Type-of-Service Routing in Dynamic Datagram Networks*

Ibrahim Matta
imm@cs.umd.edu

A. Udaya Shankar
shankar@cs.umd.edu

Department of Computer Science
University of Maryland
College Park, MD 20742

Abstract

We propose a new approach to achieving type-of-service (TOS) with adaptive next-hop routing in wide-area networks such as the Internet. We consider two traffic classes, namely delay-sensitive and throughput-sensitive. Our approach uses two FCFS queues at each outgoing link, one for each TOS; the link is shared cyclicly between the two TOS queues. The next-hops for the delay-sensitive traffic adapts to link delays of that traffic. The next-hops for the throughput-sensitive traffic adapts to overall link utilizations. We compare our approach with a traditional approach using discrete-event simulation and Liapunov analysis (for stability of routes). The proposed approach offers lower end-to-end delays as it isolates the routes of the two traffic classes.

1 Introduction

With the increasing diversity of network applications, it has become crucial for networks to meet varying requirements in terms of *type of service* (TOS), e.g., low delay, high throughput, etc. This means that a network's routing protocol should be able to determine appropriate routes for each TOS. Most of the design and analysis work on TOS assumes connection-oriented (or virtual-circuit) routing (e.g., [21, 10]). This paper is concerned with achieving TOS using next-hop (datagram) routing on wide-area store-and-forward networks. Next-hop routing, which is currently used in the Internet, is a simple and robust way to do adaptive routing.

Next-hop TOS routing is done as follows. Each node maintains for each destination node and TOS, a neighboring node id, referred to as the *next-hop*. Every data packet header contains its destination id and the TOS requested by the application. When a node receives a data packet, it forwards the packet to the next-hop for the packet's destination and TOS. The objective of the routing protocol is to choose next-hops so that the resulting routes satisfy the requested TOS. The quality of service offered by a route depends on the traffic through its links, which depends on the time-varying external load. Consequently, a routing protocol must monitor link traffic changes and adapt its next-hops. To do this, each node maintains for each outgoing link and TOS, a dynamic *link cost*, which is updated regularly according to the traffic flowing through the link. This link cost information is regularly disseminated to nodes of the network. Based on received link cost information, each node maintains and regularly updates its next-hop for each destination and TOS.

The IP layer of the Internet Protocol suite specifies different TOS [2]. Among them are the minimum delay service required for example by interactive traffic or real-time traffic (e.g., audio), and the maximum throughput service required for example by bulk transfers such as network mail or FTP. Routing protocols such as the Internet OSPF [19] and the OSI IS-IS [6] provide separate next-hops for each TOS. However, the TOS mechanism has been so far of little use, and little is known on how well it would work in practice. In addition, many current routing protocols use *static* link costs (which respond only to failures and recoveries). To our knowledge, only one approach to adaptive TOS routing has been proposed [12]. This approach, henceforth called **TOS1**, considers two TOS: *low delay* and *high throughput*. It uses measured link delays

*This work is supported in part by ONR and DARPA under contract N00014-91-C-0195 to Honeywell and Computer Science Department at the University of Maryland, and by National Science Foundation Grant No. NCR 89-04590. The views, opinions, and/or findings contained in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, ONR, the U.S. Government or Honeywell. Computer facilities were provided in part by NSF grant CCR-8811954.

as the link costs for the delay-sensitive traffic (*delay-based routing*). It uses link utilizations (or equivalently available link capacities) as the link costs for the throughput-sensitive traffic (*utilization-based routing*). In TOS1, each node maintains for each outgoing link, a *single* FCFS queue of data packets; that is, packets of every TOS share this queue. Reference [12] refers to simulation studies but does not present any quantitative results.

Traditionally, link queueing disciplines have been of the FCFS type. It appears desirable to use a more structured queueing discipline that helps "isolate" the different TOS classes, for example, by using a separate queue for each TOS class. This concept of *isolating* traffic classes using structured queueing disciplines has been used recently in flow control studies, e.g., [7, 5, 21]. In this paper, we investigate the use of a structured queueing discipline with adaptive next-hop TOS routing.

Our approach

We consider a simple *two-queue* link scheduling discipline, henceforth referred to as **type-of-service queueing**. We consider two TOS: low delay and high throughput. With type-of-service queueing, each node maintains two FCFS queues for each outgoing link, one for each type of service. The link bandwidth is allocated equally between the two queues in a round-robin fashion. (Type-of-service queueing is similar to the recently proposed fair-queueing discipline [7], except that in fair-queueing, the link bandwidth is divided equally amongst the connections using the link, rather than the types of service.)

For any link, the link cost for delay-sensitive traffic is obtained by exponentially averaging the measured delay that is experienced by delay-sensitive packets *only*. The link cost for throughput-sensitive traffic is obtained by exponentially averaging the measured utilization of the link (i.e. accounting for *both* delay-sensitive and throughput-sensitive packets). Henceforth, we refer to our approach as **TOS2**.

