Chapter 5

An Evaluation of Different Marking Algorithms in Assured Forwarding Service

T. Yang*, Z. Chen*, A. Hafid[&], D. Makrakis*

**School of Information Technology & Engineering, University of Ottawa
Colonel By Hall (A617), P.O.Box 450 Stn A, Ottawa, Ont., K1N 6N5, Canada
{tyang,zhchen,dimitris}@site.uottawa.ca*

[&]University of Western Ontario, Computer Sciences Dept., London, Ontario, Canada

Key words: Assured Forwarding, Differentiated Service, TSW, TCM, Improved-TSW.

Abstract: In this paper, we overview two popular marking algorithms that are in use in networks that support the Assured Forwarding (AF) Service of Differentiated Services (DS); the algorithms are: Time Sliding Window (TSW) algorithm and Three Color Marker (TCM) algorithm. We present the simulations we have performed to evaluate and compare the performance of these two algorithms in an AF service-capable network. Then, we propose a new marking algorithm, called Improved-TSW algorithm, which is designed by combining TSW algorithm with a token bucket. A performance evaluation, using simulations, show that the new algorithm provides a better performance, in an AF service-capable network, than the other algorithms.

1. INTRODUCTION

There are many algorithms available for metering and marking packets at the boundary nodes of a differentiated service (DS) [1] domain, among which, Time Sliding Window algorithm [2] and Three Color Marker algorithm [3] are the two most popular ones. The objective of this paper is to evaluate and compare the performance of these two marking algorithms in an AF service-capable network [4] and propose a new marking algorithm

[&] A. Hafid is currently with Telcordia Technologies (formerly Bellcore), NVC3X359, 331 Newman Springs Road, Red Bank, NJ, 07701; e-mail: ahafid@research.telcordia.com

named Improved-TSW algorithm. We performed a series of simulations to evaluate the marking algorithms using OPNET 5.1 [5], a network simulation tool developed by Mil3. In sections 1.1 and 1.2, a brief description of two existing marking algorithms (TSW and TCM) is presented.

1.1 Time Sliding Window Algorithm

Time Sliding Window (TSW) [2] algorithm is a rate estimator algorithm. TSW provides a smooth estimate of the TCP sending rate over a period of time. With the estimated rate, Avg_rate, the marking algorithm can mark packets as OUT (out of profile) packets once the traffic exceeds a certain target rate.

A rate estimator is used to smooth out the bursts of TCP traffic as well as to be sensitive to instantaneous sending rate. The design of TSW is very simple. TSW maintains three state variables: Win_length, which is measured in units of time, Avg_rate, the rate estimate upon each packet arrival, and T_front, which is the time of last packet arrival. TSW is used to estimate the rate upon each packet arrival, so variables Avg_rate and T_front are updated each time a packet arrives, but Win_length is pre-configured when the profile meter is installed (in our simulations, we chose the Win_length on the order of the RTT of the TCP connections according to [2]). Follows is a description of the main steps of TSW algorithm.

Initialization:

Win_length = a constant;
Avg_rate = connection's target rate, R_T ;
T_front = 0;

For each packet with size pkt_size arrival at time arrive_time:

 If it is an IN (in profile) packet

 Bytes_in_TSW = Avg_rate * Win_length;

 New_bytes = Bytes_in_TSW + pkt_size;

 Avg_rate = New_bytes / (arrive_time - T_front +

Win_length);

 T_front = arrive_time;

 If Avg_rate > $1.33 * R_T$

 Remark this packet as OUT packet;

 Endif

 Endif

1.2 Three Color Marker Algorithm

The Three Color Marker (TCM) [3] algorithm is a kind of token bucket algorithm. It meters an IP packet stream and marks its packets either green, yellow, or red. Marking is based on a Committed Information Rate (CIR) and two associated burst sizes, a Committed Burst Size (CBS) and an Excess Burst Size (EBS). CIR is measured in bytes per second for IP packets. The CBS and EBS are measured in bytes. A packet will be marked as green if it doesn't exceed the CBS, yellow if it does exceed the CBS, but not the EBS, and red otherwise.

