Type-of-Service Routing in Datagram Delivery Systems

Ibrahim Matta, A. Udaya Shankar

Abstract— **The Internet is expected to support various services, including best-effort services and guaranteed services. For best-effort services, we propose a new approach to achieving type-of-service (TOS) classes with adaptive next-hop routing. We consider two TOS classes, namely, delay-sensitive and throughput-sensitive. As in routing protocols such as OSPF and integrated IS-IS, each node has a different next-hop for each destination and TOS class. Traditionally, a node has a single FCFS queue for each outgoing link, and the next-hops are computed using link measurements. In our approach, we attempt to isolate the two traffic classes by using for each outgoing link a separate FCFS queue for each TOS class; the link is shared cyclicly between its 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 the traditional approach using discrete-event simulation and Liapunov analysis (for stability of routes). Our approach offers lower end-to-end delay to the delay-sensitive traffic. A related property is that the routes for the delay-sensitive traffic are more stable, i.e. less oscillations. An unexpected property is that the overall end-to-end delay is lower, because the throughput-sensitive traffic moves away to under-utilized routes.

I. Introduction

With the increasing diversity of network applications, it has become crucial for networks, such as the Internet, to offer various services, including best-effort services and guaranteed services. A guaranteed service provides bounds on performance. A best-effort service can provide qualitatively better service, but without the quantitative bounds of a guaranteed service [1]. The Internet Protocol (IP) currently provides only best-effort services. Extensive effort is underway to extend IP to support other services [2], [3], [1].

The focus of this paper is the provision of different *type-of-service* (TOS) classes of best-effort service. To offer various TOS requirements, e.g. low delay or high throughput, a network's routing protocol should be able to determine appropriate routes for each TOS class. We are concerned here with next-hop (or datagram) routing, because it has proved to be a simple and robust way to do adaptive routing of best-effort traffic [4], [5]. Source (or virtual-circuit) routing on the other hand is often used for guaranteed services.

This work is supported in part by ARPA and Philips Labs under contract DASG60-92-0055 to Department of Computer Science, University of Maryland, and by National Science Foundation Grant No. NCR 89-04590. An abbreviated version of this paper appeared in the proceedings of IEEE INFOCOM '94.

Ibrahim Matta and A. Udaya Shankar are with the Department of Computer Science, University of Maryland, College Park, Maryland 20742. E-mail: {imm, shankar}@cs.umd.edu.

Next-hop TOS routing is done as follows. Each node maintains for each destination node and TOS class, a neighboring node id, referred to as the *next-hop*. Every data packet header contains its destination id and the TOS class of the application. When a node receives a data packet, it forwards the packet to the next-hop for the packet's destination and TOS class. The objective of the routing protocol is to choose next-hops so that the resulting routes satisfy the requested type of service. 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 class, 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 class.

The IP layer of the Internet Protocol suite specifies different TOS classes [6]. Among them are the minimum delay service required for example by interactive traffic or real-time traffic, and the maximum throughput service required for example by bulk transfers such as network mail or FTP. Routing protocols such as the Internet OSPF [7] and the OSI IS-IS [8] provide separate next-hops for each TOS class. 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, typically configured by the network administrator and responding only to failures and recoveries.

To our knowledge, only one approach to adaptive TOS routing has been proposed [9]. This approach, henceforth called **TOS1**, considers two TOS classes: *low delay* and *high throughput*. We refer to traffic of the former class as *delay-sensitive* traffic, and of the latter class as *throughput-sensitive* traffic. TOS1 uses measured link delays 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 class share this queue. Reference [9] refers to simulation studies but does not present any quantitative results.

Traditionally, link queueing has been the FCFS discipline. 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. [10], [11], [12], [13], [14]. In this paper, we investigate the use of a structured queueing discipline with adaptive next-hop TOS routing.

Our approach

We consider the following simple link scheduling discipline, henceforth referred to as **TOS queueing**. We consider two TOS classes: low delay and high throughput. Each node maintains two FCFS queues for each outgoing link, one for each TOS class. The link bandwidth is allocated equally between the two queues in a round-robin fashion. (This is similar to the fair-queueing discipline [10], except that the link bandwidth is divided equally amongst the TOS classes rather than the connections using the link.)

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 [15] where the proportion of delay-sensitive traffic is small compared to the throughput-sensitive traffic. As expected, TOS2 achieves a lower end-to-end delay for delay-sensitive packets since it effectively gives them higher priority. Unexpectedly, TOS2 also yields a lower overall end-to-end delay.

We argue that this is because TOS2 achieves significantly improved routing by exploiting the scheduling structure of TOS 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 behaviors of TOS1 and TOS2, we analyze 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 view this system as a dynamical system [16]. We represent 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 apply the Liapunov function method to derive stability theorems. 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. Starting at a state outside this stability region would lead to simultaneous route oscillations for both traffic classes resulting in a bad delay performance. (A preliminary version of the Liapunov analysis that did not consider propagation delays was presented in [17], [18].)

Organization of the paper

Section II describes our discrete-event simulation model and results. Section III presents our Liapunov stability analysis. Section IV concludes the paper. Appendix I describes details of the simulations, including performance measures, scenarios, and plots. Appendix II contains details of derivations.

II. Discrete-Event Simulations

Our simulation studies were done with a discrete-event simulator, MaRS [19], which has been used for other studies of routing algorithms [20], [21]. Subsection II-A describes the simulation model. Subsection II-B presents general observations about the results.

A. 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.

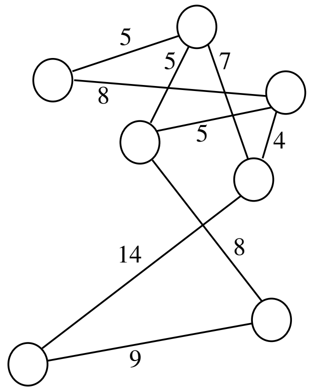


Fig. 1. The "East coast" subset of the NSFNET-T1-Backbone (7 nodes, 9 bidirectional links).

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 TOS queueing. Both TOS1 and UTIL use FCFS queueing. TOS2 uses TOS 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 SPF (Shortest Path First) used in the ARPANET [4] and OSPF (Open SPF) used in the Internet [7]. Each node maintains a time-varying cost (explained below) for each outgoing link and TOS class. Each node also maintains a view of the network topology, with a cost for each TOS class 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 [22] to choose its next-hop for each destination and TOS class. (Using a more scalable mechanism to disseminate link costs would not affect our conclusions.)

The method used to compute link costs for each TOS class 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 TOS1 and TOS2, each node maintains the following two raw-costs for each outgoing link:

- *RawUtilization*: percentage of time the communication channel is busy transmitting a packet; and
- *RawDelay*: 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:

$$\begin{aligned} LinkCost(D) &:= b \times RawDelay + \\ &\qquad (1-b) \times LinkCost(D) \\ LinkCost(T) &:= b \times RawUtilization + \\ &\qquad (1-b) \times LinkCost(T) \end{aligned}$$

where the constant b satisfies $0 < b < 1$.

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

With a utilization-based link cost metric, it seems natural to define the cost of a path as the minimum available link bandwidth (or equivalently, highest link utilization) of the links along the path. However, we found that such a path-cost metric leads to large routing oscillations and instability even at low workload. Therefore, we set the path-cost metric to the sum of the link costs along the path from the source node to the destination node and use Dijkstra's shortest path algorithm as in [23]. (For each of these two path-cost metrics, it is easy to come up with static scenarios where it outperforms the other.)

Regarding workload, this is defined in terms of ⟨source node, destination node⟩ 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 [24]. 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.

B. Observations

In this subsection, we present general observations about the simulation results. Detailed descriptions of scenarios simulated and plots of the observed performance measures are given in Appendix I.

In every scenario, the system behaves in a manner typical of open queueing networks [25]. That is, the throughput equals 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.

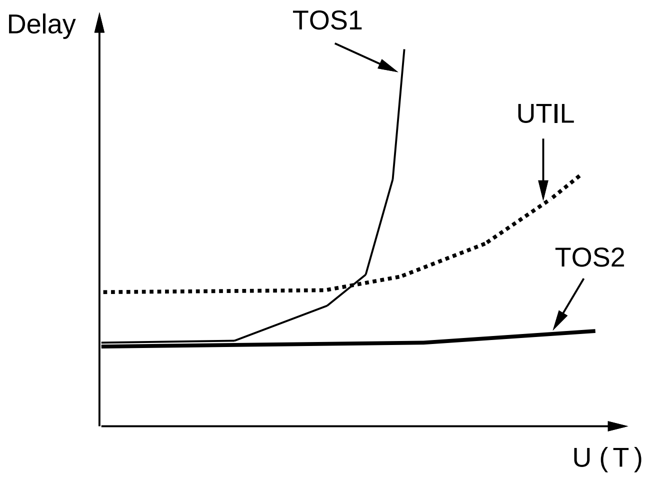


Fig. 2. A generic plot. Delay versus $U(T)$ for a fixed $U(D)$.

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 [5], [23], [26]. 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 [5]. 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 [10], 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 [5]. 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 [5].