Our discrete-event simulations on a subset of the NSFNET-T1-Backbone topology show that TOS2 performs significantly better than TOS1 in a typical situation where the proportion of delay-sensitive traffic is small compared to the throughput-sensitive traffic [14]. TOS2 not only yielded a lower end-to-end delay for delay-sensitive packets (as expected), but also a lower overall end-to-end delay (which is unexpected).

We argue that this is because with TOS2, the routing is significantly improved as we exploited the scheduling structure of type-of-service queueing when calculating link costs. In particular, the routes of the two traffic classes can be *isolated* with the delay-sensitive traffic taking the low delay routes, and the throughput-sensitive traffic taking the under-utilized routes. This results in a better overall network performance.

In fact, we find that a non-TOS scheme, which does not distinguish between the two types of traffic and applies utilization link cost to both, referred to as UTIL, performs significantly better than TOS1 at high load.

To gain more insight into the system behavior with both TOS schemes (TOS1, TOS2), we analyzed in [17] a simple model of a single source-destination node pair connected by two (parallel) paths; the first path representing low delay routes, and the second path representing high capacity routes. We viewed this system as a dynamical system [3]. We represented isolation by a *stable* state where all delay-sensitive traffic stays on the first path, and all throughput-sensitive traffic stays on the second path. We applied the Liapunov function method, and derived stability theorems ignoring propagation delays. In this paper, we derive our stability results taking into account the propagation delays. We show that for certain parameter values, the isolation state provides the best delay performance for both traffic classes. We also show that TOS2 has a larger stability region corresponding to isolation than TOS1.

Organization of the paper

The paper is organized as follows. Section 2 describes our discrete-event simulation model and results. Section 3 presents our Liapunov stability analysis. Section 4 concludes the paper. Details of the simulations and proofs are not presented here for space reasons, and can be found in [16].

2 Discrete-Event Simulations

Our simulation studies were done with a discrete-event simulator, MaRS [1], which has been used for other studies of routing algorithms [23, 22]. In Subsection 2.1, we describe the simulation model. In Subsection 2.2, we present general observations about the results.

2.1 Model

Regarding the *physical network*, we consider the "East coast" subset of the NSFNET-T1-Backbone. Figure 1 illustrates the topology. Link propagation delays in milliseconds are indicated.

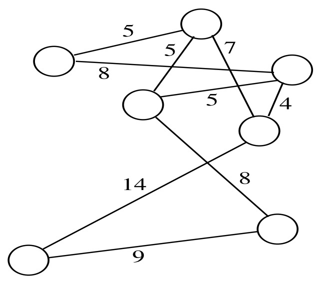

Figure 1: The "East coast" subset of the NSFNET-T1-Backbone (7 nodes, 9 bidirectional links).

We have two versions: a *low-speed* version (with NSFNET-T1 parameters) and a *high-speed* version. There are no link or node failures. All nodes have adequate buffer space for buffering packets awaiting processing and forwarding.

Regarding *link scheduling*, we consider two disciplines for the scheduling of data packets over the links: FCFS and type-of-service queueing. Both TOS1 and UTIL use FCFS. TOS2 uses type-of-service queueing. In all schemes, routing packets have priority over data packets; i.e., data packets can be scheduled for transmission only if there are no routing packets present.

Regarding *routing*, we consider a link-state algorithm like the SPF (Shortest Path First) algorithm used in the ARPANET [18] and OSPF (Open SPF) used in the Internet [19]. Each node maintains a time-varying cost (explained below) for each outgoing link and TOS. Each node also maintains a view of the network topology, with a cost for each TOS and link in the network. To keep these views up-to-date, each node regularly broadcasts the link costs of its outgoing links to all other nodes using flooding. As a node receives this information, it updates its view of the network topology and applies Dijkstra's shortest path algorithm [8] to choose its next-hop for each destination and TOS.

The method used to compute link costs for each TOS depends on the link scheduling discipline. In all cases, each node's outgoing link costs are updated regularly. A link cost is always a simple moving average of a "raw cost", which is some measure of current link traffic.

In the TOS schemes, each node maintains the following two raw-costs for each outgoing link:

- *RawUtil*: percentage of time the communication channel is busy transmitting a packet; and
- *RawDly*: In TOS1, this is the average packet delay (queueing, transmission, and propagation) in milliseconds as experienced by all data packets. In TOS2, this is the average delay in milliseconds as experienced by delay-sensitive packets only.

Let $LinkCost(D)$ and $LinkCost(T)$ denote the link cost for delay-sensitive and throughput-sensitive traffic, respectively. Then at the end of each update interval, they are updated as follows:

$$LinkCost(D) = b \times RawDly + (1 - b) \times LinkCost(D)$$
$$LinkCost(T) = b \times RawUtil + (1 - b) \times LinkCost(T)$$

where the constant $b \in (0, 1)$.

