Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

Computer Standards & Interfaces 27 (2005) 117–132

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

Extension of Multiprotocol Label Switching for long-range dependent traffic: QoS routing and performance in IP networks

Tibor Gyires[a], H. Joseph Wen[b,*]

[a] School of Information Technology, College of Applied Science and Technology, Illinois State University, Normal, IL 61790-5150, USA
[b] Department of Accounting and Management Information Systems, Donald L. Harrison College of Business,
Southeast Missouri State University, Mail Stop 5815, Cape Girardeau, MO 63701, USA

Received 21 November 2003; accepted 28 May 2004
Available online 26 June 2004

Abstract

This paper presents an extension to the Multiprotocol Label Switching (MPLS) traffic engineering in IP networks with long-range dependent traffic. The extension provides the ability to diverge traffic flows away from the shortest path calculated by the traditional IP routing protocols into a less congested area of the network. When the traffic burstiness of a packet flow exceeds a predefined threshold, the extension calculates the cost of the traffic distribution and the effectiveness of Label Switching Routers (LSPs) to minimize the number of discarded packets. The simulation results demonstrate that the extension significantly improves the overall network performance in link utilization, port processor utilization, message delay, number of dropped packets, and buffer usage level.

Keywords: Telecommunications; Simulation modeling; Traffic burstiness; MPLS; QoS; RSVP

1. Introduction

The purpose of a communications network is to support the sharing of the communications links. When too many packets are present in (a part of) the subnet, performance degrades. This situation is called congestion. When the number of packets sent into the subnet by the hosts is within the network's carrying capacity, the packets are all delivered (except for a few that are damaged or lost due to transmission errors), and the number delivered is proportional to the number sent. However, as traffic increases too far, the routers are no longer able to forward the packets, and they begin losing them. This tends to make matters worse. At very high traffic, performance collapses completely, and almost no packets are delivered. Congestion can be caused by several factors. The most dangerous cause of congestion is the burstiness of the network traffic. Recent results make evident that high-speed network traffic is burstier and its variability cannot be predicted as assumed previously. It has been shown that network traffic has similar statistical properties on many time scales. Traffic that is bursty on many or all time scales can be described statistically using the notion of long-range dependency [1,10,20,31]. Long-range depen-

* Corresponding author. Tel.: +1-573-651-2121; fax: +1-309-438-5113.
E-mail address: hjwen@ilstu.edu (H.J. Wen).

0920-5489/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2004.05.004

dent traffic has observable bursts on all time scales resulting in packet losses, extremely long response times and overrun communications link capacities.

The traditional way to control congestion is that the router detecting the congestion sends a packet to the traffic source or sources, announcing the problem. Obviously, these extra packets increase the load when the subnet is already congested. Feedback-based congestion control systems are not responsive enough to the varying capacities of transmission links and network delay. The larger the end-to-end delay, the longer it takes to inform the sending nodes that the network has become congested. It was shown in Ref. [5] that the duration of the congestion at the congested routers is directly related to the bandwidth-delay product.

In all traditional feedback schemes, the expectation is that knowledge of congestion will cause the hosts to take appropriate action to reduce the congestion. But due to the inherent network delay, the hosts will react too sluggishly to be of any real use. End-to-end feedback mechanisms introduce an unacceptable delay. The situation is getting even worse when the traffic is generated at high-speed in long bursts and the traffic characteristics cannot be predicted by traditional statistical methods [16,17].

A good traffic engineering solution that guarantees the availability of network resources for mission-critical, interactive and delay-sensitive applications is Quality of Service (QoS) [2,8]. Its features, such as queuing, RSVP and multicast services also provide the control that is needed to handle different traffic in different ways. Traffic engineering implies the use of mechanisms to avoid congestion by allocating network resources optimally, rather than continually increasing network capacities. It is accomplished by mapping traffic flows to the physical network topology along predetermined paths. Traffic engineering provides the ability to diverge traffic flows away from the shortest path calculated by the traditional IP routing protocols into a less congested area of the network avoiding congestions caused by, e.g., long-range dependent traffic.

