A UNIFIED APPROACH TO AMELIORATE ACTIVE QUEUE MANAGEMENT OF NETWORK ROUTERS

Zhongwei Zhang and David Lai*

Department of Mathematics and Computing
University of Southern Queensland
Australia

Shan Suthaharan

Department of Mathematic Sciences
University of North Carolina at Greensboro
USA

ABSTRACT

The layering technique has divided TCP/IP protocols into a few layers and assigned each layer with some special functionality. The mechanism of flow and congestion control in TCP at the transport layer can be dispersed to IP at the network layer where there used to be a simple strategy (ie. Drop-Tail) of managing the router's buffer. Active queue management can improve network efficiency by detecting impending congestions in Internet routers. The first such effort was Random Early Detection (RED) introduced in the early 1990's, and a number of active queue management algorithms have been developed since then.

The essence of AQMs is to adjust the packet dropping rate before the buffer becomes completely full. The dropping probability is a determining factor in affecting the overall throughput and transmission delay. Various AQM algorithms have been developed to produce a just-right dropping rate largely based on the statistical data. Unfortunately, all these algorithms possess some drawbacks in one way or another.

In this paper, we devise a unified approach to determining the packet dropping possibility, which is sensitive to all factors that might have some ultimate effect on the buffer's occupancy. The proposed approach has been tested on a simple network using OMNET++ [1].

1. INTRODUCTION

Computer networks have dramatically altered the way of communication; Networks such as the Internet have been overwhelmingly deployed and their efficiency has been adversely affected simply by too much traffic injected by millions of users and servers [7]. As an important component of networks, routers or switches need to be well managed, in particular the queues in the routers need to be efficiently used. The current approach of managing queues on the routers are mainly based on traffic statistics and link load. The relationship among these statistical data are complicated and dynamic, the algorithms of managing router queues have been dampening and this problem will deteriorate when more complicated traffic is injected into the networks.

Exploring an active approach to queue management has been a hot topic in past years [2, 3, 4, 5, 6]. The essence of this approach is to start reacting to the TCP senders before the possible events happen according to the previous traffic and buffer situations. The ultimate goal of AQM is to maintain the buffer at an optimal level. In other words, if there are too many packets queued up in the buffer, the AQM will increase the dropping probability, then more packets will be dropped, and the sending TCPs will reduce their sliding window size. If there are too few packets in the buffer, then AQM will cut off the dropping probability, there are less packets dropped, the sending TCPs will enlarge their sending window size. Many AQM schemes based on this idea have been devised which are largely applying statistical theory. A few representatives include RED [2], BLUE [3], GREEN [4], and their variants such as ARED, FRED. These AQM schemes are very idiosyncratic, consequently they all suffer the problem of inflexibility, in that the structure of their control systems need to be entirely reconfigured or changed if new factors are integrated.

In this paper, we will propose a unified approach to manage the router's buffer. The rest of the paper is organized into 5 sections. The new approach is presented in Section 2. In Section 3, we will use the proposed approach to simulate a simple network and its intelligent controller on the dropping probability of AQM in the routers. The feasibility of this new approach has been shown in Section 4. This paper is concluded in Section 5, and in it we point out how we can further our research in the future.

2. A NEW APPROACH FOR AQM

The new approach we are proposing in this paper is to take advantage of AI technology instead of the statistical strategies. Fist of all, there is some substantial amount of statistical data available such as the queue length, link utilization

*The first author acknowledges the support UNCG provided during his sabbatical leave at UNCG, Feb-June, 2003 which make this research possible

and congestion or packet drop rates. Secondly, there are is common knowledge we can collect, and this knowledge has been studied in control theory such as process management and scheduling. Thirdly, we are convinced in a lot of areas some management problems can be addressed much better and more effectively using AI technology than using conventional statistical methods. Based on these observations, we believe that AI technology most likely perform more effectively on the AQM problem. In our research, we explore an AI-based AQM algorithm, in particular a fuzzy logic AQM algorithm will be developed.

2.1. Identify the affecting factors

There are many factors which would impact on the network performance in terms of throughput and propagation delay. Common factors include

- link utilization: Current link utilization is a very good indication of current traffic. If the link utilization is pretty high, this indicates the traffic at the moment is very busy, and there is a high chance of the buffer filling up soon.
- fairness: There is an experiential relationship between the RTT and the dropping probability which has been used in GREEN. We will not take this relationship into consideration.
- packet loss: TCP uses the number of packet loss to adjust the sliding window size. BLUE uses the packet loss event and link idle event to adjust the dropping probability. Therefore the number of packet loss is a real indicator of congestion.
- average queue size: The average queue length has been used by many researchers. It has received a lot of criticism. We argue that this criticism is caused by overusing it where the average queue length has been used as the only determining factor of dropping probability. It is justifiable that we just take it as one factor, but not the only one.

These factors sometimes are interrelated, sometimes might be independent. For instance, link utilization is proportional to the RTT, but has nothing to do with the average queue length.