The Meter meters each packet and passes the packet and the metering results to the Marker; in our simulations, the Meter operates in Color-Aware mode [3]; this means that the Meter assumes that the packets have been pre-

marked. Then, the Marker remarks, if necessary, the packet according to the results of the Meter.

The Meter measures the packet stream by using two token buckets, C and E, that both share the common rate CIR. The maximum size of the token bucket C is CBS and the maximum size of the token bucket E is EBS. The token bucket C and E are initially (at time 0) full, i.e., the token count $TC(0) = CBS$ and the token count $TE(0) = EBS$. Thereafter, the token counts, TC and TE, are updated CIR times per second as follows:

If $TC < CBS$

TC = TC + 1;

Else if $TE < EBS$

TE = TE + 1;

Else

Neither TC nor TE will be incremented;

When a packet of size B bytes arrives at time t, the following operations are performed:

If the packet has been pre-marked as green and $TC(t) - B \geq 0$

The packet is marked green here and $TC = TC - B$;

Else if the packet has been pre-marked as green or yellow and $TE(t) - B \geq 0$

The packet is marked yellow here and $TE = TE - B$;

Else

The packet is marked red here and neither TC nor TE will be changed;

The Marker reflects the metering results by setting the DS field of the packet to a particular codepoint [6]. In case of the AF service, the color can be coded as the drop precedence of the packet. In the simulations we performed, we used only two drop precedences (IN and OUT), so we slightly changed the TCM algorithm described above to fit our simulations. We used only one token bucket; this means that marking is based on CIR and CBS only: a packet is marked IN if it doesn't exceed the CBS, OUT if it exceeds the CBS.

The rest of the paper is organized as follows. Section 2 presents the network architectures and configurations used for the simulations; Section 3 gives the evaluation and comparison of TSW and TCM marking algorithms. Section 4 presents the new marking algorithm, called Improved-TSW marking algorithm; comparison of the performance of the three algorithms is also presented. Section 5 concludes the paper.

2. NETWORK ARCHITECTURE AND CONFIGURATIONS

2.1 Network Architecture

The network architecture shown in the Figure 1 is used. Traffic is sent from the clients to the servers. Client 1, Client 2, Client 3, and Client 4 are FTP clients while Client 5 is an email client. Server 1, Server 2, Server 3, and Server 4 are FTP servers while Server 5 is an email server. The reason behind this choice is that the FTP traffic is the main part of the total traffic in this simulation and the Email traffic is just the background traffic. Traffic is sent from Client i to the Server i ($1 \leq i \leq 5$). FTP traffic uses TCP as a transport protocol and all the FTP packets are marked with IN (in profile) by

the FTP clients; packets marked with IN have low drop precedence and are protected by the AF service-capable network from other best-effort packets. Email traffic uses UDP as a transport protocol and all the Email packets are marked with OUT (out of profile) by the Email client; packets marked with OUT have high drop precedence and are dropped with high probability when there is congestion in the network. All the FTP clients have the same configurations. Marker (see Figure 1) is the boundary node of this AF service-capable network; marking algorithms are performed at Marker. In our simulations, we used TSW, TCM (see Section 1) and Improved-TSW (see Section 4) algorithms to meter and remark, if necessary, FTP traffic against the profile for this traffic. We assume that all the acknowledgements are sent from servers to clients without losses.

Figure 1. DS Domain Network Architecture for Simulations

2.2 Configurations for Simulations

2.2.1 Basic Simulation Configurations

The followings are parameters that are used in all simulations presented in this paper. Simulations are run for 1000 seconds in OPNET; it is the time needed for the network to be stable. Values for IP packet and TCP segment size are the default values set in OPNET. Application segment size is the maximum segment size the application layer can send to the lower layer. Application segment size should not be too big; otherwise, it will take more time for senders to fragment it and if one fragment in this segment is lost, receivers will not receive the whole application segment. Application segment size should not be too small either; otherwise, the system efficiency will suffer (i.e., unacceptable protocol overhead). We selected a value of 8000 bytes for the application segment size. Application start time is the time the application starts sending packets; we use the OPNET default value that is equal to 100 second. This means the FTP/e-mail traffic is sent out after 100 seconds simulation time.

