Computer Communications xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A comparison of simulated traffic conditioner performance

Tinus Strauss *, Derrick G. Kourie, Martin S. Olivier

Department of Computer Science, University of Pretoria, Pretoria 0002, South Africa

ARTICLE INFO

Article history:
Received 23 January 2007
Received in revised form 29 July 2008
Accepted 30 July 2008
Available online xxxx

Keywords:
Differentiated services
Traffic conditioners
Token bucket

ABSTRACT

In this paper, the simulated performance of a number of differentiated services traffic conditioners is studied under a range of synthetic traffic conditions. The traffic conditioners are compared using two performance measures which are defined and justified in the paper.

In the initial simulation the traffic conditioner performance differed in one measure but not the other. Further experimentation showed that one could get all the conditioners to have similar performance by setting carefully the configuration parameters of the conditioners.

1. Introduction

Quality of service (QoS) in computer networks can be defined as the ability of a network to provide selected traffic with better service. Differentiated services (Diffserv) [1,2] is a framework for providing different levels of service to various classes of traffic through the relative priorities assigned to the classes.

The Diffserv architecture consists of a number of components, one of which is the so-called traffic conditioner. The traffic conditioner is located on the boundary of a Diffserv-capable network and is responsible for measuring incoming traffic and assigning the constituent packets to different service classes based on predetermined policy rules contained in a traffic conditioning specification (TCS). Five different traffic conditioners are briefly described in Section 2.3. The conditioners are used to determine whether an incoming traffic stream is in- or out-of-profile and to assign different 'colours' to packets. These are based on the TCS which forms part of the service level agreement that has been negotiated between the network subscriber and provider.

Packets are prioritised based on the colours assigned to them. This means that, depending on their colour, some packets will have a greater probability of being discarded in the event of congestion. This prioritisation is achieved by employing different queues for the different classes of traffic and then managing some queues more aggressively than others.

Since there is a number of different traffic conditioners, a network designer is faced with the challenge of selecting one of the available conditioners to employ in a particular network. The designer might base her decision on the expected incoming traffic patterns and the subscription level at which the network will be operated. It is in this area of design that this paper aims to make a contribution.

The remainder of the paper is organised as follows. Section 2 provides descriptions of the different simulation scenarios and also defines and justifies the performance measures. The results of the experiments are discussed in Section 3 and the paper closes with Section 4.

2. Experimental setup

Simulation experiments were performed to determine whether some traffic conditioners are more appropriate in certain traffic pattern and subscription level circumstances. The experiments were performed in *ns-2* version 2.26 [3,4], a discrete event simulation environment that is freely available.

The remainder of this section describes how the simulation experiments were configured.

2.1. Simulation topology

The nodes in the simulation were connected in a simple symmetrical dumb-bell topology. The simulation topology is graphically portrayed in Fig. 1. The topology was chosen to ensure that congestion will occur at, and only at, the bottle-neck link between nodes c_1 and c_2. Section 2.2 revisits this theme when the traffic generation is detailed.

There are five different types of nodes in the simulation. Each source node s_i ($i = 1,2,\ldots,n$) sends data to each destination node d_i. Each source s_i is connected to the Diffserv domain by a direct connection to an ingress edge node ie_i. The destination nodes are similarly connected to an egress edge nodes ee_i. There are two core

* Corresponding author. Tel.: +27124202019.
E-mail address: tstrauss@cs.up.ac.za (T. Strauss).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.07.015

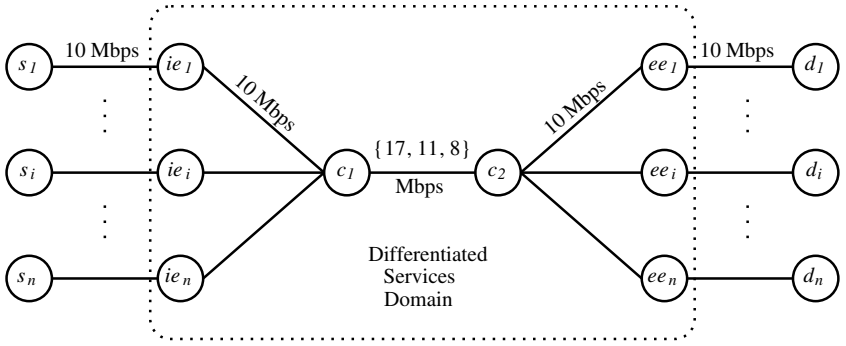


Fig. 1. Simulation topology.

nodes in the simulation – c_1 and c_2. The core nodes are connected via a single bottle-neck link. This link is assigned capacities from the set {8, 11, 17 Mbps} as explained in Section 2.4.

For the purpose of this study, $n = 10$ identical sources with identical TCSs were used in each simulation scenario. The enforcement of the TCS of source s_i is done at the ingress edge node ie_i.

2.2. Source traffic generation

One of the goals of the investigation was to determine whether or not the characteristics of the traffic arriving at a traffic conditioner affects the performance of the conditioner. For this reason traffic generation models were not chosen to represent specific applications or traffic scenarios. Rather they were chosen to represent a range of different traffic patterns. This was done in order to compare the traffic conditioners under various traffic pattern conditions, possibly corresponding to real-world situations. The models presented here range from a rather simple constant bit rate (CBR) generator to a fairly complex model used to generate self-similar, or long-range dependant traffic.

All sources generate packets of size 576 bytes using UDP as transport layer protocol. The reason why UDP was deemed appropriate is because the source traffic that was generated is not necessarily a simulated application running on a single node. The source application simulates a traffic stream. This stream could consist of multiple application 'flows'. The generated traffic should therefore not be transformed by transport layer protocol mechanisms but only by the Diffserv mechanisms.