III. 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 TOS1 and TOS2. In the following, we give the model and then apply the Liapunov function method to analyze stability.

A. Model

We model a network by a source node sending traffic to a destination node 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. There are N delay-sensitive connections, and M throughput-sensitive connections from the source node to the destination node. For every connection, packets originate at the source node according to a Poisson process, and without loss of generality we assume an arrival rate of 1 packet/time unit (alternatively we could think of the x's and y's below as the total arrival rates).

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 [27]. We assume that (some or all) connections periodically update their routes to the destination node 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 [25] 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 $(k+1)\triangle$. Recall that for an $M/M/1$ queue with offered flow f and service rate μ ($> f$), the delay (queueing + service) equals $1/(\mu - f)$ and the utilization equals f/μ.

For TOS1, with a FCFS discipline at the source node, 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 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:

$$\begin{aligned} x_{k+1} &= \begin{cases} (1-\alpha_k)\, x_k & \text{if } T_{2,k+1} < T_{1,k+1} \\ x_k + \alpha_k\, \bar{x}_k & \text{otherwise} \end{cases} \\ y_{k+1} &= \begin{cases} (1-\alpha_k)\, y_k & \text{if } \rho_{2,k+1} \le \rho_{1,k+1} \\ y_k + \alpha_k\, \bar{y}_k & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

The parameter α_k ($0 < \alpha_k \le 1$) 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 $[\alpha_{MIN}, \alpha_{MAX}]$, where $\alpha_{MAX} - \alpha_{MIN} = 0.2$, and $0.1 < \frac{\alpha_{MAX}+\alpha_{MIN}}{2} \le 0.9$.

For TOS2, with TOS queueing at the source node, the two queues at an output link are correlated, which makes the analysis difficult. A number of approximate solutions for such systems, referred to as 1-limited polling systems, have been proposed. (See [28] for a good survey.) One common approach is to approximate the system by two loosely-coupled $M/M/1$ queues [29], [30]. The service rate of each queue depends on the utilization of the other queue.

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

$$C^{eff}_{i,k} \ge 0.5 C_i \tag{4}$$

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

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

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

$$T_{1,k+1} = \frac{1}{C_{1,k}^{\mathit{eff}} - x_k} + P_1$$
$$T_{2,k+1} = \frac{1}{C_{2,k}^{\mathit{eff}} - \bar{x}_k} + P_2 \quad (6)$$

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), ... is called the *trajectory* of the system [31], [32]. The trajectory may or may not converge to a fixed point (x^*, y^*), i.e. $(x^*, y^*) = I(x^*, y^*)$. The convergence to a fixed point indicates that the system stabilizes into a particular routing pattern, i.e. a particular amount of traffic routed for each class on each path. On the other hand, non-convergence indicates that the system oscillates between different routing patterns (or has chaotic behavior). Note that I is not a continuous mapping, and well-known theorems for convergence requiring this property cannot be directly applied [33].

We represent *isolation* by 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 our iterative method converges to the fixed point $(N, 0)$. In the next section, we first derive sufficient conditions for the system to reach isolation as a function of the starting state. (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.) We then obtain less refined sufficient conditions for isolation, independent of the starting state.

B. Liapunov Function Method

We use the Liapunov function method [32] 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:

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

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

$$\bar{x}_{k+1} = (1 - \alpha_k)\ \bar{x}_k + \alpha_k\ N\ \delta_k$$
$$y_{k+1} = (1 - \alpha_k)\ y_k + \alpha_k\ M\ \beta_k \quad (8)$$

where

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

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:

$$\frac{1}{C_1 - N} + P_1 \leq \frac{1}{C_2 - M} + P_2$$
$$\frac{M}{C_2} \leq \frac{N}{C_1} \quad (10)$$

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.[1]

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

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

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 III.1: $V(\bar{x}, y)$ is a Liapunov function in D_1. That is: $V(\bar{x}, y)$ is positive definite; and for all $(\bar{x}_k, y_k)\ \epsilon\ D_1 - \{(0, 0)\}$, $\triangle V(\bar{x}_k, y_k) < 0$ and $(\bar{x}_{k+1}, y_{k+1})\ \epsilon\ D_1$. □

[1] Although the equation of the left boundary of the domain D_1 is quadratic in x and y, the resulting solution turns out to be a straight line because one of the roots is infeasible. This is confirmed by monte-carlo simulations. See numerical example in Subsection III-C.

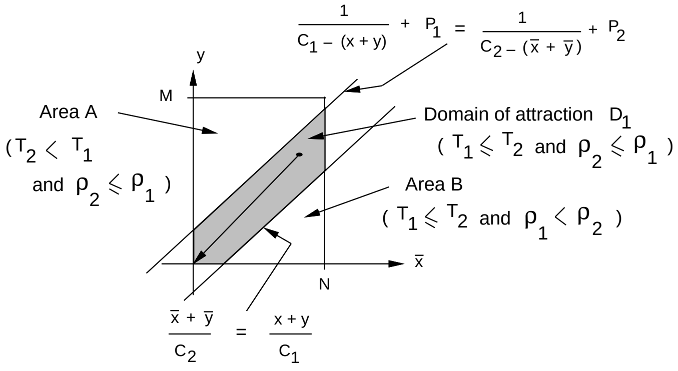


Fig. 3. Domain of attraction for TOS1.

Proof. Since $V(\bar{x}, y) > 0$ for all $(\bar{x}, y) \neq (0, 0)$ and $V(0, 0) = 0$, then $V(\bar{x}, y)$ is positive definite.

The forward difference $\triangle V(\bar{x}_k, y_k)$ is computed as follows.

$$
\begin{aligned}
\triangle V(\bar{x}_k, y_k) &= V(\bar{x}_{k+1}, y_{k+1}) - V(\bar{x}_k, y_k) \\
&= ((1 - \alpha_k)\ \bar{x}_k + \alpha_k\ N\ \delta_k)^2 + \\
&\quad ((1 - \alpha_k)\ y_k + \alpha_k\ M\ \beta_k)^2 - \\
&\quad (\bar{x}_k^2 + y_k^2) \\
&= -[1 - (1 - \alpha_k)^2]\ \bar{x}_k^2 - \\
&\quad [1 - (1 - \alpha_k)^2]\ y_k^2 + \\
&\quad (\alpha_k\ N\ \delta_k)^2 + (\alpha_k\ M\ \beta_k)^2 + \\
&\quad 2(1 - \alpha_k)\ \alpha_k\ N\ \bar{x}_k\ \delta_k + \\
&\quad 2(1 - \alpha_k)\ \alpha_k\ M\ y_k\ \beta_k \qquad (12)
\end{aligned}
$$

Consider a point $(\bar{x}_k, y_k)\ \epsilon\ D_1 - \{(0, 0)\}$. Then $\delta_k = \beta_k = 0$. From equation (12), since $0 < [1 - (1 - \alpha_k)^2] \leq 1$, we have $\triangle V(\bar{x}_k, y_k) < 0$. (This is true regardless of the randomness of α_k.) Substituting $\delta_k = \beta_k = 0$ in equations (8), we get $\bar{x}_{k+1} - y_{k+1} = (1 - \alpha_k)(\bar{x}_k - y_k)$.

Consider the case where $\bar{x}_k - y_k < 0$. Since $0 \leq (1 - \alpha_k) < 1$, we have $\frac{1}{C_1 - N + (\bar{x}_{k+1} - y_{k+1})} + P_1 < \frac{1}{C_1 - N + (\bar{x}_k - y_k)} + P_1$, and $\frac{1}{C_2 - M - (\bar{x}_k - y_k)} + P_2 < \frac{1}{C_2 - M - (\bar{x}_{k+1} - y_{k+1})} + P_2$. Hence, since $\frac{1}{C_1 - N + (\bar{x}_k - y_k)} + P_1 \leq \frac{1}{C_2 - M - (\bar{x}_k - y_k)} + P_2$, we see that $\frac{1}{C_1 - N + (\bar{x}_{k+1} - y_{k+1})} + P_1 \leq \frac{1}{C_2 - M - (\bar{x}_{k+1} - y_{k+1})} + P_2$.

Now, consider the case where $\bar{x}_k - y_k > 0$. We have $\frac{1}{C_1 - N + (\bar{x}_k - y_k)} + P_1 < \frac{1}{C_1 - N} + P_1$, and $\frac{1}{C_2 - M} + P_2 < \frac{1}{C_2 - M - (\bar{x}_k - y_k)} + P_2$. Since $0 \leq (1 - \alpha_k) < 1$, and $\frac{1}{C_1 - N} + P_1 \leq \frac{1}{C_2 - M} + P_2$ from equation (10), we see that $\frac{1}{C_1 - N + (\bar{x}_{k+1} - y_{k+1})} + P_1 \leq \frac{1}{C_2 - M - (\bar{x}_{k+1} - y_{k+1})} + P_2$.

