Chapter 42
Active Queue Management Mechanism Based on DiffServ in MPLS Networks

Yang Jiao and Li Du

Abstract Active Queue Management is the key to congestion control and enhancing IP QoS. MRED can support MPLS networks and take advantage of classifying mechanism of DiffServ and mark different businesses with different drop precedence. However, it has several limitations. In this paper, I-AMRED is proposed in order to reduce packet loss ratio and raise throughput in MPLS networks based on DiffServ and MRED. Thus, it can also adaptively control average queue length, and diminish sensitiveness to control parameters and improve stability with the service flow bursts. Two experimental schemes are designed and implemented on NS-2. Experimental results show that I-AMRED algorithm increases throughput of networks and largely decreases packet loss ratio of AF PHB traffic. When data traffic is very large or sharply increased, the performance of throughput and packet loss ratio is improved greatly, and has better stability.

Keywords DiffServ · MPLS · QoS · AQM · MRED

42.1 Introduction

In recent years, the continuous increase of traffic and service types has caused great demand in IP QoS and MPLS (Multi-Protocol Label Switching) DiffServ (Differentiated Service) model became dominant technology in solving IP QoS problems. The technological combination and interoperability [1] of MPLS and DiffServ is to take advantage of explicit routing and fast forwarding of MPLS and

Y. Jiao (✉) · L. Du (✉)
College of Information Science and Engineering, Northeastern University, Shenyang, China
e-mail: wyobj619@163.com

L. Du
e-mail: duli26@126.com

W. E. Wong and T. Ma (eds.), *Emerging Technologies for Information Systems, Computing, and Management*, Lecture Notes in Electrical Engineering 236, DOI: 10.1007/978-1-4614-7010-6_42,

the scalability of DiffServ, so choosing MPLS DiffServ environment has more practical meaning. Active Queue Management (AQM) is a mechanism which can drop packets by some strategy before router buffer is full and avoid global synchronization. RFC 2597 recommends that AQM mechanism be used to realize the multiple levels of drop precedence required in the AF PHB. MRED (Multi-level RED) [2] supporting DiffServ and MPLS is an AQM mechanism that can execute different RED policy and calculate drop probability independently for different drop precedence. In this paper, the limitations of MRED are researched and analyzed. Improved schemes are proposed and I-AMRED algorithm is raised. NS-2 platform is used to verify the effectiveness of I-AMRED algorithm. Comparisons of performance parameters such as throughput, packet loss ratio are presented between I-AMRED and MRED in two experiment schemes using NS-2.

42.2 The Limitations of MRED

MRED maintains multiple sets of RED thresholds [3]. With the continuous development of IP networks some limitations appear.

(1) Sensitiveness to RED Parameters Configuration: Configured RED parameters include the maximum threshold (max_{th}), the minimum threshold (min_{th}) and the maximum drop probability (max_p). MRED uses these configuration parameters un-self-adaptively. If RED parameters are configured improperly, the fluctuation of average queue length will become violent [4].
(2) Instability When Traffic Is Large or Sharply Increased: Essentially, linear relation between drop probability (P_b) and average queue length (avg_q) for each drop precedence leads to the instability of throughput value. When avg_q ranges from min_{th} to max_{th}, P_b will increase slowly with the increase of avg_q. When traffic sharply increases, queue buffer would have been filled up, accordingly, throughput will drop and link utilization will reduce [5].
(3) Insufficiency to Reflect Congestion Condition: MRED uses avg_q as unique evaluation parameter to reflect congestion change. However, avg_q can't always reflect congestion condition correctly. When weight value is low and the value of avg_q is high, the condition of congestion may be relieved. At this time, if deciding drop policy only according to avg_q, packet loss ratio will raise.

42.3 I-AMRED Algorithm

42.3.1 Basic Ideas of I-AMRED Algorithm

The purpose of I-AMRED is to remove the limitations of RED, and to accordingly reduce packet loss ratio.

