On The Impact of Aggregation on The Performance of Traffic Aware Routing*

A. Sridharan†	*S. Bhattacharyya, C. Diot*	*R. Guérin*†	*J. Jetcheva*‡*, N. Taft*
U. Pennsylvania	Sprint Labs	U. Pennsylvania	Sprint Labs
Philadelphia, PA	Burlingame, CA	Philadelphia, PA	Burlingame, CA

This paper investigates the impact of traffic aggregation on the performance of routing algorithms that incorporate traffic information. We focus on two issues. Firstly, we explore the relationship between *average* network performance and the coarseness (granularity) of traffic splitting across routes. Specifically, we are interested in how average network performance improves with our ability to distribute traffic arbitrarily across multiple paths. Secondly, we shift our attention from *average* to *short-term* performance, with again a focus on the impact of traffic granularity. In particular, we explore the relation between the level of traffic aggregation and its variability, which directly affects short-term routing performance. Our investigation relies on traffic traces collected from an operational network, and its results provide insight into the cost-performance trade-off associated with deploying "traffic aware" routing protocols.

1. Introduction

As IP networks become the life-line of business and commercial applications, the need for better service guarantees and improved performance are driving the deployment of service differentiation and traffic engineering in IP networks. Both typically involve *data path* mechanisms like packet classification etc. and *control path* mechanisms like signalling and extensions to routing protocols. In this paper we focus on routing, in particular, on evaluating the trade-off that exists between the added complexity and cost of the extensions required to accommodate traffic engineering, and the performance benefits it affords. We believe that such an understanding is important to decide whether or not *traffic aware* routing is worth deploying.

Traffic aware routing consists of protocols and algorithms that incorporate in the computation of routes the knowledge of both available network resources, e.g., available link bandwidth, and traffic requirements. The goal is some optimization of network usage or service guarantees. There have been many studies devoted to the design and evaluation of traffic aware routing algorithms and protocols, and they can be broadly classified in two categories. Those with a *traffic engineering* focus, and those that target an *on-demand* model (see [5,9,2] for examples of the first, and [4,1] and [8] despite its title for examples of the second). The focus of this paper is on the traffic engineering usage of routing, where a traffic matrix characterizing the bandwidth

*Authors emails: `ashwin@ee.upenn.edu, (supratik,cdiot)@sprintlabs.com, guerin@ee.upenn.edu, jorjeta@cs.cmu.edu, nina@sprintlabs.com`

†The work of these authors was supported in part by a grant from Sprint Labs and by NSF grants ANI99-06855 and ANI99-02943

‡This author is currently affiliated with CMU, but this work was done during an internship at Sprint Labs.

requirements between pairs of ingress-egress nodes is assumed known[4] and used to compute routes in an attempt to optimize network performance. The traffic information is typically obtained by *measuring* at ingress nodes the amount of traffic headed to various destinations. Our main goal is to gain a better understanding of the cost-benefit trade-off associated with incorporating traffic information into routing for traffic engineering purposes. The two main contributors to the cost of such traffic aware routing are: (1) matching traffic to routes (paths) so as to achieve "optimal" network performance; (2) updating routes to accommodate variations in traffic patterns or intensity.

The first cost can be further broken down into a traffic classification cost at the ingress router and a forwarding state cost in the core network both of which are affected by the granularity at which traffic needs to be split (classified) to achieve "optimal loads". We address this cost issue by evaluating the impact of *traffic granularity* on the performance of traffic aware routing. Traffic granularity refers to the level of traffic (and route) aggregation that constrains the load balancing ability of traffic aware routing. Coarse granularity aggregation bundles traffic into a small number of *"streams"* that must then be routed individually. This decreases classification cost but may affect routing performance by limiting its ability to arbitrarily split traffic across paths to achieve *optimal* link loads.

The second cost is related to the fact that traffic patterns, and hence traffic matrices, change over time. Frequent updating of routing tables to reflect this change is not desirable and is best kept as low as possible. To address the second cost issue regarding the frequency of routing updates, we study the influence of *time granularity* on routing performance. In particular, we assume that routes are computed on the basis of (long term) average traffic measures, and remain fixed for the duration of the experiments. We then evaluate network performance at different time scales, when traffic is distributed over this fixed set of routes. By varying the granularity of the time over which performance is measured, we want to capture the impact of the variations, from the measured long term averages, of traffic intensity over different time scales. The magnitude of those variations depends on both the length of the time interval over which we measure performance, and the granularity of the traffic streams that were available when computing routes. Our goal is to understand this relationship as traffic granularity changes, as well as investigate its impact on routing performance, and in particular short term performance, i.e., over shorter time intervals than the ones used by routing to compute optimal routes.

