The Revised ARPANET Routing Metric

Atul Khanna John Zinky

BBN Communications Corporation
150 CambridgePark Drive
Cambridge, MA 02140

Abstract

The ARPANET routing metric was revised in July 1987, resulting in substantial performance improvements, especially in terms of user delay and effective network capacity. These revisions only affect the individual link costs (or metrics) on which the PSN (packet switching node) bases its routing decisions. They do not affect the SPF ("shortest path first") algorithm employed to compute routes (installed in May 1979). The previous link metric was packet delay averaged over a ten second interval, which performed effectively under light-to-moderate traffic conditions. However, in heavily loaded networks it led to routing instabilities and wasted link and processor bandwidth.

The revised metric constitutes a move away from the strict delay metric: it acts similar to a delay-based metric under lightly loads and to a capacity-based metric under heavy loads. It will not always result in shortest-delay paths. Since the delay metric produced shortest-delay paths only under conditions of light loading, the revised metric involves giving up the guarantee of shortest-delay paths under light traffic conditions for the sake of vastly improved performance under heavy traffic conditions.

1 Introduction

Routing in the ARPANET has always been adaptive, i.e, routing where decisions are based on periodic measurements of appropriate network characteristics. In the past, these measurements have been related to the delay characteristics of links (we use the term link to refer to the simplex communication medium between two PSNs). Since the ARPANET's inception, there have been two basic routing algorithms and several associated link metrics.

We begin this paper with a brief history of ARPANET routing algorithms, followed by a description of the most recent routing scheme and its shortcomings. We then describe and explain the revised link metric, designed to combat some of these shortcomings. We analyze the behavior of the metric and compare it to the delay metric and min-hop routing. Finally, we present a brief account of the metric's success in the ARPANET.

While many sections of this paper focus on the revised metric from the perspective of the ARPANET, the metric is applicable to any network. In fact, it has been successfully deployed in several major networks, including the MILNET.

2 Routing in the ARPANET

The ARPANET PSNs make independent routing decisions about each packet they forward. Single path routing is used, i.e., at any given moment, all packets destined for a particular destination PSN always use the same outgoing link. Both the first and second algorithms used in the ARPANET computed routes with the objective of minimizing individual packet delay through the network. Note that in schemes where each packet at each PSN is routed independently, it is very important for PSNs to have consistent routes, failing which there is always the potential for formation of long-term routing loops.

2.1 Original Algorithm

The first ARPANET routing algorithm [8], designed in 1969, was a distributed version of the Bellman-Ford shortest path algorithm [5]. Each node maintained a table of its estimated shortest distance to all other nodes. These tables were exchanged between neighbors every 2/3 seconds. Each node updated its distance estimates periodically, based on infor-

mation received from neighbors and its own estimate of the distance to each of its neighbors. This latter quantity, the link metric, was simply the instantaneous queue length at the moment of updating plus a fixed constant. Thus, shortest paths to all destinations were computed by a process of repeated minimization.

This distributed version of the Bellman-Ford algorithm and the link metric it used suffered from several limitations, some of which we mention here (a more complete account can be found in [8, 10]). The link metric, which was an instantaneous sample rather than an average, was a poor indicator of expected delay on a link as the quantity being sampled fluctuated fairly rapidly at all traffic levels. Any routing algorithm would have produced potentially suboptimal routes with such a metric. In addition, while the distributed Bellman-Ford algorithm is guaranteed to converge to the shortest paths relative to a set of static costs, it resulted in the formation of persistent loops in the face of the rapidly changing link metric. Finally, the volatile nature of the metric itself resulted in routing oscillations (the positive constant added to the metric helped to alleviate this effect). Thus the the original ARPANET routing scheme produced routes that were potentially unstable and far from optimal.

2.2 SPF Algorithm

Many of the shortcomings of the algorithm above were addressed by the SPF algorithm installed in the ARPANET in May, 1979 [10, 13, 9, 11, 12]. In particular, the new algorithm resulted in loop-free and more stable routes, and used an averaged delay measurement as link cost. We now describe relevant aspects of this algorithm.

Terminology

A few words regarding terminology are appropriate here. SPF (Shortest Path First) refers to the algorithm employed by the PSN to calculate its routes to other PSNs. The SPF algorithm determines the relative appeal of network links based on a cost associated with each link. These link costs are disseminated through the network via routing updates. Note that unlike the first algorithm, routing updates contain only link cost information; no other routing information is disseminated through the network.

The term delay-SPF (D-SPF) refers to the case where routes are computed using SPF and the link metric is measured delay. The modifications to routing described in section 4 affect only the link cost. They do not involve changes to the route computation algorithm, which continues to be SPF. The term Hop-Normalized SPF (HN-SPF) refers to this case where routes are computed using SPF and the revised link metric. Section 4 will discuss HN-SPF in detail.

Route Computation

Each node or PSN in a network has full knowledge of the topology of the network, i.e., it knows about all nodes and links. In addition, it knows the cost associated with each link.

A node constructs paths to all other nodes in the network according to the Shortest Path First (SPF) algorithm due to Dijkstra [4]. The routing tree is potentially recomputed each time new routing information (about the topology or link costs) is received. The algorithm in the PSN is an incremental SPF algorithm that attempts to perform only incremental adjustments necessitated by a link cost change, e.g., if a routing update reports an *increase* in the cost for a link not in the tree, the algorithm does not recompute any part of the tree.

Link Metric

The D-SPF link cost is computed as follows. For every packet the PSN receives and forwards, it measures queueing and processing delay to which it adds tabled values of transmission and propagation delay. For each of its outgoing links, it averages this total delay over a ten-second period and compares the average to the last reported value for the link. If the difference passes a significance criterion, a routing update is generated for distribution to the rest of the network (see [13] for details regarding the distribution mechanism).

For stability reasons, the link cost has a lower bound called the bias. This bias term is a function of line speed and effectively serves to prevent an idle line from reporting a zero delay value. See [11, 3] for an analysis of the relationship between the bias term and routing stability.

As part of a scheme to ensure that every node has accurate data on which to base its SPF computation, the significance criterion gets adjusted downward each time it is not satisfied. This is done in such a way that the maximum time between routing updates for each PSN is 50 seconds. Thus, even if a PSN has no local link cost or topology changes, it will generate a routing update every 50 seconds for reasons of reliability.