(1) Import ARED Mechanism: In order to reduce sensitiveness to RED parameters configuration of MRED, import adaptive mechanism of ARED and make avg_q range from min_{th} to max_{th} as far as possible. Two judgment principles is introduced, that is reducing max_p value by dividing a factor α when congestion condition of networks is in a low level, and raising max_p value by multiplying a factor β when congestion condition of networks is in a high level. Factor α and β are both more than 1, and control RED policy should be more positive or more conservative by monitoring the change of avg_q every certain period.
(2) Change Linear Relation Between P_b and avg_q: To raise stability of MRED when traffic is large or sharply increased, linear relation between P_b and avg_q should be changed to exponential relation as in Fig. 42.1. The purpose of this change is to reduce P_b when avg_q approaches min_{th} (queue buffer of router is relatively idle) and to raise P_b when avg_q approaches max_{th} (queue buffer of router is almost filled up), accordingly relieving shortage of queue buffer of router before congestion and improving the stability of network throughput.
(3) Import Real-time Queue Length as Another Evaluation of Congestion Condition: To reflect congestion condition more sufficiently and reduce packet loss ratio, import real-time queue length ($qlen$) as another evaluation criterion of congestion condition of queue buffer of router. Three judgments are added:
    (1) $avg\_q \geq max_{th}$: If $qlen$ is less than max_{th}, use $qlen$ instead of avg_q to calculate drop probability, else, drop all arriving packets.
    (2) $qlen \geq max_{th}$: *No matter whether* avg_q *is more than* max_{th}, *drop all arriving packets.*
    (3) Use avg_q to calculate drop probability except the two conditions of the two judgments above.

When weight value of a certain Behavior Aggregate (BA) is very low and at this time congestion is relieved just now and avg_q is more than max_{th} but $qlen$ has

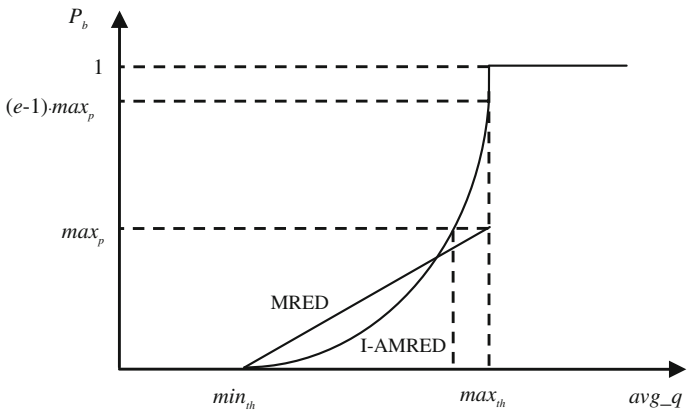

Fig. 42.1 Exponential relation of drop probability and average queue length in I-AMRED algorithm

been already less than max_{th}, it is not essential to drop arriving packets and *qlen* should be used to implement drop policy instead of drop all arriving packet. When *qlen* is more than max_{th}, queue buffer of router has been filled up, if *avg_q* is less than max_{th} and calculate P_b and let packets access, causing higher packet loss ratio, at this time, all arriving packets should be dropped. Because the three judgments above are added, congestion will be relieved earlier when *qlen* is more than max_{th} but *avg_q* is still less than max_{th}. Therefore, packet drop ratio of network will decrease totally.

42.3.2 Concrete Design of I-AMRED Algorithm

42.3.2.1 Calculation Equations

I-AMRED is an algorithm which can differentiates calculation from many BAs. In order to look convenient, variables of I-AMRED as below all aim at a certain BA.

$$avg\_q = (1 - w_q) \times avg\_q\prime + q \times w_q \tag{42.1}$$

$$P_b = max_p \times \exp\left(\frac{avg\_q - \min_{th}}{\max_{th} - \min_{th}}\right) - max_p \tag{42.2}$$

In Eq. (42.1), w_q is set as weight value of certain level drop precedence, *avg_q* is average queue length of last time and initial value of *avg_q* is zero. *q* is instantaneous queue length of sampling time. When *avg_q* ranges from min_{th} to max_{th}, P_b is calculated as in Eq. (42.2). The relation between P_b and *avg_q* in Eq. (42.2) is exponential as in Fig. 42.1.

42.3.2.2 Implementation in NS-2

NS-2 simulation platform contain DiffServ module, where MRED is programmed in the files of "dsredq.cc" and "dsredq.h" [6], so I-AMRED is implemented by modifying program code according to Eq. (42.2).