The interaction between traffic and time granularity in the context of traffic aware routing is as follows[5]. Optimal routing, ignoring for the time being granularity constraints, computes a set of "optimal paths and associated link loads" based on long term average traffic intensities, e.g., daily averages. Achieving these optimal link loads often requires a fine grain splitting of traffic into small streams, which in turn tends to increase traffic variability. As a result, fine grain splitting of traffic, besides increasing classification cost, this can produce higher short term traffic variability and, therefore, possibly more frequent transient link overloads (underloads). This could then lead to poorer short term network performance. Note that this will obviously depend on the extent to which the greater variablility of finer granularity streams also translates into more variable link loads. This will depend on the assignment of streams to paths, and this is one of the aspects we investigate. On the other hand, although using coarser traffic granularity, e.g., using supernets, may not allow us to optimally distribute traffic over links, it forces traffic to

[4] How to acquire this information is discussed in Section 2.

[5] This is described in more detail in Section 4.

remain aggregated. This may then result in smaller short term traffic fluctuations and, therefore, fewer periods of transient overload and better overall performance.

Understanding the extent to which these different parameters affect the trade-off between performance and cost in the context of traffic aware routing is the main goal of this paper. Our approach is based on evaluating the performance of two heuristic routing algorithms for "optimally" routing traffic given certain granularity constraints. We evaluate both short term and long term performance as we vary traffic granularity. There have been a number of previous studies devoted to the design and evaluation of traffic engineering protocols and algorithms [5,9,2] in the context of IP and MPLS networks that we assume here. They however represent a different setting from the one we consider in this paper since none of them focuses on the interaction of time and traffic granularity and their effect on routing performance.

The rest of this paper is structured as follows. Section 2, reviews the traffic measurement procedures we rely on to estimate the traffic matrices used in the paper. Section 3 focuses on the impact of traffic granularity on routing performance. Section 4 investigates how routing performance varies over time as a function of traffic granularity. In particular, it explores how traffic granularity affects the difference that exists between short term and long term performance. Finally, Section 5 summarizes the findings of the paper and points out potential extensions.

2. Traffic Measurement and Traffic Matrix Generation

The generation of traffic matrices for our study is based on measurements taken within Sprint's IP backbone. The Sprint IP backbone monitoring project [6] provided us with packet-level traces from a single POP (called the "monitored POP"). Optical splitters and IPMON systems [6] are used on each of the monitored links to capture the first 44 bytes of every IP packet traversing the link. These 44-byte headers include address and size information for each packet. In addition, each IP packet is time-stamped using a globally synchronized clock (GPS based). From this information, we can determine the number of bytes headed to other destinations across the Sprint network during any time interval. This data forms the basis for determining (i) traffic intensities between the monitored POP and the other 15 POPs in the Sprint backbone (see [6] for a general description of the overall topology and the internal architecture of a POP), and (ii) the variations of this traffic on different time scales and at different levels of granularity. In this work, we used a trace that is 800 minutes long and was collected at a peering link.

In our traffic matrix, each row represents an ingress POP and each column represents an egress POP. Thus in its simplest form an entry in the traffic matrix represents the total volume of traffic flowing from the ingress POP to the egress POP over the duration of the trace. We further break this data between pairs of POPs (in each direction) into multiple levles of time granularity and stream granularity and thus end up with a multidimensional traffic matrix.

The first step in building such a matrix from data consists of identifying the egress POP for each packet monitored at the ingress POP. We downloaded IBGP tables from the monitored POP at the same time that the traffic traces were gathered. Using information in the IBGP table in conjunction with detailed knowledge about the network topology, we used the method described in [11] to identify the egress POP for every packet in the trace.

Traffic granularity refers to the level of aggregation used to determine which packets are mapped onto a given stream. Packets between two POPs can be aggregated into streams according to different criteria. For example, packets can be mapped to streams based on their source and destination addresses, port numbers and protocol numbers. This would generate relatively fine granularity streams. Alternatively, coarser granularity streams can be obtained by aggregating packets on the basis of a common destination address prefix. By using multiple prefix masks of specific lengths, it is possible to vary the level of aggregation (i.e., stream granularity) over a pre-determined range. Because of its simplicity and the fact that it provides a systematic approach to varying granularity, we use this approach.

In particular, we use prefix lengths of 0, 4, 6, and 8. A prefix length of '0' corresponds to aggregating all the traffic between two POPs onto a single stream. A prefix length of 4 aggregates all packets with the same first 4 bits in the IP destination address field into a single stream. More levels of granularity are similarly defined using prefixes of length 6 and 8. (We use the notation p0, p4, p6 and p8 to refer to each of these granularity levels.) The use of longer prefix lengths leads to a larger number of individual streams which can then be routed individually. In general, as the length of the address prefix used to aggregate packets increases, so does the number of streams, and conversely, traffic granularity decreases.

At each traffic granularity, we also measured the bandwidth levels at different time granularity. The time granularity refers to the length of the measurement interval over which the average bandwidth per stream was calculated. Since our trace was 800 minutes long, an 800 minute average represents the coarsest time granularity. We also used 10 minute measurements to capture the short term variability of streams. Measuring the average bandwidth of streams on different time scales enables us to identify short-term fluctuations around longer-term averages. For example, the eighty ten-minute estimates obtained for each stream, show how the traffic associated with a given destination prefix varies around its 800 minutes average during those eighty consecutive ten-minute measurement intervals. Table 1 summarizes some typical numbers that this process yielded for our data. It indicates the range on the number of streams, and the range of average bandwidth per stream, for each of the granularity levels.