Each source generates traffic at an approximate average rate of 2 Mbps. There are thus ten 2 Mbps traffic streams, each transmitted over a 10 Mbps line, which then converge on a channel of, at most 17 Mbps. The bottle-neck link will therefore be heavily congested while the 10 Mbps links will not be congested. Packet discard is hence only expected at the bottle-neck link.

The following list contains a brief description of each of the traffic generation models used in the simulation study.

CBR. The source generates traffic at an approximately constant rate of 2 Mbps. The sources were configured to introduce some small random variation between packets. This model is one of the traffic generation applications available within *ns*.

POISSON. The Poisson arrival processes used in the simulations were created using the following technique. An inter-arrival process was created in which the times between packets are drawn from an Exponential distribution with mean 2304 μs. As a result, the number of packets arriving per unit of time is described by a Poisson process with an average packet arrival rate of 434 pps. (This rate is the inverse of the mean of the Exponential distribution, with the appropriate conversion from micro-seconds to seconds.) Since all packets are 576 bytes in size, the average sending rate is $434 \times 576 \times 8 = 2$ Mbps.

The inter-arrival process was implemented as an external program and used to generate a set of data pairs as input data for the traffic trace application in *ns*.

EXPOO. This exponential on/off (EXPOO) model is one of the source generation models built into *ns*. Traffic is generated during two periods – ON and OFF. During the ON period traffic is generated at a constant rate and during the OFF period no traffic is generated. Two exponential distributions dictate the lengths of the alternating ON and OFF periods. The average length of both ON and OFF periods was set to 500 ms. The sending rate during ON periods was 4 Mbps. As a result, this model is designed to reflect a bursty traffic situation in which, for approximately half of the time in a given time period, about twice as many packets are emitted as in the CBR model, while no packets at all are emitted during the remaining time. This, again, results in an expected average rate of 2 Mbps.

PAROO. This Pareto on/off (PAROO) model is another of the source traffic generation models built into *ns*. It functions in the same manner as the EXPOO model except that it utilises two Pareto distributions instead of Exponential distributions. It was configured along similar lines to the EXPOO model. The average length of both ON and OFF periods was set at 500 ms, and the sending rate during ON periods was set to 4 Mbps. The shape parameter of the Pareto distribution was left at the system default of 1.5, which is in the middle of the acceptable range of between one and two.

H1, H2, H3. Three self-similar traffic sources (H1, H2, and H3) were used in the simulation study. The source traffic processes were created using an implementation of [5]. The three processes differ only in the value of the Hurst parameter (H). The values of the Hurst parameter creates traffic streams with varying degrees of long range dependance. As the value of H increases the degree of long-range dependence also increases. H1 corresponds to a sample path with $H = 0.5$, H2 to $H = 0.7$ and H3 to $H = 0.95$.

The program that was used to create the sample paths, produces an arrival process, i.e. it gives the number of arrivals in a time interval. *ns*, however, requires an inter-arrival process as input for the traffic trace application. For this reason the arrival process was translated into a inter-arrival process according to the following scheme. Each entry x_i in the arrival process represents the number of packets arriving in the time-interval t_i of duration 0.1 s. The number of arrivals per interval t_i was then converted to inter-arrival times by distributing all x_i arrivals evenly across the interval t_i and specifying the size of each arrival to be 576 bytes.

Since the average rate should be close to 2 Mbps, the average number of arrivals in t_i is 44 packets of 576 bytes each. For this reason, the sample paths were created with mean 44, variance 144, and Hurst parameter one of $\{0.5, 0.7, 0.95\}$.

Let $S = \{\text{CBR}, \text{EXPOO}, \text{PAROO}, \text{POISSON}, \text{H1}, \text{H2}, \text{H3}\}$ now denote the set of source traffic generation models used in the simulations.

Please cite this article in press as: T. Strauss et al., A comparison of simulated traffic conditioner performance, Comput. Commun. (2008), doi:10.1016/j.comcom.2008.07.015

2.3. Differentiated services domain

As mentioned earlier, policies that specify the network resources to which each class of subscriber is entitled are defined in a TCS. These policies should then be monitored to determine to what extent the subscriber adheres to its side of the service-level agreement.

The TCS is policed at the ingress boundary nodes using traffic conditioners. Traffic conditioners are an important component in the Diffserv architecture and provide functions such as metering and marking of packets based on the TCS. Marking is done by assigning a value to the differentiated services code-point (DSCP) field in the IP packet header. The policing process will be inclined to assign low precedence levels to packets from subscribers whose traffic generation activity does not conform to the agreed-upon policy.

Each of the different traffic conditioners employed in the simulation study is briefly described in the following paragraphs.

TBM. The token bucket marker (TBM) meters a traffic stream based on two parameters: CIR (committed information rate) and CBS (committed burst size). The meter is specified in terms of a token bucket [6] with size CBS and token generation rate CIR. Tokens are generated at a rate equal to the CIR and are added to the token bucket provided it is not full. When a packet arrives and there are enough tokens in the bucket, the packet is deemed to be in profile and the relevant number of tokens are removed from the bucket. If there are not enough tokens in the bucket the packet is out of profile. The conditioner marks a packet 'green' when it is in profile and 'red' when it is out of profile.

TSW3CM. The time-sliding-window three colour marker (TSW3CM) is described in [7]. It meters a traffic stream based on two parameters: CTR (committed target rate) and PTR (peak target rate). It marks packets 'green', 'yellow', or 'red'. It is composed of two parts: a rate estimator and a marker. The rate estimator estimates the current rate of the traffic stream whilst taking past estimates into consideration and then the marker assigns one of the three colours to the packet, based on the current estimate.

