International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-4, Issue-11)

Research Article | November 2015

Credit Based Backward and Forward Explicit Notification for Congestion Monitoring and Mitigation (CB^2FEM^2)-IP Networks

K. Kranthi Kumar
Asst Prof in CSE&IT
SNIST, Hyderabad
Telangana, India

Namratha Peddi
B.Tech Final Year, Department of IT
SNIST, Hyderabad
Telangana, India

Abstract—

Congestion Control is a problem that is still bothersome, as more and more people use the internet every minute of every day in a global heterogeneous network, and as the number of IP addresses keep increasing. This calls for people continuing the search for an optimal method that will reach near lossless network for packet transmission.

We propose a novel approach to monitor and mitigate congestion in IP Networks - Credit Based Backward and Forward Explicit Notification for Congestion Control (CB^2FEM^2). Congestion is still a universal problem, so a method is proposed that there are a number of problems still existing in the congestion control mechanisms. Packets are still dropped when the congestion becomes unbearable, there is as yet no way around it. Also, there is no feedback mechanism combined with another congestion control mechanism. So far, effective solutions are standalone and not working hand in hand. The proposed system showcases the usage of credit based approach for IP networks by providing a mechanism by assigning credit to nodes that will help us curb the congestion problem to an extent. We do this by showcasing that results after applying credit scheme to IP networks are good, and show the simulation results and prove that the approach provides better results than existing systems.

Secondly, using the result from the first approach, we propose a preventive and reactive congestion control method, used using a dynamic, closed loop scheme and improve upon several existing methodologies and fix the loopholes. We use credit at nodes coupled with backward and forward congestion notification, and mitigation of packets when congestion occurs. So, a resource allocation scheme of calculating permitted queue length for each node is used, along with feedback mechanism, decreasing the load being sent by the sender in case of congestion, and mitigation by digression to avoid loss of packets. We display our simulation results to show the increase in the performance of the network with the employment of our strategy and believe will overcome some of the drawbacks of all the existing systems.

Keywords— Congestion Control, Credit, Backward Explicit Congestion Notification, Detouring, Packet Loss

I. INTRODUCTION

Nowadays, it is a common and everyday phenomenon for every single person to have a mobile phone. It has become a necessity to communicate with other people in real time. In addition to this, all electronic devices like microwaves, washing machines and printers possess their own IP address. This leads to a plethora of devices connected over the internet. With all these devices communicating with each other over wireless network, managing them is bound to become a problem. Congestion is one such problem that occurs, and quite possibly the main one. Congestion Control is a problem that is still troubling, as traffic only increases in a global heterogeneous network. This calls for people continuing the search for an optimal method that will reach near lossless network for packet transmission.

A main problem that persists in networks is traffic control and loss of packets in wireless networks. Packet loss can be caused by a number of factors that can corrupt or lose packets in transit, the main cause being congestion, and some others such as radio signals that are too weak due to distance or multi-path fading (in radio transmission), faulty networking hardware, or faulty network drivers. Packets are also intentionally dropped by normal routing routines (such as Dynamic Source Routing in ad hoc networks) and through network dissuasion technique for operational management purposes. The main problem is congestion, where packets are dropped simply because there is too much congestion at the node. Packet loss combined with traffic is the main reason why congestion control has become so important. There are many methods existing to curb congestion, like TCP/IP congestion avoidance, and active queue methods like random early detection, and explicit congestion notification.

In this article, we propose an approach using credits at nodes and mitigation using digression to address the problem of lost packets. We also use a feedback mechanism where the sender is notified about the congestion from the node at which the congestion takes place, so that it can decrease the load it is sending until further acknowledgement, so that the already existing congestion problem doesn't become any worse.

This article shows how credit value for each node helps in curbing congestion. Credit value is the allowable buffer space allocated for each node. In other words, it states that node's capacity. This novel method shows how this is used in IP networks. This method uses the benefit of dynamically calculating the buffer space in the congestion control algorithm by providing information to the congested node about neighboring node buffer space availability to curb the problem of packet loss.

Using the credit based approach mentioned in the previous paragraph, we develop a congestion control algorithm, using the concept of mitigation of loss using detouring of packets from a congested node. This is the area where we show how our approach handles the problem of packet dropping in areas of intense congestion and hotspots and prove through our simulation results that we are able to better address the problem of dropping of packets and provide better results.