More specifically, the traffic matrix "row" obtained as a result of this process provides us with a set of bandwidth estimates of the form $B^k_{i,j}[n, m]$, where $i = 1, \ldots, 16$ identifies the egress node; $j \in \{0, 4, 6, 8\}$ indicates the prefix length used to separate traffic into finer granularity streams; and $k \in \{10, 800\}$ identifies the time granularity at which traffic is being measured. In particular, $B^{10}_{10,8}[., .]$ is itself a "matrix" of bandwidth estimates for traffic from the monitored POP to egress POP number 10. Each row in this matrix corresponds to a single stream associated with all the packets heading towards POP number 10 with the same 8-bit destination address prefix. Each column of this matrix corresponds to one of the eighty ten-minute bandwidth estimates. As a result, $B^{10}_{10,8}[5, 22]$ gives the average traffic intensity in the 22nd ten-minute monitoring interval for stream number 5 associated with an 8-bit destination address prefix for packets from the monitored POP to POP number 10.

Because monitoring equipment is expensive and very difficult to deploy in operational networks, we were only able to obtain traces from a single POP. This generates data for a single row of our traffic matrix. In order to populate the other 15 rows of the traffic matrix we combined coarse traffic information obtained for other POPs using SNMP, with the detailed structural information provided by the measurements done at the monitored POP. The remaining 15 rows of the traffic matrix were constructed by using the complete rows as a template and creating new streams by randomly selecting a stream from the original pool and applying random cyclic shifts of the time slots and small random perturbations to the stream. Once an entire row is completed, the intensity of the streams was scaled to match the average intensity of traffic ob-

granularity level	number of streams	bandwidth ranges (Mbps)
p0	1	[1-14]
p4	[5-10]	[0-8]
p6	[10-25]	[0-4]
p8	[25-64]	[0-4]

Table 1
Traffic Characteristics for Different Granularity (800mins Time Granularity).

tained using SNMP data. The extrapolation of the original row to obtain a complete matrix is done only for the finest granularity level. Coarser granularity levels are obtained by aggregating traffic from finer granularity levels.

In a separate study [11] more focused on POP dynamics, we analyzed data from other links (of different types) within this POP. We studied both geographic and temporal properties of the data at both the POP and link level, and concluded that the link included herein is typical in that it captures two of the more salient properties observed across traffic traces gathered from all other links. The first property is that the distribution of traffic to egress POPs is highly nonuniform. A few egress POPs sink a large amount of traffic while the rest sink small and medium amounts of traffic (where the ratios of large/medium and medium/small are roughly two). The second property is that the variability of the traffic measured over half-hour time slots and headed towards a given egress POP, greatly depends on the selected POP. These two properties are present in the measured row of our traffic matrix that is used to generate the remaining rows of the matrix, so that they are preserved throughout the matrix.

Due to the lack of monitoring information from other POPs, we believe the approach described above to be a reasonable alternative since it does not introduce any particular bias in the traffic matrix other than that already present in the original row. Furthermore, multiple traffic matrices were generated to reduce the probability of such events. We acknowledge that our approximations can to some extent affect the validity of our conclusions. However, we also believe that the use of actual measurements to generate partial traffic matrix data together with “reasonable” extrapolation techniques make for a traffic model that is realistic and that at a minimum provides meaningful insights.

3. On the Impact of Traffic Granularity

In the context of traffic engineering, the goal of traffic aware routing is to distribute network traffic, so as to optimize network performance, e.g., minimize average network delay or maximum link load. In this work, we use a delay-based “cost” function that relies on a standard M/M/1 queueing delay expression (see [3, Sections 5.4 to 5.7]). Specifically, the cost function $\mathcal{C}(\gamma)$ on which we rely, is of the form

$$\mathcal{C}(\gamma) = \frac{S}{\gamma} \sum_{l \in E} \frac{B_l}{C_l - B_l}, \tag{1}$$

where S corresponds to the average packet size, γ is the average total traffic offered to the network, E is the set of links in the network, C_l are the link capacities, and B_l are the average link loads achieved by routing. There are obviously many other cost functions that are possible, but in the context of traffic engineering, minimizing the delay experienced by user packets traversing the network is a reasonable target. In addition, other "typical" cost functions like minimizing maximum link load, are known, e.g., [12] to yield results similar to those of a minimum delay cost function. Most important, however, is the convex, non-linear nature of the cost curve. In the rest of the paper, we limit ourselves to this specific cost function.

In that context, we explore the impact of traffic granularity on the achievable network delay by analyzing the behavior of two heuristics, which are briefly described later in this section. We use heuristics since it is well-known, e.g., [7], that computing optimal routes while satisfying traffic granularity constraints is an NP-hard problem.