TSW2CM. The time-sliding-window two colour marker (TSW2CM) is a simplified version of the TSW3CM since it is configured with only one rate, CTR, instead of two. Consequently it assigns only one of two colours – green or red.

srTCM. The single rate three colour marker (srTCM) is presented in [8] and measures a traffic stream according to three parameters: CIR, CBS, and EBS (excess burst size). The latter two parameters are two token-bucket capacities and the former is a token generation rate. The conditioner contains two token buckets – one of size CBS, the other of size EBS. Tokens are generated at a rate equal to the CIR. When the first bucket is filled, tokens are added to the second bucket. An arriving packet is 'green' when there are enough tokens in the first bucket, 'yellow' when there are enough in the second bucket but not enough in the first bucket, and 'red' when there are not enough in either token bucket.

trTCM. The two rate three colour marker (trTCM) [9] utilises four conditioning parameters: PIR (peak information rate), PBS (peak burst size), CIR, and CBS. These correspond to two token buckets employed to do the metering. The conditioner assigns one of three colours (green, yellow, or red) based on the states of the two internal token buckets. This trTCM differs from the srTCM in that trTCM's token buckets generate tokens independantly.

The preceding traffic conditioners can be grouped into two classes: (1) the conditioners based on token-buckets and (2) the time-sliding window (TSW) based conditioners. TBM, srTCM, and trTCM are in the former class and TSW2CM and TSW3CM are in the latter class. Let $P = \{\text{TBM}, \text{TSW3CM}, \text{TSW2CM}, \text{srTCM}, \text{trTCM}\}$ be the set containing the different traffic conditioners used in the experiments. Table 1 contains the values of the configuration parameters for each of the elements of P. All of these traffic conditioners are available as Diffserv conditioners in *ns*.

Table 1
Traffic conditioner parameter values

Parameter	TBM	TSW3CM	TSW2CM	srTCM	trTCM
CIR/CTR (Mbps)	1	1	1	1	1
CBS (B)	5760	–	–	5760	5760
PIR/PTR (Mbps)	–	1.5	–	–	1.5
EBS/PBS (B)	–	–	–	2880	2880

All the traffic conditioners are configured with a committed rate (CIR) of 1 Mbps. This rate is the average rate that the provider commits to provide to the subscriber. This committed rate will also be referred to as a subscriber's 'reserved' rate rr_i.

The traffic conditioners based on a token bucket scheme include the notion of a CBS. The burst size was set to an arbitrary 5760 bytes which translates into 10 packets.

The peak rate that is allowed (PIR) is set at 1.5 Mbps and the excess and peak burst sizes (EBS and PBS) are set to 2880 bytes, which is the CBS halved.

Notice that the source applications' sending rates all exceed the rates specified in the policies. One would therefore expect to observe dropped packets. This study intentionally simulates a situation of abuse of network resources in order to monitor the performance of various conditioners in such a case.

The initial DSCP used for packets entering the simulated Diffserv domain is '10'. The DSCP of packets that are determined to be 'green' at the ingress nodes (ie_i) is also set to '10'. The DSCP of packets determined to be 'yellow' is set to '11' and 'red' packets are marked '12'. The core nodes (c_1 and c_2) only use the DSCP to determine into which queue to place a packet; they do not manipulate the DSCP.

Each of the nodes in the simulation has one physical queue of finite capacity and three virtual queues within the physical queue. There is one virtual queue for each DSCP (10, 11, and 12) in the Diffserv domain. The maximum physical queue size is 600 packets. There can therefore only be 600 packets in a queue. In general, if any more packets were to arrive, they would be discarded irrespective of their DSCPs. However, the queue size was chosen as such to be large enough to prevent the queue from ever overflowing in the scenarios envisaged in this study.

Each of the virtual queues are managed through random early discard (RED) [10,11]. RED is a form of active queue management where randomly selected packets are discarded according to some rules. If the current average queue size of a RED queue is less than a minimum threshold min_{th}, the packet is not discarded and if it is greater than a maximum threshold max_{th}, the packet is discarded. When the average queue size is between min_{th} and max_{th}, however, the probability of discard grows as the queue size increases, up to a maximum configured probability equal to max_{p}.

The RED queues at the edge nodes and the core nodes are configured identically and the parameters used for the virtual queues are listed in Table 2.

Packets with DSCP 10 have less aggressive RED parameters than those with different DSCPs. The so-called queue weight parameter required by the RED algorithm was configured in *ns* to be −1 which

Table 2
DSCPs and the corresponding RED parameters

Colour	DSCP	min_{th}	max_{th}	max_{p}
Green	10	60	120	0.02
Yellow	11	40	60	0.5
Red	12	25	50	0.5

Please cite this article in press as: T. Strauss et al., A comparison of simulated traffic conditioner performance, Comput. Commun. (2008), doi:10.1016/j.comcom.2008.07.015

means that the parameter is automatically configured. The minimum and maximum thresholds for DSCP 10 packets are set to 60 and 120 packets, respectively, and max_p is set to a fairly low 0.02. The other virtual queues are managed more aggressively with both max_p's set to 0.5. The management also starts earlier with the min_{th} for DSCP 11 set to 40 packets and max_{th} to 60 packets. The management starts even earlier for DSCP 12 packets; min_{th} is set to 25 and max_{th} to 50 packets.

From the RED configurations above, it is clear that packets which are in-profile (DSCP 10) are policed less aggressively than out-of-profile packets (DSCP 11 and 12), with DSCP 12 packets given the lowest priority.

2.4. Simulation scenarios

Recall from Section 2.1 that three different bottle-neck link capacities (8, 11, and 17 Mbps) are used in the simulations. Since there is a reserved rate rr_i of 1 Mbps associated with each of the $n = 10$ subscribers, one can obtain a so-called reservation level for the network by calculating how much of the bottle-neck link capacity is reserved for, or committed to, subscribers. The reservation level is therefore $\sum_i rr_i$ divided by the link capacity (C). The approximate reservation level for a network with a 8 Mbps link is then 125%, for an 11 Mbps link it is 90%, and for a 17 Mbps link it is 60%. These levels, respectively, correspond to an oversubscribed, nearly fully subscribed, and an undersubscribed network. In fact, the values 8, 11, and 17 were specifically chosen to reflect these three scenarios. Let R={125%, 90%, 60%} be the set of reservation levels used in the simulation study.

