Research Article *ISSN: 2319-507X*

Tanvi Harmalkar, IJPRET, 2013; Volume 1(8):572-580 *IJPRET*

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

NEURAL NETWORK ACTIVE QUEUE MANAGEMENT FOR CONGESTION CONTROL

TANVI B. HARMALKAR, H.G. VIRANI

Dept. of Electronics & Telecommunication Engineering Goa College of Engineering.

Accepted Date:

27/02/2013Publish Date:

01/04/2013

Keywords

Congestion control,

neural network,

Routing,

Controller,

Active Queue management

Corresponding Author

Ms. Tanvi B. Harmalkar

Abstract

Rapidly growing needs for networking in the Internet and in intranets make network management increasingly important in today's computer world. Networking has influenced our everyday lives, as the interconnectivity of families and friends has changed the ways in which they communicate and seek information. In the growing world of networking, more and more emphasis is being placed on speed, connectivity, and reliability. Despite the many years of research efforts, the problem of network congestion control remains a critical issue and a high priority, especially given the growing size, demand, and speed (bandwidth) of the networks. Congestion is a complex process to define. The effect of network congestion is degradation in the network performance. And it is a problem that cannot be ignored. So in this paper we present the existing method of congestion control and enhance existing methods so as to achieve better quality of service.

Network congestion occurs when there is too much of data traffic at a node,that the network slows down or starts losing data. It degrades quality of service and also can leadto delays, lost data or eg. Dropped calls on a telephone network. Typical effects include queuing delay, packet loss or the blocking of new connections. Queuing delay is the time a job waits in a queue until it can be executed. Packet loss occurs when one or more packets of data travelling across network fail to reach destination. Network protocols which use aggressive retransmissions to compensate for packet loss tend to keep systems in a state of network congestion even after the initial load has been reduced to a level which would not normally have induced network congestion. Thus, networks using these protocols can exhibit two stable states under the same level of load. The stable state with low throughput is known as congestive collapse.

Congestion is caused when more packets were sent than could be handled by intermediate routers, the intermediate routers discarded many packets, expecting the end points of the network to retransmit the information. However, early TCP implementations had very bad retransmission behavior. When this packet loss occurred, the end points sent extra packets that repeated the information lost, doubling the data rate sent, exactly the opposite of what should be done during congestion. This pushed the entire network into a 'congestion collapse' where most packets were lost and the resultant throughput was negligible. Congestion is directly related to the amount of traffic that is present in the network. More the traffic more will be congestion. Congestion results when the traffic on the network exceeds the channel capacity. A router is able to handle optimum amounts of traffic but if the traffics exceeds the channel capacity, the entire system collapses and all or a majority of packets transmitted will be lost. To avoid congestion, we either increase the resource or decrease the load. Factors resulting in congestion are limited queue size, insufficient memory ,low band width, low CPU processing speed of router etc.

II. ROUTING WITH CONGESTION CONTROL

The neural networks are widely used to solve the routing problem and to manage the congestion in the computer networks. In communication network information is transferred from one node to another as data packets. Packet routing is a process of sending packet from its source node (s) to its destination node (d). On its way, the packet spends some time waiting in the queues of intermediate nodes while they are busy processing the packets that came earlier. Thus the delivery time of the packet, defined as the time it takes for the packet to reach its destination, depends mainly on the total time it has to spend in the queues of the intermediate nodes. Normally, there are multiple routes that packet could take, which means that the choice of the route is crucial to the delivery time of the packet for any (source, destination) pair. Routing algorithms are methods for finding the best way from a node s to another node d. This may be via a large number of other nodes or it may be in the next sub network. On a small, simple network the problem is almost trivial, statically allocating routes and defining them by hand, but when dealing with a huge internetwork such as the Internet this is not possible. Calculating the best route through such a complex system is computationally difficult and impossible to do by hand. If part of network becomes over filled with packets it can become impossible for packets to move. The queues into which they would be accepted are always full. This is called congestion. The routing problem defined by determining the optimal route for a packet from source node to destination node. The routing decision must be made under the network current conditions such that traffic load, congestion and node or link failures. Using the conventional algorithms and particular mathematical programming methods to solve this problem is not recommended for practical purposes. Because, their calculations may take long time, which may leads to slowly response of the networks.

The parallel, distributed processing structure of the neural networks and their ability to learn are justified to use it as a structure for solving the routing problem. As a result, neural networks are considered to solve such kind of optimization problem. Optimal route is obtained by routing decision which is based on observed delay function. Also, for local decision routing, neural network at each node of computer network use just local information to decide to which neighbor node should be send the packet in order to reach its destination quickly. One of the major problems for most communications networks lies in defining an efficient packet routing policy. Routing policy should be able to take into account the congestion. It sends the packet through route that may be long in terms of hops but results in shorter delivery time. The aim of the good routing algorithm is to minimize the effect of network congestion. But, the main problem of conventional routing algorithms that if costs are assigned in a dynamic way, based on statistical measures of the link congestion state, a strong feedback effect is introduced between the routing policies and the traffic patterns. This can lead to undesirable oscillations. The neural networks have available structure of implementing a learning mechanism for data communication networks. for that, a neural network can achieve great accuracy in predicting one particularnetwork problem, namely congestion.

