An Efficient Algorithm for Dynamic Bandwidth Allocation

Richard E. Martin and Kenneth M. Guild
School of Computer Science and Electronic Engineering
The University of Essex
Wivenhoe Park
Colchester

Abstract—**Dynamic networks offer an unprecedented degree of flexibility, enabled by advanced signaling and automation. Concurrently, advances in Deep Packet Inspection (DPI) techniques have made measuring and monitoring more powerful than ever before. In this paper, we propose a novel algorithm to bridge the gap between these disciplines by making networks more autonomous, with automated control of bandwidth-provisioning decisions.**

I. Introduction

A dynamic network is one in which rapid, simple re-distribution of bandwidth has been facilitated by advanced control- and data-plane architectures. The data-plane must support the flexible addition and deletion of bandwidth in a transparent manner with no loss-of-service. The control-plane must offer fast, automated interfaces which allow for signaling between the edge-network devices and the network control to request such bandwidth changes.

Within such dynamic networks, the assignment, re-dimensioning, and de-assignment of bandwidth provision is simple and fast, with reduced involvement from human operators. Concurrently, measuring and monitoring equipment is more powerful than ever, with SNMP now being reinforced with deep-packet inspection, allowing the identification of flows by application-type [1]. However, at present the bandwidth provision requires human intervention: first, to assess the need (aided by the measuring and monitoring facilities available) and then to use network management to facilitate provision. We seek to automate this process by combining measuring and monitoring with automated signalling, allowing autonomous decisions about network capacity to be made at the edge of the network by suitable traffic-monitoring algorithms.

II. About this paper

Within this paper, we use an intellectually and computationally simple algorithm that accounts for traffic variability in order to enable the autonomous optimisation of bandwidth provision, a process termed Dynamic Bandwidth Allocation (DBA). We denote this algorithm the DYnamic Linear Bandwidth Estimation through Regression of Traffic (DYLBERT). The performance of this algorithm will be compared with that of the Statistical-Decision (SD) method [2] by assessing the ability of both algorithms to correctly match the observed traffic with their own predictions of bandwidth usage over a single, point-to-point link.

III. The traffic

The traffic, defining bandwidth demands b measured in Mbps at regular intervals i denoted b_i, was created using a signal-in-noise model to match the assumptions made in [2]. A sinusoidal function was used to create the "signal" data-set, s_i.

$$s_t = \left[\sin\left(\frac{t\cdot 2\cdot \pi}{1440}\right)+2\right]\cdot\frac{2000}{3},\quad 0<r\leq 525600 \tag{1}$$

In further keeping with the paper [2], we define the noise component w_t as i.i.d. Gaussian observations thus:

$$w_t \sim N(\mu,\sigma) \tag{2}$$

where μ denotes the mean, and σ the standard deviation.

In order to emulate a range of variation, three data-sets were created using a mean of 0 with three values of the standard deviation (S) : 10, 25 and 50 Mbps, accordingly:

$$x_t = s_t + w_t \tag{3}$$

where x_t denotes the traffic utilised in Mbps at time t.

IV. The Control Plane

In order to focus on an evaluation of the performance of the algorithms, the control plane is defined very simply:

1 Determinism: The control plane always responds within the same, constant, time referred to as the delay.

2 Atomic: The control plane will not accept a request to modify bandwidth provision while another is being enacted.

Following from this definition, bandwidth request in interval i will be fulfilled in interval $i+d$; however, during the period of $(i+1, i+2, \ldots i+d)$ the control plane will not accept requests for bandwidth modification.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7261
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7261

V. The Data Plane

We consider only symmetric provision where there is sufficient capacity on the terminating interfaces to satisfy any bandwidth requests.

The prediction of a DBA algorithm may be a continuous value, B, $B \in \mathbb{R}^{+}$. As the data-plane is able to process bandwidth requests only in multiples of the data-plane granularity, the bandwidth request R is rounded appropriately using a ceiling function

VI. The DYLBERT Algorithm.

Within this paper we use a fitted linear model determined using the ordinary least squares method [3] to predict bandwidth requirements. Such a system seeks to fit a function to the data (here referred to as Y_i to follow the conventions of linear regression) as defined below:

$$y_i = a + b \cdot x_i \tag{4}$$

where a is the intercept on the x axis, and b is the gradient. b is of most import, as it tells us about the change in traffic demand over time, implying the need for greater or lesser bandwidth provision. The standard deviation of the residuals is defined as:

$$\sigma_r = sd(y_i - Y_i),\ 0 < i \leq n \tag{5}$$

This allows us to compensate for the error inherent in the least-squares fit, which may lead to packet loss.