Another major problem in congestion control methods is, even after developing a method to decrease the packets that are dropped, congestion still occurs. This is because the sender is still not aware about the congestion and continues to send packets at the normal rate which leads to more congestion at the congested node. We plan to address even this problem in our new approach by providing a feedback mechanism which is to use backward and forward congestion notification to inform the source about the congestion and to decrease the rate of sending the packets and to inform the receiver to stop sending requests. Therefore, as explained, we address all the existing problems in IP networking and propose a system which overcomes all existing problems.

Section II talks about the existing work, It talks about a Credit Based Approach in non-IP networks, Backward Explicit Congestion Notification and Forward Explicit Congestion Notification which are both feedback mechanisms, and also talks about detouring of packets at congested node. Section III talks about our proposed idea for congestion control, and how we plan to overcome the existing glitches in all the methods. The algorithm proposed is an effective method for congestion control. Section IV produces the simulation process and results of the proposed system, effectively showcasing the considerable improvement in performance on using our approach. Section V concludes the experiment. Section VI talks about the future scope of the system and how it can be improved further in the near future.

II. RELATED WORK

The credit based approach was first seen in [3]. It was initially used for non-IP networks. Credit-based flow control, implementing link-by-link, per-VC (virtual circuit) flow control, the scheme generally works over a VC link as follows. Before forwarding any data cell over the link, the sender needs to receive credits for the VC from the receiver. At various times, the receiver sends credits to the sender indicating availability of buffer space for receiving data cells of the VC. After having received credits, the sender is eligible to forward some number of data cells of the VC to the receiver according to the received credit information. Each time the sender forwards a data cell of a VC, it decrements its current credit balance for the VC by one. There are two phases in flow controlling a VC. In the first buffer allocation phase, the VC is given an allocation of buffer memory, Buf_Alloc, in the receiver. In the second credit control phase, the sender maintains a nonnegative credit balance, Crd_Bal, to ensure no overflow of the allocated buffer in the receiver. An experimental OC-12 (622-Mbps) ATM switch [6] with credit flow control was developed by BNR and Harvard. An ATM network testbed involving multiple copies of this switch and a variety of ATM host adapter cards is operational. Experiments on this testbed have confirmed the benefit of credit-based flow control. Independently, Digital Equipment Corporation has also developed a credit based ATM network. The Credit Update Protocol (CUP) [10] is an efficient and robust protocol for implementing credit control over a link. (A link can be a physical link connecting two adjacent nodes, or a virtual circuit connecting two remote nodes.)

BECN – Backward Explicit Congestion Notification was proposed as a congestion control mechanism for IP Networks. This method informs the sender about the congestion. BECN is an alternative approach to the current ECN mechanism as proposed in the internet draft. A Backward-ECN (BECN) is proposed which uses the existing IP signaling mechanism, the Internet Control Messaging Protocol (ICMP) [RFC 792] Source Quench message. The use of ICMP Source Quench (ISQ) allows a basic ECN mechanism for IP which does not require any negotiation between end systems. Congestion notification is kept at the network (IP) level. The congestion state can be reflected up to the transport layer (e.g. TCP or UDP) for appropriate action. The ISQ based approach reduces the reaction time to congestion in the network. In addition, the ISQ message can include information on the severity of the congestion allowing the end host to react accordingly so as to make maximal use of the resources while maintaining network equilibrium. Internet Draft Backward ECN for the Internet Protocol June 1998. IP currently does not have any adhered to mechanism to notify its transport protocols of network congestion problems. The ISQ message format was originally defined and documented that a disadvantage of the ISQ mechanism is that its details are discretionary stating that it is impossible for the end-system user to be sure about the conditions under which the ISQ was generated. RFC 896 [6] discusses in general terms approaches for generating ISQs and reacting to them. Among the approaches is one that considers generating ISQs adaptively before the queue is full. RFC 1016 [7] described Source Quench Introduced Delay (SQUID) where ISQs were to be generated based on threshold levels of the physical queue in the router. Packets are clocked by the sender based on inter-arrival times adjusted in response to the ISQ arrival rate. More recently, [16] explored the use of Source Quenches for controlling unresponsive sources that inject more than their fair share bandwidth into the network. There have also been proposals for using BECN within ATM networks [17]. In [18] it was affirmed that though the indiscriminate use of BECN can cause problems in ATM networks, BECN may help reduce the feedback time for paths with large delays. The proposal for BECN in IP networks [19] provided guidelines for generating ISQs and responding to them in a TCP/IP network. According to [19] a BECN TCP sender responds to ISQ congestion notification by halving its TCP window. We observed that with this proposal the BECN sender also starts increasing its window upon receipt of the next ACK after a window reduction. The immediate reaction of increasing rate of injecting packets into the network makes the proposed BECN algorithm unduly aggressive.

