A Method for RED Reconfiguration in Multi-Service Networks

W. Jarrett, L. Sacks
Department of Electronic & Electrical Engineering
University College London
{W.Jarrett@ee.ucl.ac.uk}

Abstract: Random Early Detection, or RED is a buffer management scheme that has been promoted by the IETF, and has been widely implemented by network equipment vendors. The algorithm used is initialised through a number of parameters that determine how network traffic is dealt with under periods of transient and persistent congestion. Setting these parameters is not a trivial task, as they are dependent on the estimated traffic volume and traffic profile of the network in question, entities that may change rapidly in today's heterogeneous networking environment. In this paper, we present a novel method of detecting when the configuration of RED may be sub-optimal, which provides a basis for re-initialising the RED algorithm with alternative parameters. We believe this technique may also be applicable to the configuration of other network control algorithms.

1 Introduction.

In Multi-service networks, Network, Maintenance and Restoration (or NMR) processes are associated with activities that collect and analyse data produced by network elements [1]. The desired end result is the identification and isolation of faults, and their repair. For hardware faults, this will normally constitute sub –component repair or replacement. However, users can perceive what are known as "soft" faults, which impair the performance of the service they are using. Such cases are dealt with through the analysis of management data retrieved by the same NMR processes. It is then possible to discover where network performance is less than optimal, and in such cases where possible, perform network re-configuration to return the network to normal operation.

Although IP networks may not as yet boast the same defined processes for dealing with faults & configuration, the operational characteristics are identical. Hardware failures aside, it is the correct operation of the many protocols used on a network which determine the level of service users receive. In terms of link sharing and buffer management, these amount to algorithms such as Random Early Detection [2], Weighted Fair Queuing, and Class Based Queuing, amongst others. We have focused on RED since it has been recommended for large scale network deployment by the IETF [3]. The remainder of this paper is organised as follows: Section 2 reveals additional details on RED, and why its correct configuration is important. Section 3 introduces the methodology we use to analyse the results obtained from the simulation study. This includes a brief treatment of Wavelets, and a state machine that represents the major phases of RED algorithm operation. Results and Analysis are presented in Section 4, whilst Section 5 explains our conclusions and future work.

2. Background

RED is a buffer management technique that can work both in co-operation with end host applications, or independently. As the name suggests, RED attempts to detect the onset of incipient congestion as early as possible, whilst randomising the way it selects which sources must throttle back on packet transmission to remove bias. The algorithm can set the ECN bit in a packet or discard a packet from a flow to signal congestion to an end host application. In the case of TCP based applications, this will engage mechanisms to recover from the packet loss, and will also result in a decrease of the sending rate of the application. Implementation involves the use of a number of parameters that must be configured with care to deliver optimum results. The numbers of packets (or bytes) in the queue of a router are monitored over a time interval, yielding an average queue length (avg_q). The objective is to maintain avg_q between an upper and a lower bound, known as the maximum and minimum thresholds (min_{th} & max_{th} respectively), by applying a weight (w_q) to the constantly moving average. As the moving average becomes closer to the maximum threshold, the application of a formula to calculate the maximum packet drop probability (or max_p) in conjunction with the selected weight cause a greater proportion of packets to be marked for discard. The opposite is true when the moving average approaches the minimum threshold. An estimation of the average packet size (aps) of network traffic is also used. The reader is referred to [2] for a detailed account of the algorithm.

Discovering the correct parameters for RED implementation on a given network is not a trivial task, as there are many factors to consider. Some parameters are set using an heuristic approach, whilst others depend upon the combination of the former, and an additional estimation of how traffic mix and volume. Clearly, although there may be many similarities between networks, these metrics are network dependent, with the possibility of being highly variable. Variable in the sense of distinct utilisation periods for different periods of the day or week, but also in a long-term evolutionary sense, due to an ever increasing user base and a greater proliferation of applications. Guidelines for the parameterisation of RED are outlined in [4]. Further study however, has indicated that although these parameters may yield good results, they are by no means optimal. In fact when dealing with large numbers of flows, these parameters can fail. [8]. We speculate that a single set of RED parameters may not yield optimum results over an extended period of time, given the dynamic nature of IP networks. Given this scenario, a technique for re-configuring RED (and other network control algorithms) with new parameters based on perceived network conditions seems necessary.