Simulation Duration: 1000 sec
TCP Segment Size: 536 bytes

IP Packet Size: 576 bytes
Application Segment Size: 8000 bytes

Application Start Time: 100 sec

2.2.2 FTP Client Attributes

Every FTP client generates traffic at a rate equal to 30Kbps at the application layer. This is the rate the ISP (Internet Service Provider) will try to satisfy for every client. Let us note that the actual traffic (i.e., traffic in the network) will be a little bit above 30Kbps when we take into account TCP and IP headers. The average file size should not be too small; otherwise, there will be many small FTP acknowledgements and requests in the network.

File Transfer Rate: 900 files/hour (modeled using a Poisson distribution)

Average File Size: 15000 bytes (modeled using a normal distribution)

2.2.3 Email Client Attributes

The Email client generates traffic at a rate equal to 80Kbps at the application layer. Let us note that the actual traffic (i.e., traffic in the network) will be a little bit above 80Kbps when we take into account UDP and IP headers. Email traffic will not have a profile at the boundary node of our DS domain. The total amount of TCP and UDP traffic will be larger than the bandwidth of the bottleneck link of the network. The objective is to evaluate the traffic performance of all these types of traffic under congestion.

Send Rate: 1800 messages/hour (modeled using a Poisson distribution)

Average Email Size: 20000 bytes (modeled using a normal distribution)

2.2.4 Links' Characterisation

For sake of simplicity, we assume that all links except the bottleneck link have enough bandwidth to accommodate all traffic generated by client or servers. We set the bandwidth of the bottleneck link (i.e. link between Dropper and Dropper 2 in Figure 1) to 150Kbps; the total traffic will be above 200Kbps. This setting will allow us to analyze the traffic performance of AF service-capable network when the congestion is severe.

Data Rate of all links except the bottleneck link: DS3 (44.736Mbps)

Data Rate of the bottleneck link (the link from Dropper to Dropper 2): 150000bps

2.2.5 Time Sliding Window Algorithm

The target rate is the fraction of FTP traffic the ISP is willing to protect. Actually, FTP clients, altogether, will generate traffic of a rate equal to 120 Kbps at the application layer. TSW algorithm will use this target rate to meter and remark the FTP traffic. When the target rate is exceeded, incoming packets will be remarked as OUT packets.

Target Rate: 120000bps

2.2.6 Three Color Marker Algorithm

The CIR is the token generation rate for the token bucket. The bucket size and the token generation rate for this token bucket are computed based on

the allowed burst size and the target rate of this traffic that are set in the profile for this traffic.

Bucket Size: 64000 bits

CIR: 120000bps

2.2.7 Improved Time Sliding Window Algorithm

Improved Time Sliding Window Algorithm is a new marking algorithm that combines TSW algorithm and a token bucket; a detailed description of this algorithm is presented in Section 4. The target rate is set for TSW algorithm and the CIR and Bucket Size are set for the token bucket.

Target Rate: 120000bps

Bucket Size: 64000 bits

CIR: 120000bps

2.2.8 RIO (RED [7] with IN and OUT) Algorithm

RIO stands for RED algorithm with IN/OUT bit. RIO uses twin RED algorithms for dropping packets, one for IN packets and one for OUT packets. In RIO [2], packets with different drop priorities will have different thresholds and different probabilities to be dropped. The settings of the parameters of RIO algorithm are aimed to produce simulations where OUT packets have higher dropping probability than IN packets.

