



Faculty Publications

2001-05-01

Effective Bandwidth for Traffic Engineering

Mark J. Clement
clement@cs.byu.edu

Rob Kunz

Seth Nielson

Quinn O. Snell
snell@cs.byu.edu

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Computer Sciences Commons](#)

Original Publication Citation

Effective Bandwidth for Traffic Engineering, Rob Kunz, Seth Nielson, Mark Clement, Quinn Snell, Proceedings of the IEEE Workshop on High Performance Switching and Routing (HPSR 21), Dallas, TX, May 21.

BYU ScholarsArchive Citation

Clement, Mark J.; Kunz, Rob; Nielson, Seth; and Snell, Quinn O., "Effective Bandwidth for Traffic Engineering" (2001). *Faculty Publications*. 573.
<https://scholarsarchive.byu.edu/facpub/573>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.

Effective Bandwidth for Traffic Engineering

Rob Kunz, Seth Nielson, Mark Clement, Quinn Snell

Brigham Young University
Department of Computer Science
Brigham Young University
Provo, UT 84602

{robby, darkman, clement, snell}@cs.byu.edu

Abstract- In today's Internet, demand is increasing for guarantees of speed and efficiency. Current routers are very limited in the type and quantity of observed data they can provide, making it difficult for providers to maximize utilization without the risk of degraded throughput. This research uses statistical data currents provided by router vendors to estimate the impact of changes in network configuration on the probability of link overflow. This allows service providers to calculate in advance, the effect of grooming on a network, eliminating the conservative trial-and-error approach normally used. These predictions are made using Large Deviation Theory, which focuses on the tails of the distribution, giving a better estimate than average and peak values.

I. INTRODUCTION

In today's Internet, demand is increasing for guarantees of speed and efficiency. Service providers are motivated to find a way to multiplex as much traffic as possible without breaking any service guarantees and to also find a way to charge for the guaranteed bandwidth they are providing. The Theory of Effective Bandwidth is designed to accurately estimate the resources that a particular source requires and the degree of statistical multiplexing that can occur while maintaining a reasonable packet drop probability.

This research uses traces from AT&T frame relay trunks to determine the effective bandwidth of individual connections through the network. Individual connections can be moved to new routes within the network in order to increase traffic balance. The principle metric used is the probability of overflow for each link, computed from the effective bandwidth equations.

Large trunk routers are limited in the amount of trace data that they can practically obtain and maintain. Much of this research has focused on using data that routers currently provide, rather than making optimal calculations with data that is not generally available. Much of this research focuses on converting gross trace data from several thousand trunks and several hundred thousand circuits in the AT&T network into detailed traces for each circuit.

II. EFFECTIVE BANDWIDTH

Using Large Deviation Theory (which focuses on the impact of the tails of a distribution), Effective Bandwidth computes a bandwidth that is a better estimate than just average and peak values. The Effective Bandwidth is computed by using source characteristics along with the link through which it is passing.

The effective bandwidth of a source given the operating point of a link is given by the following equation.

$$\alpha_j(s, t) = \frac{1}{st} \log E \left[e^{sX_j[0,t]} \right] \quad (1)$$

In the equation for Effective Bandwidth, the two most enigmatic variables are the 's' and 't' parameters. One source states, "Particular values of (s,t) can be taken to characterize the *operating point* of a link," [1]. This same source characterizes the 's', or space, parameter as the degree of statistical multiplexing potential of a link. A large value of 's' indicates that a low degree of statistical multiplexing can be performed, whereas a small value indicates a large degree of potential statistical multiplexing. It should be noted that the 's' parameters is in units of kb^{-1} .

The 't' parameter is used to characterize the most probable duration of the buffer busy period prior to overflow and also represents the minimum granularity necessary for traces to capture the statistical properties that affect buffer overflow [1]. It should be noted that the 't' parameter is in units of milliseconds.

Taken together, these parameters form the 'st' parameter which is in units ms/kb . Conceptually, this represents the amount of multiplexing that can happen in a period prior to overflow. These two parameters are dependent on all of the characteristics of the link: buffer, capacity, QoS and scheduling. It is also interesting to note that they are dependent (also) on the sources, but with a large degree of multiplexing, this definition 'loop' is broken [1].