3. Methodology

Our overall approach to this problem can be divided into three distinct phases. Initially, the collection of traffic data from the network. Secondly, a set of analysis tools that allow us to obtain the goodput of the network, but principally how this relates to the current RED parameter settings. Thirdly, a method of selecting alternative parameters to reconfigure the RED algorithm if this is deemed necessary. For the remainder of this paper we focus on the second phase, which involves Wavelet Multi-Resolution Analysis.

Wavelets have found use in several scientific/engineering fields including De-noising of signals, Visual Image Processing, Signal Compression, and analysing self-similarity. Specifically,[5] [6] they are a series of mathematical functions that can be used to divide a raw time-amplitude based signal into a series of signals, each of which exist in the time-frequency domain. Each of these sub-signals can be analysed individually at its given scale (1/frequency) or a combination of all signals can be viewed graphically on a 3D plane or image map to reveal how the original signal behaves in time, frequency and amplitude. Fourier analysis and its variants have been the dominant technique in the frequency analysis domain for decades, but they have limitations in the analysis of non-stationary signals where not only frequency information is required, but also the time interval during which each frequency is present. Frequency information is important, as the correct configuration of RED is desired to control the build up of congestion in a timely manner. This translates to RED interaction (packet marking/dropping) at different frequencies under varying traffic conditions. Clearly, if applications are responding to the operation of RED, then arrival rates of traffic at a RED-enabled router should oscillate accordingly as TCP window sizes grow and shrink. If the configuration of RED is sub-optimal, it may lead to sub-optimal packet marking/dropping. This may be for a variety of reasons, e.g. the min_{th} or max_{th} are incorrectly set, resulting in the avg_q to grow too much/little before packet marking commences, etc. Packet marking should be such that we avoid packet loss through buffer overflow whilst maintaining high link utilisation. Our technique is based upon associating different levels of goodput with a particular set of RED parameters. For each case, Wavelets are used to extract any features from collected data (which include traffic arrival & departure rates at routers) that relate to the level of goodput. Finding a consistent set of features allows goodput to be determined only from Wavelet Analysis (most likely using a threshold technique). At this point, the original values used to initialise RED could be replaced by a set known to offer better goodput although what these values should be will not be addressed at this time. We believe network traffic arrival rates to be one of the more significant metrics, since it is one of few quantities that can be retrieved from real routers.

The second part of our methodology involved the development of a state machine that is representative of the main phases of operation of the RED algorithm. We look at how the packets are distributed between the four states, as this may act as a pointer to the dominant behaviour of the traffic, which has implications for the goodput. These are as follows; state 1 represents no packet drops; state 2 represents random packet drops between min_{th} and max_{th}; state 3 represents forced packet drops between max_{th} and q_{lim}; state 4 represents forced packet drops because of buffer overflow.

The third part of our methodology involved the design of a series of Monte-Carlo style simulations to discover the degree with which one can deviate from the suggested RED parameters and still achieve reasonable performance. In these simulations each of the RED parameters was chosen using a uniform random number generator. The limits chosen for each parameter were such that they explored. The ns-2 simulator [7] has been used to conduct the experiments described in this paper. The simulation topology consists of a single bottleneck link (with bandwidth of 10 Mb/s and propagation delay of 30ms.) providing a connection between varying numbers of traffic sources and traffic sinks (each with a bandwidth of 1Mb/s and propagation delay of 10ms.). In these experiments we have used 300 FTP flows to form the basis of the traffic profile. We also use a small number of low bandwidth telnet flows that provide a degree of randomness to the traffic profile. These serve only as preliminary tests to form a basis for comparison. Other simulations including web traffic profiles are being conducted and results of these will follow subsequently.