3 Limitations of the Delay Metric

In this section we discuss the problems associated with D-SPF. We show how it results in routing oscillations and explain the ill-effects of such oscillations.

With D-SPF, routing decisions are based on actual measured link delay values which were calculated during a pre-

vious interval and propagated via routing updates. The underlying assumption here is that the measured packet delay on a link is a good predictor of the link delay encountered after all nodes re-route their traffic based on this reported delay. Thus, it is an effective routing mechanism only if there is some correlation between the reported values and those actually experienced after re-routing. With D-SPF, the correlation between successive measured delays is high when a network is lightly loaded. However, the predictive value of measured delays declines sharply under heavy traffic loads.

3.1 Light Network Traffic Loading

The packet delay measured on a link has three components: queueing delay, transmission delay, and propagation delay. Of these three, only queueing delay depends on the utilization of the link. Under conditions of light loading, packet queueing is minimal, and thus queueing delay is largely negligible. Furthermore, because a change in routing tends to result in routing changes for small volumes of traffic, link delays continue to remain dominated by the propagation and transmission delay terms. Thus, under conditions of light loading, the reported delay values are fairly good predictors of the delay encountered after re-routing and in fact routing tends to be fairly independent of traffic conditions.

Under moderate loading, queueing delay is no longer negligible. However, routing changes result in moderate traffic shifts; consequently, queueing delays don't change too drastically and the delay metric remains a useful predictor of expected delay.

3.2 Heavy Network Traffic Loading

Whereas under low traffic conditions queueing delay is minimal, on a heavily-loaded network queueing delay can exert a significant influence on the delay computed for a link. Three factors in particular contribute to the ineffective performance of D-SPF under heavy traffic conditions:

1. The range of permissible delay values is too wide. For example, in a network consisting solely of 56 kb/s lines a highly loaded line can appear 20 times less attractive than a lightly loaded one, while in a network with both 9.6 and 56 kb/s lines a heavily loaded 9.6 kb/s line can appear 127 times less attractive than a lightly loaded 56 kb/s line.

   A range this wide is problematic in that a link reporting a high value can look unattractive to all sources. For example, it is conceivable that a 127-hop path can look more attractive than a single-hop path. *While it is certainly the case that high delay links should be avoided, they shouldn't necessarily be avoided to the point that they aren't used by any active routes.*

2. There is no limit on the variation of reported delays in successive updates for a particular link.

3. All the nodes in a network adjust their routes (and thus their flows) in response to a link metric update simultaneously. This is not the consequence of some explicit synchronization scheme, but rather of the fact that routing update processing is a high priority process within the PSN. Note that routing updates are generated at intervals on the order of tens of seconds, while network packet transit times are typically much less than a second.

The following example illustrates how these factors combine to cause oscillatory routing behavior in heavily loaded networks.

3.3 Routing Oscillations

Consider a network consisting of two regions which are connected by two links, A and B, with the same propagation delay and bandwidth (see figure 1). All routes originating in one region and destined for the other must use one of these links. Assume that most of these routes happen to be using link A at a given instant. Packet queueing delays on this link will be significant and a high delay value will be reported. Because of the wide range of allowable delay values and the lack of any restrictions on their movement, it is likely that this value will be high enough to make link B the preferred route for most, if not all, inter-region traffic. Because all nodes adjust their routes simultaneously, most or all inter-region traffic shifts at the same time to link B. Now the roles of links A and B have reversed, and the next measurement interval will yield a high reported delay on link B and a low reported delay on link A. This process of links A and B alternating (instead of cooperating) as traffic carriers will continue to feed on itself until the traffic volumes subside.

The behavior of D-SPF under heavy traffic conditions described in the preceding paragraph can occur in almost any network topology and is undesirable for several reasons:

1. A significant portion of available network bandwidth is unused. In particular, at any given moment, it is likely that some network links will be over-utilized while others are under-utilized. In the example above, only 50% of the available inter-region bandwidth is available during a particular interval.

2. The over-utilization of subnet links can lead to the spread of congestion within the network.

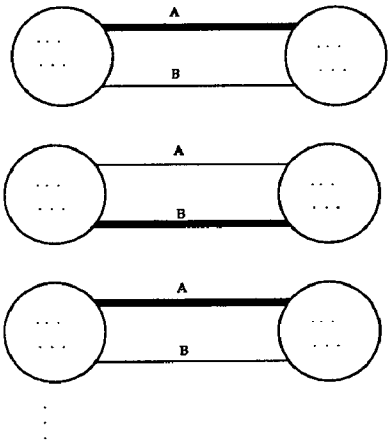

Figure 1: Routing Oscillations

3. For a given node-to-node traffic flow, the route taken through the network could oscillate between a short-hop path and a long-hop path. Some of this use of longer paths could be unnecessary and thus constitute a waste of network bandwidth.

4. The large swings in reported values of delay result in the frequent satisfaction of the update generation threshold criterion. This leads to a greater number of routing updates on the network, leading to increased consumption of link bandwidth by network control traffic.

5. Because these updates typically contain values that are significantly different from previously reported values, the route-computation module of the PSN is invoked more often, resulting in increased PSN CPU utilization.

It should be noted that the performance of D-SPF was far superior to that of the Bellman-Ford algorithm. It was only under conditions of heavy utilization that the unstable behavior described above occurred.

4 The Revised Link Metric

The key to understanding SPF is to normalize the link cost in terms of *hops*. When a link reports a cost, the cost is *relative* to the costs of alternate links. For example, when a link reports a cost of 91 units while the rest of the links in the network report 30 units, the implication is that an alternate path with 2 *additional* hops should be used before using that link. When there are many alternate paths, most of the routes will move off this link. An interpretation which normalizes the reported cost by dividing it by the *ambient* cost of alternate links takes into account the effect of the reported cost relative to other links.

The general interpretation of the delay metric is as an *absolute* measure of path length. When a PSN chooses the path, it does so in greedy fashion and takes the shortest path available without regard to how its choice will affect other users. When traffic is light, this approach works fine. When traffic levels increase, however, these greedy routes interfere with each other. Under heavy loads, the goal of routing should change to give the *average* route a good path instead of attempting to give *all* routes the best path. Some of the routes should be diverted to longer paths so that remaining routes can make effective use of the overloaded link. The diverted routes should be those that have alternate paths which are only slightly longer.