Recall that UTIL does not use any TOS facility. The utilization metric is used to compute one next-hop for every destination at each network node (i.e., $LinkCost(T)$ is used for all traffic).

Regarding the *workload*, it is defined in terms of $\langle source\ node, destination\ node\rangle$ pairs. In each pair, the source produces data packets to be delivered to the destination. A source produces data packets according to a packet-train model [13]. The workload consists of two parts, a delay-sensitive workload and a throughput-sensitive workload. For both parts, we use a *uniform* distribution of source-destination pairs over the nodes of the network. Let parameter $U(D)$ ($U(T)$) denote the average number of source-destination pairs between every two nodes for delay-sensitive (throughput-sensitive) traffic. (We have also investigated *skewed* distribution of source-destination pairs, and obtained similar results.)

2.2 Observations

In this subsection, we present general observations about the simulation results. Detailed descriptions of scenarios simulated (parameters settings, etc.) and plots of the observed performance measures can be found in [16].

In every scenario, the system behaves in a manner typical of *open* queueing networks [4]. That is, the throughput is equal the workload as long as the workload is less than the system capacity; for workload higher than the system capacity the system is unstable. With increasing workload, the delay increases at first slowly until a point where the system starts becoming saturated; we refer to this point as the *saturation point*. Further increase in the workload beyond this point causes the delay to increase dramatically (with increasing rate) until the system becomes unstable.

Fixing $U(D)$ and varying $U(T)$ in a range where the delay-sensitive traffic constitutes almost 25%-30% of the total traffic, we found that TOS2 performs significantly better than TOS1 with respect to delays; TOS1 reaches saturation sooner. (See Figure 2.)

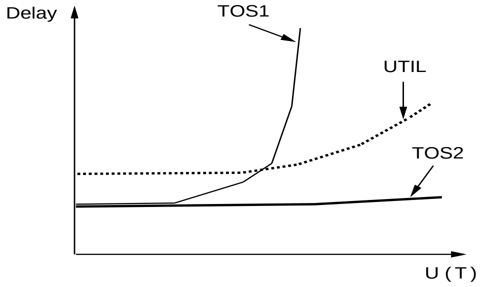

Figure 2: A generic plot. Delay versus $U(T)$ for a fixed $U(D)$.

Our explanation is as follows: It is well known that delay-based routing does not perform well at high load when queueing delay is a significant part of measured link delay, which consists of queueing, transmission, and propagation delays [15]. This is mainly because from the classical delay-utilization curve, around saturation, a small increase in utilization corresponds to a large increase in link delay. This dramatic change can result in the link becoming unattractive and thus being avoided by *all* delay-sensitive sources. Consequently, at the next routing update the link reports a very low cost and becomes attractive again. This leads to oscillatory behavior, which in turn degrades performance [15]. This is the case with TOS1 due to the use of the FCFS link scheduling discipline.

In TOS2, because delay-sensitive packets have a lower queueing delay under type-of-service queueing [7], the measured link delay becomes dominated by transmission and propagation delays. Thus, the reported delay link costs do not change dramatically and the delay metric remains a good indicator of expected link delay after updating the routes [15]. This improves the performance of delay-based routing of delay-sensitive packets and results in more stable routes for that traffic class. Meanwhile, the link utilization metric makes the throughput-sensitive traffic move away from the delay-sensitive traffic, taking the under-utilized routes. This has the effect of *isolating* the two traffic classes, resulting in a better overall network performance.

Intuitively, isolation is desirable since otherwise it becomes more likely that both traffic classes will move away from a highly loaded link (i.e. with high delay and utilization) at the same routing update. Such simultaneous traffic shifts degrade the overall performance, and in particular result in a higher delay and network under-utilization.

Ignoring low values of $U(T)$, UTIL also provides lower delays than TOS1. However, UTIL performs worse than TOS2 over the whole range of $U(T)$. This is expected, since the utilization-based metric does not necessarily result in minimum delay routes (especially at light load) [15].

3 Stability Analysis

In this section, we consider a simple network model to gain more insight into the complicated behavior we observed in our simulation experiments. We derive stability conditions for the routes of the two traffic classes under both TOS schemes (TOS1, TOS2). In Subsection 3.1, we give the model. In Subsection 3.2, we apply the Liapunov function method to analyze stability. Details of derivations and proofs can be found in [16].

3.1 Model

We model a network by a node S sending traffic to node D along two paths. Path 1 represents low delay routes, and path 2 represents high capacity routes. Path i ($i = 1, 2$) has propagation delay P_i time units and average transmission capacity C_i packets/time unit.[1] There are N delay-sensitive connections, and M throughput-sensitive connections from S to D. For every connection, packets originate at S according to a Poisson process, and without loss of generality we assume an arrival rate of 1 packet/time unit.