Similarly, we see that $\frac{M + (\bar{x}_{k+1} - y_{k+1})}{C_2} \leq \frac{N - (\bar{x}_{k+1} - y_{k+1})}{C_1}$. Therefore, $(\bar{x}_{k+1}, y_{k+1})\ \epsilon\ D_1$, and $\delta_{k+1} = \beta_{k+1} = 0$. Consequently, starting at any point in D_1, the trajectory stays inside D_1 approaching the origin (as shown in Figure 3). □

From lemma III.1 and the Liapunov stability theory [32], we have the following theorem:

Theorem III.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 [16], [33] because it constitutes a set of starting states for which the iteration converges to the origin. In D_1, the iteration is said to be a *contraction*, since $V(\bar{x}_{k+1}, y_{k+1}) < V(\bar{x}_k, y_k)$ for all $(\bar{x}_k, y_k) \neq (0, 0)$ along the trajectory. It is important to observe that the domain of attraction contains all the system states for which $T_1 \leq T_2$ and $\rho_2 \leq \rho_1$. Also, note that starting at any point in D_1, $\delta_k = \beta_k = 0$ for all $(\bar{x}_k, y_k)$ along the trajectory.

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

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

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

$$
\begin{aligned}
\frac{1}{C_1 - N} + P_1 &\leq \frac{1}{C_2 - 0.5M} + P_2 \\
\frac{M}{C_2} &\leq \frac{N}{C_1}
\end{aligned}
\qquad (13)
$$

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. The goal is to show that starting at any point in D_2, $\delta_k = \beta_k = 0$ for all $(\bar{x}_k, y_k)$ along the trajectory. $\frac{1}{C_{1,k}^{eff} - N + \bar{x}_k} + P_1 \leq \frac{1}{C_{2,k}^{eff} - \bar{x}_k} + P_2$ implies $\delta_k = 0$. $\frac{M + (\bar{x}_k - y_k)}{C_2} \leq \frac{N - (\bar{x}_k - y_k)}{C_1}$ implies $\beta_k = 0$. Because the expressions for $C_{i,k}^{eff}$ are hard to work with, we try to find simpler expressions, say $C_{i,k}^{simp}$, such that $\frac{1}{C_{1,k}^{eff} - N + \bar{x}_k} + P_1 \leq \frac{1}{C_{1,k}^{simp} - N + \bar{x}_k} + P_1$, and $\frac{1}{C_{2,k}^{simp} - \bar{x}_k} + P_2 \leq \frac{1}{C_{2,k}^{eff} - \bar{x}_k} + P_2$ (then $\frac{1}{C_{1,k}^{simp} - N + \bar{x}_k} + P_1 \leq \frac{1}{C_{2,k}^{simp} - \bar{x}_k} + P_2$ implies $\delta_k = 0$). We would then define $D_2 = \{(\bar{x}, y) : \frac{1}{C_1^{simp} - N + \bar{x}} + P_1 \leq \frac{1}{C_2^{simp} - \bar{x}} + P_2 \wedge \frac{M + (\bar{x} - y)}{C_2} \leq \frac{N - (\bar{x} - y)}{C_1} \wedge 0 \leq \bar{x} \leq N \wedge 0 \leq y \leq M\}$, and attempt to show that $V(\bar{x}, y)$, defined in (11), is a Liapunov function in D_2.

To do this, we need an upper bound on $C_{2,k}^{eff}$, and a lower bound on $C_{1,k}^{eff}$. From equations (5), we see that $C_{2,k}^{eff} \leq C_2 - 0.5(\bar{y}_k - \bar{x}_k)$. We could not find an appropriate lower bound on $C_{1,k}^{eff}$. So we made the approximation $C_{1,k}^{eff} \approx C_1 - 0.5 y_k$ (the accuracy of this is discussed below).

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

Lemma III.2: Assuming $C_{1,k}^{eff} \approx C_1 - 0.5 y_k$, $V(\bar{x}, y)$ is a

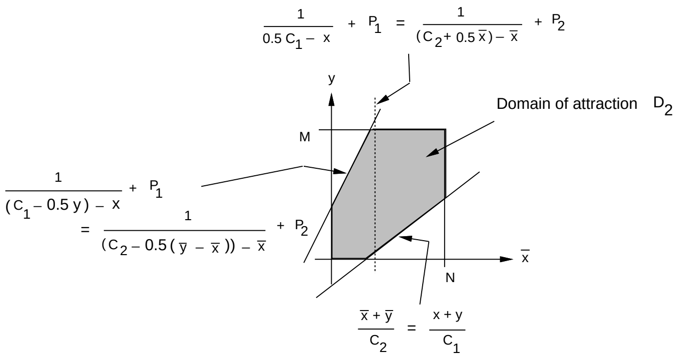


Fig. 4. Domain of attraction for TOS2.

Liapunov function in D_2. □

The proof of the above lemma is similar to the proof of Lemma III.1, and is given in Appendix II. From lemma III.2 and the Liapunov stability theory [32], we have the following theorem:

Theorem III.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. □

C. Effect of α_k on system behavior

The domains of attraction we have just found (cf. Theorems III.1 and III.2) are not the largest, i.e. there 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. The proof is given in Appendix II.

Theorem III.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

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

□

Theorem III.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 [34].)

Consider the simple case where $\alpha_k = 1$, for all k. It can be seen that isolation, whenever possible[2], provides the optimal performance for both traffic classes [35]. 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 II-B.

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 $C^{eff}_{1,k} \approx C_1 - 0.5y_k$, which was made in Theorem III.2 for TOS2. In fact, it can be shown that $C^{eff}_{1,k} \leq C_1 - 0.5y_k$. Regardless of that, we found that our approximation is only slightly optimistic. In particular, our monte-carlo simulations [17], [18] show that starting at any point in D_2 satisfying $\frac{1}{0.5C_1 - N + \bar{x}} + P_1 \leq \frac{1}{(C_2 + 0.5\bar{x}) - \bar{x}} + P_2$, the iteration indeed leads to the origin. Those points also satisfy $\frac{1}{(C_1 - 0.5y) - N + \bar{x}} + P_1 \leq \frac{1}{(C_2 - 0.5(\bar{y} - \bar{x})) - \bar{x}} + P_2$. This is because $\frac{1}{(C_1 - 0.5y) - N + \bar{x}} + P_1 \leq \frac{1}{0.5C_1 - N + \bar{x}} + P_1$ (with equality occurring when $y = C_1$), and $\frac{1}{(C_2 + 0.5\bar{x}) - \bar{x}} + P_2 \leq \frac{1}{(C_2 - 0.5(\bar{y} - \bar{x})) - \bar{x}} + P_2$. Some points in D_2 not satisfying $\frac{1}{0.5C_1 - N + \bar{x}} + P_1 \leq \frac{1}{(C_2 + 0.5\bar{x}) - \bar{x}} + P_2$ may not, however, lead to the origin. 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.

Figures 5 and 6 show the domains of attraction corresponding to isolation for TOS1 and TOS2, respectively, obtained by monte-carlo simulations for $C_1 = 30, C_2 = 50, P_1 = 0.5, P_2 = 1, N = 20, M = 30$, and $\alpha_k \in [0.8, 1]$.

Theorem III.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. Figure 7 depicts the *load region*, i.e. values of (N, M), defined by $\{(N, M) : \frac{1}{C_1 - N} + P_1 \leq \frac{1}{C_2} + P_2 \wedge \frac{M}{C_2} \leq \frac{N}{C_1}\}$. In this region, the necessary conditions for isolation, defined in (10) and (13), for both TOS1 and TOS2 are satisfied. This region thus defines sufficient conditions to reach isolation for low enough α_k for both TOS1 and TOS2. This is in agreement with our monte-

[2] 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.

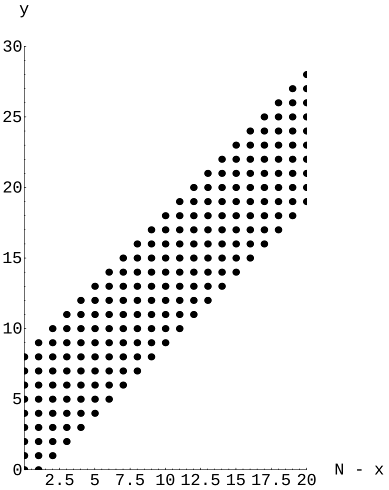


Fig. 5. Numerical example: Domain of attraction for TOS1.

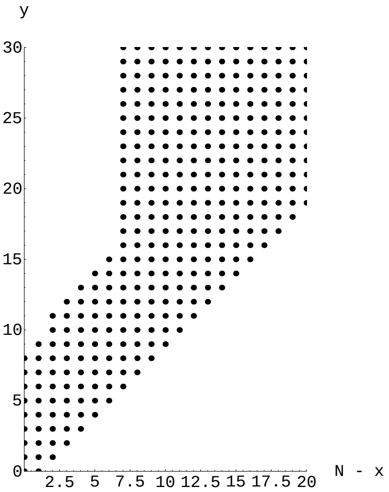


Fig. 6. Numerical example: Domain of attraction for TOS2.

carlo simulations obtained in [17], [18] which also show that in this region, TOS2 gives much better transient performance than TOS1, i.e. less oscillations and much faster convergence to isolation (we have not yet been able to obtain transient measures analytically).