Detour Induced Buffer Sharing is a method used for packet loss in congested networks. Modern data center networks (DCNs) are built with shallow buffered switches. These switches are cheaper than their deep buffered counterparts, and they also reduce the maximum queuing delays packets may suffer. However, shallow buffers can lead to high packet loss under bursty traffic conditions (often caused by the "incast" traffic pattern [9]). DCTCP [3] attempts to solve this problem by using ECN markings to throttle the flows early, to avoid buffer overflow. However, DCTCP cannot prevent packet loss if the traffic bursts are severe, and short-lived. To share buffers among switches, this is a system that a switch detour excess packets to other switches – instead of dropping them – thereby temporarily claiming space available in the other switches' buffers. This approach is called detour-induced buffer sharing (DIBS). Spare capacity: If all links in the network are fully utilized, DIBS will not work, since there will be no capacity to handle the extra load caused by detoured packets. However, in DCNs, congestion is usually transient and local [13]. Thus, when we detour packets away from one congestion hotspot, the network will typically have capacity elsewhere to handle the traffic. We are working to derive a limit on network utilization, beyond which detouring can be detrimental. DIBS can be implemented alongside traffic spreading schemes such as ECMP (flow or packet level) or MPTCP [8]. These schemes reduce the possibility of extreme congestion, but do not eliminate it on their own.

III. PROPOSED SYSTEM

The proposed system architecture is described in Fig.1, as follows

A. Proposed system architecture:

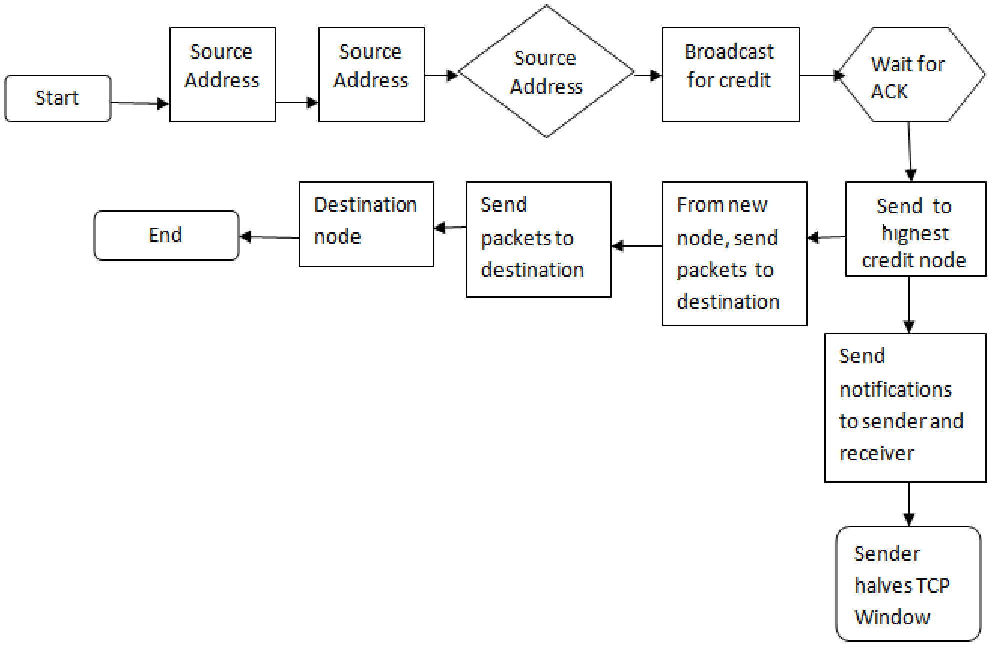

Fig.1 Architecture depicting the proposed method

B. Proposed System Algorithm

The below section discusses about proposed system algorithm in a step wise manner.