At any instant, we describe the state of the network by the tuple (x, y), where x is the number of delay-sensitive connections on path 1, and y is the number of throughput-sensitive connections on path 1. To model routing updates, we use a discrete-time flow approach as in [4]. We assume that (some or all) connections periodically update their routes to D every $\triangle$ time units, where $\triangle$ is long enough for the network to reach steady-state after a routing update. Routes, and hence the network state, are updated at discrete time instants $(k+1)\triangle$, $k = 0, 1, 2, \cdots$. Let (x_k, y_k) be the network state immediately after time $k\triangle$. At an update instant $(k+1)\triangle$, we use steady-state $M/M/1$ results [4] to estimate link costs based on (x_k, y_k). Using these link costs, routes are updated and consequently the new network state (x_{k+1}, y_{k+1}) is obtained. We denote by $T_{i,k+1}$ and $\rho_{i,k+1}$ the delay and utilization cost of path i, respectively, at time $(k+1)\triangle$.

For TOS1, with a FCFS discipline at S, we can write the delay link costs as follows:

$$\begin{aligned} T_{1,k+1} &= \frac{1}{C_1 - (x_k + y_k)} + P_1 \\ T_{2,k+1} &= \frac{1}{C_2 - (\bar{x}_k + \bar{y}_k)} + P_2 \end{aligned} \tag{1}$$

where $\bar{x}_k = N - x_k$ and $\bar{y}_k = M - y_k$ denote the number of delay-sensitive connections and the number

[1] Thus we assume $\frac{1}{C_1} + P_1 < \frac{1}{C_2} + P_2$ and $C_2 > C_1$.

of throughput-sensitive connections, respectively, on path 2.

The utilization link costs are

$$\begin{aligned} \rho_{1,k+1} &= \frac{x_k + y_k}{C_1} \\ \rho_{2,k+1} &= \frac{\bar{x}_k + \bar{y}_k}{C_2} \end{aligned} \tag{2}$$

The network state is updated using the costs of the two paths as follows:

$$x_{k+1} = \begin{cases} (1-\alpha_k)\ x_k & \text{if } T_{2,k+1} < T_{1,k+1} \\ x_k + \alpha_k\ \bar{x}_k & \text{otherwise} \end{cases}$$

$$y_{k+1} = \begin{cases} (1-\alpha_k)\ y_k & \text{if } \rho_{2,k+1} \leq \rho_{1,k+1} \\ y_k + \alpha_k\ \bar{y}_k & \text{otherwise} \end{cases} \tag{3}$$

The parameter α_k $(0 < \alpha_k \leq 1)$ reflects the amount of traffic rerouted. It can also be thought of as the degree of routing update synchronization at different nodes. Unless otherwise indicated, we assume that α_k is uniformly distributed over a small interval $[\alpha_{MIN}, \alpha_{MAX}]$.

For TOS2, with a type-of-service queueing discipline at S, the two queues at an output link are correlated, which makes the analysis difficult. A number of approximate solutions for such systems have been proposed. One common approach is to approximate the system by two loosely-coupled $M/M/1$ queues [24]. The service rate of each queue depends on the utilization of the other queue.

Define $C_{i,k}^{eff}$ as the effective capacity available for the delay-sensitive traffic on path i after time $k\triangle$. We have

$$C_{i,k}^{eff} \geq 0.5 C_i \tag{4}$$

with the worst case occurring when the other queue is *always* not empty. Assuming each queue is $M/M/1$, we obtain the following:

$$\begin{aligned} C_{1,k}^{eff} &= \frac{(C_1 - 0.5(y_k - x_k)) + \sqrt{(C_1 - 0.5(y_k - x_k))^2 - 2C_1 x_k}}{2} \\ C_{2,k}^{eff} &= \frac{(C_2 - 0.5(\bar{y}_k - \bar{x}_k)) + \sqrt{(C_2 - 0.5(\bar{y}_k - \bar{x}_k))^2 - 2C_2 \bar{x}_k}}{2} \end{aligned} \tag{5}$$

Thus we have the following delay link costs with type-of-service queueing:

$$\begin{aligned} T_{1,k+1} &= \frac{1}{C_{1,k}^{eff} - x_k} + P_1 \\ T_{2,k+1} &= \frac{1}{C_{2,k}^{eff} - \bar{x}_k} + P_2 \end{aligned} \tag{6}$$

The utilization link costs are as defined in (2). The network state is updated as in (3).