From the definition of Delay in §IV, it is clear that bandwidth requested in interval i will be fulfilled in interval $i + delay$. Therefore, the bandwidth requested at i will have to suffice for the interval $i + delay$ to $i + (2 \cdot delay)$. To account for this, the bandwidth requested at interval i, denoted B_i, is defined as follows:

$$B_i = max(a + b \cdot t + 3 \cdot \sigma_r),\quad i + delay < t < i + (2 \cdot delay) \tag{6}$$

VII. The SD Method

Recently, a mechanism for bandwidth allocation in TDM networks has been proposed based on specifying thresholds for bandwidth alteration which are linked to the statistical properties of the observed traffic [2], along with the granularity of the data-plane. The purpose of the algorithm is to reduce Unstable Determination, defined §IX-C.

Formally, the threshold to increase traffic (T_1) and to decrease traffic (T_2) are linked to the capacity of each TDM link (which they denote L_0), the number of such links already provided (N), and the standard deviation (σ) thus:

$$\begin{aligned} T_1(N) &= N \cdot L_0 - 3 \cdot \sigma && (7) \\ T_2(N) &= (N-1) \cdot L_0 - 3 \cdot \sigma && (8) \end{aligned}$$

The algorithm defines a window of the previous 64 data-points which it uses to determine path addition or deletion. The number of times that the traffic exceeds or falls below a threshold is counted; exceeding the threshold for addition adds to a variable they denote x, while falling below the threshold for path deletion adds to a variable they denote y thus:

$$\begin{aligned} w &= \{\, x_{(t-63)},\ x_{(t-62)}\ \cdots\ x_t \,\} && (9) \\ x &= |\{\, x :\ x \in w,\ \ x > T_1(N) \,\}| && (10) \\ y &= |\{\, x :\ x \in w,\ \ x < T_2(N) \,\}| && (11) \end{aligned}$$

Following on from a thorough mathematical analysis, the authors of the paper define cricital values of x and y which cause path addition and deletion, defining the threshold of x as 20 and y as 44.

VIII. About the Simulator

The data describing the traffic demands was used as the input to a DBA algorithm simulator implemented in multi-threaded ISO C99 which models the control plane and the network interfaces, as well as providing implementations of the algorithms themselves. The SD mechanism was implemented based on the information provided within the original paper [2]. The DYLBERT algorithm builds up the ordinary-least-squares linear-regression functions provided by the GNU Scientific Library [4].

IX. Defining Performance Statistics

Internally, the simulator generates a vector B_t, where B indicates the bandwidth assigned at time t, having taken into account the effect of the control- and data-plane properties on the requests from the DBA algorithms. By comparing this with x_t we can define several metrics which allow for the analysis of the performance of the algorithms.

A. Under-provisioning

Under-provisioning is indicated by intervals during which the algorithm underestimates demand. In a real network scenario, this may cause packet loss and unacceptable congestion on the link, as well as a violation of the Service-Level Agreement (SLA). For this reason, it is worth of study.

We define Underprovisioning, denoted $\hat{U}$, as the sum of all negative values of $B - x$ as a percentage of the total bytes, such that:

$$\hat{U} = 100. \left(\frac{\sum \left[H(B_i - x_i) \cdot (B_i - x_i) \right]}{\sum x_i} \right) \tag{12}$$

where $H(x)$ denotes the Heaviside step function [5] and n is the number of intervals.

This definition allows us to study Under-provisioning in relation to a guaranteed service level; for example, the performance of an algorithm in relation to a five-nines SLA is easy to evaluate following this definition, as it corresponds to an Under-provisioning level of 10^{-3}.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7261
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7261

B. Analysing Over-provisioning

Over-provisioning is indicated by intervals during which the algorithm overestimates demand. This is less problematic than Under-provisioning; Over-provisioning will have no negative effects on the quality of service percieved by the client; however, an excessive Over-provision indicates that greater bandwidth savings could be made, leading to better network utilisation and therefore efficiency.

Over-provisioning, denoted $\hat{O}$, is defined similarly to Under-provisioning:

$$\hat{O} = 100. \left(\frac{\sum \left[H(x_i - B_i) \cdot (B_i - x_i) \right]}{\sum x_i} \right) \tag{13}$$

C. Unstable Determination

The metric of unstable determination was proposed in [2], defined as any unnecessary change in bandwidth. For example, an increase in provision followed immediately by a decrease (or the oppsite scenario) is considered "unstable".

We adjust this metric to incorporate delay by defining unstable determination as having occurred when an increase follows a decrease (or the inverse occurs) and the holding time for the associated provision is equal to the delay.