Recalling that S = {CBR, EXPOO, PAROO, POISSON, H1, H2, H3} and P = {TBM, TSW3CM, TSW2CM, srTCM, trTCM}, the different scenarios that were simulated can be represented by the Cartesian product of the sets mentioned before: $SIMS = R \times S \times P$. Thus, for example, one simulation scenario assumed a 125% reservation level on the bottle-neck link, where CBR traffic was generated (at 2 Mbps) at each of the n source nodes, this traffic being policied at the ingress nodes according to, say, the TBM policing policy; another corresponded to the foregoing, but assumed a 90% reservation level, etc. This means that a total of $3 \times 5 \times 7 = 105$ scenarios were simulated.

Every simulation scenario element in *SIMS* was run 10 times. The simulations were run for $t = 60$ s (simulation time).

After each simulation run the following data were collected: the number of packets generated by each source ($N_{in,i}$) and the number of packets per DSCP that arrived at each destination ($N_{10,i}, N_{11,i}, N_{12,i}$).

2.5. Performance measures

Two performance measures were employed in the present study. The one measure relates the extent to which the subscriber achieved its so-called target rate. The other measure determines how well the number of DSCP 10 packets corresponded to the reserved rate. The ith subscriber's target rate is its reserved rate plus its share of the excess capacity: $tr_i = rr_i + \frac{1}{n}(C - \sum_{j=1}^{n} rr_j)$.

The first measure's relevance is fairly obvious if one assumes that the provider network should provide the subscriber with a fair share of excess capacity. To determine how well the target rate is achieved, one divides the actual average achieved rate ar_i by the target rate:

$$\text{TargetRatio}_i = \frac{ar_i}{tr_i}.$$

The achieved rate is determined by counting all the packets that arrived at the destination and translating this count into an average rate over the simulation. Let $\rho(c,t)$ be a function that translates a packet count c over the whole simulation duration t into an average bit rate. Then, $ar_i = \rho(N_{10,i} + N_{11,i} + N_{12,i}, t)$.

The closer to one that the observed TargetRatio_i is, the better the traffic conditioner is deemed to have performed. If the ratio is larger than one, then the subscriber achieved an average rate greater than it is allowed. If, under stress conditions, one subscriber obtains a better rate, then another will typically achieve a lower than one ratio, since there is a fixed sized bottleneck link in the network. For some subscribers to win, some have to lose.

Although the observed target rate performance of the traffic conditioners is important, there might be other differences that matter to both the subscriber and the provider. For example, in an extreme case, a subscriber might achieve her target rate even though almost none of the packets are marked as in-profile. This could potentially have negative cost implications for the subscriber, depending, of course, on exactly how the agreement was specified. The subscriber might have to pay more for the out-of-profile packets, since she might be penalised for exceeding the agreed reserved rate.

From the provider perspective, the same issues are relevant, although the subscriber and provider have opposing commercial interests. When a subscriber achieves her target rate by using more than her reserved rate, then the provider would typically prefer to charge the subscriber differently for delivering the packets that exceeded the reserved rate.

For the above-mentioned reasons a second performance measure was used to compare the traffic conditioners.

Recall that the subscriber and provider enter into an agreement regarding the service to be expected from the provider. The provider commits to deliver packets from the subscriber while the offered traffic rate is less than, or equal to, the agreed upon CIR (also referred to as reserved rate). The subscriber is allowed, however, to offer more traffic to the network on the understanding that the traffic exceeding the CIR will receive treatment different from the traffic not exceeding the CIR. For this reason, it is important that the correct number of packets are marked with DSCP 10, the correct number being a number that corresponds to the reserved rate. To determine how well the reserved rate was achieved with 'green' (DSCP = 10) packets, the following ratio was calculated:

$$\text{GreenRatio}_i = \frac{\rho(N_{10,i})}{rr_i}.$$

Ideally, if a subscriber's average sending rate is less than, or equal to, the reserved rate, then all of the packets originating from said subscriber ought to arrive as green at the destination. However, there will inevitably be short term variations in sending rate, which are determined by the way in which the source generates traffic. These variations will affect the way in which a traffic conditioner determines whether a particular packet should be marked as in-profile or not. As a consequence, even if the average sending rate taken over the full simulation falls below the reserved rates, these short term variations may nevertheless cause a proportion of the packets to be marked as out-of-profile, so that the ratio of green packet rate transmission to reserved rate is less than it might have been.

Since the simulated sources in the present study generate traffic exceeding their reserved rates, the ratio should not be less than one. Of course, if a source generates traffic at an average rate less than the reserved rate, the ratio will be less than one. The ratio should also not be greater than one since that would mean that the subscriber obtained 'better' service than specified in the service level specification.

These two performance measures are used in the remainder of the article to determine whether some traffic conditioners are better than others in the scenarios considered. The closer the observed value of the ratio is to one, the better the traffic conditioner performed.

Please cite this article in press as: T. Strauss et al., A comparison of simulated traffic conditioner performance, Comput. Commun. (2008), doi:10.1016/j.comcom.2008.07.015

3. Observations

An earlier work [12] details an analysis of the differences in the performance of the conditioners based on the TargetRatio metric. The analysis employed the Kruskal-Wallis non-parametric test [13] as implemented in R [14], a system for statistical computation and graphics.