We designed several modifications to the delay metric to combat many of the limitations of D-SPF discussed in the section 3. These modifications perform some processing on the delay value measured by the PSN, so that the value reported in the routing update is no longer delay, but rather a function of delay. The reported cost is normalized to take into account how the network will respond to it. As will be shown in section 5, the network is extremely responsive to changes in the reported cost. Because of this, the revised metric limits the relative value so that the largest value it can report is only two additional hops in a homogeneous network. In addition, the dynamic behavior of SPF has been changed so that routes are shed from an oversubscribed link in a gradual manner. Routes with slightly longer alternate paths are shed first. If this does not relieve the oversubscription, then progressively longer alternate paths are tried in successive routing periods.

We will now describe the implementation of the revised metric. First we will discuss how the new software fits within the PSN architecture. Next we will describe how the metric was normalized and how its dynamic behavior was changed. We will also show the specific normalization used in the ARPANET and MILNET, which is tuned to handle heterogeneous line types. As indicated earlier, the term Hop Normalized SPF (HN-SPF) refers to the case where the SPF algorithm computes routes based on the revised link metric. We use the term HNM (HN-SPF Module) to refer to the module which computes the revised metric.

4.1 Overview of the Revised Metric

Figure 2 shows the modifications relative to the existing routing update code. The HN-SPF module takes the value of the measured delay and transforms its value. The new value is passed on to the flooding subsystem which disseminates

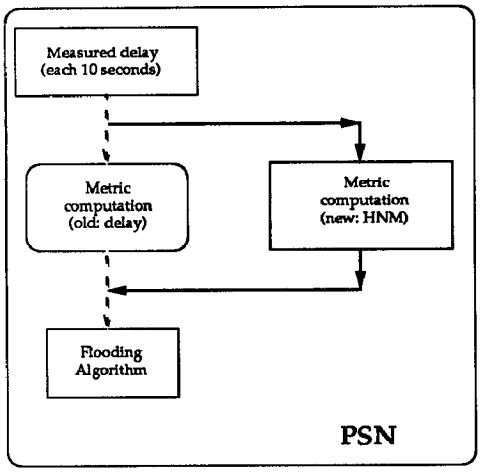

Figure 2: Relationship between the HNM and the Routing Update Code

the new link cost to all the other nodes in the network. Thus the HN-SPF module fits easily into the existing network software; no other modifications to the routing mechanism were made.

Pseudocode for this transformation is given in figure 3. The value of delay is first transformed into an estimate of the link utilization. A simple M/M/1 queueing model is used with the service time being the network-wide average packet size (600 bits/packet) divided by the trunk's bandwidth. The result is then averaged with previous utilization estimates using a recursive filter. Next, the average utilization goes through a linear transformation to normalize the metric. The change in reported cost from one update to the next is limited both in how little and much the cost can change. The final component of the HN-SPF Module enforces absolute limits on the value of the metric which is part of the normalization process.

The transformations are parameterized based on the link's line-type. Each logical link between nodes is assigned a line-type based on the combined bandwidth of the trunks making up the link. Up to eight different line-types are allowed, each one corresponding to a variety of line configurations. Also a history of past behavior is kept for each link. This information is held in the averaging filter for utilization and the cost reported in the last routing update. These modifications are described in exact detail in [7].

We should state here that our changes were restricted to the metric for reasons related to simplicity of implementation. Changes to the overall routing scheme were beyond the scope of our work. Our goals were limited to damping

```
Function HN-SPF(Measured Delay, Line Type) returns Reported Cost
 Sample Utilization = Delay to Utilization[Measured Delay]
 Average Utilization = .5 * Sample Utilization + .5 * Last Average
  Last Average = Average Utilization  (stored for each link)
 Raw Cost = Slope[Line Type] * Average Utilization + Offset[Line Type]
 Limited Cost = Limit Movement(Raw Cost, Last Reported, Line Type)
 Revised Cost = Clip(Limited Cost,Maximum[Line Type],Minimum[Line Type])
  Last Reported = Revised Cost  (stored for each link)
Return(Revised Cost)
```

Figure 3: HN-SPF Pseudocode

routing oscillations and reducing routing overhead on link bandwidth and PSN CPU. It should be noted that not all of the problems associated with D-SPF can be solved by modifying the metric. For example, consider the problem of simultaneous route re-computation. The current packet-forwarding scheme takes advantage of the fact that shortest-paths are hereditary (every subpath of a shortest path is also a shortest path) and that all PSNs have a consistent view of network link costs. This allows the packet header to contain only the identity of the destination node, as opposed to the entire path, but only if all PSNs recompute routes as soon as they receive new routing updates. Thus, to stagger PSN route re-computation would require extensive changes to the packet-forwarding system as well as to the packet header.

4.2 Normalizing the Cost to Hops

Figure 4 illustrates the effect of this transformation for the case of 56 kb/s lines. The link cost in this figure has been normalized by the value reported by an idle line, for the purpose of making a meaningful comparison. In particular, the metric has been divided by 30 routing units for HN-SPF and 2 units for D-SPF (this is the delay metric's bias value for a 56 kb/s line). Note how the curve for the D-SPF cost is much steeper than that for the HN-SPF cost at high utilization levels. As we explain in section 5, it these *relative* costs that force all of the traffic to be shed, which in turn causes routing oscillations.

For a 56 kb/s link the minimum reported cost is 30 units and the maximum cost is 90 units. This limits a links relative cost to be no greater than two additional hop in a homogeneous network. For example, if a link reports the maximum cost while all other links report the minimum cost, then the link will have a relative weight of two additional hops.

