in the packet. If so, the router overrides the corresponding entries in the packet and places its own router ID, packet loss, and sequence number into the header. In this manner, after traversing the whole path, each packet records information from the most congested link.<sup>3</sup> At the end of interval  $\Delta$ , the router approximates the combined arriving rate  $X(n) = \sum_{i=1}^N x_i(n-D_i)$  by averaging  $S$  over time  $\Delta$ :

$$\bar{X} = \frac{S}{\Delta} \quad (8)$$

Based on this information, the router computes an estimate of packet loss  $p(n)$  using

$$\bar{P} = (\bar{X} - C) / \bar{X} \quad (9)$$

where  $C$  is the capacity of the outgoing link known to the router (these functions are performed on a per-queue basis). After computing  $\sim p$ , the router increments its packet-loss sequence number (i.e.,  $seq = seq + 1$ ) and resets variable  $S$  to zero. Newly computed values  $seq$  and  $\sim p$  are then inserted into qualified packets arriving during the next interval  $\Delta$  and are subsequently fed back by the receiver to the sender. The latter adjusts its sending rate as we discuss in the next section.

### C. The User

MKC employs the primal algorithm at the end users who adjust their sending rates based on the packet loss generated by the most congested resources of their paths. However, to properly implement MKC, we need to address the following issues. First, most existing congestion control algorithms are window-based, while MKC is a rate-based method. This means that, instead of sending out a window of packets at once, each MKC user  $i$  needs to properly pace its outgoing packets and maintain its sending rate at a target value  $x_i(n)$ . We implement this mechanism by explicitly calculating the interpacket interval  $\delta_i(k)$  of each packet  $k$ :

$$\delta_i(k) = \frac{S_i(k)}{X_i(n)} \quad (10)$$

where  $s_i(k)$  is the size of packet  $k$  of user  $i$ . Second, notice that ACKs carrying feedback information continuously arrive at the end-user and for the most part contain duplicate feedback (assuming  $\Delta$  is sufficiently large). To prevent the user from responding to redundant or sometimes obsolete feedback caused by reordering, each packet carries a sequence number  $seq$ , which is modified by the bottleneck router and is echoed by the receiver to the sender. At the same time, each end-user  $i$  maintains a local variable  $seq_i$ , which records the largest value of  $seq$  observed by the user so far. Thus, for each incoming ACK with sequence  $seq$ , the user responds to it only when  $seq > seq_i$ . This allows MKC senders to pace their control actions such that their rate adjustments and the router's feedback occur on the same timescale. Third, MKC requires both the delayed feedback  $\eta_i(n)$  and the delayed reference rate  $x_i(n-D_i)$  when deciding the next sending rate.

## VII. Conclusion

This paper investigated the properties of Internet congestion controls under non-negligible directional feedback delays. We focused on the class of control methods with radial Jacobians and showed that all such systems are stable under heterogeneous delays. To construct a practical congestion control system with a radial Jacobian, we made two changes to the classic discrete Kelly control and created a max-min version we call MKC. Combining the latter with a negative packet-loss feedback, we developed a new controller EMKC and showed in theory and simulations that it offers smooth sending rate and fast convergence to efficiency. Furthermore, we demonstrated that EMKC's convergence rate to fairness is exponential when the network provider scales the number of flows  $N$  as  $\Theta(C)$  and linear otherwise. From the implementation standpoint, EMKC places very little burden on routers, requires only two local variables per queue and one addition per arriving packet, and allows for an easy implementation both in end-to-end environments and under AQM support. Our future work involves improvement of the convergence speed to fairness and design of pricing schemes for EMKC to achieve loss-free performance regardless of the number of flows  $N$ . It is generally accepted that future communication and computer networks will be characterised by high-speed and long-distance connectivity, and by the requirement to carry a wide variety of network services and traffic types. These demands create new challenges for network designers and researchers. It is widely recognised that transport layer enhancements are essential if high performance next generation networks are to be realized.

Our objective here is to develop a systematic framework for modifying the basic TCP algorithm that renders it suitable in a variety of network types.. We describe a new TCP-variant that is suitable for deployment in high speed and long distance networks, as well as conventional networks. The new TCP variant, H-TCP, is shown to be fair when deployed in homogeneous networks, to be friendly when competing with conventional TCP sources, to rapidly respond to changes in available bandwidth, and to utilise link bandwidth efficiently.

## References

[1] R. Bronson, *Schaum's Outline of Theory and Problems of Matrix Operations*. McGraw-Hill, 1988.