4. Results & Analysis

The Monte-Carlo simulation tests yielded goodput results over a range of different values. Over one thousand tests were done to provide base information. Typically, goodput was in the range of 85% to 98%. For comparison, we focus our attention on two simulation runs. Simulation Number 74 achieved a goodput score of 98% whilst the simulation number 394 achieved 85.6%. The RED parameters that gave rise to these results together with RED state information are shown in Table 1.

SimNo	w_q	max_p	aps	min_{th}	max_{th}	q_{lim}	Thruput	Goodput	State1	State2	State3	State4
74	0.001	9	1240	9	33	115	102.62	98	1926	8722	159	601
394	0.0008	6	122	12	41	171	91.89	85.6	2716	3811	8370	61

Table 1: RED Parameters & Results.

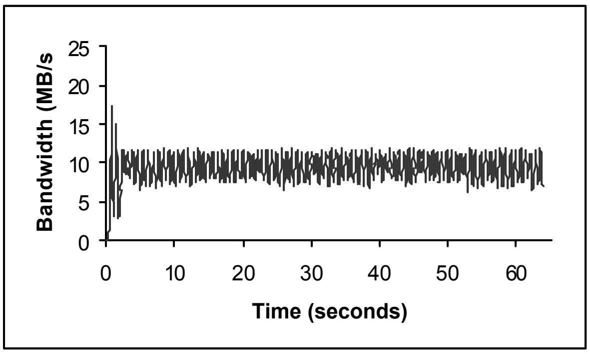

Figure 1: Simulation 394 Arrival Rates

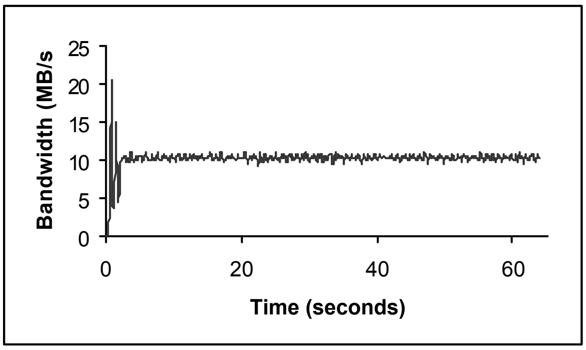

Figure 2: Simulation 74 Arrival Rates

In Figures 1 and 2, we show the arrival rates at the bottleneck router of simulation runs 394 and 74 respectively. Notably, the arrival rates for simulation 74 are much smoother compared to that of simulation 394, The immediate implication here is that the RED parameters chosen for the former are a significantly better since there is less fluctuation in the behaviour of the traffic as RED intervenes. A more interesting measurement is revealed from construction the image maps resulting from the wavelet transform of both arrival rate traces. These are shown in Figures 3 and 4 respectively.

The image map displays results in three dimensions. Firstly, the scale or frequency level is depicted along the left of the graph, with 0.25 being the first aggregation level. Time is located along the bottom of the graph, where the duration of each simulation was 64 seconds. The colour of each adjacent band reveals the difference