We process $\boldsymbol{B}$ into two vectors, $\hat{\boldsymbol{B}}$ giving the bandwidths (omitting repetitions), and $\hat{\boldsymbol{H}}$ giving the associated holding time. From this, we identify all potentially-unstable allocations thus:

$$t = \; \{ \; t : \; t \in \mathbb{N}^{+}, \; \hat{\boldsymbol{B}}_{t-2} = \hat{\boldsymbol{B}}_t \wedge \hat{\boldsymbol{B}}_t \neq \hat{\boldsymbol{B}}_{t-1} \; \} \tag{14}$$

allowing us to define v, below, indicating the number of unstable determinations:

$$v = \; | \{ \; x : \; x \in t, \; (\hat{H}_x - delay) = 0 \; \} | \tag{15}$$

It is more-useful to express unstable determination as a percentage of the total signalling; in such a manner, the percentage of errant signals can be studied independently from the change in the total number of signals sent. Therefore, we define Unstable Determination, $\hat{D}$, as

$$\hat{D} = 100 \cdot \frac{v}{\hat{S}} \tag{16}$$

where $\hat{S}$ indicates the total number of signals, which can be trivially defined from $\boldsymbol{B}$ - any adjacent values which are not equal ($B_i \neq B_{i+delay}$) require signalling. $\hat{S}$ is therefore the number of times this occurs, defined in set builder notation thus:

$$\hat{S} = \; | \{ \; t : \; t \in \mathbb{N}^{+}, \; B_t \neq B_{t+1} \; \} | \tag{17}$$

X. Results

A. Granularity

The SD mechanism operates by the thresholds outlined in §VII; therefore, the performance should be affected by granularity (L_0), as well as traffic variance. We expect DYLBERT to deliver more-deterministic performance. The graph in Fig. 1 plots the performance of the SD and DYLBERT algorithms for the different traffic-variances with varying data-plane granularity. On the Under-provisioning graph (logarithmic y-axis), Under-provisioning equivalent to a 99.999% availability is shown.

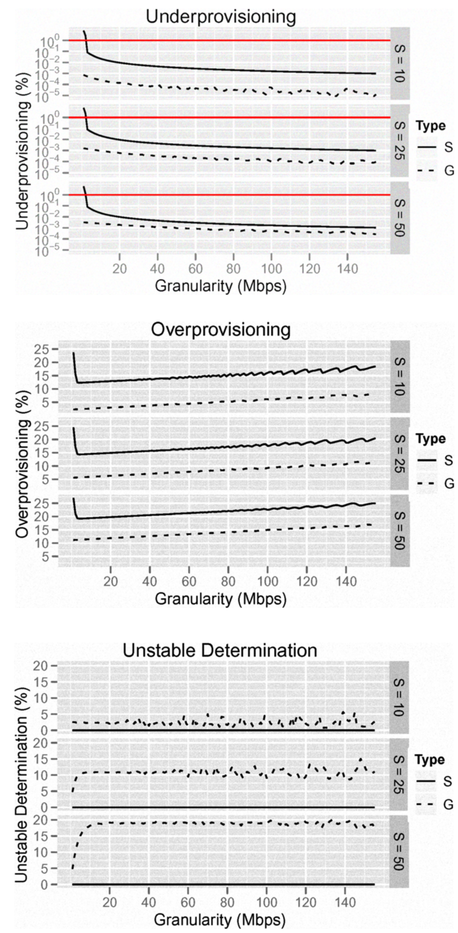

Fig. 1. Graphs showing how the performance of the SD and DYLBERT algorithms vary for three different variances of traffic and with varying data-plane granularity

The aim of the SD algorithm of achieving high determinism has been achieved. This compares favourably with the performance of the DYLBERT algorithm, where Unstable Determination increases significantly with increased traffic-variance, up to almost 20% in the worst case.

However, it is also clear that the exceptional performance in this area has come at the cost of lower performance in others; the amount of Under-provisioning is almost 100 times higher than DYLBERT in the worst-case, and Over-provisioning is between two and five times higher.

XI. Delay

The second investigation is into the effect of delay on performance. In order to do so, a data-plane granularity of 20 was chosen on analysis of the data in Fig. 1; the rapid initial decrease in Under-provisioning of the SD

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7261
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7261

algorithm has been overcome by this point, as well as the rapid rise in Unstable Determination of the DYLBERT algorithm. Based on this, the results of varying the control-plane delay can be seen in Fig. 2. Once more, on the Under-provisioning graph (logarithmic y-axis), Under-provisioning equivalent to a 99.999% availability has been highlighted.