D. Sufficient conditions independent of the starting state

In Theorems III.1 and III.2, we derived sufficient conditions for the system to reach isolation as a function of the starting state. Now, using these theorems, we derive sufficient conditions for isolation independent of the starting state and the values of α_k.[3] Figure 8 depicts the load regions defined by these sufficient conditions.

Theorem III.4:

(i) For TOS1, the conditions $N+M \leq C_1 - \frac{1}{(\frac{1}{C_2}+P_2-P_1)}$ and $\frac{M}{C_2} \leq \frac{N}{C_1}$ (regions B and C in Figure 8) are sufficient for isolation (regardless of the starting state and the values of α_k).

[3] These conditions were obtained in [17], [18] in a different way using a worst-case analysis.

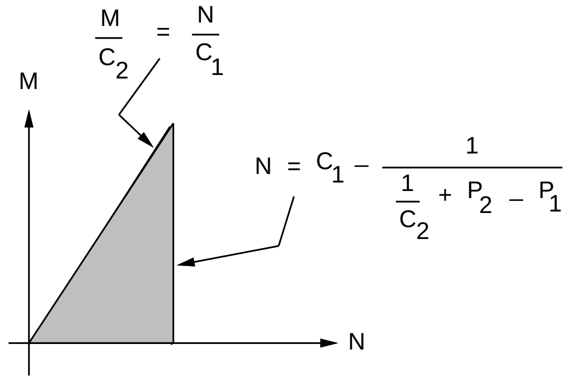


Fig. 7. Load region for isolation for low enough α_k for both TOS1 and TOS2.

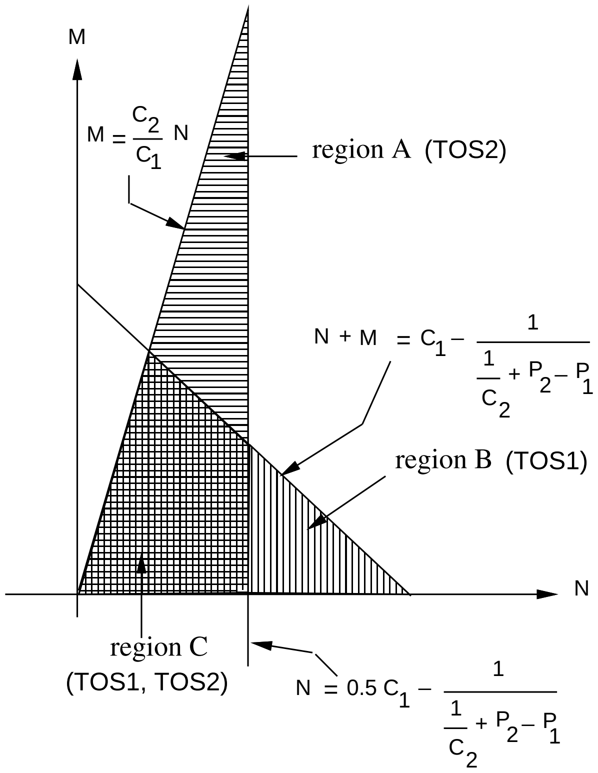


Fig. 8. Load regions satisfying sufficient conditions for isolation independent of starting state.

(ii) For TOS2, the corresponding conditions are $N \leq 0.5C_1 - \frac{1}{(\frac{1}{C_2}+P_2-P_1)}$ and $\frac{M}{C_2} \leq \frac{N}{C_1}$ (regions A and C in Figure 8).

□

The proof of Theorem III.4 is given in Appendix II. Referring to Figure 8, we observe that for N less than $0.5C_1 - \frac{1}{(\frac{1}{C_2}+P_2-P_1)}$, TOS2 provides isolation (regardless of the starting state and the values of α_k) for higher values of M than TOS1. This effect becomes more pronounced as C_2 increases.

Note that the conditions of Theorem III.4 capture less information than the ones given by Theorems III.1 and III.2; i.e. there can be a system configuration that does not satisfy the sufficient conditions of Theorem III.4, but still reaches isolation for some starting states or values of α_k.

IV. 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 structured link queueing, reduces queueing delays for the delay-sensitive traffic, hence improving the performance of delay-based routing and providing more stable routes for this traffic. 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 this paper, we assumed unbounded buffer space at the links. In practice, the buffer space would have to be allocated to the different TOS classes. We would expect the same conclusions to hold. We also assumed a datagram (best-effort) service model of the sort currently used in the Internet. Extensive effort is currently underway to extend this Internet service model to support other services, including guaranteed service [2], [3], [1]. In such an extended service environment, our findings would still apply to traffic classes requiring best-effort service. In addition, our approach appears scalable as each router needs to maintain information only about each TOS class rather than each connection. This also makes the implementation of the non-FCFS queueing discipline in the routers simple and inexpensive.

In addition to simulation studies, we examined the behavior of the TOS schemes 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 more detailed network models with arbitrary topologies.

It is obvious that TOS2 relies upon the delay-sensitive traffic being a small proportion of the overall traffic. When the proportion of delay-sensitive traffic is not small, TOS2 performs worse than TOS1 because the delay-sensitive traffic in TOS2 suffers from higher queueing delay than in TOS1 [36]. However, the performance of TOS2 can be restored by simply allowing, for example, the delay-sensitive queue to get two turns for every turn of the throughput-sensitive queue. The natural extension of this would be to give delay-sensitive packets absolute priority over throughput-sensitive packets. However, giving delay-sensitive packets absolute priority may not be desirable because throughput-sensitive packets may not get transmitted (i.e. blocked) causing queues of this traffic class to build up, and consequently providing low throughput for the traffic class requesting high throughput service. Future work remains to similarly analyze such situations and propose a scheduling discipline that dynamically allocates turns to the TOS queues based on traffic estimates.

There might be other possible approaches to adaptive next-hop TOS routing. One approach [37] 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.

It is interesting to observe the relationship between our work here and work in the virtual-circuit literature. We took the concept of isolating link resources, often used in virtual-circuit models, and applied it to a datagram model. This induced isolation of routes for the TOS classes, resulting in an improved overall network performance. It would be interesting to see if the concept of route isolation is also applicable in virtual-circuit models. A related idea in virtual-circuit models is the packing of low-bandwidth connections over a subset of the routes leaving the remaining routes free for high-bandwidth connections, thus in effect isolating both types of connections [38].

Regarding the analysis techniques, in datagram models it is reasonable to make the quasi-static assumption that steady-state is reached between two successive routing updates. This is because packet transmission times are small compared to the routing update interval. This assumption does not seem reasonable for virtual-circuit models because the lifetimes of connections are typically larger than the routing update interval. This suggests that analysis techniques used here might not be applicable to virtual-circuit models.

Appendix

I. Simulation Details

In this appendix, we describe the simulation parameters, and the performance measures. We also give details of simulated scenarios and plots.

Parameters

All links have bandwidths of 1.5Mbit/sec for the low-speed version, and 100Mbit/sec for the high-speed version. Each node's outgoing link costs are updated regularly, with inter-update time uniformly distributed with mean 10 seconds [4] and standard deviation 1 second.[4] The factor b, used in the link cost calculation, is 0.8.[5] A data source is represented as a Markov chain with two states: a busy state

[4] Update intervals at different nodes are independent.

[5] We chose the factor 0.8 after experimenting with other values. A small value such as 0.5 makes the routing algorithm adapt slowly. Whereas a high value such as 1.0 may result in an unstable behavior.

and an idle state. In the busy state, the source produces a (geometrically distributed) number of data packets with some constant inter-packet generation time. The source then stays idle for an exponentially distributed duration before starting the transmission of the next train (burst) of packets. (This traffic model has been used in many studies, e.g. [39].) Unless otherwise indicated, all sources have the following parameters: for the low-speed case, the data packet length equals 128 bytes, the inter–packet generation time is 150 msec, the average train size is 100 packets, and the average idle duration is 2 seconds (this corresponds to an average packet rate of about 0.006 packet/msec.); for the high-speed case, the data packet length is 5000 bytes, the inter–packet generation time is 50 msec, the average train size is 1000 packets, and the average idle duration is 2 seconds (this corresponds to an average packet rate of about 0.02 packet/msec.).

Performance Measures

We consider average measures of throughput, delay and load. An average measure is based on statistics collected over a large measurement interval, which is the duration of the simulation except for an initial "startup interval" (to eliminate transient effects due to empty initial network). Thus:

- *Throughput.* Total number of data bytes received at destinations during the measurement interval divided by the length of the measurement interval.
- *Delay.* Total delay of all data packets received at destinations during the measurement interval divided by the number of those data packets, where *delay of a data packet* is defined to be the time difference between sending a packet and receiving it at the corresponding destination.
- *Data Load.* Fraction of the network capacity, i.e. sum of all link capacities, used by data packets during the measurement interval.
- *Throughput(T).* Total number of data bytes received at destinations for the throughput-sensitive traffic during the measurement interval divided by the length of the measurement interval.
- *Delay(D).* Total delay of all delay-sensitive data packets received at destinations during the measurement interval divided by the number of those data packets.