Step 1: Obtain Source and Destination Addresses.

Step 2: Start performing the data transfer.

Step 3: If buffer space for a node is full, then congestion is occurring. Go to step 5.

Step 4: If buffer space for node is still available, no congestion, go to step 10.

Step 5: Using credit based approach explained below, send out a broadcast to nodes in the network asking them to calculate their credit and reply back with an acknowledgement and credit number if they have available buffer space.

Step 6: Wait for specific timeframe until timeout for acknowledgement.

Step 7: On receipt of acknowledgements, detour the packets to node that has sent acknowledgement with the highest credit.

Step 8: From the new node, send the packets via nodes in the packet switched network to the destination.

Step 9: To prevent further congestion from taking place while and before the packets are being detoured, send a backward congestion notification to the source and a forward congestion notification to the destination.

Step 10: Forward the packets to the destination.

Step 11: Stop data transfer.

Step 12: End.

In this system, we present a novel approach - Credit Based Backward and Forward Congestion Monitoring and Mitigation an approach to the congestion control problem, and our improvement over the mentioned congestion control methods, and how we plan to address their drawbacks. Our approach is working at prevention and reactivity to congestion using a closed loop scheme where sources are informed dynamically about congestion state of the network.

Initially, as shown in Fig.1, over a connection-oriented network without virtual circuits, the sender starts the data transmission. At a particular node, if the buffer space is completely full, congestion occurs. In this case, we propose a novel system for congestion control. If there is congestion at a node, we plan to mitigate the packets by digression to neighboring nodes in the network, which have available buffer space. It becomes counterproductive to send packets to neighboring nodes if they themselves are congested. So far, this has been a problem, a pre-computed routing algorithm was used which did not dynamically inform about buffer space availability in nearby nodes. So, how do we overcome this? We've provided the solution in the next section.

Once we determine which node to detour our packets to, we send our packets there. Consider our approach in reaching from our home to our place of work. Usually, one would take the shortest path that would take them there, the route that we frequently follow. But nowadays, since getting stuck in traffic is a very likely phenomenon, we use Google Maps to tell us if there is traffic on our usual route. If there is traffic, Maps gives us an alternate route to follow. When we take the alternate route, it automatically reroutes us from that route to our destination. This path might not be the shortest path, but considering the traffic, it will take us to our destination much quicker. Using this mechanism, we reroute the packets to the node and send our packets through the network to our destination.

Now, we come to how the selection of which node to transmit the packets takes place. This is done by using a credit system. Credit is the amount of available buffer space in a node. In other words, it is the node's capacity calculated dynamically. When there is congestion at a node, it sends out a broadcast to all the nodes in the network asking for buffer availability and their credit. Then is the waiting period, it waits till a timeout for acknowledgements. After the timeout, from the available acknowledgements, it detours its packets to the node with the highest credit. The credit sent back to the node from the remaining nodes in the network is calculated by the respective nodes dynamically, at that particular instant. This reduces the risk of detouring to a node with congestion by a considerable amount. Combining these approaches, we can tackle the problem of dropping packets immensely.

The problem doesn't stop here. While the detouring process is taking place, the sender continues to send packets at the same rate. This only increases the congestion and leads to an irrevocable situation. The problem is handled using a feedback mechanism to inform the sender that congestion has occurred. We use the backward congestion notification mechanism. This causes the sender to half its TCP window until further acknowledgement. This gives the congested node enough time to tackle the congestion problem without dropping any packets. This mechanism is achieved through ICMP Source Quenches. We also forward congestion notification to tell the destination to stop sending requests for packets. This is also done with the help of ICMP Source Quenches. Our suggested algorithm introduces some modifications to the original BFCN algorithm [14]. It ensures that the BECN sender is not unduly aggressive by creating a delay of one RTT before sender starts to increase its congestion window after a window reduction. This gives the network time to alleviate the state of congestion before packet injection rate is increased. The modified algorithm requires the use of a single bit to differentiate between an ISQ due to a marked packet and an ISQ due to a dropped packet. An ISQ due to a marked packet would have this bit set so that the sender on receipt of the ISQ detects it should wait an RTT before increasing its window according to the TCP algorithm. When the bit is unset, the sender assumes that the ISQ was due to a dropped packet.