3.1. Heuristic Routing Algorithms

The two heuristics we describe next attempt to approach unconstrained, optimal performances, while accounting for the traffic granularity constraints imposed by the existence of individual unsplittable streams. We use them to evaluate the impact of traffic granularity on our ability to approach the performance of an optimal routing algorithm. Note that our primary intent is **not** to demonstrate the superior performance of one heuristic over the other, but instead to assess the impact of traffic granularity on routing performance, and hence provide motivations, or lack thereof, for moving towards finer granularity in the context of traffic aware routing. Due to lack of space, and given the fact that our main focus is on the impact of *granularity*, we only give a brief description of the algorithms. The interested reader is referred to [10] for a more detailed discussion.

3.1.1. Heuristic 1

The first heuristic is a simple greedy method that adds streams one at a time, while each time selecting a path that minimizes delay. This heuristic is inspired from methods used in an "on-demand" model of traffic aware routing where requests are generated dynamically and need to be routed one at at time. The main variations for this first heuristic are in terms of the *order* in which individual streams are routed, e.g., in ascending, descending, or random orders in terms of their traffic intensity. In our experiments we found that sorting the streams in descending order gives the best performance (for both heuristics) among the three orderings. Henceforth we shall implicitly assume such an ordering.

A key feature of this heuristic is that it is independent of the actual traffic offered between different nodes. This clearly makes for greater simplicity, but also points to a limitation of the approach, as it does not exploit key information that is available to the routing algorithm.

3.1.2. Heuristic 2

The second heuristic takes into account the knowledge of the total traffic matrix, and in particular the output (set of paths and associated loads) generated by an optimal routing algorithm, that ignores granularity constraints imposed by streams. The motivation for using this information is that it represents the best performance achievable.We give a brief outline of the heuristic and refer the reader to [10] for more details.

The optimal routing problem can be set up as a straightforward multi-commodity flow problem [10] which is solved using PPRN[6]. The heuristic proceeds by assigning streams to (optimal)

[6] The PPRN package is available at `http://www-eio.upc.es/~jcastro/pprn.html` and was developed at the Statistics and Operations Research Dept. at Universitat Politecnica de Catalunya, Barcelona, Spain, by Jordi

paths in a manner that attempts to get as close as possible to the link loads achieved by the optimal algorithm. This relies on a two phase procedure. The first phase involves routing streams one-by-one, as in the first heuristic, but on a network with (fictitious) link capacities initially set equal to the desired optimal loads. As each stream is added, it is routed on the path that yields the minimum "delay" given the assumed link capacities. In order to take the granular nature of traffic into consideration we relax the "fictitious" capacity constraint while routing over this network, by allowing a stream to be routed if it does not exceed the capacity of any link on that path by more than a factor of $(1 + \Delta)$. The parameter Δ controls the amount by which the link constraint is violated. In the second phase, any stream for which no feasible[7] path was found during the first phase, is routed using a standard minimum delay algorithm, but using now the actual link capacities together with the link loads that resulted from the first phase.

3.2. Performance Evaluation

In this section, we investigate the impact of traffic granularity on routing performance, where performance is measured in terms of the total network average delay computed using the long-term average load. First, we compare the performance of the two heuristics against the optimal routing algorithm, while assuming the finest granularity available from our traffic measurements. i.e., the use of an address prefix mask of length 8. This provides some insight into the differences in performance that exist between heuristics that incorporate knowledge of the traffic matrix (Heuristic 2) and those that don't (Heuristic 1). Unless specified, the stream ordering used for both the heuristics was in decreasing fashion, i.e., larger streams were routed first.

In order to compare the performance of the heuristics, we scaled the *average* intensity of a randomly selected set of source-destination pairs in the Traffic Matrix to create hot spots. We show the performance of the two heuristics and of the optimal routing algorithm in Figure 1. Note that while the two heuristics were constrained to routing individually streams generated from length 8 prefixes, the optimal algorithm did not follow any such constraint. As can be seen, Heuristic 2 outperforms Heuristic 1 and closely follows the optimal till the 'knee', thereafter the granularity of the streams forces a different sub-optimum allocation. The main reason for the inability of both heuristics to approach the optimal performance, even at the finest granularity level (p8) can be found in Table 1, which shows that large bandwidth streams remain. Those large bandwidth streams affect the load balancing ability of any algorithm. This factor notwithstanding, the better performance of heuristic 2 illustrates the fact that using the information available from the traffic matrix can yield better performance. In what follows, we explore further the impact that traffic granularity has on routing performance.

We start this comparison by using the different levels of traffic granularity generated from our traffic measurements. Specifically, traffic can be aggregated onto streams using prefix of lengths 0, 4, 6, or 8, which, as shown in Table 1, translate into average number of streams ranging from 1 (prefix length of 0) to 64 (prefix length of 8). Clearly, one can expect a larger number of streams to result in better performance, as it gives routing more flexibility in assigning traffic to paths. The aspect we want to explore is the *evolution* of routing performance as the number of streams varies, i.e., what is the magnitude of performance improvements as the number

Castro and Narcs Nabona for solving multi-commodity network flow problems with linear and non-linear cost functions.