- [2] D.-M. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1–14, Jun. 1989.
- [3] M. Dai and D. Loguinov, "Analysis of Rate-Distortion Functions and Congestion Control in Scalable Internet Video Streaming," in *Proc. ACM NOSSDAV*, Jun. 2003.
- [4] S. Deb and R. Srikant, "Global Stability of Congestion Controllers for the Internet," *IEEE Trans. on Automatic Control*, vol. 48, no. 6, pp. 1055 – 1060, Jun. 2003.
- [5] S. Floyd, "High-speed TCP for Large Congestion Windows," *IETF RFC 3649*, Dec. 2003.
- [6] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," in *Proc. ACM SIGCOMM*, Aug. 2000.
- [7] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, App. 397–413, Jan. 1993.
- [8] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.
- [9] C. Jin, D. Wei, and S. H. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance," in *Proc. IEEE INFOCOM*, Mar. 2004.
- [10] R. Johari and D. K. H. Tan, "End-to-End Congestion Control for the Internet: Delays and Stability," *IEEE/ACM Trans. on Networking*, vol. 9, no. 6, pp. 818– 832, Dec. 2001.
- [11] K. Kar, S. Sarkar, and L. Tassiulas, "A Simple Rate Control Algorithm for Maximizing Total User Utility," in *Proc. IEEE INFOCOM*, Apr. 2001.
- [12] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth Delay Product Networks," in *Proc. ACM SIGCOMM*, Aug. 2002.
- [13] W. G. Kelley and A. C. Peterson, *Difference Equations*. Harcourt / Academic Press, 2001.
- [14] F. P. Kelly, "Charging and Rate Control for Elastic Traffic," *European Trans. on Telecommunications*, vol. 8, no. 1, pp. 33–37, Jan. 1997.
- [15] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237– 252, Mar. 1998.
- [16] T. Kelly, "Scalable TCP: Improving Performance in High-speed Wide Area Networks," in *Proc. First International Workshop on Protocols for Fast Long-Distance Networks*, Feb. 2003.
- [17] S. Kunniyur and R. Srikant, "Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management," *ACM SIGCOMM*, Aug. 2001.
- [18] S. Kunniyur and R. Srikant, "A Time-Scale Decomposition Approach to Adaptive Explicit Congestion Notification (ECN) Marking," *IEEE Trans. on Automatic Control*, vol. 47, no. 6, pp. 882 – 894, Jun. 2002.
- [19] S. Kunniyur and R. Srikant, "End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks," *IEEE/ACM Trans. on Networking*, vol. 11, no. 5, pp. 689–702, Oct. 2003.
- [20] S. Kunniyur and R. Srikant, "Stable, Scalable, Fair Congestion Control and AQM Schemes that Achieve High Utilization in the Internet," *IEEE Trans. on Automatic Control*, vol. 48, no. 11, pp. 2024–2029, Nov. 2003.
- [21] D. Loguinov and H. Radha, "End-to-End Rate-Based Congestion Control: Convergence Properties and Scalability

Analysis," *IEEE/ACM Trans. on Networking*, vol. 11, no.5, pp. 564–577, Aug. 2003.

[22] S. H. Low, "A Duality Model of TCP and Queue Management Algorithms," *IEEE/ACM Trans. on Networking*, vol. 11, no. 4, pp. 525–536, Aug. 2003.

[23] S. H. Low and D. E. Lapsley, "Optimization Flow Control I: Basic Algorithm and Convergence," *IEEE/ACM Trans. on Networking*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[24] L. Massoulié, "Stability of Distributed Congestion Control with Heterogeneous Feedback Delays," *IEEE/ACM Trans. on Networking*, vol. 47, no. 6, pp. 895–902, Jun. 2002.

[25] T. Mori, N. Fukuma, and M. Kuwahara, "Delay-Independent Stability Criteria for Discrete-Delay Systems," *IEEE Trans. on Automatic Control*, vol. 27, no. 4, pp. 964–966, 1982.

[26] F. Paganini, J. Doyle, and S. H. Low, *A Control Theoretical Look at Internet Congestion Control*, ser. Multidisciplinary Research in Control: The Mohammed Dahleh Symposium 2002. Springer-Verlag, Berlin, 2003.

[27] G. Vinnicombe, "On the Stability of End-to-End Congestion Control for the Internet," Cambridge University, Technical Report CUED/FINFENG/ TR.398, Dec. 2000.

[28] G. Vinnicombe, "On the Stability of Networks Operating TCP-like Protocols," in *Proc. IFAC*, Aug. 2002.

[29] G. Vinnicombe, "Robust Congestion Control for the Internet," Cambridge University, Technical Report, 2002.

[30] B. P. Wydrowski, L. L. H. Andrew, and I. M. Y. Mareels, "MaxNet: Faster Flow Control Convergence," *Networking*, vol. 3042, pp. 588–599, May 2004.

[31] B. P. Wydrowski and M. Zukerman, "MaxNet: A Congestion Control Architecture for Maxmin Fairness," *IEEE Communication Letters*, vol. 6, no. 11, pp. 588–599, Nov. 2002.

[32] L. Xu, K. Harfoush, and I. Rhee, "Binary Increase Congestion Control for Fast, Long Distance Networks," in *Proc. IEEE INFOCOM*, Mar. 2004.

[33] Y. R. Yang and S. S. Lam, "General AIMD Congestion Control," in *Proc. IEEE ICNP*, Nov. 2000.

[34] L. Ying, G. E. Dullerud, and R. Srikant, "Global Stability of Internet Congestion Control with Heterogeneous Delays," in *Proc. American Control Conference*, Jun.2004.

[35] Y. Zhang, S.-R. Kang, and D. Loguinov, "Delayed Stability and Performance of Distributed Congestion Control," in *Proc. ACM SIGCOMM*, Aug. 2004.

[36] Y. Zhang, D. Leonard, and D. Loguinov, "JetMax: Scalable Maxmin Congestion Control for High-Speed Heterogeneous Networks," in *Proc. IEEE INFOCOM*, Apr. 2006.

[37] Y. Zhang and D. Loguinov, "Local and Global Stability of Symmetric Heterogeneously-Delayed Control Systems," in *Proc. IEEE CDC*, Dec. 2004.

## Author



**Vikram A** received B.E degree in Electrical and Electronics Engineering, M.E. degree in Computer Science and Engineering from V.S.B Engineering College in 2006 and from Pavender Bharathidasan College of Engineering and Technology in 2008 and Master of Business Administration in Project Management from Alagappa University in 2012 respectively. Currently he is working as Assistant Professor in Saranathan Engineering College, Tiruchirappalli, Tamil Nadu.