We refer to the iteration defined in (3) as I, i.e., $(x_{k+1}, y_{k+1}) = I(x_k, y_k)$. I is a mapping from a set G into itself, where $G = \{(x, y) : 0 \leq x \leq N \wedge 0 \leq y \leq M\}$. The sequence of points (x_1, y_1), (x_2, y_2), $\ldots$ is called the *trajectory* of the system [20]. The trajectory may or may not converge to a fixed point (x^*, y^*), i.e., $(x^*, y^*) = I(x^*, y^*)$.

Isolation corresponds to a stable state where every delay-sensitive connection stays on path 1, and every throughput-sensitive connection stays on path 2. This is equivalent to say that I converges to the fixed point $(N, 0)$. In the next section, we derive sufficient conditions for the system to reach isolation as a function of the starting state (x_0, y_0). (In a real network, the starting state would be the result of arrivals of new connections, departures of old connections, failure/recovery of links, etc.)

3.2 Liapunov Function Method

We use the Liapunov function method [20] to obtain sufficient conditions for stability and convergence to a fixed point without actually solving the system equations. The basic idea is to find a positive-definite scalar function $V(\mathbf{S})$, where $\mathbf{S}$ is the system state, such that its forward difference $\triangle V(\mathbf{S})$ taken along a trajectory is always negative. $V(\mathbf{S})$ is said to be a Liapunov function, and is regarded as a measure of the distance of the state $\mathbf{S}$ from the fixed point. As time increases, $V(\mathbf{S})$ decreases and finally shrinks to zero, i.e., the fixed point is approached.

It is more convenient to deal with the fixed point $(0, 0)$ rather than $(N, 0)$. Thus, we define the network state by $(\bar{x}, y)$ instead of (x, y). Then, the iteration I defined in (3) becomes:

$$\bar{x}_{k+1} = \begin{cases} (1-\alpha_k)\ \bar{x}_k & \text{if } T_{1,k+1} \leq T_{2,k+1} \\ \bar{x}_k + \alpha_k\ x_k & \text{otherwise} \end{cases}$$

$$y_{k+1} = \begin{cases} (1-\alpha_k)\ y_k & \text{if } \rho_{2,k+1} \leq \rho_{1,k+1} \\ y_k + \alpha_k\ \bar{y}_k & \text{otherwise} \end{cases} \tag{7}$$

Combining (1), (2), and (7), the system behavior with TOS1 is described by the following:

$$\begin{aligned} \bar{x}_{k+1} &= (1-\alpha_k)\ \bar{x}_k + \alpha_k\ N\ \delta_k \\ y_{k+1} &= (1-\alpha_k)\ y_k + \alpha_k\ M\ \beta_k \end{aligned} \tag{8}$$

where

$$\begin{aligned} \delta_k &= \begin{cases} 0 & \frac{1}{C_1 - N + (\bar{x}_k - y_k)} + P_1 \leq \frac{1}{C_2 - M - (\bar{x}_k - y_k)} + P_2 \\ 1 & \text{otherwise} \end{cases} \\ \beta_k &= \begin{cases} 0 & \frac{M + (\bar{x}_k - y_k)}{C_2} \leq \frac{N - (\bar{x}_k - y_k)}{C_1} \\ 1 & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

At the fixed point (which is now the origin), $\bar{x}_k \rightarrow 0$, $\bar{x}_{k+1} \rightarrow 0$, $y_k \rightarrow 0$, and $y_{k+1} \rightarrow 0$. Consequently, for the equations (8) to be satisfied, $\delta_k \rightarrow 0$ and $\beta_k \rightarrow 0$, which imply the following necessary (but not sufficient) conditions for convergence to the origin:

$$\begin{aligned} \frac{1}{C_1 - N} + P_1 &\leq \frac{1}{C_2 - M} + P_2 \\ \frac{M}{C_2} &\leq \frac{N}{C_1} \end{aligned} \qquad (10)$$

Define $D_1 = \{(\bar{x}, y) : \frac{1}{C_1 - N + (\bar{x} - y)} + P_1 \leq \frac{1}{C_2 - M - (\bar{x} - y)} + P_2 \ \wedge\ \frac{M + (\bar{x} - y)}{C_2} \leq \frac{N - (\bar{x} - y)}{C_1} \ \wedge\ 0 \leq \bar{x} \leq N \ \wedge\ 0 \leq y \leq M\}$. (See Figure 3.)

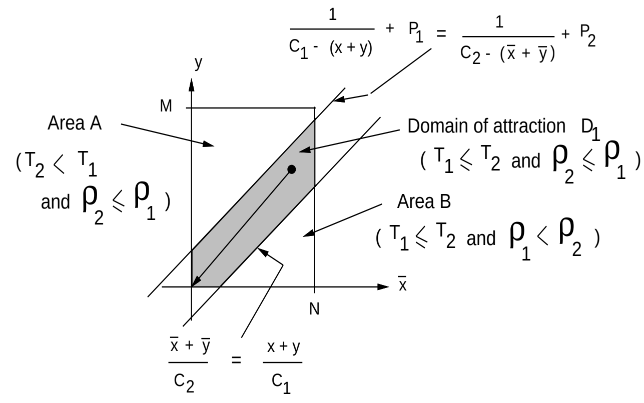

Figure 3: Domain of attraction for TOS1.