Multiprotocol Label Switching (MPLS) has emerged as a technology that can provide connection oriented traffic engineering and QoS. Many future core networks will use MPLS, including converged data and voice networks. Traffic engineering is achieved by forwarding large volumes of voice and data between Label-Switched Routers (LSRs) together with bandwidth reservation for traffic flows with various Quality of Service requirements. MPLS uses a label switching technique to forward data. A fixed-format label is inserted in front of each data packet on entry into the MPLS network. At each hop, the packet is routed based on the value of the incoming label and sent out on an outgoing interface with a new label value. The path that data traverses through a network is defined by the transition in label values, as the label is swapped at each LSR. Such a path is called a Label Switched Path (LSP). In order to distribute the labels along an LSP, a signaling protocol is required, such as the Resource Reservation Protocol (RSVP) [4,9].

The objective of our paper is to present an extension to MPLS in networks with long-range dependent traffic. We describe a network architecture for minimizing the packet loss due to congestions caused by long-range dependent traffic. When the traffic burstiness of an LSP increases at an LSR, the router distributes the traffic flow over n new LSPs leading to the destination(s) of the original traffic. The selection of the new LSPs is based on cost estimates of the flow distribution and routers' effectiveness in reducing the number of dropped packets in previous flow distributions. We define an architecture for the components of our MPLS network and the flow distribution protocol. Using a discrete event simulation model, we demonstrate how our extension improves the MPLS technology.

The second section introduces QoS, MPLS and RSVP. The third section gives an overview of long-range dependent traffic. The fourth section presents our Extended MPLS framework and the proposed flow distribution protocol followed by the conclusion.

2. Overview of QoS, MPLS, and RSVP

2.1. Quality of service (QoS)

A recent survey by infonctics [26] listed QoS as the second leading concern of IT managers, behind "security" in its importance in their network design decisions. QoS is the ability of a network to differentiate between different types of traffic and prioritize accordingly. It is the cornerstone of any convergence strategy. Voice, video, and data display very different traffic patterns in the network. Voice and video are very delay-dependent and have very predictable patterns, whereas data is very bursty and is less delay-sensitive. If all three types of traffic occur on a network, the data traffic usually interferes with voice and video and causes it to be unintelligible. Fig. 1 shows bandwidth requirement and delay-sensitivity of different types of business applications.

Networked applications are evolving far faster than the infrastructure that supports them. All this is creating a problem-poor application performance in the enterprise. Many enterprises today are faced with an aged-old dilemma that seems to creep into every network: Throw more bandwidth at the problem, or manage the existing bandwidth more effectively. QoS can effectively improve the usage of existing bandwidth [21,22]. It concerns itself with four different categories of services-bandwidth, latency, jitter, and loss. The first service category, bandwidth, concerns itself with how the network manages the entire stream of data packets flowing through it, particularly in times of network congestion. The second service category is latency, the end-to-end delay of a flow. Numerous applications, including voice and video, have a specific end-to-end delay budget. If a packet is delayed beyond the allocated budget, the data becomes stale or is no longer relevant. The third category addresses the need to control jitter, the variations in latency between packets. The final category concerns the need to manage packet loss. As a consequence of congestion, packet loss has two purposes. First, reducing the number of packets competing for output link can relieve the level of congestion. Second, when sending hosts notice that some packets are being discarded, they usually reduce the volume of traffic they are injecting into the network. While this is an effective approach for managing congestion, many applications can only tolerate a small amount of packet loss. The proposed MPLS extension is mainly to minimize packet loss and to control jitter.