In the file"dsredq.h", add enumeration variable *status* and integer variable *qlen*, and declare function *updateIAMREDMaxP()*.

In the file"dsredq.cc", define the function *updateIAMREDMaxP()* in order to update max_p parameter periodically according to I-AMRED judgments. Modify the calculation formulae to exponential form and add judgments about *qlen* into the function *enque()*. Add the function *updateIAMREDMaxP()* into the function *calcAvg()* in order to update max_p parameter periodically for every BA.

42.4 Experiments on NS-2 Platform

42.4.1 Experiment 1

The network topology of experiment 1 is set as a simple network. Four source nodes send data to four destination nodes through a bottleneck link which is from a core router to an edge router. Three TCP agents are set on three source nodes which are marked with different drop precedence. An UDP agent is set on one source node, in which traffic is the sum of the three TCP sources. Links between edge routers and core router adopt TSW3CM. All nodes are configured as MPLS nodes.

Two schemes are set in experiment 1. First, let four source nodes send data in a high constant rate at the beginning of simulation time. Second, let two source nodes send data at the beginning of simulation time, and then at the 3rd second let other two source nodes send data. The purpose is to verify stability when traffic sharply increases.

42.4.2 Experiment 2

The network topology of experiment 2 is set as Fig. 42.2, and the purpose is to examine whether I-AMRED will be stable when the network scale is large. C is core router, E1, E2, …, E11 are source edge routers, E0 is destination edge router, S1, S2, …, S11 are source nodes, D is destination node. Every source node has a TCP agent marked by different drop precedence. These TCP agents use random number generator and adopt exponential distribution generating a number as a time interval of TCP transmission. Pareto model is used to generate a random number to assign a size of file needed to be transmitted. The arrival of packet obeys Poisson distribution. The link between C and E0 is bottleneck link. Accordingly, uncertainties of actual network distribution and data transmission can be simulated and stability of I-AMRED can be verified more practical.

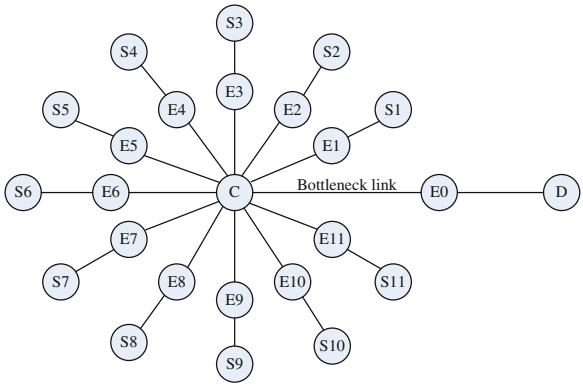

Fig. 42.2 Network topology of experiment 2

42.5 Results and Discussions

42.5.1 Experiment 1

When four source nodes send data in constant rate at the same time, set simulation time as 10 s, and throughput comparison figure between MRED and I-AMRED is shown in Fig. 42.3. So I-AMRED improves throughput on the whole, and is more stable than MRED at the beginning of simulation time. Packet loss ratio comparison between MRED and I-AMRED is shown in Table 42.1.

When letting two TCP sources send data at the beginning of simulation time, and then at the 3rd second let another TCP source and UDP source send data at the second scheme in experiment 1, throughput comparison figure between MRED and I-AMRED is shown in Fig. 42.4. So it is clear that when traffic is sharply increased, throughput using I-AMRED increases as simulation time goes on.

Packet loss ratio comparison between the two algorithms is as Table 42.2, so I-AMRED algorithm make packet loss ratio of TCP (AF PHB) reduce much, but UDP packet loss ratio raises, for drop precedence of UDP packets is high, and traffic of UDP is the sum of three TCP sources. As a whole, I-AMRED make average packet loss ratio decrease by over 10 %.