Results

Here, we present details of scenarios simulated along with plots of the observed (steady–state) performance measures, namely *throughput*, *delay*, *throughput*(T), *delay*(D), and *data load*. In our simulations, 95% confidence intervals were computed using the method of independent replications [40], [41]. In particular, a measure, say x, is obtained as $\frac{x_1+x_2+\cdots+x_n}{n}$, where the $x_1, x_2, \ldots, x_n$ are the measures obtained using the different independent simulation runs. In all cases, the size of the confidence intervals is less than 2% of the mean.

We now present our simulation results. Although we show results only for uniform workload, we obtained similar results for the skewed workload we investigated. We also obtained similar results for workload that has different parameters for each traffic type. In case (C) below, we consider a smaller packet size for the delay-sensitive sources, namely 64 bytes rather than 128 bytes. We have scaled the delay plots for clarity, so delay values higher than 100 milliseconds are not shown.

(A) Low-speed, varying U(T), fixed U(D), equal packet sizes.

Figure 9 shows *delay*(D) and *delay* versus $U(T)$ in the range 20 to 26, for a fixed $U(D) = 8$. The delay-sensitive traffic constitutes almost 25% of the total traffic. Ignoring differences at low values of $U(T)$, TOS1 performs the worst, becoming saturated around $U(T) = 22$ (corresponding to almost 60% data load). Interestingly, UTIL which does *not* use any TOS facility performs better than TOS1. TOS2 performs the best. At $U(T) = 24$, UTIL's *delay* is 53% higher than TOS2's, and TOS1's *delay* is 1322% higher than TOS2's. Note that UTIL performs worse than TOS2 over the whole range of $U(T)$. At low values of $U(T)$, UTIL performs the worst. At $U(T) = 20$, UTIL has about 32% higher *delay* than both TOS2 and TOS1.

Figure 10 shows *data load*, *throughput*(T), and *throughput* versus $U(T)$. Observe that the data load for TOS1 increases as TOS1 becomes saturated. This indicates the use of longer routes, and consequently higher delay. Both *throughput*(T) and *throughput* increase linearly with the workload, and they are the same for all schemes. This shows that the system is stable for all schemes.

(B) High-speed, varying U(T), fixed U(D), equal packet sizes.

Figure 11 shows *delay*(D) and *delay* versus $U(T)$ in the range 8 to 18, for a fixed $U(D) = 4$. The delay-sensitive traffic constitutes almost 25% of the total traffic. As observed in the low-speed case (A), ignoring differences at low values of $U(T)$, TOS2 performs the best, followed by UTIL, and then TOS1. TOS1 reaches saturation sooner, around $U(T) = 16$ (corresponding to a data load of almost 61%). TOS1 has about 758% higher *delay* than TOS2 at $U(T) = 18$. Note that for the same data load, the difference in delay is less significant than in case (A). This is due to the fact that in a high-speed network, queueing delay is less significant due to small transmission times. For example, in the high-speed network, transmission time of the 5000-byte data packet on the 100Mbit/s link is 0.0004 sec. Whereas in the low-speed network, transmission time of the 128-byte data packet on the 1.5Mbit/s link is 0.7 sec. This fact reduces the effect of bad oscillations inherent in delay-based routing when queueing delays are significant.

Figure 12 shows *data load*, *throughput*(T) and *throughput* versus $U(T)$. Again, as in case (A), *throughput*(T) and *throughput* increase linearly with the workload for all schemes. Henceforth, we do not show plots for throughput. We also do not show plots for UTIL since the utilization-based metric, as we have observed, does not necessarily result in minimum delay routes.[6]

(C) Low-speed, varying U(T), fixed U(D), unequal packet sizes.

Figure 13 shows *delay*(D) and *delay* versus $U(T)$ in the range 16 to 24, for a fixed $U(D) = 16$. The delay-sensitive traffic constitutes almost 28% of the total traffic. As observed in (A), TOS2 performs better than TOS1. TOS1 reaches saturation sooner, around $U(T) = 18$ (corresponding to a data load of almost 55%). TOS1 has about 1800% higher *delay* than TOS2 at $U(T) = 20$.

Figure 14 shows *data load* versus $U(T)$. Observe that TOS1 reaches saturation at a data load which is smaller than in (A). This is because here delay-sensitive traffic has smaller packet sizes, thus suffering higher delays with TOS1.

II. Proofs and Derivations

Computation of effective capacities

Let's consider two queues Q_X and Q_Y with arrival rates X and Y, respectively, *equally* sharing a link with capacity C. Define C_X^{eff} and C_Y^{eff} as the effective link capacity available for traffic X and Y, respectively. Also, let n_X and n_Y be the number of packets in Q_X and Q_Y, respectively. Thus,

$$\begin{aligned} C_X^{eff} &= C \times Prob[n_Y = 0] + 0.5C \times Prob[n_Y > 0] \\ C_Y^{eff} &= C \times Prob[n_X = 0] + 0.5C \times Prob[n_X > 0] \end{aligned}$$

Note that $C_X^{eff} \geq 0.5C$ ($C_Y^{eff} \geq 0.5C$), with the worst-case occurring when $Q_Y(Q_X)$ is *always* not empty.

Assuming each queue is $M/M/1$, and substituting

$$\begin{aligned} Prob[n_X = 0] &= 1 - Prob[n_X > 0] = 1 - X/C_X^{eff}, \\ Prob[n_Y = 0] &= 1 - Prob[n_Y > 0] = 1 - Y/C_Y^{eff} \end{aligned}$$

we obtain two equations in the two unknowns C_X^{eff} and C_Y^{eff}. Solving them, we get

$$\begin{aligned} C_X^{eff} &= \frac{(C - 0.5(Y-X)) + \sqrt{(C-0.5(Y-X))^2 - 2CX}}{2} \\ C_Y^{eff} &= C_X^{eff} + 0.5(Y - X) \end{aligned}$$

Proof of Lemma III.2

Consider a point $(\bar{x}_k, y_k)$ ϵ $D_2 - \{(0,0)\}$. $\frac{1}{(C_1-0.5y_k)-N+\bar{x}_k} + P_1 \leq \frac{1}{(C_2-0.5(\bar{y}_k-\bar{x}_k))-\bar{x}_k} + P_2$ implies $\frac{1}{C_{1,k}^{eff}-N+\bar{x}_k} + P_1 \leq \frac{1}{C_{2,k}^{eff}-\bar{x}_k} + P_2$, and hence $\delta_k = 0$. Let's rewrite $\frac{1}{(C_1-0.5y_k)-N+\bar{x}_k} + P_1 \leq \frac{1}{(C_2-0.5(\bar{y}_k-\bar{x}_k))-\bar{x}_k} + P_2$ as $\frac{1}{C_1-N+(\bar{x}_k-0.5y_k)} + P_1 \leq \frac{1}{C_2-0.5M-0.5(\bar{x}_k-y_k)} + P_2$. Since $\delta_k = \beta_k = 0$ then $\bar{x}_{k+1} - 0.5y_{k+1} = (1-\alpha_k)(\bar{x}_k - 0.5y_k)$, and $\bar{x}_{k+1} - y_{k+1} = (1-\alpha_k)(\bar{x}_k - y_k)$.

Consider the case where $\bar{x}_k - 0.5y_k < 0$. This implies that $\bar{x}_k - y_k < 0$. Since $0 \leq (1-\alpha_k) < 1$, we have $\frac{1}{C_1-N+(\bar{x}_{k+1}-0.5y_{k+1})} + P_1 < \frac{1}{C_1-N+(\bar{x}_k-0.5y_k)} + P_1$, and $\frac{1}{C_2-0.5M-0.5(\bar{x}_k-y_k)} + P_2 < \frac{1}{C_2-0.5M-0.5(\bar{x}_{k+1}-y_{k+1})} + P_2$. Hence, since $\frac{1}{C_1-N+(\bar{x}_k-0.5y_k)} + P_1 \leq \frac{1}{C_2-0.5M-0.5(\bar{x}_k-y_k)} + P_2$, we see that $\frac{1}{C_1-N+(\bar{x}_{k+1}-0.5y_{k+1})} + P_1 \leq \frac{1}{C_2-0.5M-0.5(\bar{x}_{k+1}-y_{k+1})} + P_2$.

Now, consider the case where $\bar{x}_k - 0.5y_k > 0$. In this case, either $\bar{x}_k - y_k > 0$ or $\bar{x}_k - y_k < 0$. First, consider the case where $\bar{x}_k - 0.5y_k > 0$ and $\bar{x}_k - y_k > 0$. We have $\frac{1}{C_1-N+(\bar{x}_k-0.5y_k)} + P_1 < \frac{1}{C_1-N} + P_1$, and $\frac{1}{C_2-0.5M} + P_2 < \frac{1}{C_2-0.5M-0.5(\bar{x}_k-y_k)} + P_2$. Since $0 \leq (1-\alpha_k) < 1$, and $\frac{1}{C_1-N} + P_1 \leq \frac{1}{C_2-0.5M} + P_2$ from equation (13), we see that $\frac{1}{C_1-N+(\bar{x}_{k+1}-0.5y_{k+1})} + P_1 \leq \frac{1}{C_2-0.5M-0.5(\bar{x}_{k+1}-y_{k+1})} + P_2$. We also see that the latter inequality also holds for the case where $\bar{x}_k - 0.5y_k > 0$ and $\bar{x}_k - y_k < 0$.