IV. SIMULATION RESULTS

Compared to DIBS alone, DIBS+ Credit approach +BECN improve query completion time (QCT), with little collateral damage to background flows (FCT). (Background inter arrival time: 110ms; query arrival rate: 300 qps; response size: 20KB). TCP has its own window-based flow control mechanism [9] which interprets lost or delayed packets as evidence of congestion. Studies by Fang and Chen of Sandia National Laboratories show that TCP connections over credit get close to fair shares of bandwidth and achieve full link utilization. With our approach using credit, the performance was improved by 50% and the packet loss was greatly reduced. The experimental results confirm that detouring with feedback and credit-based flow control substantially increases efficiency under congestion. For example, for multiple TCP sessions compete for bandwidth over a single link through a switch with a 100-packet buffer, the efficiency is only 30 percent, and with this approach, the efficiency rises to over 98 percent.

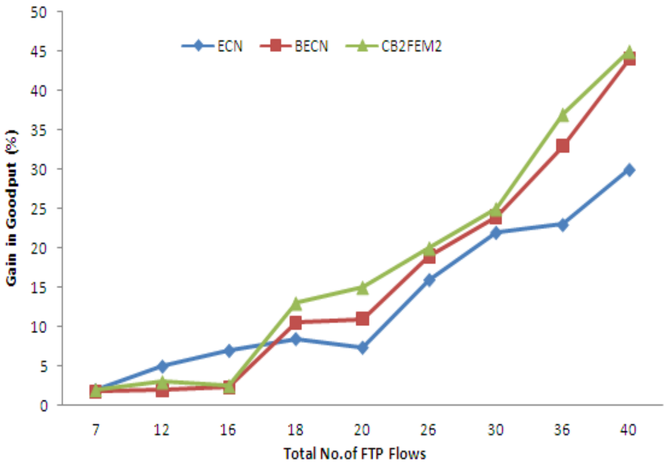

Fig .2 Result showing goodput in ECN, BECN, CB^2FEM^2 for corresponding FTP Flows

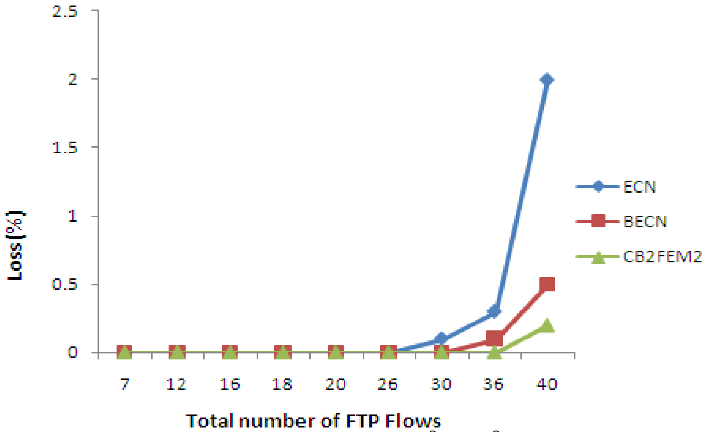

Fig.3, Result showing loss (%) in ECN, BECN, CB^2FEM^2 for corresponding FTP Flows

From Fig.2, we deduce that the goodput (number of data packets received by the receiver) for the approach CB^2FEM^2, and from the Fig.3, we can see that the loss (%) is our approach is much lesser than that of other methods. We can successfully conclude that our approach, CB^2FEM^2, provides better results and performance.

V. CONCLUSION

From our simulations using the proposed mechanism, we conclude that integrating all these methods in a novel way, we are decreasing the delay by a considerable amount, in sending the packets from source to destination. The rerouting after detouring coupled with the Backward and Forward Explicit Congestion Notification and the idea of credit based approach decreases delay in transmission as well as avoid loss of packets by detouring them, not dropping them.