The HN-SPF Module imposes upper and lower bounds on the values that can be reported by lines, which are set on a line-type basis. The lower bound also depends on the configured propagation delay of the line, but is less sensitive to it than the delay metric. In particular, the lower bound is a slowly increasing function of the configured propagation

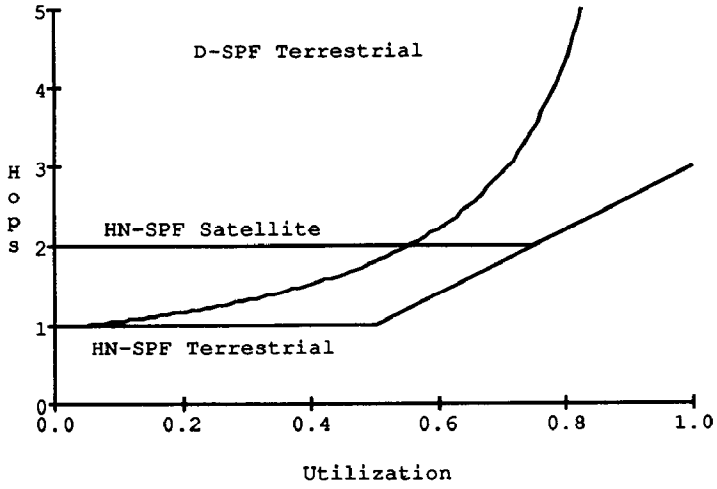

Figure 4: Comparison of Metrics (Normalized) for a 56 Kb/s Line

delay. For example, it is set higher for a satellite line than a terrestrial line of the same speed, to discourage use of the former *under light traffic conditions.*

When a link is lightly utilized, there is little reason to shed traffic from the link. The HN-SPF metric is constant until the utilization gets above a threshold that depends on the line-type. For example, it is 50% for a 56 kb/s terrestrial link. At higher utilizations the cost of the link is allowed to rise in order to shed some of its traffic. The effect of HN-SPF is to make routing reasonably sensitive to the propagation, queueing and transmission delays of links at low utilizations and insensitive to propagation and queueing delays at high utilizations.

4.3 Limits on Relative Changes

The modifications include three mechanisms that control the change between successively reported update values for a particular link. Two of these prevent the value from changing by too much, while one prevents the change from being too little.

Averaging The measured link delay is averaged over a single 10-second period. The revised metric is computed using an averaging process that encompasses link condition information from previous periods. Averaging increases the period of routing oscillations, thus reducing routing overhead.

Maximum Change The maximum amount by which the reported value (for a given link) can vary is limited to a little more than a half-hop (relative to the minimum value for the line type). In particular, there are two limits per line-type on the allowed upward and downward change in the reported value. These limits are essential for limiting the amplitude of routing oscillations and are discussed further in section 5.

Minimum Change The revised metric enhances the mechanism that prevents the generation of frivolous routing updates. A change in the links cost is allowed only if the change is above a certain threshold. This threshold is a little less than a half-hop (relative to the minimum value for the line type). This feature has the effect of reducing both routing related computation and routing-related link bandwidth consumption.

4.4 Heterogeneous Trunking

Both the ARPANET and MILNET have heterogeneous trunking. Both use satellite and multi-trunk lines, while the MILNET also uses different link bandwidths. To address the needs of these networks, we normalized the HN-SPF metric to handle heterogeneous links. While these values have been successful on the ARPANET and MILNET, they are not necessarily appropriate for all network topologies. We designed the HN-SPF module so that these values would be easy to change, and envisioned that parameter sets would be tailored to the needs of individual networks.

Consider figure 5, which illustrates the behavior of the revised metric as a function of line utilization for four different lines: 9.6 kb/s terrestrial, 9.6 kb/s satellite, 56 kb/s terrestrial and 56 kb/s satellite.

While a 56 kb/s terrestrial line is favored over a 56 kb/s satellite line during periods of low utilizations, the two are treated equally when highly utilized. This ensures that satellite bandwidth is utilized when the network is heavily loaded. Also note that, for the same utilization level, a 56 kb/s satellite trunk can appear no more than twice as expensive as its terrestrial counterpart. This has the effect of decreasing path lengths vis-a-vis those with the delay metric, since short paths incorporating satellite lines do not appear as unfavorable relative to longer paths consisting entirely of terrestrial lines as they do with D-SPF.

Also note that a fully utilized 9.6 kb/s line can report a value only about 7 times greater than that by an idle 56 kb/s line, as opposed to approximately 127 times with the delay metric. This should make it more likely that some traffic flows will continue to use it despite its previous heavily utilized state, which is preferable to the scenario where all routes tend to move away from it once it advertises its condition.

Finally, note how an idle 56 kb/s satellite line appears more favorable than an idle 9.6 kb/s line, as opposed to ap-

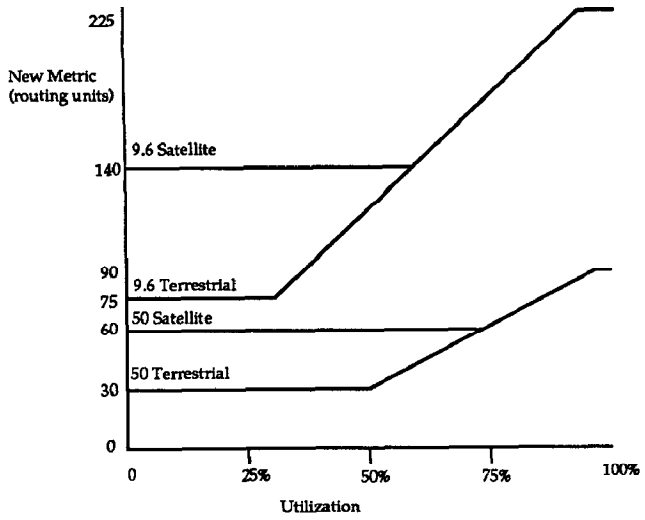

Figure 5: Absolute Bounds

Utilization estimated from delay using the M/M/1 queueing model with an average packet size of 600 bits.

pearing about twice as expensive with the delay metric. This is once again motivated by a desire to efficiently use network resources, especially high-speed satellite bandwidth.

In general, the normalizations were chosen such that the maximum value for a particular line is approximately three times the minimum value for a zero-propagation-delay line of the same type. This is based on our value judgment that traffic should not be routed around a heavily utilized line by more than two additional hops, in networks similar in size and topology to the ARPANET. Thus, if the shortest path between two nodes consists of two 56 kb/s links, then HN-SPF will never route traffic between the two nodes over a 56 kb/s path consisting of more than 6 links.

4.5 Limits of HN-SPF

It should be mentioned here that HN-SPF can only accomplish load-sharing indirectly, by affecting the number of paths using a link; whether or not the path is active is not a major factor. Thus, while HN-SPF should vastly improve load-sharing and general performance vis-a-vis D-SPF in many situations, it will be most effective when network traffic consists of several small node-to-node flows. To accomplish load-sharing when network traffic is dominated by several large flows would require a multi-path routing algorithm (e.g., see [6]). In general, single path routing algorithms are fairly ineffective in dealing with such traffic patterns.