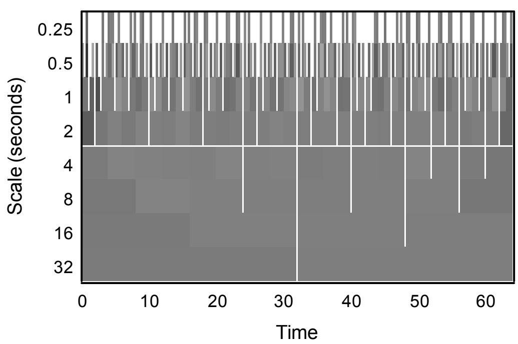

Figure 3: Simulation 394 Wavelet Coefficients

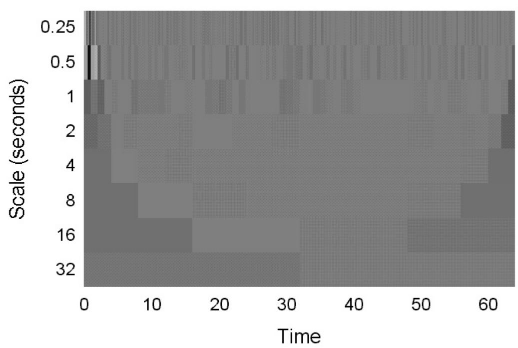

Figure 4: Simulation 74 Wavelet Coefficients

in amplitude of two successive wavelet coefficients. Where the colour changes are smooth, the less difference between successive coefficients, whereas distinct bands of colour are representative of larger differences between coefficients. The exact amplitude is conveyed by the intensity of the colour (where black represents the minimum and white the maximum). As seen from the image maps, at the second decomposition at a frequency of 2 Hz., there is a distinct difference between the two sets of arrival rates. To understand what has caused these differences, we can consider the state results for both simulations. From Table 1, we can see that a significant amount of packets were treated in state 2 for simulation 74, whereas simulation 394 had more packets treated in state 3. Remembering that for optimum results we wish to control ave_q whilst it is between the min_{th} and max_{th}, the result we see for simulation 394 is undesirable.

Analysing additional results from the random simulations highlights the inverse relationship between the queue weight parameter, and the maximum packet drop probability parameter. If w_q is too low [4], the queue cannot respond quickly enough to the onset of congestion, although it would appear we could compensate for this by increasing the level of the max_p parameter. If both the max_p and w_q parameters are too low, the queue will fill rapidly and then be subject to vigorous packet drops as max_p increases towards 1. This severely impairs the thruput/goodput of the bottleneck link, as many TCP flows will halve their send windows in response to the congestion indication. In contrast, setting both parameters at a high level removes the ability of the algorithm to distinguish instances of transient congestion from incipient congestion. We believe that the large number of packets that are dealt with in state 3 for simulation. 394 support these points, in that the setting of w_q and max_p are both too low, hence the algorithm oscillates between performing timely packet drops and performing vigorous packet drops.

5. Conclusions & Future Work

The Multi-resolution Analysis technique appears successful in determining the difference in the level of goodput obtained by an aggregated traffic flow. Work is being done to associate different levels of goodput with the value of the coefficients that are produced from the wavelet transform. This will allow link utilisation statistics to be inferred from looking only at the wavelet co-efficient plot. This will require testing under a number of different traffic scenarios to ascertain a reliable mapping.

To continue the development of the third part of our methodology, we will perform cluster analysis on the parameters that were used for the Monte Carlo simulations. A notable result from these simulations was that even when the RED parameters generated deviated significantly from the suggested values, high levels of goodput could still be achieved (sometimes above 95%). Performing cluster analysis on these parameters would allow the discovery of similar groups of parameters that yield good results. Such data could then be used as a tool for increasing our understanding of the relationship between the parameters and the values, and selecting alternative parameter sets when experiencing persistent congestion.

We continue to investigate different aspects of Multi-Resolution to ascertain if we can see when both w_q and max_p have been set too high or too low. This significant step may give an indication as to what parameters need to be changed to achieve higher goodput, as well as how the parameter should be changed.

References.

[1] The TeleManagement Forum. ***"The SMART TMN Telecomms Operation MAP"*** Page 44, Evaluation Version Release 1.0, December 1998.

[2] S Floyd, V Jacobson. ***"Random Early Detection Gateways for Congestion Avoidance"*** August 1993. IEEE ACM Transactions on Networking.

[3] B. Braden et al., ***"Recommendations on queue management and congestion avoidance in the Internet"*** RFC 2309, April 1998.

[4] S Floyd, "**RED: Discussions of Setting Parameters"** http://www.aciri.org/floyd/REDparameters.txt, November 1997.

[5] J. Walker. ***"Wavelets and their Scientific Applications"*** Chapman & Hall/CRC 1999. ISBN 0-8493-8276-9

[6] G Strang, T Nguyen, ***"Wavelets and Filter Banks"*** 2nd revised edition, December 1996 Wellesley-Cambridge Press. ISBN: 0961408871

[7] UCB/LBLN/***VINT "The Network Simulator - ns-2"*** http://www.isi.edu/nsnam/ns/

[8] R Morris, ***"Scalable TCP Congestion Control"*** Proceedings of IEEE INFOCOM, pp 1176-1183, April 2000