Define the scalar function $V(\bar{x}, y)$:

$$V(\bar{x}, y) = \bar{x}^2 + y^2 \qquad (11)$$

We show that $V(\bar{x}, y)$ is a Liapunov function in D_1, which implies that starting from any point in D_1, the trajectory stays inside D_1, and converges to the origin.

Lemma 3.1 *$V(\bar{x}, y)$ is a Liapunov function in D_1. That is: (1) $V(\bar{x}, y)$ is positive definite. (2) For all $(\bar{x}_k, y_k) \in D_1 - \{(0,0)\}$, $\triangle V(\bar{x}_k, y_k) < 0$ and $(\bar{x}_{k+1}, y_{k+1}) \in D_1$.*

From lemma 3.1 and the Liapunov stability theory [20], we have the following theorem:

Theorem 3.1 *For TOS1, any starting state in D_1 (the shaded area in Figure 3) leads to the origin, regardless of the values of α_k.*

The region D_1 is called *domain of attraction* corresponding to the origin [3] because it constitutes a set of starting states for which the iteration converges to the origin.

With TOS2, the system behavior is described by the same difference equations (8) except that δ_k is defined as

$$\delta_k = \begin{cases} 0 & \text{if } \frac{1}{C^{eff}_{1,k} - N + \bar{x}_k} + P_1 \leq \frac{1}{C^{eff}_{2,k} - \bar{x}_k} + P_2 \\ 1 & \text{otherwise} \end{cases}$$

At the fixed point, $x_k \rightarrow N, \bar{x}_k \rightarrow 0$, $y_k \rightarrow 0, \bar{y}_k \rightarrow M$. Then, from equations (5), we have $C^{eff}_{2,k} \rightarrow C_2 - 0.5M$, and $C^{eff}_{1,k} \rightarrow C_1$. For equations (8) to be satisfied at the fixed point, $\delta_k \rightarrow 0$ and $\beta_k \rightarrow 0$. Then necessary conditions for convergence to the origin are:

$$\begin{aligned} \frac{1}{C_1 - N} + P_1 &\leq \frac{1}{C_2 - 0.5M} + P_2 \\ \frac{M}{C_2} &\leq \frac{N}{C_1} \end{aligned} \qquad (12)$$

As we have done with TOS1, we want to show that $V(\bar{x}, y)$, defined in (11), is a Liapunov function in some region around the origin. Call this region D_2. Because the expressions for $C^{eff}_{i,k}$ are hard to work with, we made the approximation $C^{eff}_{1,k} \approx C_1 - 0.5y_k$ (the accuracy of this is discussed below) leading to the following definition of D_2.[2]

Define $D_2 = \{(\bar{x}, y) : \frac{1}{(C_1 - 0.5y) - N + \bar{x}} + P_1 \leq \frac{1}{(C_2 - 0.5(\bar{y} - \bar{x})) - \bar{x}} + P_2 \ \wedge\ \frac{M + (\bar{x} - y)}{C_2} \leq \frac{N - (\bar{x} - y)}{C_1} \ \wedge\ 0 \leq \bar{x} \leq N \ \wedge\ 0 \leq y \leq M\}$. Figure 4 depicts this region.

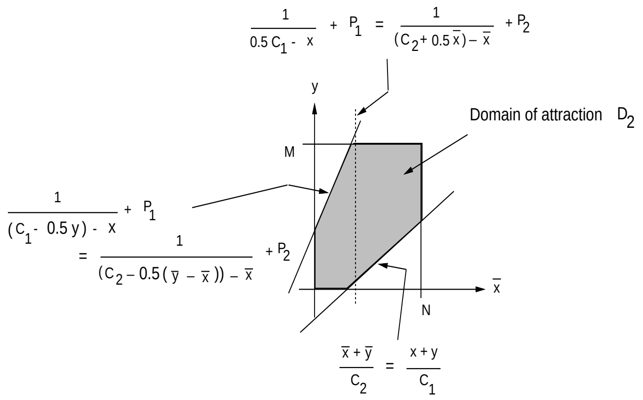

Figure 4: Domain of attraction for TOS2.

Lemma 3.2 *Assuming $C^{eff}_{1,k} \approx C_1 - 0.5y_k$, $V(\bar{x}, y)$ is a Liapunov function in D_2.*

From lemma 3.2 and the Liapunov stability theory [20], we have the following theorem:

Theorem 3.2 *For TOS2, assuming $C^{eff}_{1,k} \approx C_1 - 0.5y_k$, any starting state in D_2 (the shaded area in Figure 4) leads to the origin, regardless of the values of α_k.*

Effect of α_k on system behavior

The domains of attraction we have just found (cf. Theorems 3.1 and 3.2) are not the largest, i.e., there

[2] A justification for this can be found in [16].

may be points outside the domains which lead to the origin. This depends on the values of α_k. In particular, the domains are indeed largest for high enough values of α_k. On the other hand, they are not for small values of α_k. The following theorem shows this for TOS1.