5 Behavior of SPF

Earlier we showed that D-SPF is unstable under heavy loads and that the major cause of this instability is that it can report a link cost which results in the shedding of all its routes. HN-SPF stabilizes routing by limiting both the magnitude of the reported cost and the amount it can change between routing updates. In terms of control theory, HN-SPF changes both the equilibrium point and the gain of the routing algorithm.

In this section we model the equilibrium behavior of the SPF algorithm itself using topology and traffic information from an operational network, and show how this behavior is a complex interaction between the network topology, the traffic matrix and the metric. We use this model to compare the behavior of three SPF schemes and show that HN-SPF lies between the extremes of min-hop routing and D-SPF. In particular, we show that HN-SPF's equilibrium point allows more traffic on the link than that of D-SPF, especially under conditions of overload.

We also explain the dynamic behavior of the SPF algorithm, i.e., the manner in which it converges to an equilibrium. While D-SPF can be unstable even at moderate loads, HN-SPF is stable under most conditions. HN-SPF can oscillate around its equilibrium and several techniques are used to damp these oscillations. However, unlike D-SPF, the amplitude of these oscillations is limited so that not all traffic is shed from the link.

Note that all the examples in what follows use the July 1987 ARPANET topology and peak hour traffic matrix. The modelling technique is general, however, and doesn't depend on the specifics of the topology and traffic used. Also note that all utilization-to-delay and delay-to-utilization transformations are based on an M/M/1 queueing model, again for illustrative purposes.

5.1 Model of Equilibrium SPF Behavior

A network's response to a change in link cost can be broken down into a series of transformations (Figure 6). After comparing the reported cost to all other link costs, the SPF algorithm decides on the routes to be sent over the given link. The sum of the traffic on these routes gives rise to a link utilization. This link utilization is converted into a cost which is reported to the network. The cycle then repeats itself. If the new cost is the same as the old cost the link is at equilibrium. We define the network to be at equilibrium when all its links reach equilibrium.

The complex nature of the interactions between SPF, the topology and the traffic matrix makes it difficult to analyze the system as a whole. In particular, note that the process of

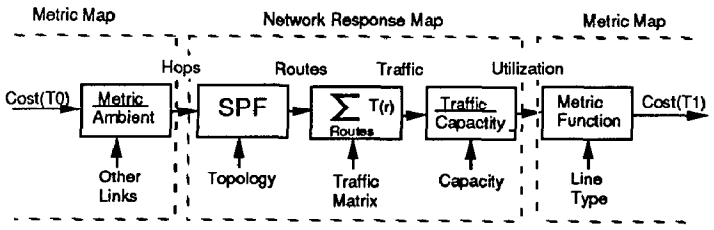

Figure 6: Mappings from Reported Cost to Traffic

calculating equilibrium for a given link consists essentially of successively varying its cost and then recomputing the routes over it until its utilization converges to some value. Since a change in one link's routes affects the utilization and thus potentially the cost of other links, thus affecting its own choice of routes during the next iteration, any exact determination of equilibrium would have to consider this interplay between the links. Furthermore, this would have to be done simultaneously for all links, clearly a task of considerable complexity.

We choose instead to model the system from the view of an "average" link. We assume that all links except the one under consideration report the same ambient value; this ambient value can be considered a hop. The underlying assumption we make is that the flow of traffic on and off the considered link during the process of determining its equilibrium does not significantly affect the costs of other network links. This assumption is exact in the case of min-hop routing, and is very good in the case of HN-SPF, since the HN-SPF metric is essentially constant until link utilization exceeds 50%. The assumption is weaker but still reasonable in the case of D-SPF, where the metric is more sensitive to traffic.

Note that even though we fix all but one of the links in the network, we are interested in the case where the entire network is active. The equilibrium values determined using our method provide a reasonable way of comparing the relative performance of the different routing schemes.

5.2 SPF Model Transforms

We now examine the transformations mentioned above in detail.

Each link is taken one at a time and statistics are collected relating the reported cost needed (in hops) to shed each route and its traffic. Ties are always broken in favor of using the given link. The statistics are aggregated over the whole network to get the characteristics of the "average link". The results are shown in Figure 7. The characteristics of individual links differ from the "average" link, so the

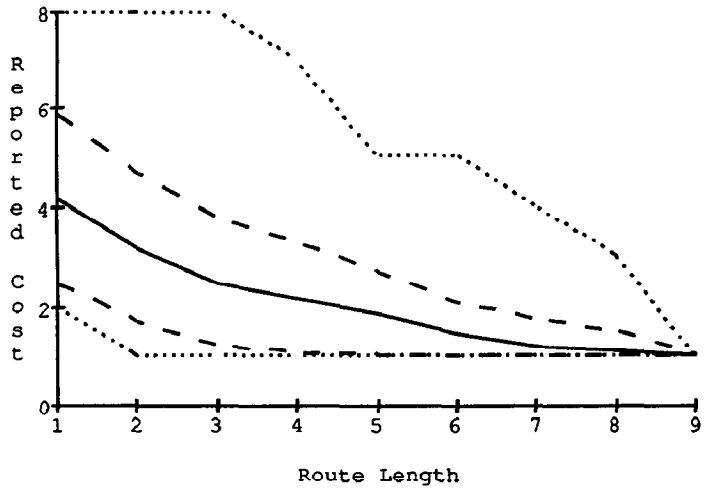

Figure 7: Reported Cost Needed to Shed Routes

standard deviation (dashed line) and maximum/minimum values (dotted line) are also provided.

The ARPANET topology is rich with alternate paths. Figure 7 shows that long routes have alternate paths that are only slightly longer. The X-axis of the figure is the particular path length under consideration. The Y-axis shows the average reported cost needed to shed all routes of that length. Note that in the case of a one-hop route, the maximum reported cost needed to shed the route is eight hops. Since SPF is hereditary, if the 1-hop route does not use the link, then no other routes will use it either. So if a link reports more than eight hops, then it will shed *all* of its routes. The average reported cost needed to shed all routes is four hops. This can happen with D-SPF if a link is more than 75% utilized over the measurement interval (assuming an M/M/1 relationship between delay and utilization). Since HN-SPF is limited to reporting at most 3 hops, this does not happen for the average link.

All the routes that flow over a link do not carry the same traffic. Since SPF chooses routes without regard to how much traffic actually flows over them, a network traffic matrix is necessary to evaluate the amount of traffic going over a link as a function of reported costs.