Therefore, since in all cases, $\frac{1}{(C_1-0.5y_{k+1})-N+\bar{x}_{k+1}} + P_1 \leq \frac{1}{(C_2-0.5(\bar{y}_{k+1}-\bar{x}_{k+1}))-\bar{x}_{k+1}} + P_2$ then $\frac{1}{C_{1,k+1}^{eff}-N+\bar{x}_{k+1}} + P_1 \leq \frac{1}{C_{2,k+1}^{eff}-\bar{x}_{k+1}} + P_2$. Similarly, we see that $\frac{M+(\bar{x}_{k+1}-y_{k+1})}{C_2} \leq \frac{N-(\bar{x}_{k+1}-y_{k+1})}{C_1}$. Therefore, $(\bar{x}_{k+1}, y_{k+1})$ ϵ D_1, and $\delta_{k+1} = \beta_{k+1} = 0$. Thus, we see that the iteration is a contraction in D_2.

Proof of Theorem III.3

First, we note that $\frac{M+(\bar{x}_k-y_k)}{C_2} \leq \frac{N-(\bar{x}_k-y_k)}{C_1}$ iff $\bar{x}_k - y_k \leq L_2$. Also, $\frac{1}{C_1-N+(\bar{x}_k-y_k)} + P_1 \leq \frac{1}{C_2-M-(\bar{x}_k-y_k)} + P_2$ iff $\bar{x}_k - y_k \geq L_1$. $L_1 \leq 0$ and $L_2 \geq 0$ are necessary for the domain of attraction D_1 to surround the fixed point $(0,0)$ and thus convergence to be possible.[7]

If $\bar{x}_k - y_k < L_1$ (i.e. $\delta_k = 1, \beta_k = 0$) then $\bar{x}_{k+1} - y_{k+1} = (1-\alpha_k)\,(\bar{x}_k - y_k) + \alpha_k\ N \geq -(1-\alpha_{MIN})\ M + \alpha_{MIN}\ N$. If $\bar{x}_k - y_k > L_2$ (i.e. $\delta_k = 0, \beta_k = 1$) then $\bar{x}_{k+1} - y_{k+1} = (1-\alpha_k)\,(\bar{x}_k - y_k) - \alpha_k\ M \leq (1-\alpha_{MIN})\ N - \alpha_{MIN}\ M$. Thus, the following conditions cause TOS1 to lock into a limit cycle for any $(\bar{x}_0, y_0)$ not in the domain of attraction D_1:

$$\begin{aligned} -(1-\alpha_{MIN})\ M + \alpha_{MIN}\ N &> L_2 \\ (1-\alpha_{MIN})\ N - \alpha_{MIN}\ M &< L_1 \end{aligned}$$

This implies $\alpha_{MIN} > max(\frac{N-L_1}{M+N}, \frac{M+L_2}{M+N})$, and part ($i$) is proved.

If $\bar{x}_k - y_k < L_1$ then $\bar{x}_{k+1} - y_{k+1} = (1-\alpha_k)\,(\bar{x}_k - y_k) + \alpha_k\ N \leq \alpha_k\ N \leq \alpha_{MAX}\ N$. We also have $\bar{x}_{k+1} - y_{k+1} > \bar{x}_k - y_k$.

If $\bar{x}_k - y_k > L_2$ then $\bar{x}_{k+1} - y_{k+1} = (1-\alpha_k)\,(\bar{x}_k - y_k) - \alpha_k\ M \geq -\alpha_k\ M \geq -\alpha_{MAX}\ M$. We also have $\bar{x}_{k+1} - y_{k+1} < \bar{x}_k - y_k$.

Thus, the following conditions force the iteration to eventually enter the domain of attraction D_1 and converge to the origin for any $(\bar{x}_0, y_0)$:

$$\begin{aligned} \alpha_{MAX}\ N &\leq L_2 \\ -\alpha_{MAX}\ M &\geq L_1 \end{aligned}$$

This implies $\alpha_{MAX} \leq min(\frac{L_2}{N}, \frac{-L_1}{M})$, and part ($ii$) is

[6] We have also studied a non-TOS scheme similar to UTIL except that it uses the delay metric rather than the utilization metric (i.e. $LinkCost(D)$ is used for all traffic); however we do not show plots for it since it performed badly at heavy load, as expected.

[7] We note that $L_2 \leq N$ since otherwise we get $-M > N$, which contradicts the fact that $N, M \geq 0$. Referring to Figure 3, L_1 is assumed to be greater than $-M$.

proved.

Proof of Theorem III.4

Referring to Figure 3 for TOS1, if the point $(\bar{x}, y) = (0, M)$ is inside the domain of attraction, i.e. $(\bar{x}, y) = (0, M)$ satisfies $\frac{1}{C_1-(x+y)} + P_1 \leq \frac{1}{C_2-(\bar{x}+\bar{y})} + P_2$, then we have $N + M \leq C_1 - \frac{1}{(\frac{1}{C_2}+P_2-P_1)}$. This implies $T_{1,k} \leq T_{2,k}$, for every k. This ensures that path 1 is always attractive to delay-sensitive traffic, and eventually all delay-sensitive traffic will be on path 1.

Given that all delay-sensitive connections remain on path 1, we see from (2) and (3) that the condition $\frac{M}{C_2} \leq \frac{N}{C_1}$ is enough for all throughput-sensitive traffic to eventually move to path 2. This proves part (i).

Referring to Figure 4 for TOS2, if $(\bar{x}, y) = (0, 0)$ is inside the domain of attraction, i.e. $(\bar{x}, y) = (0, 0)$ satisfies $\frac{1}{0.5C_1-x} + P_1 \leq \frac{1}{(C_2+0.5\bar{x})-\bar{x}} + P_2$, then we have $N \leq 0.5C_1 - \frac{1}{(\frac{1}{C_2}+P_2-P_1)}$. This implies $T_{1,k} \leq T_{2,k}$, for every k. Therefore, similar to part (i), we see that the conditions $N \leq 0.5C_1 - \frac{1}{(\frac{1}{C_2}+P_2-P_1)}$, and $\frac{M}{C_2} \leq \frac{N}{C_1}$ are sufficient for isolation. This proves part (ii).

Acknowledgment

We would like to thank the anonymous reviewers and the editor for their constructive comments, which have improved the overall presentation and the concluding discussion of the paper.

References

[1] H. Schulzrinne, J. Kurose, and D. Towsley, "An evaluation of scheduling mechanisms for providing best-effort real-time communication in wide-area networks", in *IEEE INFOCOM*, Toronto, Ontario, Canada, June 1994, pp. 1352–1361.

[2] B. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: An overview", Internet Draft, October 1993.

[3] D.D. Clark, S. Shenker, and L. Zhang, "Supporting real-time applications in an integrated services packet network: Architecture and mechanism", in *SIGCOMM '92*, Baltimore, Maryland, August 1992, pp. 14–26.

[4] J. McQuillan, I. Richer, and E. Rosen, "The new routing algorithm for the ARPANET", *IEEE Transactions on Communications*, vol. COM-28, no. 5, pp. 711–719, May 1980.

[5] A. Khanna and J. Zinky, "A revised ARPANET routing metric", in *ACM SIGCOMM '89*, September 1989, pp. 45–56.

[6] P. Almquist, "Type of service in the Internet Protocol suite", Tech. Rep. RFC-1349, Network Working Group, July 1992.

[7] J. Moy, "OSPF version 2", RFC 1247, Network Information Center, SRI International, July 1991.

[8] R. Callon, "Use of OSI IS-IS for routing in TCP/IP and dual environments", Tech. Rep. RFC-1195, Digital Equipment Corporation, December 1990.

[9] M.L. Gardner, I.S. Loobeek, and S.N. Cohn, "Type-of-service routing with loadsharing", in *GLOBECOM '87*, Tokyo, Japan, November 1987.

[10] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm", in *ACM SIGCOMM '89*, Austin, Texas, September 1989, pp. 1–12.

[11] S. Keshav, A. Agrawala, and S. Singh, "Design and analysis of a flow control algorithm for a network of rate allocating servers", in *IFIP WG 6.1/WG 6.4 Second International Workshop on Protocols for High-Speed Networks*, Palo Alto, CA, November 1990, pp. 55–72.

[12] J. Bolot and A.U. Shankar, "A discrete-time stochastic approach to flow control dynamics", in *GLOBECOM '92*, Orlando, Florida, December 1992.

[13] A. Parekh, "A generalized processor sharing approach to flow control in integrated services networks", Tech. Rep. LIDS-TR-2089, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 1992.

[14] D. Ferrari and D.C. Verma, "Quality of service and admission control in ATM networks", Tech. Rep. TR-90-064, International Computer Science Institute, Berkeley, California, December 1990.