Taking 's' and 't' into account, the effective bandwidth equation begins to become clear. The Effective Bandwidth of a source is characterized by the number of cells that it produces divided by the 'st' parameter. It should be noted that the number of cells produced by the Effective

Bandwidth is computed using a logarithmic moment generating function [2]. A moment generating function produces the ‘central moments’ about a probability density function. The first moment is the mean. In this equation, the moment generating function assumes that the distribution of cells of a source follows some probability density function or some random variable $X[0,t]$ where $[0,t]$ is the time interval in which the cells are produced. A *logarithmic* moment generating function is used as opposed to a moment generating function because it allows for better estimates in the tails of the distribution, or in the peaks. Once this logarithmic moment value has been computed, it is divided by the ‘st’ parameter. Basically, this means that effective bandwidth is equal to the most probable number of cells produced in a given $[0,t]$ time period divided by the amount of multiplexing that can be performed in the same time period.

Of course, it is difficult to know the particular probability density function of your cell distribution. Fortunately, we can obtain an estimate using empirical data. The modified Effective Bandwidth equation is defined using an average of the production of cells in various time periods [1]. Using this equation, the logarithmic moment generating function can be computed using trace files. The empirical form of the Effective Bandwidth bears a strong resemblance to research into Self-Similarity. For a given value of ‘t’, all of the data sent during the time interval between $(i-1)t$ and it is aggregated into a single value. A search can then be made for the interval that produces the most likely buffer utilization prior to overflow. The empirical form for the effective bandwidth equation [5] is

$$\alpha_j(s,t) = \frac{1}{st} \log \left[\frac{1}{T/t} \sum_{i=0}^{T/t} e^{sX_j[(i-1)t, it]} \right]. \quad (2)$$

At this point, we integrate the Effective Bandwidth of the sources into a link with a capacity C and a buffer B. The sup-inf formula (3) is used to estimate the values of ‘s’ and ‘t’ given empirical data.

$$\begin{aligned} & \lim_{N \rightarrow \infty} \frac{1}{N} \log Q(Nc, Nb, Nn) \\ & = \sup_t \inf_s \left[st \sum_{j=1}^N n_j \alpha_j(s,t) - s(ct+b) \right] = -I \end{aligned} \quad (3)$$

This formula integrates the effective bandwidths of the various sources being multiplexed and evaluates them using the capacity and buffer of the link. This equation is equal to -I and the Probability of Overflow is equal to $e^{-NI + o(N)}$ and $o(N)$ goes to 0 as N is large [5].

It should be noted that N represents the number of combined sources that are entering the link. That is, if the

link has three types of sources, source A, B and C, and A comprises 20% of the flow, B comprises 40% and C comprises 40%, then $n_A = .2$, $n_B = .4$ and $n_C = .4$ and N is the number of these streams entering the system.

Once the ‘s’ and ‘t’ parameters are computed from the sup-inf formula (3), the effective bandwidth equation (2) can be completed. It has been shown through theoretical and empirical studies that changes in traffic mix have a very light impact on these parameters. Once they have been computed, as long as the traffic mix does not change drastically, new sources’ bandwidth could be easily computed without re-computing the entire sup-inf (which would require trace files and some computational time).

As an aside, the sup-inf formula (3) requires that the formula be computed using two optimizations. First, for each ‘t’, an ‘s’ is found that minimizes the inside of sup-inf. The largest value resulting from each of these ‘t’s is the result of the function. Remember that ‘t’ represents the most probable time of the buffer being busy prior to overflow and ‘s’ represents multiplexing. In searching for a smallest ‘s’ we are actually searching for a worst case scenario: the smallest value of ‘s’ that will still cause overflow for that value of ‘t’. The sup-inf formula effectively looks at the tail of the distribution rather than in the center. This allows for greater protection against over optimistic appraisals of link utilization. On the down side, these equations have been found to calculate cell loss rates and probability of overflow higher than in practice or simulation. In their defense, you probably will never loose a customer because you allow him too much bandwidth.

The ‘s’ parameter can be found because of the convexity of the equation. The Effective Bandwidth equation is determined by ‘s’ in a logarithmic manner and has a slow increase. The ‘s(b+ct)’ is determined by ‘s’ but is constantly increasing. The minimal value of the difference between the effective bandwidth and the ‘s(b+ct)’ terms can be computed using a Golden Section Search.