Drop Probability for IN Packets ($P_{\max_in}$): 0.03

Drop Probability for OUT Packets ($P_{\max_out}$): 0.08

Maximum Threshold for IN Packets ($\max_in$): 40 packets

Maximum Threshold for OUT Packets ($\max_out$): 30 packets

Minimum Threshold for IN Packets ($\min_in$): 20 packets

Minimum Threshold for OUT Packets ($\min_out$): 10 packets

Queue Weight: 0.002

Queue Length: 60 packets

3. TCM VS. TSW

In Figure 2, the curve named best-effort quantifies the amount of FTP traffic successfully received by the receivers in current best-effort network. The curves named tcm, tsw and improved-tsw quantifies the amount of FTP traffic successfully received by the receivers in AF service-capable network using TCM algorithm, TSW algorithm and improved-TSW (at Marker) respectively.

Figure 2 shows that the amount of FTP traffic, successfully received by the receivers, has increased using AF service-capable network instead of best-effort network (Y-axis represents the amount of FTP traffic that has been successfully transferred; the unit of Y-axis is 10000 * bytes/sec). Based on these observations, we can state that the TCP traffic has been protected in AF service-capable network from the non-adaptive UDP traffic. Next, an evaluation of the performance of different marking algorithms is presented.

In the current simulations, the same packet rate is configured for both marking algorithms, i.e., Three Color Marker (TCM) algorithm and Time Sliding Window (TSW) algorithm. In TCM, CIR (Committed Information Rate) is set to 120 Kbps; in TSW, Target Rate is set to 120Kbps as well. Thus, we are able to analysis the performance of these two algorithms under the same conditions, e.g.,

if a customer signs up for 120 Kbps with an ISP, we aim to identify the marking algorithm that could provide better service to this customer.

Figure 2. FTP traffic received

Figure 3 shows the number of IN packets remarked to OUT packets at the boundary node Marker using these two algorithms (the unit of Y-axis is packets/sec). In these simulations, we generated very bursty traffic. Although the TCM algorithm allows bursts in the traffic, it tolerates only small bursts. If the traffic burst is very high, i.e., the peak rate of the traffic is much higher than the subscribed rate, the bucket of TCM algorithm will become full very quickly. Then the upcoming IN packets will be remarked as out of profile packets. Figure 3 shows that when using TCM algorithm IN packets change to OUT packets continuously. The reason behind this is that the token generation rate of the bucket in TCM algorithm is only 120Kbps, but the peak rate, i.e., burst, of the traffic is much higher than 120Kbps. Furthermore, the generated traffic includes a large number of bursts making the bucket “always” full when packets arrive. A solution is to increase the bucket size in TCM to allow bigger bursts in the traffic. This is based on the computation of the burst size: $\text{Burst Size} = \text{CIR} * T + \text{Bucket Size}$, where T is the time for the burst. Nevertheless, since the CIR is generally much lower than the peak rate (burst) of the traffic, the recovery rate of the bucket will be still much lower than the peak rate. Then, after the bucket reaches its full, the upcoming packets will be remarked continuously although the bucket size is bigger than before. Figure 4 shows the mean value of the number of packets that change from IN to OUT at boundary node Marker with bucket sizes of 64K and 160K respectively. After increasing the bucket size from 64K to 160K, fewer packets are remarked to OUT at the Marker. However,

the rate of IN to OUT is still high; Rate of IN to OUT = total packets remarked from IN to OUT / total IN packets. Figure 5 illustrates that the traffic performance with the bucket size of 160K is slightly better than the traffic performance with bucket size of 64K. However, increasing the bucket size does not improve the performance as much as increasing CIR (see Figure 6 in which CIR is changed from 120Kbps to 160Kbps). Furthermore, increasing the bucket size requires more system resource (memory) of the router.

Figure 3. IN packets change to OUT packets at Marker by using TSW and TCM algorithm

Actually, there are two techniques that can be used to improve the traffic performance when using TCM algorithm.