[15] D. Johnson, "NSFnet report", in *19th IETF*, University of Colorado, National Center for Atmospheric Research, December 1990, pp. 377–382.

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

[17] I. Matta and A.U. Shankar, "On the interaction between gateway scheduling and routing", in *IEEE MASCOTS*, Durham, North Carolina, January 1994, pp. 84–88.

[18] I. Matta and A.U. Shankar, "On the interaction between gateway scheduling and routing", Tech. Rep. CS-TR-3102, Department of Computer Science, University of Maryland, College Park, MD 20742, July 1993, Available by anonymous ftp at ftp.cs.umd.edu:pub/MaRS/Papers.

[19] C. Alaettinoğlu, A. U. Shankar K. Dussa-Zieger, and I. Matta, "Design and implementation of MaRS: A routing testbed", *Journal of Internetworking: Research and Experience*, vol. 5, no. 1, pp. 17–41, March 1994.

[20] A.U. Shankar, C. Alaettinoğlu, I. Matta, and K. Dussa-Zieger, "Performance comparison of routing protocols using MaRS: Distance-vector versus link-state", in *ACM SIGMETRICS/PERFORMANCE*, Newport, Rhode Island, June 1992, vol. 20, pp. 181–192.

[21] 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.

[22] E. Dijkstra, "A note on two problems in connection with graphs", *Numer. Math.*, vol. 1, pp. 269–271, 1959.

[23] D. Glazer and C. Tropper, "A new metric for dynamic routing algorithms", *IEEE Transactions on Communications*, pp. 360–367, March 1990.

[24] R. Jain and S.A. Routhier, "Packet trains - measurements and a new model for computer network traffic", *IEEE JSAC*, vol. 4, no. 6, pp. 986–995, September 1986.

[25] L. Kleinrock, *Queueing Systems*, vol. I and II, New York: Wiley, 1976.

[26] D.P. Bertsekas, "Dynamic behavior of shortest path routing algorithms for communication networks", *IEEE Transactions on Automatic Control*, vol. 27, no. 1, pp. 60–74, February 1982.

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

[28] H. Takagi, "Queueing analysis of polling models: An update", in *Stochastic Analysis of Computer and Communication Systems*, pp. 267–318. H. Takagi (Editor). Elsevier Science Publishers B.V. (North-Holland), 1990.

[29] N. Mitrou and D. Pendarakis, "Cell-level statistical multiplexing in ATM networks: Analysis, dimensioning, and call-acceptance control w.r.t. QOS criteria", in *Queueing, Performance and Control in ATM (ITC-13)*, pp. 7–12. J. Cohen and C. Pack (Editors). Elsevier Science Publishers B.V. (North-Holland), 1991.

[30] W. Zhu and S. Chanson, "Adaptive threshold-based scheduling for real-time and non-real-time traffic", in *IEEE RTSS '92*, Phoenix, Arizona, December 1992, pp. 125–135.

[31] B. Kuo, *Automatic Control Systems*, Prentice-Hall, Inc., fourth edition, 1983.

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

[33] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, Brooks/Cole Publishing Company, 1991.

[34] S. Floyd and V. Jacobson, "The synchronization of periodic routing messages", in *ACM SIGCOMM '93*, San Francisco, California, September 1993.

[35] A. Economides and J. Silvester, "Optimal routing in a network with unreliable links", in *IEEE INFOCOM '88*, August 1988, pp. 288–297.

[36] I. Matta and A.U. Shankar, "Type-of-service in adaptive next-hop routing", Tech. Rep. CS-TR-2963, Department of Computer Science, University of Maryland, College Park, MD

20742, September 1992, Available by anonymous ftp at ftp.cs.umd.edu:pub/MaRS/Papers.

[37] J. Zinky, "Private communication", 1992.

[38] S. Gupta, *Performance Modeling and Management of High-Speed Networks*, PhD thesis, University of Pennsylvania, Department of Systems, 1993.

[39] L. Zhang, "VirtualClock: A new traffic control algorithm for packet switching networks", in *SIGCOMM '90*, Philadelphia, Pennsylvania, September 1990, pp. 19–29.

[40] S. S. Lavenberg, *Computer Performance Modeling Handbook*, Academic Press, 1983.

[41] H. A. Taha, *Operations Research : An Introduction*, Macmillan publishing Co., second edition, 1976.

[42] I. Matta and A.U. Shankar, "Type-of-service routing in dynamic datagram networks", in *IEEE INFOCOM*, Toronto, Ontario, Canada, June 1994, pp. 992–999, Tech. report available by anonymous ftp at ftp.cs.umd.edu:pub/MaRS/Papers.

Ibrahim Matta received the B.Sc. and M.Sc. degrees in Computer Science and Automatic Control from Alexandria University, Egypt in 1986 and 1989, respectively. He is now completing his Ph.D. degree in Computer Science at the University of Maryland, College Park. His current research interests are in computer communication networks, routing protocols, and performance evaluation. He is a member of the ACM and IEEE.

A. Udaya Shankar received the Ph.D. degree in Electrical Engineering from the University of Texas at Austin. He is an Associate Professor in the Computer Science Department and the Institute for Advanced Computer Studies at the University of Maryland at College Park. His research interests include the modeling and analysis of distributed systems and network protocols, from both correctness and performance aspects. He is currently working on assertional methods for layered systems and real-time systems, and on numerical-analytical methods for evaluating integrated-services networks.

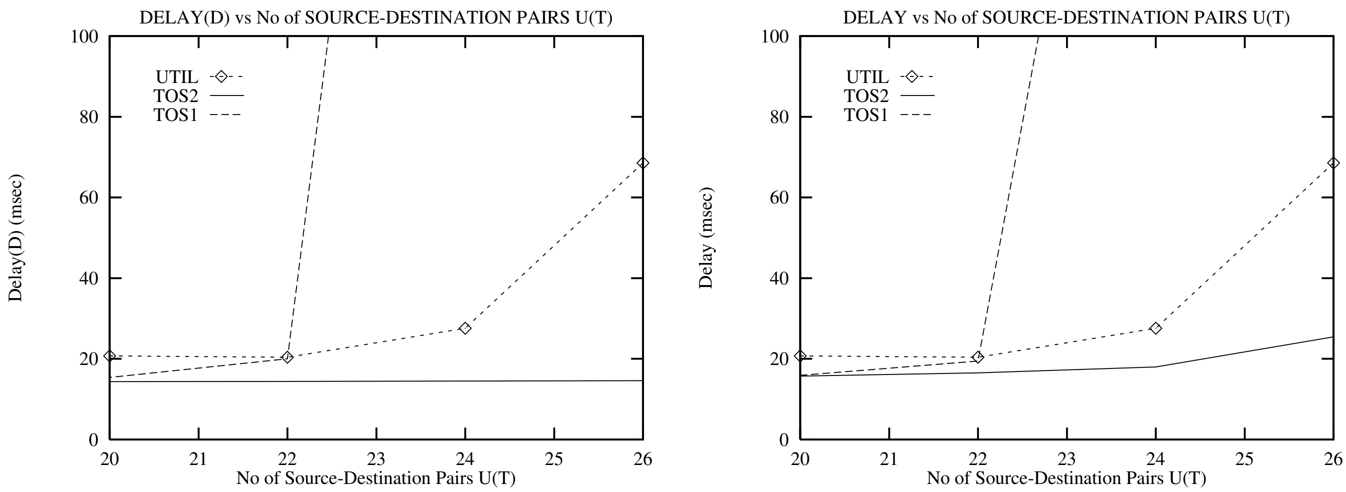


Fig. 9. Low-speed. Equal packet sizes. $Delay(D)$ and $delay$ vs $U(T)$ for $U(D) = 8$.

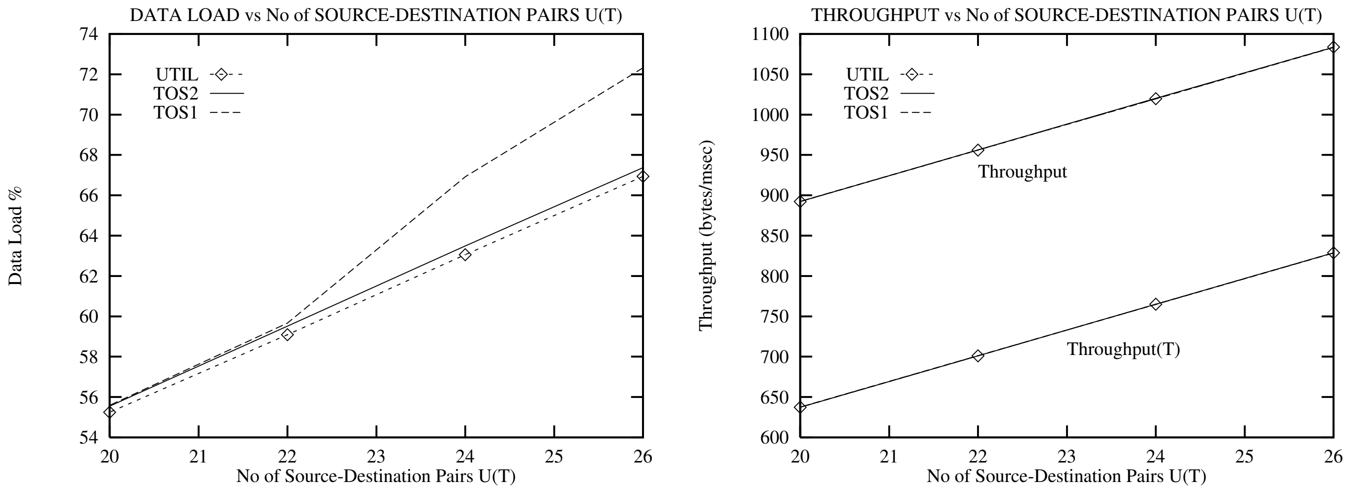


Fig. 10. Low-speed. Equal packet sizes. $Data\ load$ and $throughput$ vs $U(T)$ for $U(D) = 8$.