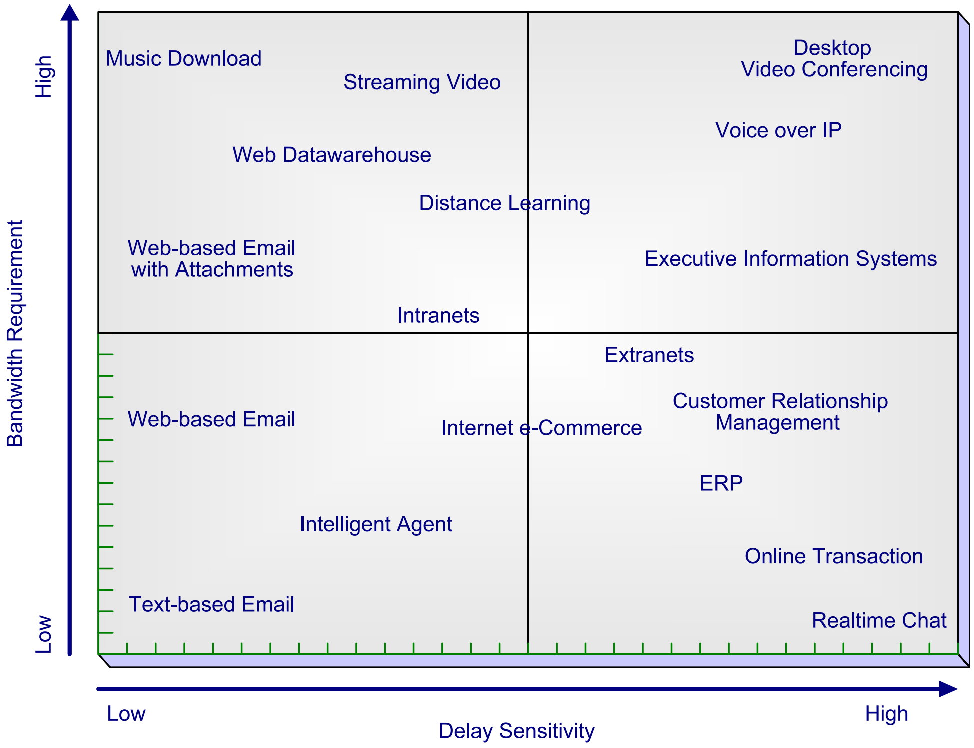


Fig. 1. QoS requirements of various computer applications.

2.2. Multiprotocol Label Switching (MPLS)

MPLS has emerged as an enabling technology for new public networks due to its support for traffic engineering. It is capable to forward large volumes of voice and data between Label-Switched Routers (LSRs). Because MPLS can simultaneously support multi-layer services (Layers 2 and 3), it has gained support from many standards organizations, equipment providers and service providers. Telecommunications carriers can integrate multiple single-service networks onto an integrated MPLS core. There is already a large MPLS market because the technology provides the foundation of many IP Virtual Private Network (VPN) services. The next generation of MPLS development will improve traffic engineering, offer new services and integrate existing service platforms. Each development will require a new infrastructure because existing equipment will not be powerful enough to support the new services. Vendors have started to develop extensions to MPLS that are focused on particular applications. Our proposed extension to MPLS is unique in its traffic engineering category. The addition of the new features largely depends on the resource and QoS requirements of the implementation in the routers and switches in an MPLS network.

MPLS does not replace IP routing, but will work together with existing routing technologies to provide very high-speed data forwarding between LSRs. A fixed-format label is inserted in front of each data packet on entry into the MPLS network. At each hop across the network, the packet is routed based on the value of the incoming label and sent out on an outgoing interface with a new label value. The path that data traverses through a network is defined by the transition in label values, as the label is swapped at each LSR. Since the mapping between labels is constant at each LSR, the path is determined by the initial label value. Such a path is called a Label Switched Path (LSP). The basic operation of an MPLS network is depicted in Fig. 2 similar to the one in Ref. [6].