The results in [12] suggest that for the very specific scenarios under study, the experiments did not yield significantly different target rate results for the different traffic conditioners. This constitutes *prima-facie* evidence that the choice of traffic conditioner does not have a significant impact on the extent to which the target rate is achieved. Of course, to gain greater confidence in this assertion, a more comprehensive study would have to be undertaken that considers a range of scenarios not covered by the present simulation.

However, the GreenRatio analysis in [12] points to a different conclusion. The observations indicated that the two time-sliding window traffic conditioners yielded higher GreenRatio values than did the three token bucket conditioners. This should not be construed to mean that the time-sliding window conditioners are always a better choice. Indeed, it was seen that they are sometimes too lenient, i.e. they result in green ratios that are greater than one. This is potentially to the detriment of the service provider, since it implies a potential loss of revenue in that the client is getting more of a service than was agreed upon per contract.

The reason for the lenience in the time-sliding window conditioners might be due to the smoothing features to estimate the current traffic rate that are inherent in the TSW algorithm. In the case of a token bucket conditioner, one would expect that the bucket's size determines the extent to which the short-term fluctuations in the traffic stream can be absorbed: a larger token bucket allows for larger short-term fluctuations in the sending rate before packets are marked as out-of-profile.

In the present work, the traffic conditioner performance is investigated further. Two scenario were selected for analysis based on $|1\text{-GreenRatio}_i|$ performance: one in which the time-sliding window conditioners performed better; and another in which the token bucket conditioners performed better.

It was decided to select the 60%, PAROO and 60%, $H = 0.95$ scenarios as representative scenarios for further analysis since the time-sliding window conditioners performed better in the PAROO case while the token bucket conditioners were better in the $H = 0.95$ case. Further did the time-sliding window conditioners achieve higher average GreenRatios in both these cases.

3.1. Traffic pattern

Fig. 2(a) shows a plot of the traffic pattern generated by one single PAROO source superimposed on the traffic pattern generated by one single $H = 0.95$ source. The graph shows the average number of bytes sent in 0.2 s intervals. From the plot it is clear that the PAROO traffic stream contains more variation than the $H = 0.95$ traffic stream. The fact that the time-sliding window conditioners performed better in the PAROO case, suggests that when the traffic pattern is highly variable, a conditioner based on a time sliding window is more effective than a token bucket conditioner.

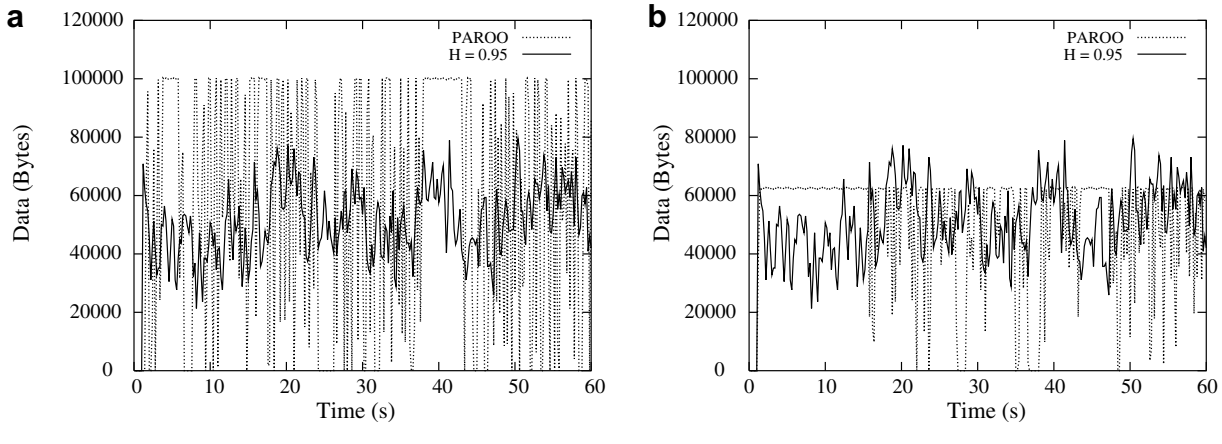
In order to gather additional evidence as to whether the traffic pattern plays a role in the relative performance of the conditioners, an additional experiment was carried out. The 60%, PAROO simulations were repeated with slightly different parameter values for the PAROO traffic generator. Specifically, the parameters were chosen to reduce the short-term variation in the PAROO traffic streams. The PAROO traffic model was configured with a 800 ms average ON period and a 200 ms OFF period instead of 500 ms for both. The constant sending rate during the ON period was set to 2.5 Mbps as opposed to the original 2 Mbps. Fig. 2(b) shows the resulting modified PAROO traffic stream with the original $H = 0.95$ traffic stream superimposed on it. It is clear from the plot that the modified PAROO traffic stream contains less variation than the original traffic stream.

The performance of the traffic conditioners in the modified case was compared to the original performance. The time-sliding window conditioners still performed better than the token bucket ones. The difference in performance was, however, smaller. This indicates not only that the traffic pattern does indeed affect the performance of the various traffic conditioners, but provides additional evidence to suggest that time sliding window technology is better than token bucket technology in conditions of high variability.

3.2. Token bucket size

The foregoing conclusions were predicated on token bucket experiments that had a bucket size of $5760 = 10 \times 576$ bytes. It was decided to determine whether the token bucket traffic conditioner performance would improve if the bucket size was modified. Consequently, the following simulation scenarios were repeated using a range of token bucket sizes.

The experiments were limited to 60% reservation level with PAROO (as described in Section 2.2 and not the modified version



(a) Original (b) Modified

Fig. 2. PAROO and $H = 0.95$ traffic patterns at a single source.