The network responds to an increase in the reported cost by moving some of the routes and associated traffic off the link. The amount of traffic that remains on the link depends on the value of the reported cost relative to the ambient cost. By normalizing the reported cost in terms of hops, we can characterize the network response independent of the metric used.

Figure 8, which we call the Network Response Map,

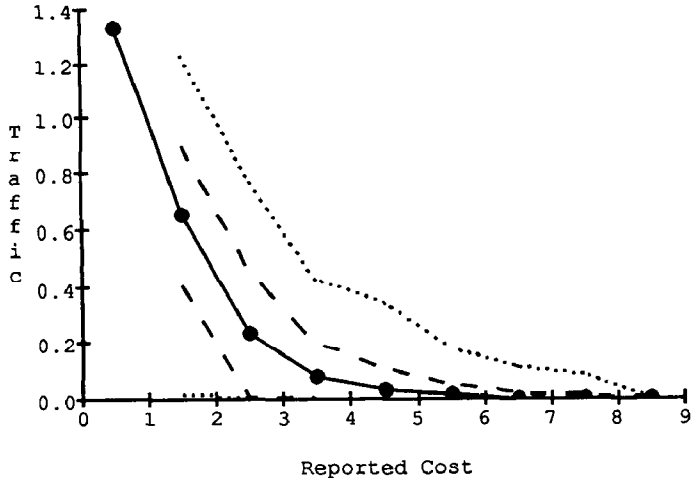

Figure 8: Overall Network Response To Reported Cost

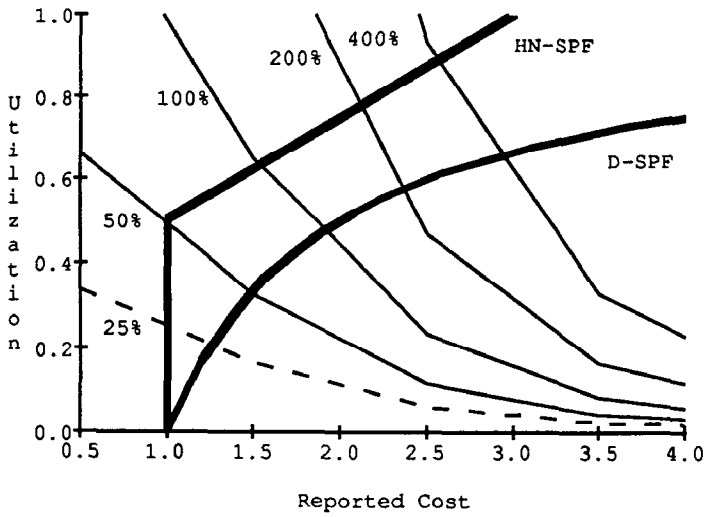

Figure 9: Equilibrium Calculation

shows the amount of traffic on the "average" link as a function of different reported costs. The Y-axis is normalized so that base traffic (1) is the traffic when the reported cost is one hop. The figure is best explained with an example. The point at x=1.5 represents two cases: the case where the link reports a cost of 1 and all path-length ties are broken against using the link considered, and the case where the link reports a cost of 2 and all path-length ties are broken in favor of using the link. In other words, the point represents the maximum amount of traffic when the link reports two and the minimum when it reports one. From the figure it should be evident that a very small change in the reported cost can cause large changes in traffic. Consider, for example, the large difference between the traffic at x=0.5 and x=1.5. Potentially all of this traffic can be shed from the link with a very small change in reported cost. We call this the *epsilon* problem.

The amount of traffic being routed over a link depends on the global interaction between the current reported cost and the costs of other links. Current traffic does not depend on local factors, such as link capacity or the routing metric, though these do define the next reported cost. Figure 8 shows how small the reported cost needs to be in order to shed most of the link's traffic. If the link reports a cost of 4, then over 90% of its base traffic will be shed. The effect of the traffic on the link depends on the capacity of the link and the routing metric. For example, if the base traffic is 75% of the link's capacity, then D-SPF would report a cost of 4, whereas HN-SPF would report a value of 2. D-SPF would shed over 90% of its traffic, while HN-SPF would shed less than 30%.

5.3 Equilibrium Calculation

We now calculate the equilibrium points for different SPF routing metrics.

Figure 8 defines the mapping of reported cost to utilization (Network Response map) and Figure 4 defines the mappings from utilization to reported cost (Metric map) for different routing metrics for a 56 kb/s link. Equilibrium is achieved when the reported cost from one period results in a traffic level on the link that in turn results in the same cost for the next period. Thus both the traffic on the link and the reported cost will be the same from one period to the next. To find the equilibrium point, we combine the two mapping functions and solving for $Cost(t_i) = Cost(t_{i+1})$. Because of the extremely non-linear nature of both the Network Response map and the Metric map, solving these equations using analytical techniques is not feasible. Instead we present only the solution which was obtained using numerical techniques.

Figure 9 depicts graphically the method we use to calculate equilibrium. Two metric maps are shown, one for HN-SPF and one for D-SPF. A family of Network Response maps are shown, representing different traffic levels. The percentage figure corresponding to each Network Map represents the percentage the "average link" would be utilized if min-hop routing were in effect; it is a measure of the offered load to the link relative to its capacity.

The equilibrium point changes with different offered loads. When designing a network, one matches the network topology and link capacity to match cost and performance requirements. This is done by adjusting topology and ca-