With the first technique we should let the packets pass a shaper before entering the TCM marker. By using a shaper, both the burst size and the peak rate of the traffic are under control. Another one or two token buckets (one token bucket is used in TCM algorithm) combined with this shaper will allow to control the traffic's burst size and peak rate. After passing the shaper, the traffic will not have big bursts; thus, fewer packets will be remarked to OUT when using TCM algorithm. At the mean time, we can let traffic with low burst enter the AF service-capable network to release the burden of interior routers (e.g. less buffer size may be used in interior routers). This will help the traffic behaviour into the AF service-capable network to be more predictable. For example, in our current simulations, even if we mark a burst of IN packets into our AF service-capable network, the IN packet will likely be dropped in the interior nodes because of resource, i.e., buffers, shortage at the Dropper (queues are full). However, if we remove the big bursts from the traffic using a shaper, this problem will disappear. A shaper will be added in front of the marker, to improve the

traffic performance of the AF service-capable network, in our future simulations.

With the second technique we should use two token buckets (instead of one) at the boundary nodes to meter and mark traffics. One bucket is used to control the average rate of the coming traffic; there is only one token in this token bucket. The CIR of this token bucket is the average rate of the traffic. When a packet arrives and there is a token in this token bucket, this packet enters the AF service-capable network without passing the second token bucket; otherwise the packet enters the second token bucket. The second token bucket is used to control the bursts of the coming traffic; the bucket size of the second token bucket controls the burst size of the traffic and its CIR controls the frequency of the bursts in that traffic. If a packet enters the second token bucket and there are no enough tokens in this bucket, the packet will be remarked as OUT packet. By using two token buckets to meter and mark traffics at the boundary nodes, we should get better traffic performance.

Figure 3 shows that much fewer IN packets have been remarked to OUT packets by using TSW algorithm than TCM algorithm; in fact, in TSW algorithm only the packets with the average rate above $1.33 \cdot R_T$ ($1.33 \cdot 120\text{Kbps}$) are remarked as OUT packets. The bursts of IN packets whose rate is below 160Kbps pass the Marker without remarking. Figure 3 also shows that all the IN packets that have been changed to OUT packets by using the TSW algorithm are those packets whose average rate is above 160Kbps. This means that only big bursts are remarked to OUT packets in the TSW algorithm.

Figure 4. Packets remarked from IN to OUT with different bucket size

Figure 5. FTP traffic received with different bucket size

Figure 6. How CIR in TCM affect traffic performance

Figure 2 shows that TSW algorithm performs better than TCM algorithm. This means that more FTP traffic has been successfully received by the receivers when using TSW algorithm. This is rooted in the fact that less IN

packets have been remarked at the boundary node Marker in TSW than in TCM algorithm. Therefore, more IN packets (FTP packets) have been allowed into our AF service-capable network and thus protected from OUT packets (Email packets). In fact the simulations show that when the traffic's burst is very high and the congestion is very severe in the bottleneck link, using TSW algorithm results in better traffic performance than using TCM algorithm. However, it is not always true that TSW algorithm performs better. For example, when metering a CBR (Constant Bit Rate [8]) source, a profile meter using TSW algorithm would allow the source to transmit at a sustained rate higher than the contracted one. TCM algorithm does not permit this [9].

4. IMPROVED-TSW ALGORITHM

In order to get better traffic performance than TSW and TCM, we designed a new marking algorithm by combining TSW algorithm and a token bucket. Figure 3 shows that when using TSW algorithm, the packets remarked from IN packets to OUT packets are those ones whose rate is above $1.33 \cdot R_T$. A way to improve the performance of this algorithm is to reduce the number of packets remarked from IN packets to OUT packets; token bucket can be used to realize this. If an IN packet's rate is above $1.33 \cdot R_T$, we first examine whether there are enough tokens in this token bucket. If there are tokens left in the bucket, the packet will be not remarked; otherwise, the packet will be remarked as OUT packet. Thus, more IN packets will be allowed to enter the AF service-capable network. This new algorithm is a combination of TSW algorithm and a token bucket algorithm: we call it Improved Time Sliding Window algorithm (Improved-TSW).