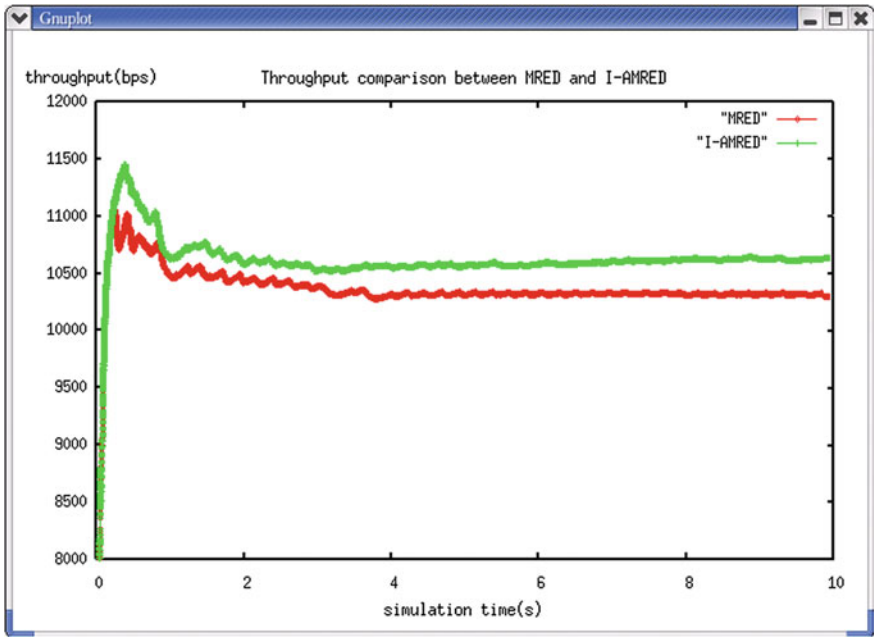

Fig. 42.3 Throughput comparison when sending rate is constant in experiment 1

Table 42.1 Drop packet ratio comparison when sending rate is constant

	TCP1 (%)	TCP2 (%)	TCP3 (%)	UDP (%)
MRED	5.699177	5.215420	6.076389	6.785214
I-AMRED	2.628697	3.158488	2.604699	8.526743

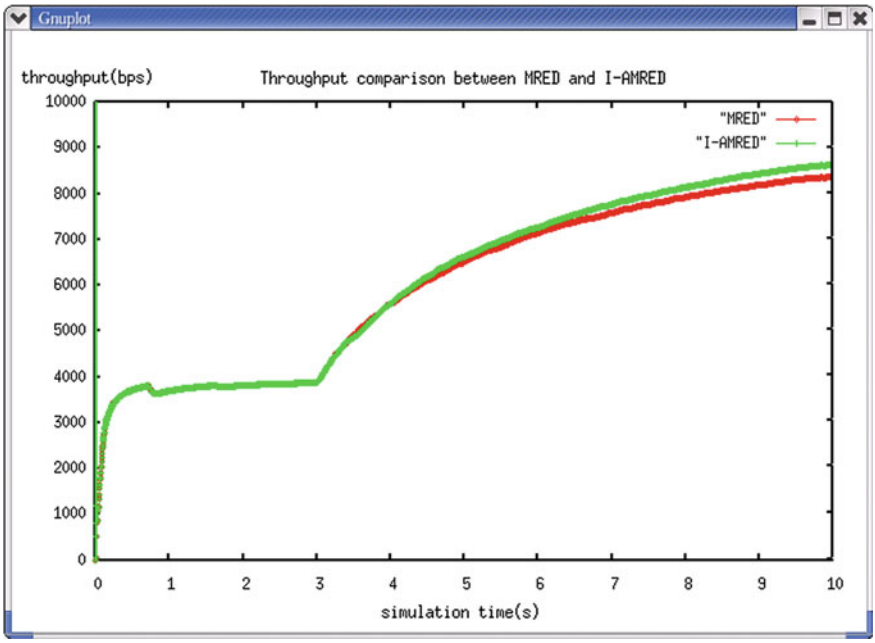

Fig. 42.4 Throughput comparison when traffic increases sharply in experiment 1

Table 42.2 Drop packet ratio comparison when traffic sharply increases at 3rd second

	TCP1 (%)	TCP2 (%)	TCP3 (%)	UDP (%)
MRED	6.156234	6.538661	5.343511	5.996960
I-AMRED	4.559118	4.552129	2.422407	10.309650

42.5.2 Experiment 2

In experiment 2, throughput comparison figure between MRED and I-AMRED is shown in Fig. 42.5, so I-AMRED make throughput even more stable and higher when network topology is more complex. Packet loss ratio comparison between MRED and I-AMRED is shown in Table 42.3.