At the entry to the network, called ingress, each packet is examined to determine which LSP it should use based on the destination address, the quality of service requirements, and the current state of the network. The group of similar packets forwarded in the same way is known as a Forwarding Equivalence Class (FEC). One or more FEC may be mapped to a single LSP. The diagram above shows two data flows from host A: one to host B and one to host C. Two LSPs are depicted. LSR1 is the ingress point to the network to transmit data from and to host A. When LSR1 receives packets from A, it determines the FEC for each packet, determines the LSP to use, and inserts a label to the packet. LSR1 then forwards the packet on the appropriate outgoing interface defined for the LSP. LSR2 examines the labeled packet received on the incoming interface and refers to a lookup table to decide the outgoing interface to send out the packet. As shown in Fig. 2, each packet with label 15 will be sent out on the interface to LSR3 with label 36. Label 23 will be swapped with label 19 and the packets will be sent to LSR4. LSR3 and LSR4 are exit routers, called egress LSRs from the MPLS network. The egress LSRs remove the labels from the packets and forward them using traditional IP routing.

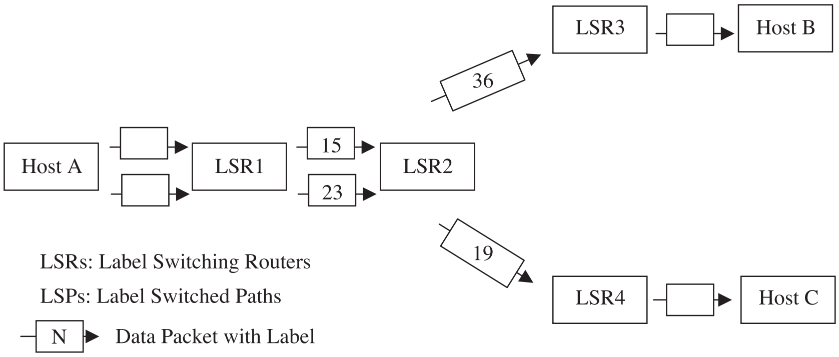


Fig. 2. LSPs and LSRs in an MPLS network.

LSPs are setup using the lookup tables at each LSR that map the pairs (incoming interface, label) to the pairs (outgoing interface, label) in the table entries. The process that updates and creates the table entries is called label distribution. The label distribution protocol that establishes LSPs between LSRs can play an important role in implementing Traffic Engineering functions. The class of service and quality of service required for traffic flows can also be included in the Traffic Engineering process. There are currently two label distribution protocols that provide support for Traffic Engineering: RSVP and Constrain-based routed Label Distribution Protocol (CR-LDP). The main differences between the two protocols are the reliability of the transport protocol and direction of the resource reservation, i.e., the reservation is done in the forward or reverse direction. Each protocol has its supporters and opponents, and the protocols are still being developed by the Internet Engineering Task Force (IETF). We chose RSVP to demonstrate our Traffic Engineering protocol for reasons discussed below.

2.3. Overview of RSVP

Originally, RSVP was developed as the signaling protocol of the Integrated Services (IntServ) framework to provide QoS support in IP based networks [27,29]. The framework defines a flow as a stream of related datagrams between hosts that is created by a single application or user activity and requires the same QoS. The physical path of the flow across a network is determined by conventional destination-based routing. A flow is a simplex mode of packet transmission. Quality of Service is implemented in a router and a host for a particular data flow by a module called traffic control. It includes the Packet Classifier, Admission Control, Policy Control, Packet Scheduler, and the Estimator [27,29].