2.2. Discover the relationship between the congestion and the factors

The relationship between these factors and the congestion level is implicit. It needs careful human observation and expert insightful analysis to discover these relationships. These relationships are represented with fuzzy rules in the form of "**if x is High and y is Positive, then z should be Small**".

Having observed the congestion, we realized that the dropping probability and the average queue length (AQL) has the following relationship – the inverse proportional relation.

$$\text{dropping probability} \propto 1/AQL$$

- if AQL is big, the buffer is very close to being full, the dropping probability should be increased significantly.
- if AQL is small, this indicates the buffer is still relatively empty, then the dropping probability should be smaller than its current level.

Similarly, dropping probability has an inverse proportional relationship to the number of dropped packets and idle events.

$$\text{dropping probability} \propto \frac{\text{\# of idle event}}{\text{\# of dropping packets}}$$

- if the number of dropping packets is big, the buffer has started dropping packets and the buffer is full already, the dropping probability should be increased significantly.
- if an idle event occurs, this indicates the buffer has been empty, then the dropping probability can be set to a lower value.

The difference between the current nominal link utilization and its transmission capacity (ie. Δu) can also be used to tune the dropping probability. For instance,

- if Δu is a large positive, then the dropping probability should be raised up significantly.
- If Δu is a large negative, then the dropping probability should be cut down significantly.

2.3. Expand the base rule-base

The base rule-base is a rule set with four rules in it initially, shown in Table 1. We then translate the knowledge into rules and insert these new rules in the rule-base. The new rule-base can be further expanded if new knowledge is discovered.

	z	Δz	δdp
1	SMALL	SMALL	SMALL
2	SMALL	BIG	NORMAL
3	BIG	SMALL	NORMAL
4	BIG	BIG	BIG

Table 1. rule base

2.4. Prototype Intelligent control

Fuzzy logic and the theory of fuzzy systems provide a formal and computationally oriented basis for dealing with approximate reasoning and intrinsically imprecise terms. They have been proved excellent tools to cope with uncertain and possibly contradictory information in a simple, elegant and reliable way.

The first step to implement a fuzzy control system is to determine the membership functions for each linguistic value. This step includes identifying the linguistic variable's domain and membership function's supporting parameters.

The second step is to evaluate which fuzzy operators such as AND or OR are suitable for doing this fuzzy reasoning. AND and OR could be implemented as MIN or MAX, respectively.

The third step is to defuzzify the consequent linguistic variable's value to a crisp value, in which different methods give slightly different results. The center point of gravity method is the most popular one, but it is also the most complex one.

3. SIMULATION USING OMNET++

We use the OMNeT++ simulation environment to design a network with two clients and one server, a router is connected with the clients and the server is in the middle.

Following this approach, we implement an intelligent controller for this network using OMNeT++. It is a case study on the proposed approach. The control feasibility of this intelligent control system is demonstrated by running a few simulations. Other objectives of the simulations include: (1) to test the feasibility of introducing AI technology in the OMNeT++ environment. (2) to evaluate the performance of the AI based AQM against the statistical methods.

3.1. Design network using NED

In OMNeT++, a network is represented as a compound module which defines many submodules in NED language. Note that a submodule could be composed of many other submodules, eventually all these compound modules and submodules are composed of a few simple modules.

3.2. Implementing the controller in AQM

The fuzzy controller for determining the dropping probability is implemented in the router's module, in particular in the *activity* function.

Figure 1 depicts the structure of the intelligent controller, which is composed of multiple fuzzy engines. Each fuzzy engine is a simple fuzzy reasoning system and is responsible for each identified factor.

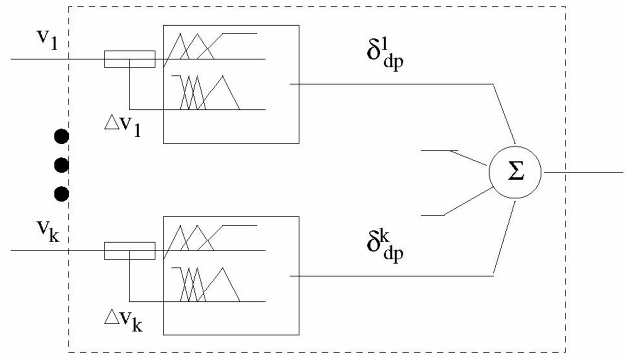

Fig. 1. The fuzzy logic system

3.2.1. *Intelligent controller*

An intelligent controller consists of several fuzzy logic systems. Each fuzzy logic controller is responsible for an individual control factor. The overall packet dropping probability is the averaged sum of all dropping probability. I.e. $\delta p = \sum_1^K \frac{1}{k_i}\delta_i$ where $k_i \geq 0, i = 1, 2, \cdots, K, \sum_1^K k_i = 1$.

3.2.2. *Fuzzify the crisp values*

Within each fuzzy engine, suppose there are m linguistic values for the linguistic variable U, and n for V. The fuzzified value of crisp variable x on the linguistic vale N_1 will be $N_1(x_0)$.