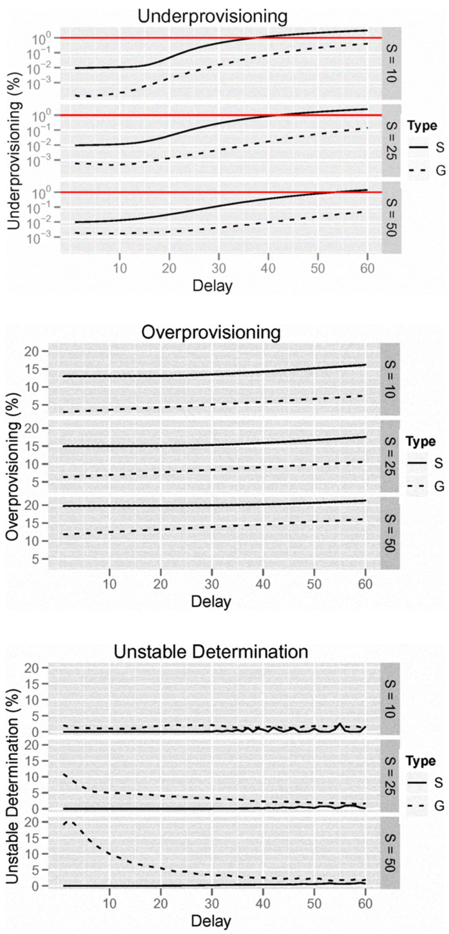

Fig. 2. Graphs showing how the performance of the SD and DYLBERT algorithms vary for three different variances of traffic and with varying control-plane delay

Both algorithms show a clear degradation in performance with increasing delay. This is to be expected; SD no ability to account for delay, while the gradient-based algorithm allows for a limited forecasting with increasing margin-of-error as delay increases. The DYLBERT algorithm maintains its performance advantage over the SD method even as delay increases, showing around 10 times better Under-provisioning performance, and between 10 and 5% less Under-provisioning.

The SD method maintains its strong performance in the Unstable Determination metric, with only a small percentage of unstable determinations being incurred at high delays.

XII. Discussion

The purpose of the SD method is to reduce the frequency of unstable determination. While we have shown this, the excellent performance in this area has come at the cost of others. The algorithm suffers from a serious tendency to under and over-provision bandwidth. The DYLBERT algorithm performs much better, resulting in lower Under-provisioning and Over-provisioning, leading to higher utilisation of the provided bandwidth and network equipment. There is, therefore, a trade-off to be had between more-accurate signalling and more-accurate predictions.

From the graphs in Fig. 1, artificially making the granularity more-coarse than the data-plane permits (for example, specifying a granularity of 20Mbps on an Ethernet network, where the minimum granularity is measured in kilobytes) will markédly decreased signalling traffic while also acting to decrease Under-provisioning, and therefore may make more sense than adopting an algorithm which reduces control plane load at the expense of more-general performance.

XIII. Conclusion

Within this paper we have used a novel mechanism for bandwidth allocation within autonomic networks, and compared its performance with the SD algorithm. While we have shown that both algorithms have their advantages and disadvantages, it is clear that for most scenarios the advantages of the SD method do not outweigh its disadvantages of increased Under-provisioning, and Over-provisioning when compared to our algorithm.

Acknowledgment

This work was funded by the UK Technology Strategy Board (TSB) and Engineering and Physical Sciences Research Council (EPSRC) under the Heterogeneous IP Networks (HIPNet) project (EP/E002382/1). The authors are with the School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, U.K. (e-mail: rmarti@essex.ac.uk, kguild@essex.ac.uk).

References

[1] S. Kumar, J. Turner, and J. Williams, "Advanced algorithms for fast and scalable deep packet inspection," in *ANCS '06: Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems*, (New York, NY, USA), pp. 81–92, ACM, 2006.

[2] S. Yamamoto, I. Shake, T. Kurahashi, Y. Tarui, W. Imajuku, Y. Tsukishima, K. Sasayama, O. Moriwaki, and M. Fukutoku, "Theoretical and experimental study of statistical-decision method for link-capacity adjustment," *Journal of Lightwave Technology*, vol. 25, pp. 2864 – 2873, September 2007.

[3] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures.* 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742: Chapman & Hall/CRC (Taylor & Francis Group), fourth ed., 2007.

[4] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, and F. Rossi, *GNU Scientific Library Reference Manual (3rd Ed.).* GNU, 2007.

[5] R. Bracewell, *The Fourier Transform and its Applications.* Boston: McGraw-Hill, third ed., 2000.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7261
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7261