RSVP reserves resources in routers along the traffic flow as requested by an application. RSVP can be considered the configuration protocol for IntServ. The sending host sends a PATH message to the destination(s) of the traffic flow. It carries the flow specification of the offered load: the identity of the application, the traffic characterization (bandwidth, burstiness, average and peak transmission rate), and data for traffic classification. When the PATH message arrives at the destination(s), it responds by sending RESV messages identifying the requested flow specification: the receiving application, the type of services (latency bound, minimum bandwidth guarantee, etc.) and the amount of resources requested by the destination(s). RESV messages return to the source along the reverse path of the PATH messages. As RESV messages arrive at the routers, the devices decide whether there are sufficient resources available for the traffic flow. If there are enough resources and there is no conflict between the request and local policies established for the destination, the request is admitted. Otherwise, the request is rejected and an error message is sent to the network devices along the path notifying other RSVP-aware devices that the request was not admitted.

RSVP has become a proposed standard and is currently widely implemented in IP networking equipment. However, RSVP has not been widely used in service provider networks because of the lack of its scalability and the overhead required supporting large number of host-to-host flows. A number of extensions were added to the original RSVP specification to reduce the effect of the deficiencies above. The designers of MPLS chose to further extend RSVP into a signaling protocol to create LSPs that could be automatically routed away from network congestions. The main reasons that the designers of MPLS chose to extend RSVP rather than design a new signaling protocol to support traffic engineering requirements are the following [30]:

(1) RSVP was designed for resource reservation across a set of multicast or unicast Internet traffic paths. Reservation is an important component of traffic engineering that has made RSVP a good candidate to implement it.
(2) RSVP allows carrying opaque objects in its messages processed by the various modules in a router. This feature makes it easier to add new RSVP objects that can be used to create and maintain distributed state information other than resource reservation. The traffic engineering extensions can easily be developed to extend RSVP to support explicit routing and label distribution that are essential traffic engineering requirements.

(3) The extended RSVP is backward compatible with traditional RSVP implementations. It can differentiate between LSP signaling and traditional RSVP reservations by examining the objects contained in the signaling messages.

3. Long-range dependent traffic

For more details of long-range dependency in time series and the associated statistical tests, see Refs. [19,23,31]. We follow the definitions of these papers.

3.1. Definitions

Let X=(X_t: $t=0, 1, 2, \ldots$) be a covariance stationary stochastic process. Such a process has a constant mean $\mu=E[X_t]$, finite variance $\sigma^2=E[(X_t-\mu)^2]$, and an autocorrelation function $r(k)=E[(X_t-\mu)(X_{t+k}-\mu)]/E[(X_t-\mu)^2]$ ($k=0, 1, 2, \ldots$) that depends only on k. It is assumed that X has an autocorrelation function of the form:

$$r(k) \sim \alpha k^{-\beta},\ \ k = 1, 2, 3, \ldots \tag{1}$$

where $0<\beta<1$ and α is a positive constant. Let $X^{(m)}$=($X^{(m)}_{(k)}$: $k=1, 2, 3, \ldots, m=1, 2, 3, \ldots$) represent a new time series obtained by averaging the original series X over non-overlapping blocks of size m. For each $m=1, 2, 3, \ldots$, $X^{(m)}$ is specified by $X_k^{(m)}=1/m(X_{km-m+1}+\ldots+X_{km})$, $(k\geq 1)$. Let $r^{(m)}$ denote the autocorrelation function of the aggregated time series $X^{(m)}$.

The process X is called exactly self-similar with self-similarity parameter $H=1-\beta/2$ if the corresponding aggregated processes $X^{(m)}$ have the same correlation structure as X, i.e., $r^{(m)}(k)=r(k)$, for all $m=$l, 2, … ($k=$l, 2, 3, …).

A covariance stationary process X is called asymptotically self-similar with self-similarity parameter $H=1-\beta/2$ if for all k large enough $r^{(m)}(k)\rightarrow r(k)$, $m=1, 2, 3, \ldots, 0.5\leq H\leq 1$.

A stationary process is called long-range dependent if the sum of the utocorrelation values approaches infinity: $\Sigma_k r(k)\rightarrow\infty$.