3.2.3. *Fuzzy reasoning*

In the fuzzy engine, the reasoning is carried out by a fuzzy operator called "AND". Which means two linguistic values will be ANDed, ie. the minimal value will be selected. $W = min(U, V)$.

3.2.4. *Defuzzifying*

For the same crisp value of z and δz, a number of linguistic values will be generated by running through all the rules in the rule base. We will choose a simple approach to get the overall fuzzied value, with the simple approach is to get its mean value as follows. $z = \frac{1}{R}\sum_{i=1}^R W_i$.

4. EXPERIMENT RESULT

A controller developed by following the proposed approach has been tested on the simple network. The parameters of the simulation are described in Table 2.

The simulated network is shown in Figure 2. Note that within the simulation, the traffic generated by the clients are based on the Poison-Pareto model [7]. Different kinds of traffic are used to test the intelligent controller.

simulation time	1000 seconds
mean message length	120
mean time interval	1
buffer length	10000
router speed	260 bits

Table 2. Simulation parameters

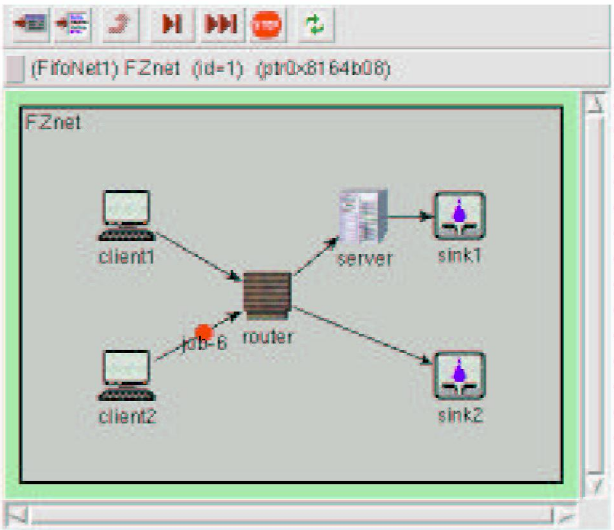

Fig. 2. The network simulated using OMNeT++

For a simple case where only one factor is considered, the intelligent controller relies on one fuzzy engine. It actually is a generic and fuzzified version of either RED, or BLUE or GREEN. However, the special case for a multi-engine controller is a two engine controller. This is entirely new. Figure 3 shows some preliminary results.

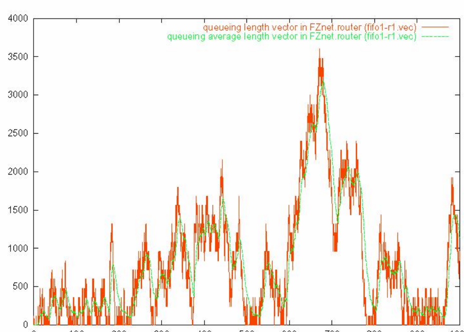

Fig. 3. The dynamics of queue length and average queue length

5. CONCLUSION

This paper has proposed a unified approach to calculate the dropping probability in AQM schemes. This method is scalable in the sense that any new traffic and router variables, which have some influence on the network performance, can be easily incorporated into the queue management without changing the structure of the control system.

The unified approach has laid out a systematic procedure of manufacturing a fuzzy controller to address complex queue management by either varying the supporting parameters of the membership functions and/or expanding the fuzzy rule base.

The developed intelligent controller has been tested on a simple network using OMNeT++. Preliminary results have shown some promising perspectives. In our experiments, no UDP traffic is introduced and the number of connections is relatively small. Future work will include the consideration of a large number of TCP connections and the integration of UPD traffic over wireless networks.

6. REFERENCES

[1] Andras Varga, “Omnet++: Discrete event simulation system,” .

[2] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *ACM/IEEE Transactions on Networking*, vol. 1(4), pp. 397–413, August 1993.

[3] Wu chang Feng, Kang G. Shin, Dilip D. Kandlur, and Debanjan Saha, “The blue active queue management algorithms,” *IEEE/ACM Transaction on Networking*, vol. 10(4), pp. 51–528, August 2002.

[4] Wu chun Feng, Apu Kapadia, and Sunil Thulasidasan, “Green: Proactive queue management over a best-effor network,” in *IEEE BlobeCom (GLOBBCOM)*, Taipei, Taiwan, November 2002, pp. 1–5.

[5] B. Wydrowski and M. Zukerman, “Green: An active queue management algorithm for a self managed internet,” in *Proceedings of ICC 2002*, New York, 2002, vol. 4.

[6] R. Pletka, M. Waldvogel, and S. Mannal, “Purple: predictive active queue management utilizing congestion information,” in *Proceedings of Local Computer Networks (LCN), pages 21–30,*, Oct 2003.

[7] M. Zukerman, T. Neame, and R. Addie, “Internet traffic modeling and future technology implications,” in *Proc. of IEEE Infocom’03*, S. Francisco, CA, USA, Apr. 1–4 2003.