Theorem 3.3 *For TOS1, starting at any point in area A or area B (Figure 3), the following hold:*

(i) If $\alpha_{MIN} > max(\frac{N-L_1}{M+N}, \frac{M+L_2}{M+N})$ then the iteration does not converge to the origin and locks into a limit cycle oscillating between states in areas A and B.

(ii) If $\alpha_{MAX} \leq min(\frac{L_2}{N}, \frac{-L_1}{M})$ then the iteration converges to the origin.

where

$$L_1 = \frac{H\ S - 2 + \sqrt{(H\ S-2)^2 + 4\ H[H\ (C_1-N)\ (C_2-M)+S]}}{2H}$$
$$H = P_1 - P_2$$
$$S = C_2 - M - C_1 + N$$
$$L_2 = \frac{C_2 N - C_1 M}{C_1 + C_2}$$

Theorem 3.3 indicates that for high enough values of α_k, the system may not converge to isolation, and rather oscillates with both traffic classes shifting *simultaneously* at each routing update. Such simultaneous traffic shifts result in a bad performance, i.e. higher delay and network under-utilization. This effect increases as α_k increases. (In practice, α_k depend on several factors, and can be quite high [11].)

Consider the simple case where $\alpha_k = 1$, for all k. It can be seen that isolation, whenever possible[3], provides the optimal performance for both traffic classes [9]. In this case, we are constrained to use a single path for each traffic class. Thus, in order to maximize the throughput of the throughput-sensitive traffic, we should send its packets over the maximum capacity link, i.e. path 2. Then, in order to minimize the packet delay of the delay-sensitive traffic, we should send its packets over the minimum packet delay link, i.e. path 1. Note that routing the delay-sensitive traffic (also) on path 2 would result in a higher delay compared to the delay of (the unused) path 1.

Note that the above result agrees with our argument about the benefits of isolation, which was made in Subsection 2.2.

Referring to Figures 3 and 4, we can conclude that for high enough α_k, TOS2 has a larger domain of attraction corresponding to isolation than TOS1. This conclusion is not affected by our approximation $C_{1,k}^{eff} \approx C_1 - 0.5y_k$, which was made in Theorem 3.2 for TOS2. In fact, it can be shown that $C_{1,k}^{eff} \leq C_1 - 0.5y_k$. Regardless of that, our monte-carlo simulations show that our approximation is only slightly optimistic [17]. I.e., some points in D_2 may not actually converge to isolation. However, such points constitute a small part of D_2, and hence the approximation does not affect our conclusion that TOS2 has a larger domain of attraction corresponding to isolation. In particular, isolation occurs for higher values of y_0 with TOS2 than with TOS1.

Theorem 3.3 also indicates that for small enough values of α_k, the system reaches isolation for all starting states and for all system configurations satisfying the necessary conditions for isolation. For such configurations and with such α_k, our monte-carlo simulations show that TOS2 gives much better transient performance, i.e. less oscillations and much faster convergence to isolation, than TOS1 [17] (we have not yet been able to obtain transient measures analytically).

4 Conclusions and Future Work

We found our proposed scheme (TOS2) very effective in providing lower end-to-end delays in a typical situation where the proportion of delay-sensitive traffic is small compared to the throughput-sensitive traffic. This is because our scheme, by using a structured link queueing discipline (type-of-service queueing), reduces queueing delays for the delay-sensitive traffic, hence improving the performance of delay-based routing (i.e. more stable routes) and providing this traffic with low delay service. At the same time, the link utilization metric isolates the throughput-sensitive traffic which takes the under-utilized routes, resulting in a better overall network performance. In general, we have shown how in an integrated services environment, routing with some form of non-FCFS scheduling support can provide significant performance improvement.

In addition to simulation studies, we examined this behavior analytically. We obtained stability conditions using the Liapunov function method. We are currently extending our analysis to obtain transient characteristics such as convergence time. Analyzing the interactions between adaptive routing and other link queueing algorithms is an open research area. Future work is also needed to explore the interaction between all components of the congestion control problem, namely scheduling, flow control, and routing, on arbitrary network topologies.

[3] Which means that the necessary conditions to reach isolation are satisfied, i.e., the delay of path 1 carrying all delay-sensitive traffic is less than or equal the delay of path 2 carrying all throughput-sensitive traffic, with path 2's utilization being less than or equal path 1's utilization.