The Hurst parameter H represents the speed of decay of a process' autocorrelation function. As $H\rightarrow 1$, the extent of both self-similarity and long-range dependence increases. It can also be shown that for self-similar processes with long-range dependency H>0.5 [31].

There are various network models for characterizing the bursty nature of network traffic. Most of the models focus on the Hurst parameter only and do not consider the time scale from which the degree of long-range dependency, the burstiness of the traffic, begins to appear. As it has been demonstrated in Ref. [28], the Hurst parameter alone is not sufficient to completely characterize traffic burstiness. The Hurst parameter only shows how fast or slowly the burstiness increases or decreases but does not capture the range of time scales over which the long-range dependency nature of the network traffic becomes visible. The paper describes that the time scales can be captured by a parameter called Fractal Onset Time Scale (FOTS). It shows the beginning of the time scale from which the bursty nature of the traffic becomes evident. As the FOTS becomes a smaller and smaller scale, the traffic burstiness increases. Therefore, the traffic burstiness can be characterized not just by the Hurst parameter alone but by three parameters, namely the average arrival rate of the network traffic, the Hurst parameter, and the FOTS. After the authors of the paper [28], we call them the Three Fundamental Parameters (TFPs). The paper illustrates the use of the TFPs in various Fractal Point Process models that have been implemented in the OPNET network modeling tool in the form of Raw Packet Generators. For technical reasons, we chose the M/Pareto model to characterize traffic burstiness. We implemented the model and the TFPs in the COMNET modeling tool based on captured bursty traffic traces. See the details in Appendix A.

4. MPLS extension

Our framework is similar to the one in Ref. [15] applied for MPLS network. It extends MPLS by adding new functionalities to the framework. The new functionalities further augment the functions of the LSR's Estimator. We assume that the Traffic Control database of an LSR has the following knowledge of flows terminating at or crossing the router: flow identifier, required resources, priority, and the maximum number of flows that can be established on outgoing links.

Reassignment cost: Assuming a packet flow T_i. In the subsequent paragraphs, we will explain that T_i is going to be distributed over some number of new LSPs if the traffic burstiness exceeds a threshold value. Let us denote the distributed portion of the packet flow T_i on path j by T_{ij}. Each router that is in a distribution path may have to make some resource and routing reassignments. $A_{ij}(R)$ denotes the cost of such a reassignment at LSR R for a distributed flow T_{ij}. It includes the costs for assigning buffers to the flow being distributed, rerouting lower priority flows, etc. The LSR's Estimator estimates this cost based on predefined entries in a cost table created by the network service provider.

Discarded packets: Each Estimator can estimate the packet loss in the new, distributed flow. $P_{ij}(R)$ denotes the estimated number of packets discarded at LSR R in the distributed flow T_{ij}.

E-parameter: The E-parameter for LSR R, denoted by $E_{ij}(R)$ is a measure of its efficiency in rerouting a distributed portion T_{ij} of the packet flow T_i on path j. Small (close to zero) values of the E-parameter indicate high quality and efficiency, and high values indicate that router R is unreliable and unstable in rerouting of distributed traffic. The Estimator recalculates a router's E-parameter after each flow distribution using the E values from previous flow distributions. After each flow distribution, it is determined how realistic an LSR's cost estimate has been with respect to the actual cost. By testing a statistical hypothesis, it can be determined if it has been unrealistic. If the corresponding hypothesis H_0 is rejected, no request messages will be sent to this router in the future for diverting distributed flow. If a router's cost estimate has been realistic, i.e., the corresponding hypothesis H_0 is accepted, the E-parameter is recalculated. The closer the estimated cost is to the actual cost, the smaller is the E-parameter. A local network administrator provides initial values of E. Subsequent values of the E-parameter are calculated using the statistical sampling method given in Ref. [12]. The motivation for using the E-parameter is to enable the traffic distribution process to learn from past performances and use this knowledge to select the most efficient distribution paths.