[7]Note that feasibility is only in the context of the fictitious link capacities, and a feasible path can typically be found when using the *real* link capacities.

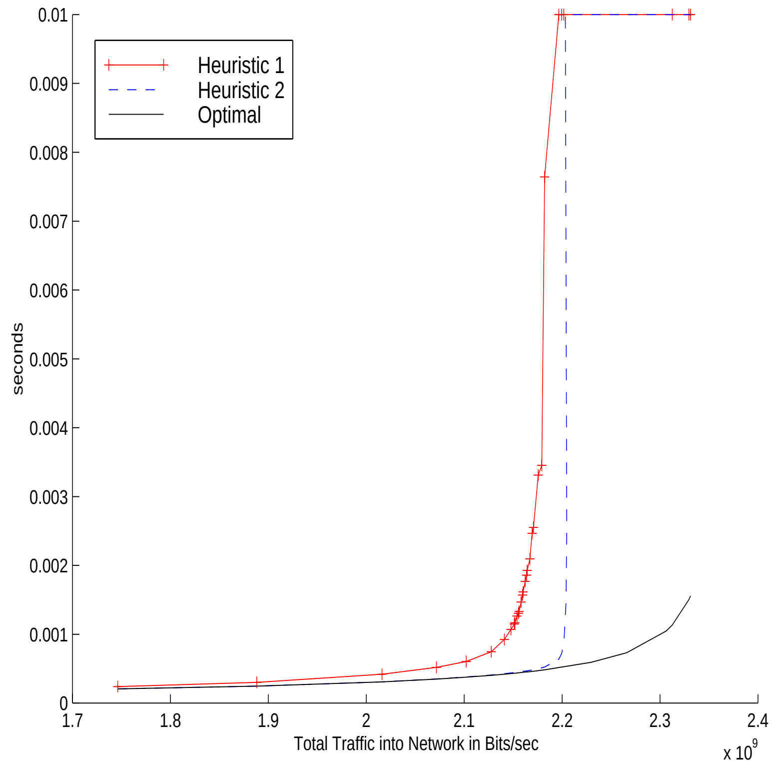


Figure 1. Optimal Routing vs Routing Under Granularity Constraints.

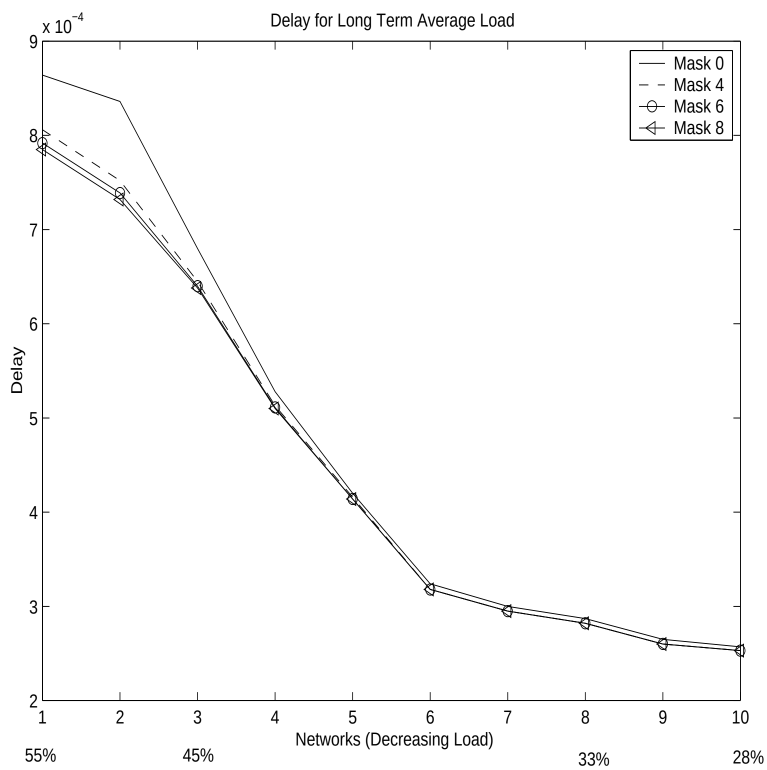


Figure 2. On the Impact of Traffic Granularity. (Delay Computed for Long Term Average Load).

Granularity	Average No. of Distinct Paths	Max. No. of Distinct Paths
Mask 0 ($p0$)	1	1
Mask 4 ($p4$)	2	7
Mask 6 ($p6$)	2.3	12
Mask 8 ($p8$)	2.5	16

Table 2
Statistics for The Number of Distinct Paths Used at Each Granularity Level for Topology 1 (Most Heavily Loaded).

of streams varies. We proceeded to do so, by computing the mean delay averaged over 30 traffic matrices for each granularity level, each of which was routed on 10 networks with the same topology but increasing capacities (decreasing loads). Each traffic matrix was generated using the method described in Section 2 and the total average intensities for each S-D pair (for each matrix) were scaled to the values obtained using SNMP data to obtain a fair comparison. Different load values were generated by scaling down link capacities.