Please cite this article in press as: T. Strauss et al., A comparison of simulated traffic conditioner performance, Comput. Commun. (2008), doi:10.1016/j.comcom.2008.07.015

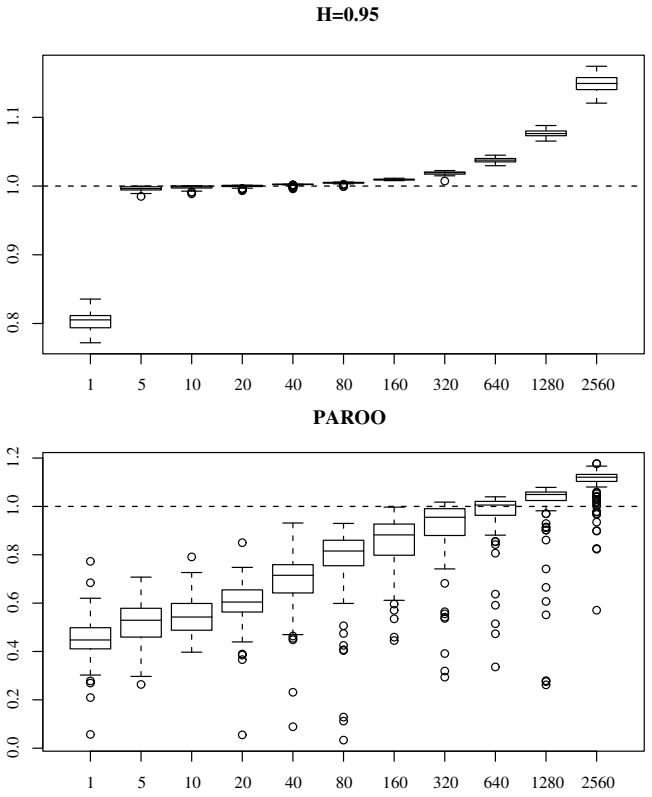


Fig. 3. GreenRatio values for different bucket sizes.

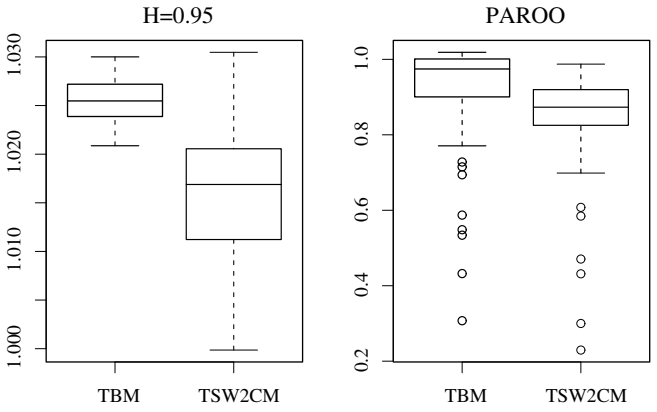


Fig. 4. Green ratios for TBM and TSW2CM.

above) and $H = 0.95$ traffic using the TBM traffic conditioner. The bucket size started at 2880 bytes (5 packets) and was doubled up to 1,474,560 bytes (2560 packets). The case of a bucket size equal to the size of one packet, 576 bytes, was also simulated. Each simulation scenario was again repeated 10 times. Fig. 3 contains graphical representations of the observed green ratios. Note that the *x*-axis is in units of packet size, where 1 packet is 576 bytes.

From Fig. 3, it can clearly be seen that the bucket size does, as expected, influence the green ratio performance of the token bucket traffic conditioner. However, it affects the performance in the two traffic patterns differently. From the plots it can be seen that a bucket size of approximately 640×576 bytes is required in the PAROO case to achieve a green ratio of one, while this ratio is already attained in the $H = 0.95$ model when the bucket size is very small – approximately 5×576 bytes.

In fact, in the $H = 0.95$ traffic case the performance is not very sensitive to changes in the bucket size for sizes less than 320×576 bytes.

It can also be seen that the token bucket based conditioners can be as lenient as the time-sliding window based conditioners. This is demonstrated in the simulation scenarios described below.

Fig. 4 contains box plots of the green ratios observed for both PAROO and $H = 95\%$ traffic patterns, when conditioned by both a TBM conditioner with a 250,000 byte bucket (i.e. approximately 434 packets) and by a TSW2CM conditioner. The rationale for choosing this bucket size is that it corresponds to the number of bytes that is generated in one second, assuming a traffic generation rate of 2 Mbps – the average rate at which sources generated traffic in the simulations. One second is the length of the smoothing window in the time sliding window algorithm used in the TSW2CM.

From the plot it can be seen that the TBM achieved, on average, greater green ratios than the TSW2CM. This is true for both the PAROO and $H = 95\%$ traffic patterns. It is therefore possible to obtain level of leniency that are similar to time sliding window conditioners when using a token bucket based conditioner.

3.3. Packet size

Since it was illustrated above that both the traffic model and the token bucket size influence the observed GreenRatio performance of the traffic conditioners, it seems relevant to investigate further the effect of the traffic on the conditioners' performance. In this section the impact of the size of the packets in the traffic stream is investigated. Two traffic models – CBR and PAROO – were selected for this purpose. These two models were chosen because they represent two opposite ends of the spectrum in terms of traffic burstiness. CBR traffic is not bursty at all while PAROO traffic is very bursty (see Fig. 2). It was decided to compare a small packet size of 64 bytes, the original size of 576 bytes, a medium size of 1536 bytes, and a large packet size of 4096 bytes.

A new batch of simulation runs are required to analyse the impact of the packet size. Let $S' = \{\text{CBR}, \text{PAROO}\}$ be the set of traffic models and $T = \{64, 1536, 4096\}$ be the set of packet sizes. Recall that $R = \{125\%, 90\%, 60\%\}$ and $P = \{\text{TBM}, \text{TSW3CM}, \text{TSW2CM}, \text{srTCM}, \text{trTCM}\}$. The new set of scenarios that were simulated can then be expressed as $SIMS' = R \times S' \times P \times T$. Each of these 90 scenarios was run as described for the original scenarios. The traffic generation rate in bits per second was kept the same as in the original configuration scenarios – only the size of the packets was varied.