III. ACTIVE QUEUE MANAGEMENT(AQM)

An Active Queue Management system is used to control the length of a queue so that it does not run full, adding its maximum (usually bloated) delay under load. In current TCP Congestion Control, performance degrades since multiple packets are lost, low link utilization and Congestion collapse. Here the role of the router becomes important because controlling congestion is effective in networks and to allocate bandwidth fairly. The problems with current router algorithm are that it uses FIFO based tail-drop (TD) queue management. The two drawbacks with TD are lock-out and full-queue. Lock-out is small number of ows that monopolize usage of buffer capacity Full-queue means the buffer is always full (high queuing delay). So the possible solution is to use Active Queue Management (AQM): A group of FIFO based queue management mechanisms to support end-to-end congestion control in the Internet The goals of AQM are first to reduce the average queue length by decreasing end-to- end delay and second to reduce packet losses by more efficient resource allocation. The methods to achieve this is to drop packets before buffer becomes full and Use (exponentially weighted) average queue length as an congestion indicator. In Passive Queue Management, there are various alternatives when a packet arrives to a full queue, drop the first packet in line (drop-head). When a packet arrives to a full queue, drop random any packet (random drop). Passive Queue Management helps to avoid the lock-out problem, but keep queues full and do not solve global synchronization. In active approach, packets are dropped early before the congestion arises. AQM give sources enough time to react to congestion before queues fill up and do not keep queues full. It drop packets selectively to avoid global synchronization

A. Neural Network AQM

Active Queue Management (AQM) has been widely used for congestion avoidance in Transmission Control Protocol (TCP) networks. Although numerous AQM schemes have been proposed to regulate a queue size close to a reference level, most of them are incapable of adequately adapting to TCP network dynamics due to TCPs non-linearity and time-varying stochastic properties. To alleviate these problems, an AQM technique based on a dynamic neural network using the Back-Propagation (BP) algorithm was introduced. The dynamic neural network is designed to perform as a robust adaptive feedback controller for TCP dynamics after an adequate training period. The performances of the proposed neural network AQM approach were evaluated. The proposed approach yields superior performance with faster transient time, larger throughput, and higher link utilization compared to two existing schemes: Random Early Detection (RED) and Proportional-Integral (PI)-based AQM.

The essence of congestion control strategies for TCP networks is to rapidly recover from network congestion, or to prevent an incipient congestion. This can be achieved by dynamically adjusting window size at the source side or controlling incoming packets to a router at the link side. Numerous TCP schemes that optimally adjust window size for congestion avoidance have been explored in the last decade. The first widely used scheme, TCP Tahoe, was later modified to TCP Reno, currently the most popular TCP. The congestion window in these protocols is based on the Additive Increase Multiplicative Decrease (AIMD) algorithm: congestion window size is increased by one packet per acknowledgement (ACK) but is halved if a source receives three duplicate ACK signals or does not receive any ACK within a given round-trip time. AQM provides congestion information acquired from the link side to the sources. The objective of an AQM is primarily to proactively respond to network congestion as its queue begins to increase. Rather than simply waiting for a congested queue to overflow and then tail drop all subsequently arriving packets, it maintains queue size at a predefined level in the router. RED is a popular example of an AQM scheme. In RED, the router calculates the drop probability using a current queue size. The incoming packets are passed, dropped or marked, based on this probability. By discarding or marking a single packet, the router sends an implicit or explicit warning to the source. As a response to the warning, the source is expected to adjust the congestion window size to reduce its transmission rate. The drop probability is often linearly proportional to queue length. Although RED is an effective TCP congestion control, it can induce network instability and major traffic disruption if not properly configured. Hence, optimal parameter selection for RED design under different congestion scenarios has been a problem. Moreover, even if optimally selected, the parameter values must be adjusted in real-time implementations because TCP dynamics change with the number of active TCP flows. A more sophisticated adaptive control strategy for AQM in TCP networks using a dynamic artificial neural network AQM control was presented. The control can promptly adapt its operation to the nonlinear time-varying and stochastic nature of TCP networks. Neural networks have been widely applied in the last two decades in a variety of engineering fields such like signal processing, process control, communication systems, etc. They are iteratively trained by a proper learning algorithm to minimize a selected performance measure. As a result, unlike

RED and classical linear control based approaches, neural networks are able to determine the optimal AQM system parameters values autonomously after adequate training. Following training, the neural network operates as an adaptive and robust controller that can provide excellent performance even for environmental conditions not included in the training data set. The dynamic neural network controller presented is trained to regulate the actual queue size close to a reference value determined by network requirements. After training, the neural network operates as an adaptive controller under changes in TCP dynamics.

We choose a multi-layer recurrent (including feedback) dynamic neural model because of its well-known advantages. There are mainly two methods for training recurrent neural networks: a back-propagation-trough-time algorithm and a real-time recurrent learning algorithm. For simplicity, we derive a learning procedure by the general backpropagation (BP) method.