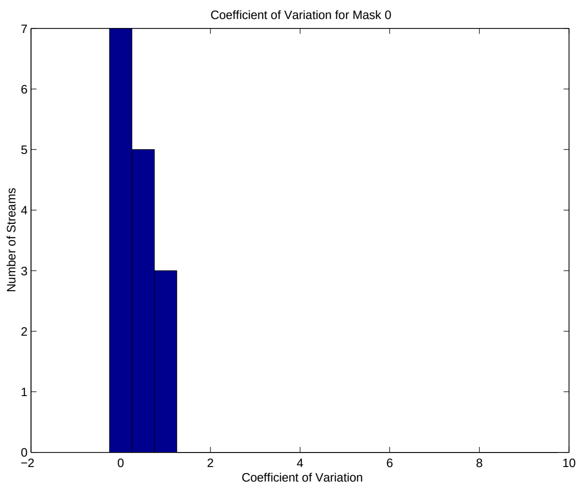
The results are plotted in Figure 2, which shows the average network wide delay as a function of network sizing. We immediately see that a large fraction of the performance gain is achieved when going from a prefix length of 0 to one of 4, and subsequent improvements are much more modest. This stands to reason, since the number of paths through the network is limited, so that beyond a certain level, additional streams are still routed over the same set of paths. We justify this claim by providing, in Table 2, statistics regarding the actual number of distinct paths used for each granularity level. We notice that the average number of paths used doubles from Mask 0 to Mask 4, but increases slowly thereafter indicating the limited availability of paths for all the source-destination pairs. We also note that the improvement in performance decreases rapidly as we increase the capacity of the network, which is not unexpected.

4. On the Impact of Time Granularity

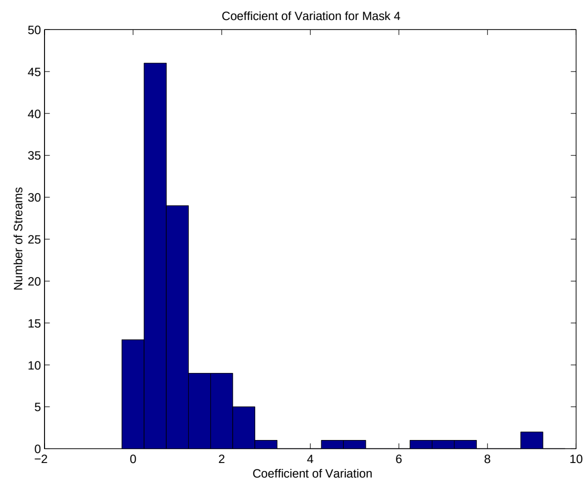
The purpose of this section is to explore another dimension of how performance is affected by the granularity at which routing is performed. Specifically, we saw in Section 3 that finer granularity leads to a lower *average* load on the links, which translates into lower *average* delay, that is, better long term performance. However, performance measured over finer time scales can be significantly different. The traffic, and hence link loads measured at finer time scales fluctuate around the long term average values. This combined with the non-linear nature of the delay function can result in significantly different performance as compared to that obtained using long term average load. This is because fluctuations at the higher end of the curve contribute much more to the delay than those at the lower end due to the non-linear convex nature of the curve. Hence, both the mean load *and* the variability of the traffic determine how routing performance, measured over short time intervals (10 minutes), differs from its expected value based on the long term average load (800 minutes). The goal of this section is to shed some light on how these different factors interact and ultimately affect routing performance. Especially since, as we show in the next sub-section, splitting traffic into finer streams *increases* their variability. This can potentially offset the advantage of a lower operating link load, and result in a higher delay over smaller time scales as compared to that obtained with coarser granularity streams.

4.1. Evaluating the Impact of Time Variability

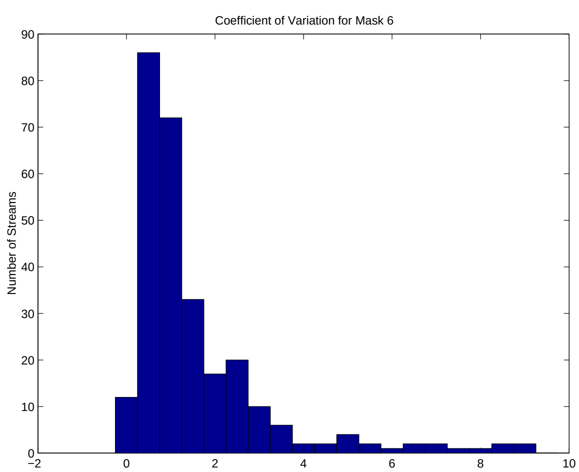
In order to investigate this potential trade-off between improved average performance and greater traffic variability, we first proceed to evaluate how traffic granularity affects its variability. We do so by computing the coefficient of variation of the traffic intensity of individual streams for different levels of granularity over 80 10-minute-interval measurement samples. The results are summarized in Figure 3, which clearly indicates that as the number of streams increases, i.e., from Mask 0($p0$) to Mask 8 ($p8$), so does the variability of individual streams.