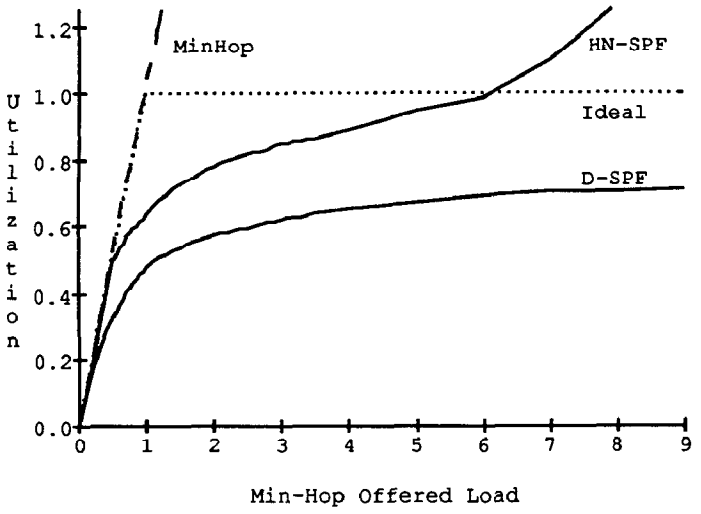

Figure 10: Equilibrium Traffic for a Heavily Utilized Line

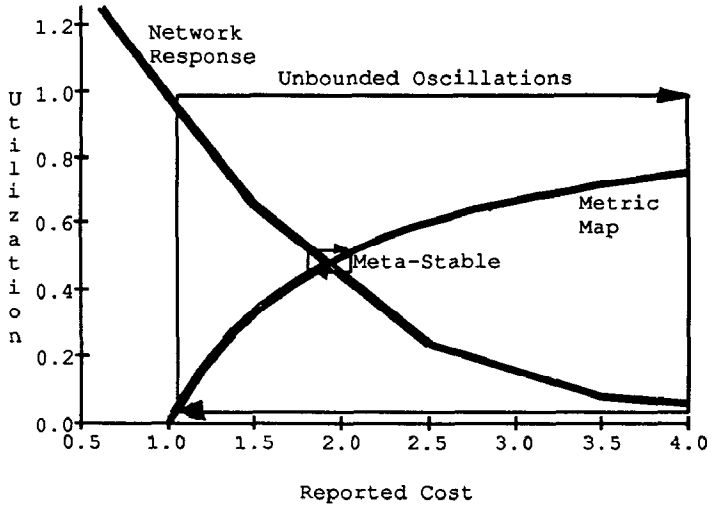

Figure 11: Dynamic Behavior of D-SPF

pacity as a function of expected traffic. A major operational issue is to make sure that the network can adapt to the variance in traffic and still provide adequate service. For static routing like min-hop, there is no such adaptation. In the case of traffic-sensitive routing like D-SPF and HN-SPF, where load balancing is dynamic, one can ask to what extent can routing handle variance in the network traffic.

Figure 10 shows the equilibrium link utilization for different offered loads. The ideal routing would be to route traffic over the link until it reached 100% and then to shed additional traffic to maintain this level as the offered load increased. Since min-hop is not traffic-sensitive, it becomes oversubscribed once the offered load reaches 100%. Figure 10 shows that HN-SPF can sustain higher link utilization levels than D-SPF, especially under high loads. HN-SPF is between min-hop and D-SPF: it acts like min-hop until the link utilization exceeds 50% and then starts shedding traffic, but still maintains higher link utilizations than D-SPF.

Operationally, HN-SPF is the safety net that compensates for bad network designs and unexpected changes in traffic patterns. It makes good use of network bandwidth and can automatically handle variations in traffic that are several times the designed traffic level. Min-hop does not offer any of these adaptive features and D-SPF does not effectively utilize network bandwidth.

5.4 Dynamic Behavior

Dynamic behavior describes how the system converges to its equilibrium. The previous section showed the equilibrium points for different routing algorithms, but did not describe how or if the system achieved equilibrium. We will show that for heavy offered loads D-SPF is unstable and will oscillate between being oversubscribed and idle. HN-SPF will usually converge to its equilibrium though it may oscillate around the equilibrium with a bounded amplitude.

We illustrate the concept of dynamic behavior using Figures 11 and 12. These graphs show the Network Response map and the Metric map for offered loads of 100%. The equilibrium routing is defined by the point where the two maps intersect. The dynamic behavior of the system can be traced by starting at a certain traffic level and finding the corresponding reported cost on the Metric map. This reported metric will result in a new traffic level which can be found from the Network Response map. The dynamic behavior can be found by repeating this process.

Under heavy offered loads, D-SPF usually operates in an unstable fashion. As seen in Figure 11, the behavior of D-SPF depends on the initial starting point. If the reported cost is close to the equilibrium point, the system will converge to the equilibrium, while if the starting point is away from the equilibrium, the system will diverge and oscillate between its maximum and minimum values. The equilibrium is considered meta-stable because a slight perturbation can knock the system off its equilibrium and into the realm of instability.

HN-SPF, on the other hand, will converge to the equilibrium and may oscillate around it with a bounded amplitude. This is because the maximum change is bounded by a half-hop. Without this bound, HN-SPF would oscillate with a much larger amplitude, but still would not be unstable like D-SPF. The averaging filter used by HN-SPF also affects

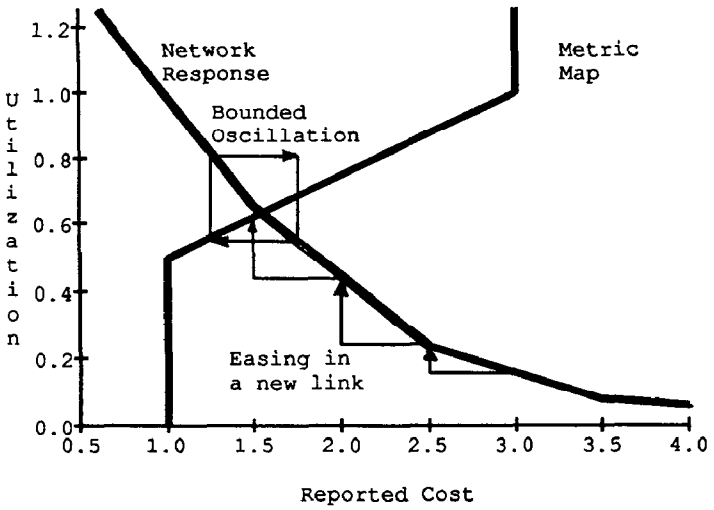

Figure 12: Dynamic Behavior of HN-SPF

Date	May 87	Aug 87
Internode Traffic (kbps)	366.26	413.99
Round Trip Delay (ms)	635.45	338.59
Rtng. Updates per Trunk/sec.	2.04	1.74
Update Period per Node (sec)	22.06	26.32
Internode Actual Path (hops/msg)	4.91	3.70
Internode Minimum Path	3.97	3.24
Path Ratio (Actual/Min.)	1.24	1.14

Table 1: ARPANET: Network-wide Performance Indicators

the behavior. Since it essentially averages the cost over the last two routing periods, it slows down the *frequency* of the oscillations.