We can implement a number of other techniques into our method, as follows. If all the channels in a network are congested, then implementing DIBS isn't possible. So, we could incorporate priority based detouring, assigning priorities to packets, and detouring only those packets which are highly and moderately important, and not detouring the packets with very low priority. The DIBS process is iterative, meaning that it continues to detour at different places to avoid congestion. This could greatly increase the transmission time. Hence, we could limit the number of times a packet can be detoured.

We could explore our proposed algorithm in internetworks and large scale networks as well, in the future.

REFERENCES

[1] Andrew S.Tanenbaum ,Computer Networks, 4th ed.,
[2] Michael Welzl, Network Congestion Control, managing internet traffic
[3] H. T. Kung and Robert Morris - Credit based Flow Control for ATM Networks in IEEE Network Magazine
[4] Blackwell, et al. "An Experimental Flow Controlled Multicast ATM Switch," Proceedings of First Annual Conference on Telecommunications R&D in Massachusetts, Vol. 6, pp. 33-38, October 25, 1994.
[5] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data center TCP (DCTCP). In SIGCOMM, 2010.
[6] J.Nagle – Congestion control in TCP/IP internetworks, RFC 896, January 1984
[7] W.Prue, J.Postel "Something a host could do with a source quench : The source quench introduced delay (SQuID), Internet RFC 1016, July 1987
[8] S. Kandula, J. Padhye, and P. Bahl. Flyways to de-congest data center networks. In HotNets, 2009.
[9] V. Vasudevan, A. Phanishayee, H. Shah, E. Krevat, D. Andersen, G. Ganger, G. Gibson, and B. Mueller. Safe and effective fine-grained TCP retransmissions for datacenter communication. In SIGCOMM, 2009.
[10] H. T. Kung, T. Blackwell and A. Chapman, "Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing," Proc. ACM SIGCOMM '94 Symposium on Communications Architectures, Protocols and Applications, August 31- September 2, 1994, pp. 101-114.
[11] Frank Akujobi, Ioannis Lambadaris, Rupinder Makkar, Nabil Seddigh, Biswajit Nandy – BECN for congestion control in TCP/IP Networks: Study and comparative evaluation
[12] Matt Calder, Rui Miao, Kyriakos Zarifis, Ethan Katz-Bassett, Minlan Yu, Jitendra Padhye – Don't Drop, Detour!
[13] C. Raiciu, S. Barré, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving datacenter performance and robustness with multipath TCP. In SIGCOMM, 2011.
[14] J.Hadi Salim, B.Nandy, N.Seddigh , "A proposal for backward ECN for the Internet Protocol(IPv4/IPv6) ", July 1998
[15] V. Jacobson, "Congestion Avoidance and Control," Proc. SIGCOMM '88 Symposium on Communications Architectures and Protocols, Aug. 1988.
[16] A.Rangarajan, A.Acharya "Early regulation of unresponsive best effort traffic ", ICNP 99, Oct 1999
[17] P.Newman "Traffic management for ATM local area networks " IEEE communications magazine, vol.32 no.8, pp-44-50, Aug 1994.

[18] R.Jain, S.Kalyanaraman, R.Vishwanath "The OSU scheme of congestion avoidance in ATM networks Lessons learnt and extenstions", issue on traffic control in ATM Networks, vol.31, Nov 1997.

[19] S. J. Hadi, U.Ahmed, Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks, RFC 2884, July 2000.

AUTHORS PROFILE

Kranthi Kumar K. currently with Sreenidhi Institute of Science and Technology, Hyderabad, India, working as Asst. Professor in the department of Information Technology. He has 12 years of experience in Teaching. Graduated in B.Tech (CSE) from JNTU Hyderabad in 2003. M.Tech (CSE) from JNTU, Anantapur, AP in 2007. He is about to complete his Ph.D in Computer Science & Engineering from JNT University, Hyderabad. He has 50 publications in various national/international journals and conferences. His area of research is Image Processing- Image Retrieval Systems.

Namratha Peddi, is a B.Tech (IT) Final year student in Sreenidhi Institute of Science and Technology, Yamnampet, gatkesar, Hyderabad. She is studying in the Department of Information Technology. Her Area of Interest are Computer Networks, Network Security, Data warehouse and Data mining. She Published Three international journal papers and her research area is Congestion control in IP and Non IP networks.