The relative performance of the traffic conditioners were compared using to the two performance metrics TargetRatio and GreenRatio. It was found that the traffic conditioners yielded similar TargetRatio results for each scenario in *SIMS'*. This is the same observation that was made in the original simulations. One can consequently deduce that the packet sizes used in the traffic stream does not influence whether or not a source achieves its target rate.

The GreenRatio performance measure did result in differences between the traffic conditioners. Differences were observed in all cases except in the cases where CBR traffic was used in an oversubscribed network. In all other cases, at least one of the conditioners performed significantly different from the others. The pattern of differences for each of the new packet sizes is similar to that observed for the original 576 byte packets size. In the remainder of the section this assertion is illustrated using a number of examples from the observations. In each case a plot of GreenRatio performance over the five conditioners is given. The conditioners are always plotted in the following order (from left to right) TSW2CM, TSW3CM, TBM, srTCM, and trTCM.

Please cite this article in press as: T. Strauss et al., A comparison of simulated traffic conditioner performance, Comput. Commun. (2008), doi:10.1016/j.comcom.2008.07.015

Please note that the scales on the *y*-axes of the plots are omitted since we are only concerned about the differences in performance between the traffic conditioners within a single plot. Furthermore, the scales vary *between* plots.

3.3.1. Small packet size

The first column of Fig. 5 contains plots of the GreenRatios obtained by the traffic conditioners in the original experiment for the oversubscribed case with CBR and PAROO traffic.

The second column contains the GreenRatios obtained for precisely the same scenarios, except for one aspect – the packets are now 64 bytes in size.

In the oversubscribed case (125% reserved), the traffic conditioners yielded very similar GreenRatio results for the non-bursty CBR traffic. Even when bursty PAROO traffic was used, the conditioners also exhibited the earlier observed behaviour: the times-sliding window based conditioners achieved higher GreenRatios than the token bucket based conditioners. From the plots it is clear that the relative performance of the traffic conditioners remained the same. The other scenarios are not shown since they lead to the same conclusion. The smaller packet size did not change the relative performance of the conditioners.

3.3.2. Medium packet size

Given that the small packets did not affect the traffic conditioners' relative performance, one should consider a packet size larger than the original to determine whether a larger packet size leads to a different conclusion. A size of 1536 bytes was used for the traffic sources in the present scenario. This size corresponds the maximum size of packets originating from an Ethernet network.

In this scenario the Kruskal-Wallis tests show that the traffic conditioners do not all achieve the same GreenRatio levels.

Fig. 6 shows plots for the nearly fully subscribed (90% reserved) network scenario. Observe that the non-bursty traffic – for both the original and the larger packets sizes – resulted in a wide range of GreenRatio observations for time sliding window based conditioners. The token bucket based conditioners, however, yielded much less variation. The important observation is not so much the differences between the two classes of conditioners, but the similarity between the original plot and the new plot. Similarly, when one considers the plots for the bursty traffic, one observes that the pattern of conditioner performance is again the same in the two plots. The plots therefore suggest that the larger packet size also does not affect the relative performance of the traffic conditioners.

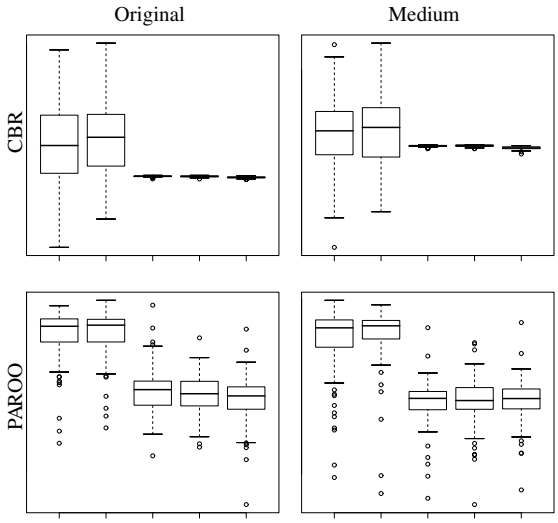


Fig. 6. 90%, original vs medium.

3.3.3. Large packet size

It is only in the case of the large packets that there was observed a difference in the performance of the traffic conditioners.

Fig. 7 presents the relative performance of the conditioner with the original packet size and the large packet size of 4096 bytes in the undersubscribed (60% reserved) cases. One pronounced difference between the original experiments and the new experiments is visible in the plots – that of the performance of the trTCM. It always scored zero on the GreenRatio metric. The reason for this is that the size of the packet is larger than the capcity of the peak burst size bucket used in the conditioner. This then causes the conditioner to immediately classify the packet a non-green. Since all the packets are larger than this bucket, they are all classified non-green and this, in turn, causes zero green packets to arrive at the destination, resulting in a GreenRatio of zero.

The srTCM does not exhibit this behaviour even-though it also uses a bucket that is smaller than the present packet size. The difference is due to the order in which the comparisons are made. The

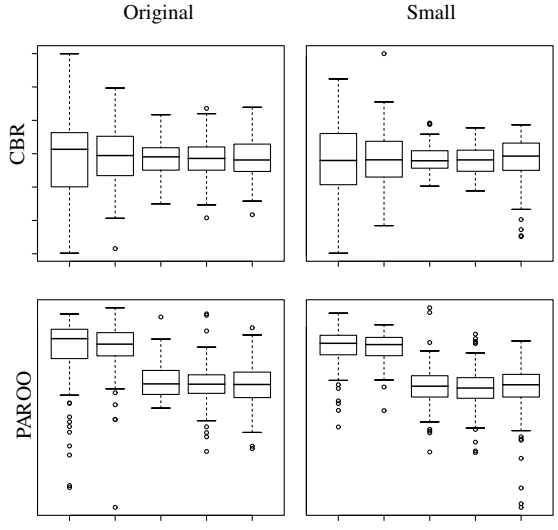


Fig. 5. 125%, original vs small.