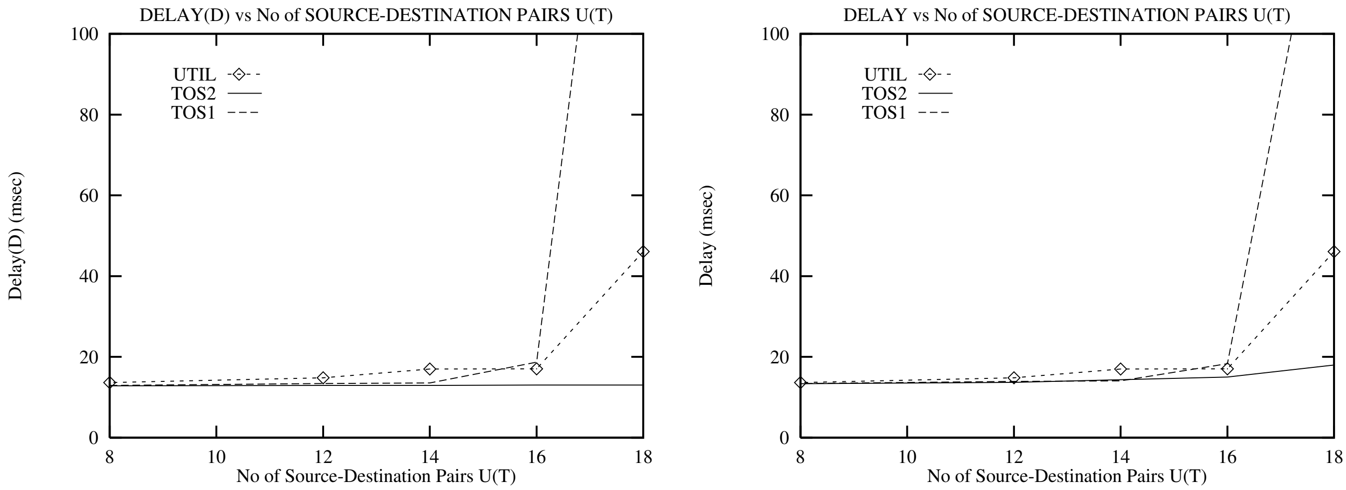
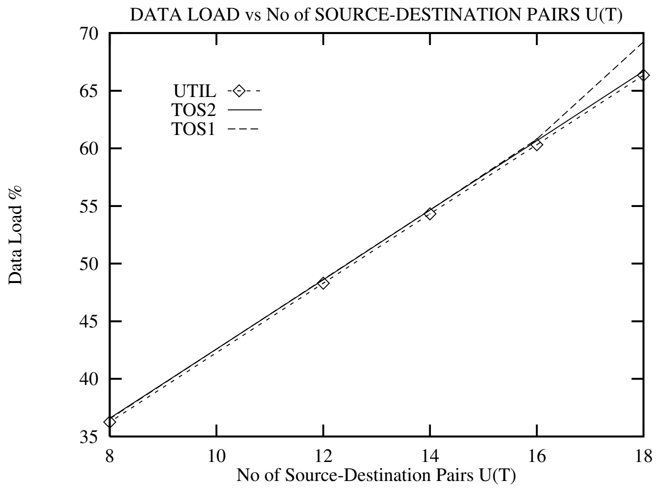


Fig. 11. High-speed. Equal packet sizes. $Delay(D)$ and $delay$ vs $U(T)$ for $U(D) = 4$.



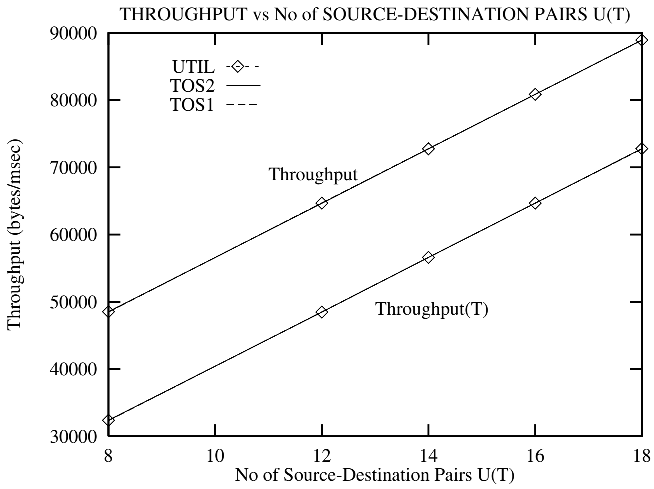
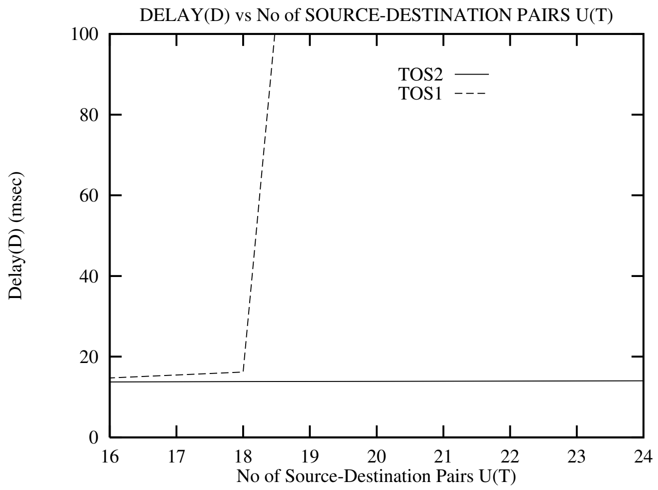


Fig. 12. High-speed. Equal packet sizes. *Data load* and *throughput* vs $U(T)$ for $U(D) = 4$.



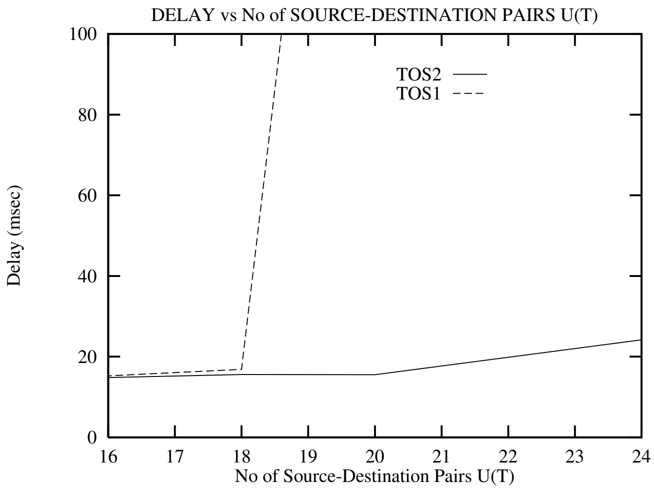


Fig. 13. Low-speed. Unequal packet sizes. $Delay(D)$ and $delay$ vs $U(T)$ for $U(D) = 16$.

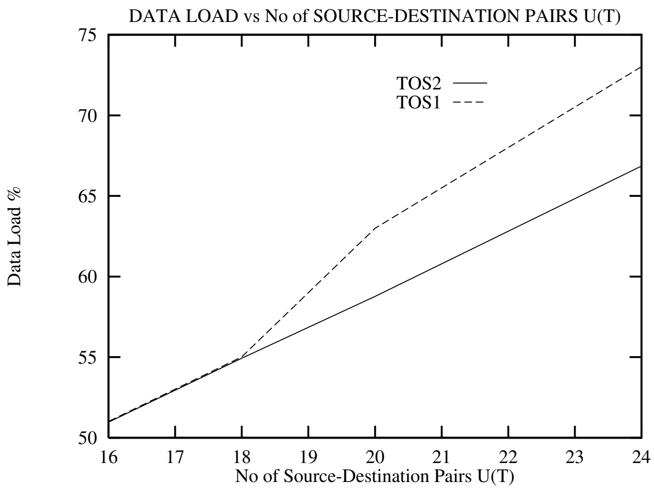


Fig. 14. Low-speed. Unequal packet sizes. *Data load* vs $U(T)$ for $U(D) = 16$.