Improved Time Sliding Window Algorithm

Initially:

Win_length = constant;
 R_T = Target Rate;
 Avg_rate = R_T , Connection's Target Rate;
 $T_front = 0$;
 CBS = Bucket Size;
 TC = CBS;
 Update time = $1 / CIR$;

For each Update time:

 If TC is less than CBS, then
 TC is incremented by one;

 Endif

For each packet with size pkt_size arrival at time arrive_time:

 If it is an IN packet

 Bytes_in_TSW = Avg_rate * Win_length;

 New_bytes = Bytes_in_TSW + pkt_size;

 Avg_rate = New_bytes / (arrive_time - T_front +

Win_length);

T_front = arrive_time;

 If Avg_rate > $1.33 \cdot R_T$ and TC > pkt_size

 TC = TC - pkt_size;

 Else

```

If Avg_rate > 1.33 * RT and TC <= pkt_size
    Remark this packet as OUT packet;
Endif

```

```

Endif

```

```

Endif

```

TC is the current token number in the token bucket; CBS is the token bucket's size; T_{front} is the time of the last packet arrival; Avg_rate is the rate estimated upon each packet arrival.

The first part of Improved-TSW is the same as the TSW algorithm. The second part realizes the token bucket algorithm; when the average rate of the IN packet is above $1.33 * R_T$, the algorithm checks whether there are enough tokens in the token bucket. If the response is yes the packet is kept as IN packet. Otherwise, the packet is remarked as OUT packet. Next we present an evaluation of the performance of the Improved-TSW and compare it with TSW and TCM.

Figure 7 shows that the average number of packets that are remarked from IN packet to OUT packet (at the boundary node Marker) using TSW, TCM and Improved-TSW respectively; the average number of packets changed from IN to OUT by using Improved-TSW algorithm is the lowest. TCM algorithm remarks more packets to OUT packets than TSW and Improved-TSW. Thus, TCM algorithm has the worst traffic performance. More IN packets are allowed to enter the AF service-capable network by using Improved-TSW algorithm; this means that more IN packets are protected by using Improved-TSW algorithm rather than TCM or TSW. Figure 2 shows that with Improved-TSW algorithm we get the best traffic performance, i.e., the amount of FTP traffic received by the receivers is the highest.

Figure 7 Comparison of three algorithms at Marker

5. CONCLUSIONS

In this paper, we compared two marking algorithms (TSW and TCM) that are currently used at the boundary nodes of Assured Forwarding networks. We proposed a new marking algorithm named Improved TSW marking algorithm that is designed by combining the TSW algorithm with a token bucket. We also proposed some other mechanisms to improve the performance of TCM algorithm.

This paper shows that selecting an appropriate marking algorithm is very important to AF service-capable networks. Our simulations show that

1. If we use TCM algorithm, a large amount of IN packets will be remarked to OUT at the boundary nodes of an AF service capable network;
2. If we use TSW algorithm or Improved-TSW algorithm traffic performance improves significantly; and
3. If we use Improved-TSW algorithm, we get better traffic performance (than using TSW) in our AF service-capable network with high burst and severe congestion at the bottleneck.

REFERENCES

- [1] D. Black, S. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for Differentiated Services", IETF RFC 2475, December 1998.
- [2] D. Clark and W. Fang, "Explicit Allocation of Best Effort Delivery Service", IEEE/ACM Trans. on Networking, vol. 6, no. 4, pp. 362-373, August 1998.
- [3] J. Heinanen, et al., "A Three color Marker", IETF Internet Draft, <draft-heinanen diffserv-tcm-01.txt>, February 1999.
- [4] Juha Heinanen, Fred Baker, Walter Weiss and John Wroclawski, "Assured Forwarding PHB Group", IETF RFC 2597, June 1999.
- [5] OPNET, <http://www.mil3.com>.
- [6] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [7] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, Vol. 1, No. 4, pp397 – 413, August 1993.
- [8] Andrew S. Tanenbaum, "Computer Networks", 3rd Ed, 1996, Prentice Hall, Inc., pp458 – 459.
- [9] J. Ibanez and K. Nichols, "Preliminary Simulation Evaluation of an Assured Service", IETF Internet Draft <draft-ibanez-diffserv-assured-eval-00.txt>, August 1998.