The block diagram of TCP congestion control with the neural network is shown in Fig (1)

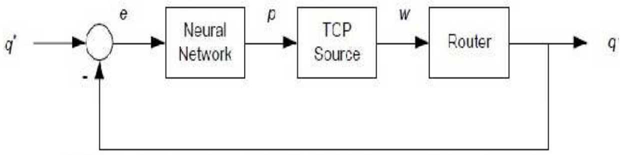

Fig (1): Neural Network AQM of TCP Network

In Fig.(1), the congestion window size, w of the TCP source is determined by the probability, p calculated from the neural network. Queue dynamics at the link side is affected by w. The neural network control system minimizes the error signal, e between the actual queue size, q and the reference queue target value, q*. The loss probability, p is the control input to the TCP source. Recurrent neural network for AQM is shown in Fig (2).

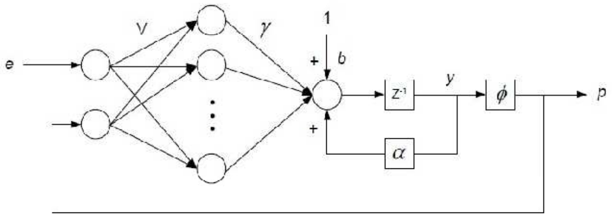

Fig (2): Recurrent Neural Network for AQM

A dynamic recurrent neural model including one feedback connection and a three-layer perceptron is selected. The input vector of this neural network includes the error signal, e and the probability, p as a feedback signal from its output. From the various parameters shown in Fig (2) the network output is obtained.

B. Neural Network Model Predictive Controller

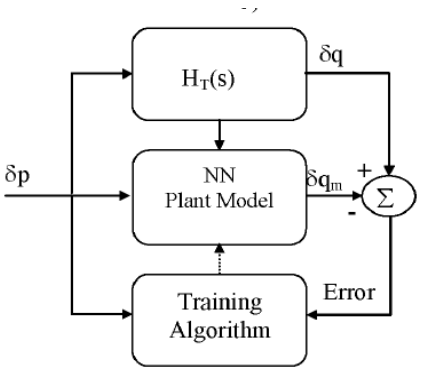

Fig (3) Predictive Controller

Plant forward dynamics identification is at the first stage of any model predictive control structure. Fig (3) shows a

Neural structure for the system model identification. The neural plant model and the actual system outputs are compared and an error signal is generated which is used to train the neural network. The plant parameters, and the network weights are changes such that the error signal is minimized. The main objective of the identification process is to find the function h. In this application we train a neural network to approximate the nonlinear function h.

IV. CONCLUSIONS

With the growing number of network users, more emphasis is placed on the maintenance and reliability of the networks. When network problems occur, they often catastrophically break the service for those enterprises or individuals that depend on the network connection. Sometimes, such breaks of service are just annoying, but for companies and commercial users they often mean lost revenues on the order of thousands, or even millions, of dollars. Such breaks have become a significant problem in all forms of electronic commerce. To address this difficulty, a system is needed to insure network availability and efficiency by preventing such costly network breakdowns. The Neural Network AQM and Neural Network Model Predictive Controller are discussed in this paper. The adaptive nature of the MPC controller algorithm exhibits a stable behavior even under severe modeling error. For better quality of service the simulation were carried out based on queue size parameters.

REFERENCES

1. Cho, H.C.; Fadali, M.S.; Hyunjeong Lee; , "Neural Network Control For TCP Network Congestion," American Control Conference, 2005. Proceedings Of The 2005 , Vol., No., Pp. 3480- 3485 Vol. 5, 8-10 June 2005 Doi: 10.1109/ACC.2005.1470511

2. Rahnamai, K.; Gorman, K.; Gray, A.;, "Model Predictive Neural Control Of TCP Flow In AQM Network," Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual Meeting Of The North American , Vol., No., Pp.493-498, 3-6 June 2006 Doi:10.1109/NAFIPS.2006.365459

3. Abdulkareem Y. Abdalla , Turki Y. Abdalla,Khloud A. Nasar;,"Routing With Congestion Control In Computer Network Using Neural Networks",International Journal Of Computer Applications (0975 – 8887) Volume 57– No.2, November 2012

4. Vishal Mishra, Wei-Bo Gong, And Don Towsley, "Fluid-Based Analysis Of A Network Of AQM Routers Supporting TCP Flows With An Application To RED," In Proceedings Ofacm/SIGCOMM, 2000.

5. Hollot C., Misra V., Towsley D., Gongw., "A Control Theoretic Analysis Of RED," Proceedings Of IEEE Infocom 2001.

6. S. Floyd, “Recommendations On Using The Gentle Variant Of RED,” Http://Www.Aciri.Org/Floyd/Red/Gentle.Html, 2000.

7. V. Misra, W. B. Gong, And D. Towsley, “Fluid-Based Analysis Of A Network Of AQM Routers Supporting TCP Flows With An Application To RED,” Proceedings Of acm/SIGCOM, Pp. 151- 160, 2000.