Java code has been written to take a trace file and generate the effective bandwidth and the ‘s’ and ‘t’ parameters. These values can then be used to predict overflow probabilities and to perform other modeling tasks. We have tested the code on existing traces and have been able to calculate similar values to other researchers in order to verify our techniques.

III. TRACE DATA

Before the effective bandwidth of sources can be determined, trace data must be available. Chris Chase from AT&T provided us with a large quantity of trace data summarizing several thousand trunks and several hundred thousand connections. The trace data contains hourly averages and 5 minute peak values for each of the trunks. Given this data, we had to derive specific connection values from those gross trunk values.

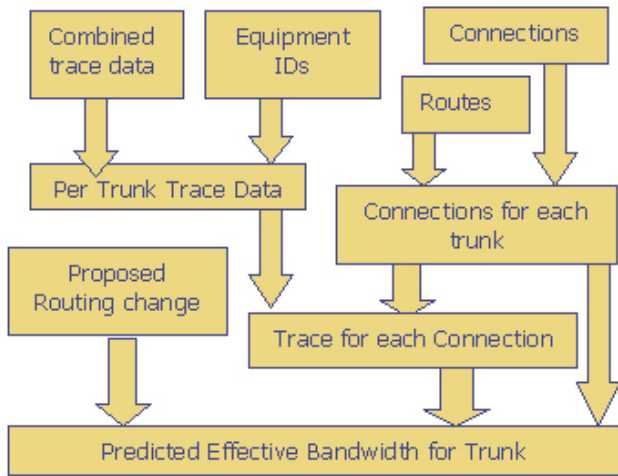


Fig. 1. Traffic Engineering overview.

Fig. 1 provides an overview of the trace generation process. First we extracted trunk statistics and circuit information from the statistic files. We then use Gaussian elimination and Linear Programming to calculate individual circuit values from the trunk statistics. This results in per-circuit trace files. These files can be used to compute the effective bandwidth contributions of each circuit. The circuits can then be moved to a different path and the new effective bandwidth can be predicted. It is hoped that this technique can be used to come up with a more balanced network load across all of the trunks.

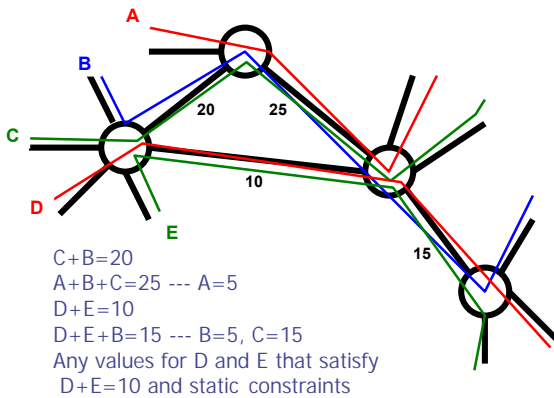


Fig. 2. Example network with 4 trunks with current throughput values of 20,25,10,15 and 5 circuits A, B, C, D.

$$\begin{matrix}
 & A & B & C & D & E \\
 \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} & \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} & = & \begin{bmatrix} 20 \\ 10 \\ 25 \\ 15 \end{bmatrix}
 \end{matrix}$$

Fig. 3. Network in fig. 2 represented as a matrix.

IV. DERIVING CONNECTION DATA

In order to derive connection data from trunk data, the contributions of each circuit must be determined from trunk statistics. Fig 2 shows a hypothetical network with trunk throughput shown in the links and five circuits (A,B,C,D,E). We must estimate the throughput contribution of each circuit given only trunk values.

For example, only paths E and D cross the section of weight 10 so we know that $E+D=10$. Paths C and B cross the section of weight 20 so $C+B=20$. Paths $A+B+C = 25$ and paths $D+B+E=15$. With these 4 equations and 5 unknowns we are able to solve for all but E and D. Although this simple example can be solved by observation, a mechanized approach must be developed for systems with 5,000 equations and 400,000 unknowns

The most plausible algorithm we've found is Gaussian elimination. Fig. 2 can be represented as a matrix with trunks as rows, circuits as columns and a 1 in the matrix if the circuit passes through the given trunk (fig. 3).