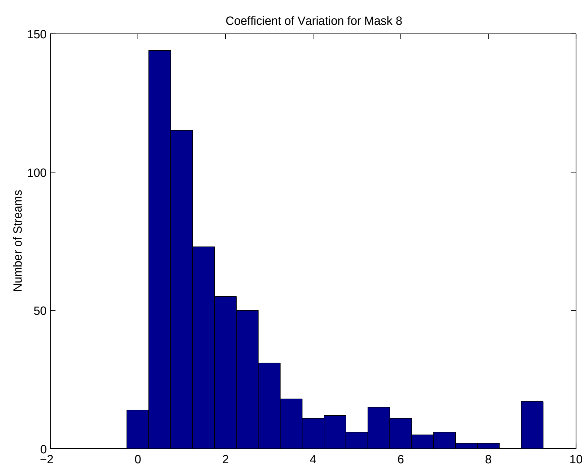
(a) $(P0)$



(b) $(P4)$



(c) $(P6)$



(d) $(P8)$

Figure 3. Relation Between Traffic Variability and Stream Granularity.

This implies that the potential benefits of improved average performance may be offset by the impact of this greater stream variability which could result in higher link load variations, if it also translates into greater variability at the *link* level, i.e., after streams have been assigned to paths. Note that greater stream variability need not necessarily imply greater *link* traffic variability. For example, routing all streams from the same S-D pair on the same path would obviously result in (link) traffic characteristics that are identical to those observed without first splitting the traffic into streams. However, such a scenario is unlikely, as the load balancing decisions of routing will typically result in streams from the same S-D pair being routed over a distinct set of paths. In that context, it is unclear how aggregating streams from different S-D pairs will affect the variability of link traffic.

In order to asses the relative effect of these competing factors, we evaluated the network delay averaged over all the 80 10-minute measurement intervals, for the 30 traffic matrices routed over the 10 networks used in Figure 2. Recall that different network loads were achieved by scaling link capacities and not traffic matrices, hence, preserving temporal traffic characteristics across experiments. The results are shown in Figure 4.

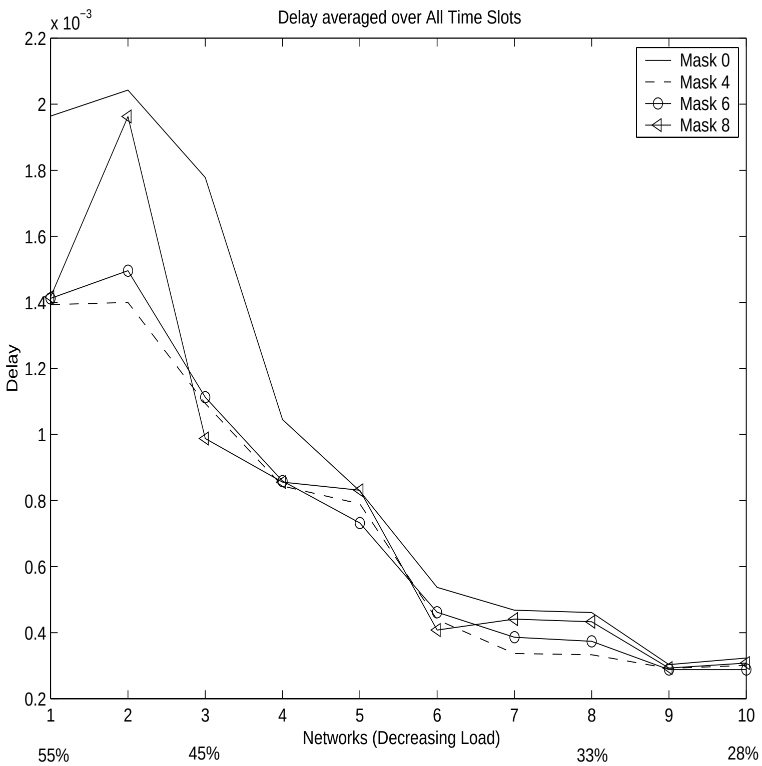


Figure 4. Delay Averaged Over Time Slots.