Another feature of HN-SPF is that it gently eases in new lines. When a line comes up, it abruptly adds new capacity to the network. If routing is allowed to over-react to this new bandwidth, it may knock some of the links out of their meta-stable states and cause oscillations. To address this issue, when a link comes up it starts with its highest cost. Routing will converge to its equilibrium slowly by pulling in a little more traffic with each routing period (Figure 12).

Another feature of HN-SPF is a heuristic way of getting the routing to fall into a meta-stable state. As the link metric oscillates around the equilibrium point, for each cycle HN-SPF reports a slightly different cost. The maximum down value is one unit less than the maximum up value. Thus, for each cycle the reported cost marches up one unit. This has the effect of spreading the reported costs for lines with the same utilizations, especially when lines are lightly utilized. This spreading help overcome the *epsilon problem* by reducing the number of equal length paths.

6 Performance in the ARPANET

In this section we provide selected results from a study conducted by BBNCC on the effectiveness of the revised metric in the ARPANET. Further details can be found in [1, 14]. An extensive study of the results of deploying the HNM in the MILNET can be found in [2].

Table 1 shows indicators of network performance based on peak hours before and after the installation of the HNM. Note the 46% reduction in round-trip delay despite a 13% increase in network throughput. While part of this reduction in delay can certainly be attributed to the 18% decrease in minimum path length between the two sets of traffic, most of the reduction is the result of the improved load-sharing and routing stability associated with HN-SPF, especially given the increased traffic level. This belief is further strengthened by the 8% decrease in the ratio of actual to minimum path length. Note also the 19% reduction in number of routing updates generated.

The effectiveness of HN-SPF in reducing the likelihood of network congestion is illustrated rather dramatically in figure 13, which shows the total number of packets dropped due to congestion for weekdays just before and after installation of the HNM. The sharp drop in the number of dropped packets after the deployment of the patch is a clear indication of reduced levels of congestion. Indeed, the drop is accomplished despite ever-increasing traffic levels on the ARPANET.

7 Conclusions

The HNM has substantially improved the performance of routing in the ARPANET. HN-SPF retains many desirable features of SPF, such as dynamically routing around down lines and destination-based addressing. It has overcome some of the major defects of D-SPF, including routing oscillations and the reduction of effective bandwidth. Under light traffic loads, HN-SPF behaves in similar fashion to D-SPF, giving each route a low delay path. Under heavy loads it changes its criteria to give the "average" route a good path. It does this by diverting some routes to slightly

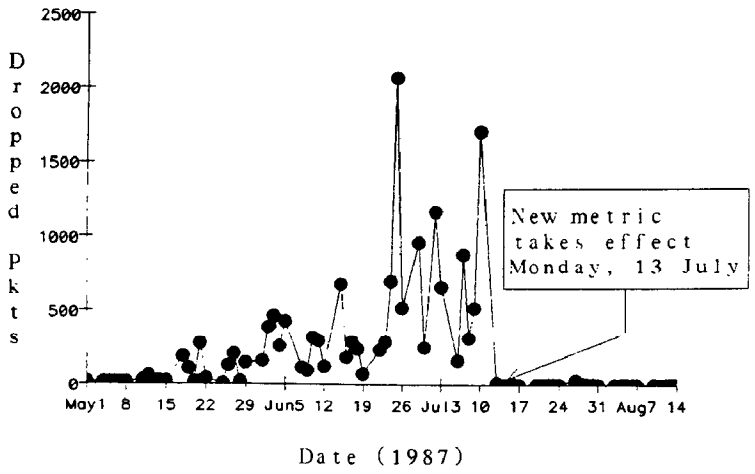

Figure 13: ARPANET: Dropped Packets (1987)

longer paths, allowing the remaining routes to efficiently use the link. T-SPF has raised the effective capacity of the network by an estimated 25% and is one of the reasons the ARPANET has survived large growths in traffic without the benefit of increased bandwidth.

8 Acknowledgments

The authors would like to thank Frederick Serr for his contributions to the work described here. This work was performed for the Defense Communications Agency under contract no. DCA 200-85-C-0023.

References

[1] *ARPANET Performance Analysis Report.* Quarterly Report 11, BBN, Aug. 1987.

[2] *MILNET Routing Improvements: Measurements and Analysis of the SPF Metric Patch.* BBN Report 6719, BBN, Feb. 1988.

[3] D. P. Bertsekas. Dynamic Behavior of Shortest Path Routing Algorithms for Communication Networks. *IEEE Transactions on Automatic Control*, AC-27:60–74, Feb. 1982.

[4] E. W. Dijkstra. A Note on Two Problems in Connection with Graphs. *Numerische Mathematik*, 1:269–271, 1959.

[5] R. Gallager and D. Bertsekas. *Data Networks.* Prentice-Hall, 1987.

[6] V. Haimo, M. Gardner, I. Loobeek, and M. Frishkopf. *Multi-Path Routing: Modeling and Simulation.* BBN Report 6363, BBN, Sep. 1986.

[7] A. Khanna. *Short-Term Modifications to Routing and Congestion Control.* BBN Report 6714, BBN, Feb. 1988.

[8] J. M. McQuillan, G. Falk, and I. Richer. A Review of the Development and Performance of the ARPANET Routing Algorithm. *IEEE Transactions on Communications*, 1802–1811, Dec. 1978.

[9] J. M. McQuillan, I. Richer, and E. C. Rosen. *ARPANET Routing Algorithm Improvements: First Semiannual Technical Report.* BBN Report 3803, BBN, Apr. 1978.

[10] J. M. McQuillan, I. Richer, and E. C. Rosen. The New Routing Algorithm for the ARPANET. *IEEE Transactions on Communications*, 711–719, May 1980.

[11] J. M. McQuillan, I. Richer, E. C. Rosen, and D. P. Bertsekas. *ARPANET Routing Algorithm Improvements: 2nd Semiannual Technical Report.* BBN Report 3940, BBN, Oct. 1978.

[12] J. M. McQuillan, I. Richer, E. C. Rosen, and J. G. Herman. *ARPANET Routing Algorithm Improvements: 3rd Semiannual Technical Report.* BBN Report 3940, BBN, Oct. 1978.

[13] E. C. Rosen. The Updating Protocol of ARPANET's New Routing Algorithm. *Computer Networks*, 4:11–19, Feb. 1980.

[14] J. A. Zinky, A. Khanna, and G. Vichniac. Performance of the Revised Routing Metric for ARPANET and MILNET. Submitted to MILCOM 89, March 1989.