Through Gaussian elimination, we can create an upper diagonal matrix where the diagonal is all ones and anything below the diagonal is zero (fig. 4).

Using back substitution, we can entirely solve the system of equations, once the bottom row has been satisfied. In our small example, D and E can be set to 5 since $D + E = 10$. If there were static constraints (D is a T1 connection and E is a DS3) we could make better estimates for these unconstrained variables. We then proceed to previous row where $C - D - E = 5$. Since $D + E = 10$, we know that $C = 15$. Row 2 indicates that $B + C = 20$. Since $C = 15$, we know that $B = 5$. On the top row, $A+B+C=25$. Since $B+C = 20$ we know that $A = 5$.

This is the basic algorithm. The two keys are providing good estimates for unconstrained variables and resolving inconsistencies. Estimates are made using static constraints for the connections in connection with linear programming. This gives us a proven statistical basis for our estimates.

$$\begin{array}{ccccc}
 & A & B & C & D & E \\
 \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} & \begin{bmatrix} C \\ A \\ B \\ D \\ E \end{bmatrix} & = & \begin{bmatrix} 25 \\ 20 \\ 5 \\ 10 \end{bmatrix}
 \end{array}$$

Fig. 4. Network matrix after Gaussian Elimination.

For example, an OC-3c connection will have a much higher possible value than a T1 circuit. An inconsistency occurs when it is impossible to solve one of the rows in the matrix while satisfying the static constraints. For example, if a row required a 56k connection to be 64k we have an inconsistency. In order to deal with this problem, we backtrack to the point where the other parameters were set and change them by raising or lowering them as well as all other nodes on the line so that each solution still works. We then work the changes into each proceeding row and repeat this process until the system of equations is solved.

By setting the trunk values on the right to the contents of the traces and making each row a connection with a weight that depends on its speed, we can create trace files for each of the 400,000 connections. The size of the individual trace depends on the number of samples in the original trace. Since our current files contain 24 measurements * 4 days, we can generate trace files with 96 values. From these trace files we can compute the effective bandwidth and determine probabilities of overflow. We can also get an idea of the traffic flows that can be expected in the network.

Our next task is to use linear programming to determine the statistically best solutions to the problem. We will use the 5-minute peak values to help determine the contribution of each circuit to trunk statistics.

V. RESULTS

We performed some experiments to determine the potential of this approach in traffic grooming. In order to do this we picked a trunk from the trace data and moved a T1 circuit into the trunk and out of the trunk to determine the impact as measured by effective bandwidth. These statistics can be used to determine the appropriate balance between loading on different trunks.

Effective Bandwidth analysis provides information on the probability of packet loss given a specified number of buffers and line capacity. The roughly translates into queue length and delay for each circuit. If the aggregate probability of loss is minimized across the network, then delay will also

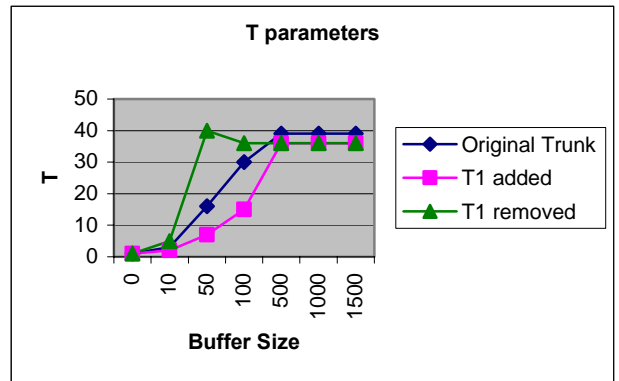


Fig. 5. The T parameter is the most probable period of buffer utilization prior to overflow. When a T1 line is removed, the T parameter is higher for a given Buffer Size showing that there is a longer buffer busy period prior to overflow.

be minimized [3]. Delay and jitter are usually seen as important values to optimize in making traffic grooming choices [4].

Fig 5 shows the impact of adding and removing a T1 line on the T parameter. When a T1 line is removed, the T parameter is higher for a given Buffer Size showing that there is a longer buffer busy period prior to overflow. For a buffer size of 50 packets, the original trunk had a T value of 16. When the T1 was removed, the T parameter increased to 40 showing that the busy period prior to overflow was higher due to decreased utilization. When a T1 was added, the T parameter dropped to 7 indicating that the busy period prior to overflow was lower, roughly indicating that the probability of loss is greater with the additional T1 line.