We are also investigating other approaches to adaptive next-hop TOS routing. One approach [25] is to maintain for delay-sensitive traffic two minimum propagation delay routes and use the secondary route when the primary route becomes congested. This approach attempts to avoid the bad effect that queueing delay may have when it dominates measured link delay in delay-based routing. However, it is not clear how to split the delay-sensitive traffic between the two available routes without causing severe oscillations while at the same time reducing queueing delays to provide a low delay service. Further research is needed to explore this area.

References

[1] C. Alaettinoğlu, K. Dussa-Zieger, I. Matta, and A.U. Shankar. MaRS (Maryland Routing Simulator) – Version 1.0 User's Manual. Technical Report UMIACS-TR-91-80, CS-TR-2687, Department of Computer Science, University of Maryland, College Park, MD 20742, June 1991.

[2] P. Almquist. Type of Service in the Internet Protocol Suite. Technical Report RFC-1349, Network Working Group, July 1992.

[3] K-H. Becker and M. Dorfler. *Dynamical Systems and Fractals*. Cambridge University Press, 1989.

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

[5] J. Bolot and A.U. Shankar. A Discrete-Time Stochastic Approach to Flow Control Dynamics. In Proc. *GLOBECOM '92*, Orlando, Florida, December 1992.

[6] R. Callon. Use of OSI IS-IS for routing in TCP/IP and Dual Environments. Technical Report RFC-1195, Digital Equipment Corporation, December 1990.

[7] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In Proc. *ACM SIGCOMM '89*, pages 1–12, Austin, Texas, September 1989.

[8] E. Dijkstra. A Note on Two Problems in Connection with Graphs. *Numer. Math.*, 1:269–271, 1959.

[9] A. Economides and J. Silvester. Optimal Routing in a Network with Unreliable Links. In Proc. *IEEE INFOCOM '88*, pages 288–297, August 1988.

[10] D. Ferrari and D.C. Verma. A Scheme for Real–Time Channel Establishment in Wide–Area Networks. *IEEE JSAC*, 8(3):368–379, April 1990.

[11] S. Floyd and V. Jacobson. The Synchronization of Periodic Routing Messages. In Proc. *ACM SIGCOMM '93*, San Francisco, California, September 1993.

[12] M.L. Gardner, I.S. Loobeek, and S.N. Cohn. Type-of-Service Routing with Loadsharing. In Proc. *GLOBECOM '87*, Tokyo, Japan, November 1987.

[13] R. Jain and S.A. Routhier. Packet Trains - Measurements and A New Model for Computer Network Traffic. *IEEE JSAC*, 4(6):986–995, September 1986.

[14] D. Johnson. NSFnet Report. In Proc. *of the Nineteenth IETF*, pages 377–382, University of Colorado, National Center for Atmospheric Research, December 1990.

[15] A. Khanna and J. Zinky. A Revised ARPANET Routing Metric. In Proc. *ACM SIGCOMM '89*, pages 45–56, September 1989.

[16] I. Matta and A.U. Shankar. Type-of-Service Routing in Dynamic Datagram Networks. CS-TR-3147, University of Maryland, College Park, MD 20742, October 1993. Available by anonymous ftp at ftp.cs.umd.edu:pub/MaRS.

[17] I. Matta and A.U. Shankar. On the Interaction between Gateway Scheduling and Routing. In Proc. *IEEE MASCOTS '94*, pages 84–88, Durham, North Carolina, January 1994. Available by anonymous ftp at ftp.cs.umd.edu:pub/MaRS.

[18] J. McQuillan, I. Richer, and E. Rosen. The New Routing Algorithm for the ARPANET. *IEEE Transactions on Communications*, COM-28(5):711–719, May 1980.

[19] J. Moy. OSPF Version 2. RFC 1247, Network Information Center, SRI International, July 1991.

[20] K. Ogata. *Discrete-Time Control Systems*. Prentice-Hall, Inc., 1987.

[21] A. Parekh. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks. Technical Report LIDS-TR-2089, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 1992.

[22] A.U. Shankar, C. Alaettinoğlu, K. Dussa-Zieger, and I. Matta. Performance Comparison of Routing Protocols under Dynamic and Static File Transfer Connections. *ACM Computer Communication Review*, October 1992.

[23] A.U. Shankar, C. Alaettinoğlu, I. Matta, and K. Dussa-Zieger. Performance Comparison of Routing Protocols using MaRS: Distance-Vector versus Link-State. In Proc. *ACM SIGMETRICS/PERFORMANCE*, volume 20, pages 181–192, Newport, Rhode Island, June 1992.

[24] W. Zhu and S. Chanson. Adaptive Threshold-based Scheduling for Real-Time and Non-Real-Time Traffic. In Proc. *IEEE RTSS '92*, pages 125–135, Phoenix, Arizona, December 1992.

[25] J. Zinky. Private Communication, 1992.