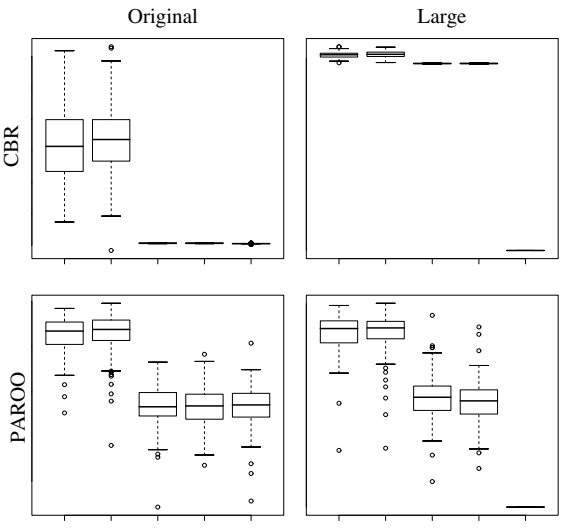


Fig. 7. 60% original vs large.

Please cite this article in press as: T. Strauss et al., A comparison of simulated traffic conditioner performance, Comput. Commun. (2008), doi:10.1016/j.comcom.2008.07.015

srTCM first compares a newly arriving packet against the committed rate bucket and after that, against the second, smaller bucket. The committed rate bucket is larger than the packet which means the packet could be classified as green. If the packet was found not to be green, the second bucket is used to determine whether the packet should be marked as yellow or red. In this case the packet will always be red since the packet is larger than the second bucket. Consequently, the srTCM will classify no packet as yellow. One should also note that the specifications of the srTCM and trTCM [8,9] point out that, for them to work according to expectation, the sizes of the buckets in the conditioners should be at least as large as the largest expected packet.

From the above one can deduce that the size of the packet does not generally affect the relative performance of the traffic conditioners. To avoid degenerate behaviour as seen above, one should take into consideration the possible size of packets when determining the sizes of the token buckets in the token bucket based conditioners.

4. Conclusion

When performance was measured in terms of the TargetRatio metric, there was no convincing evidence to suggest that the choice of traffic conditioner affects a source's achieved rate. Therefore, if a network designer is only interested in the relative target rate performance of the traffic conditioners, she is, based on the evidence at hand, free to choose any one of the studied traffic conditioners.

Traffic conditioner performance differed when compared in terms of the GreenRatio metric. The experimental results suggest that the traffic patterns do indeed play a role in how well the various traffic conditioners perform. A network designer should be aware of this fact and should choose an appropriate traffic conditioner. In general, the various versions of token bucket conditioners performed similarly to one another in a specific traffic pattern context. The same was true for the various versions of time sliding window conditioners.

However, within the context of the earlier experiments, the time sliding window conditioners tended to achieve more easily a green ratio greater than one than the token bucket conditioners. Later experiments suggested that the same ratios could be attained by token bucket conditioners if the bucket size is made large enough. The attainment of such ratios is an indication that the subscriber is being treated too leniently and that the network is possibly being deployed non-optimally from the provider's point of view. The network provider should therefore be cognisant of this observed difference between conditioners and should monitor the green ratios to ensure that appropriate levels are attained.

The results presented suggest that setting parameters for traffic conditioners appropriately is important. The results emphasise that network designers should take cognisance not only of the fact that traffic patterns and traffic conditioner parameter values affect the performance of the network; but also of the fact that these may ultimately impact on both provider revenue and on subscriber satisfaction. The methodology employed in this study – simulation of traffic sources in a simple network topology to study the impact of various traffic conditioners, characterised by various parameters – would appear to be a relatively inexpensive and practical approach to grappling with the real-world problem of arriving at an optimal network design.

References

[1] K. Nichols, S. Blake, F. Baker, D. Black, Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers, RFC 2474.
[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for differentiated services, RFC 2475.
[3] S. McCanne, S. Floyd, ns Network Simulator (Feb 2003). URL http://www.isi.edu/nsnam/ns/.
[4] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu, Advances in network simulation, IEEE Computer 33 (5) (2000) 59–67.
[5] V. Paxson, Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic, ACM SIGCOMM Computer Communications Review 27 (5) (1997) 5–18.
[6] W. Stallings, High Speed Networks: TCP/IP and ATM Design Principles, Prentice-Hall, Englewood Cliffs, NJ, 1998.
[7] W. Fang, N. Seddigh, A time sliding window three colour marker (TSWTCM), RFC 2859.
[8] J. Heinanen, R. Guerin, A single rate three color marker, RFC 2697.
[9] J. Heinanen, R. Guerin, A two rate three color marker, RFC 2698.
[10] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking 1 (4) (1993) 397–413.
[11] D.D. Clark, W. Fang, Explicit allocation of best effort packet delivery service, IEEE/ACM Transactions on Networking 6 (4) (1998) 362–373.
[12] T. Strauss, D.G. Kourie, M.S. Olivier, A simulation study of traffic conditioner performance, in: Proceedings of SAICSIT 2005, 2005, pp. 171–181.
[13] A. Steyn, C. Smith, S. du Toit, C. Strasheim, Modern Statistics in Practice, Van Schaik, 1994.
[14] R Development Core Team, R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, 3-900051-07-0 (2004). URL http://www.R-project.org.

Please cite this article in press as: T. Strauss et al., A comparison of simulated traffic conditioner performance, Comput. Commun. (2008), doi:10.1016/j.comcom.2008.07.015