Four TCP agents selected are examined. It is clear that packet loss ratios of four TCP sources all decrease after using I-AMRED, but the decrease of low drop

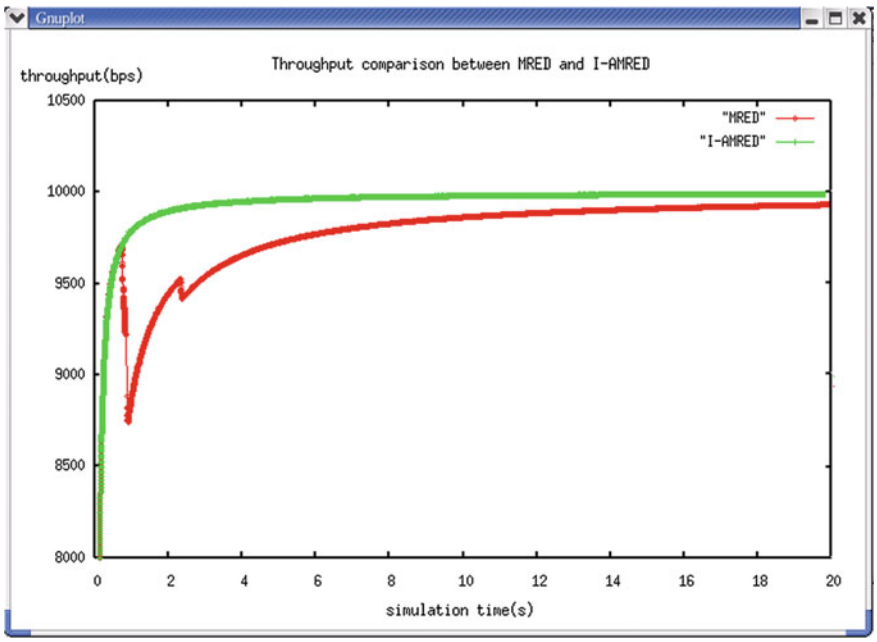

Fig. 42.5 Throughput comparison of MRED and I-AMRED in experiment 2

Table 42.3 Drop packet ratio comparison in experiment 2

	TCP1 (%)	TCP2 (%)	TCP3 (%)	TCP4 (%)
MRED	1.800000	2.824859	8.387097	9.409190
I-AMRED	0.908998	1.821934	6.895765	8.134652

precedence is more obvious, the decrease of high drop precedence is less. As a whole, average packet loss ratio decreases notably.

42.6 Conclusion

The results of experiment 1 and 2 indicate that I-AMRED raises throughput especially when traffic is large. And it sharply increases compared with MRED, and notably reduces packet loss ratio of AF PHB. Because of the improvement of throughput and packet loss ratio, sensitiveness to RED control parameter decreases, while stability of network throughput increases. These advantages make I-AMRED algorithm very suitable for current IP networks in which the amount and type of user and traffic expand rapidly. In summary, I-AMRED algorithm is ideally suitable for MPLS networks based on DiffServ.

Acknowledgments This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant Nos. N100204003.

References

1. Rahimi, M., Hashim, H., Rahman, R.A.: Implementation of quality of service (QoS) in multi protocol label switching (MPLS) networks. In: IEEE Conference Publications, pp. 98–103 (2009)
2. Qadeer, M.A., Sharma, V., Agarwal, A., Husain, S.S.: Differentiated services with multiple random early detection algorithm using ns2 simulator. In: Computer Science and Information Technology ICCSIT 2nd IEEE International Conference, pp. 144–148 (2009)
3. Makkar, R., Lambadaris, I., Salim, J.H., Seddigh, N., Nandy, B., Babiarz, J.: Empirical study of buffer management scheme for DiffServ assured forwarding PHB. In: IEEE Conference Publications, pp. 632–637 (2000)
4. Yang, W.: Research on Congestion Control Mechanism Based on Red algorithm. Nanjing Information Engineering University, Nanjing (2010)
5. Hollot, C.V., Misra, V., Towsley, D.: A control theoretic analysis of RED. Proc. IEEE INFOCOM **33**, 1013–1015 (2004)
6. Cao, Z.B.: Optimization and simulation of RED algorithm. Comput. Technol. Dev. **20**, 188–191 (2010)