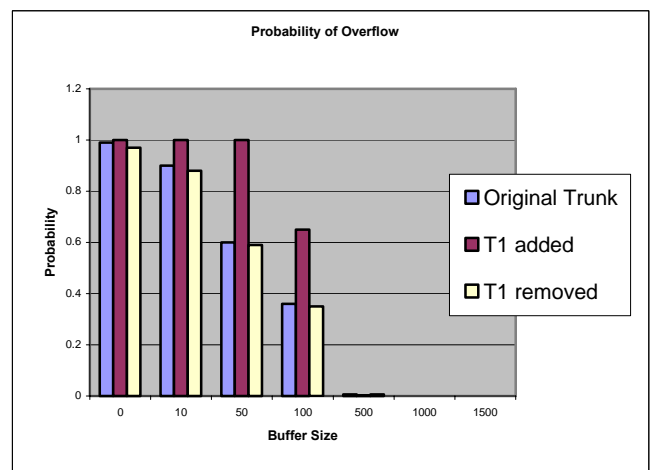


Fig. 6. The probability of overflow for different Buffer sizes as traffic is groomed.

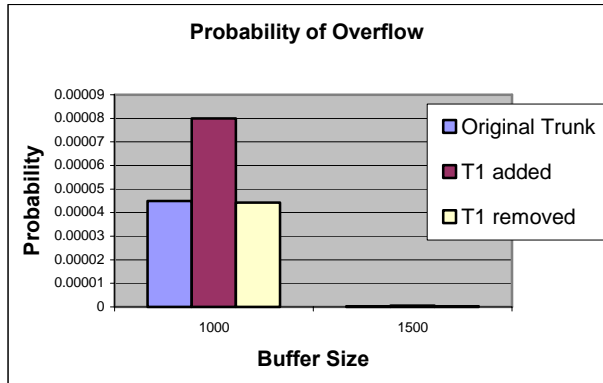


Fig. 7. Overflow probability for larger buffer sizes.

When a T1 line connection is added to a trunk, the probability of overflow increases. The effect is non-linear though, and is dependent on the buffer size and the burstiness of the traffic. Several examples of this effect are shown for different buffer sizes in Fig 6. (Fig 7 shows an expanded view of the overflow probability for larger buffer sizes.)

VI. CONCLUSION

Traffic Engineering and grooming are difficult to perform without tools to provide predicted results for potential changes. This research uses trace data that is readily available from trunk routers to determine the impact of routing changes on network balance. Large Deviation theory is used to examine the impact of changes on overflow probability, allowing a much more detailed analysis than mean values would provide.

Application of this research should allow traffic engineers to increase the utilization of their trunks without degrading Quality of Service for their customers.

ACKNOWLEDGMENT

This research is funded by AT&T Corporation. Thanks to Chris Chase who provided trace files and helpful suggestions.

REFERENCES

- [1] V.A. Siris, "Large deviation techniques for traffic engineering," <http://www.ics.forth.gr/netgroup/msa>, Last Access: 3/17/2000
- [2] C. Coutcoubetis, A. Dimakis, and G.D. Stamoulis, "Application and evaluation of large deviation techniques for traffic engineering in broadband networks," *In Proc. of ACM SIGMETRICS '98/PERFORMANCE '98*, Madison, Wisconsin, June 1998
- [3] Srinivas Vutukury, J.J. Garcie-Luna-Aceves, "A traffic engineering approach based on minimum-delay routing," *In Proc. of the Ninth International Conference on Computer Communications and Networks*. Las Vegas, Nevada, October 2000.
- [4] A. Feldmann, A. Greenberg, C. Lund, N. Reingold and J. Rexford "NetScope: Traffic engineering for IP networks," *In IEEE Network*, March/April 2000.
- [5] F. P. Kelly, "Notes on effective bandwidths," *Stochastic Networks: Theory and Applications Telecommunications Networks*, volume 4 of Royal Statistical Society Lecture Notes Series, pp. 141-168, Oxford University Press.