From comparing Figure 4 with Figure 2, we immediately observe that the benefits of lower average link loads and, therefore, delay, do not always translate into visibly better performance when time variability is taken into account. The figure shows the delay averaged over the 80 10-minute slots for the different levels of traffic granularity under consideration, i.e., masks 0, 4,6, and 8. We observe that when routing is limited to a single stream (mask 0), performance is poor even on the shorter 10-minute time scale. This is because although link traffic may exhibit smaller short term load variations, the higher average link loads amplify those variations when it comes to delay because of the non-linear nature of the curve. As traffic granularity decreases, i.e., to masks 4 or 6, we observe that the resulting lower average link loads manage to also improve short term performance. This is because, despite the potentially larger short term traffic fluctuations, the lower overall average link loads sufficiently dampen the impact of those variations on short term delay. However, this improvement does not readily extend as traffic granularity decreases further. Specifically, we see that routing using mask 8 streams often results in worse average short term delays than when masks 4 or 6 are used. This is because, as seen from Figure 2, the improvement in average link loads that mask 8 streams afford, is marginal compared to what is achievable with mask 4 or 6. On the other hand, mask 8 streams exhibit higher short term traffic fluctuations that result in degraded average short term delays. Note that this behavior is observed even though, as shown in Table 2, the number of paths used with mask 8 is similar to what is used with masks 4 and 6. This indicates that although the number of paths is similar, the assignment of traffic (streams) to them is different.

The findings of Figure 4 indicate the existence of a trade-off between short-term and long-term performance, when decreasing traffic granularity to achieve lower average link loads. The figure suggests using the coarsest possible traffic granularity that achieves a “significant” decrease in average link loads, e.g., a mask of 4 in the current environment. Further reductions in traffic granularity improve average loads only marginally, and the greater short term traffic variability they induce often becomes the dominant effect, worsening short term performance.

5. Conclusion

We have investigated a new aspect of traffic aware routing in IP networks, namely, the impact of traffic and time granularity on routing performance. Our performance measure was based on a traditional delay based cost function, but we expect comparable findings with other cost functions. The investigation was carried out using actual traffic and flow data collected on an operational Internet backbone.

The main contributions of the paper consist of: (1) quantifying the impact of traffic granularity on routing performance, and in particular that the bulk of the improvement occurs with a relatively small number of streams; (2) designing and evaluating a routing heuristic (Heuristic 2) that approximates the performance of “optimal” routing reasonably well, while incorporating traffic granularity constraints; (3) observing that while finer granularity routing improves average performance, this does not always carry over to smaller time scales, where the greater variability of finer grain traffic can offset most of the resulting performance improvements.

This has been a preliminary investigation into the potential benefits and trade-offs related to traffic aggregation. We are currently collecting more network traces that span over a few days rather than hours to further investigate this topic and base our findings on a firmer footing. Specifically, we intend to : (1) verify the stability of the traffic matrix (which affects routing computation) over a large enough time scale ; (2) Explore the trade-off between long-term and short-term performance over a sufficiently big data set; (3) extend the aggregation scheme to use routing prefixes which is a practical and deployable alternative.

Acknowledgements

The authors would like to thank Jordi Castro for his help in using the PPRN software package and applying it to the relatively large optimization problems of this paper.

REFERENCES

1. G. Apostolopoulos, R. Guérin, S. Kamat, and S. Tripathi. Quality-of-service based routing: A performance perspective. In *Proceedings of SIGCOMM*, pages 17–28, Vancouver, Ontario, CANADA, September 1998.
2. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic engineering over MPLS. Request For Comments (Informational) RFC 2702, Internet Engineering Task Force, September 1999.
3. D. Bertsekas and R. G. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1992.
4. S. Chen and K Nahrstedt. An overview of quality-of-service routing for next generation high-speed networks: Problems and solutions. *IEEE Network Magazine, Special Issue on Transmission and Distribution of Digital Video*, 12(6):64–79, November/December 1998.
5. B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings of INFOCOM'2000*, Tel Aviv, Israel, March 2000.
6. C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owezarski, D. Papagiannaki, and F. Tobagi. Design and deployment of a passive monitoring infrastructure. In *Passive and Active Measurement Workshop*, Amsterdam, April 2001.
7. B. Gavish and S. Hantler. An algorithm for optimal route selection in SNA networks. *IEEE Trans. Commun.*, COM-31(10), October 1983.
8. M. Kodialam and T.V. Lakshman. Minimum interference routing with applications to MPLS traffic engineering. In *Proceedings of INFOCOM'2000*, Tel Aviv, Israel, March 2000.
9. T. Li and Y. Rekhter. A provider architecture for differentiated services and traffic engineering (PASTE). Request For Comments (Informational) RFC 2430, Internet Engineering Task Force, October 1998.
10. A. Sridharan, S. Bhattacharyya, C. Diot, R. Guérin, J. Jetcheva, and N. Taft. On the impact of aggregation on the performance of traffic aware routing. Technical report, University of Pennsylvania, July 2000. Available at http://www.seas.upenn.edu/˜guerin.
11. N. Taft, S. Bhattacharyya, J. Jetcheva, and C. Diot. Understanding traffic dynamics at a backbone POP. Technical Report TR01-ATL-020201, Sprint Labs, February 2001. Available at http://www.sprintlabs.com/PEOPLE/nina/papers.html.
12. K. S. Vastola. A numerical study of two measures of delay for network routing. M.S. Thesis, U. Illinois, Dept. Elec. Eng., Urbana, IL, 1979.