Distribution cost: The cost associated with a distributed flow T_{ij} of flow T_i is the sum of reassignment costs, the estimated discarded packets, and the E-parameters along a path j. The cost of a distributed flow $T_{ij}(R,P)$ between two remote LSRs R and P is denoted by $C(T_{ij}(R,P))$ and defined recursively by:

$$C(T_{ij}(R,P)) = [C(T_{ij}(Q,P) + E_i(Q) + A_{ij}(Q) + P_{ij}(Q)],$$

where R and Q denote adjacent LSRs along the path j.

4.1. Flow distribution protocol

IntServ recommends the use of the Token Bucket scheme to characterize the degree of burstiness. A Token Bucket is defined by two parameters: a token rate T that specifies the sustainable data rate of the traffic flow, and the bucket size B that specifies the amount by which the data rate can exceed T for short periods of time. However, measurements of real traffic indicate that burstiness is present on a wide range of time scales that can be described statistically using the notion of long-range dependency. Long-range dependent traffic has observable bursts on all time scales. As we discussed it in the previous section, an appropriate way of measuring traffic burstiness is based on the estimate of the TFPs of the network traffic. Therefore, characterizing burstiness based on the TFPs is more appropriate than the Token Bucket scheme that is based on measurements of only short periods of time. We are going to apply the statistical method developed in Ref. [11] to estimate the FOTS of bursty traffic flows.

In our flow distribution protocol, the specification of a new packet flow and subsequent control packets carry the anticipated traffic's TFPs. It is calculated from previous connections and applications' traffic patterns at the sending host. It can also be estimated and updated by intermediate LSRs. Different LSRs can accommodate the same bursty traffic differently. For instance, an LSR can reserve more input buffers for the incoming packet flow; another router can assign more processing power for handling the flow, etc. We also assume that the specification of a new packet flow contains the average traffic volume V in bits per second that the router can handle based on empirical observations. (Note that this measurement is not necessarily the forwarding capability of the router).

When the TFPs of the flow specification of a new request or the traffic burstiness of an existing flow exceed certain thresholds, the router divides the traffic into n new LSPs. Instead of using a single path, the LSR will divide the traffic flow over n number of new paths, leading to the destination(s) of the original traffic. (For simplicity, we assume unicast instead of multicast sessions. The protocol below can easily be expanded for multicast as well.) The n new LSPs will reunite at the destination using some existing techniques, such as packet sequence numbering. The flow distribution is implemented in the following steps.

Step 1. Assume that LSR X detected that the traffic burstiness of a flow or the TFPs of a new request exceed certain thresholds. X estimates the volume requirement of the flow based and sends a PATH message and the flow specification to the destination LSR Y requesting resources for the flow. The PATH message also request LSR Y to provide a label for the new LSPs.

Step 2. LSR Y sends an RESV packet back to X on all paths towards X. The RESV packet can follow existing LSPs or any paths determined by standard IP routing. The RESV packet contains the label that the downstream LSR communicates to its upstream neighbor. The RESV packets are reproduced at each subsequent LSR along the paths to X. The RESV packet is sent upstream towards LSR X. Each LSR that receives a RESV packet with a label will use the received label for outgoing traffic along the new LSP.

Step 3. Each RESV stores all visited LSRs along the path in the packet's data field.

Step 4. Each RESV carries the flow specification requesting resources for flow distribution at each LSR. An LSR's Estimator estimates the resource requirement of the flow. If there are enough resources, the request is tentatively admitted and reservation is made for a portion or for all of the requested resources. If the request is rejected, reject messages are sent along the path, notifying MPLS-aware routers of the failure. (Note that the reservation does not guarantee that packets will not be lost because it is based on an estimate only.) The reservation is valid only for a limited t time interval. If there is no confirmation arriving for the reservation in time t, the resources can be reallocated for other purposes.

Step 5. Each RESV collects estimates of the reassignment costs, discarded packets, the E-parameter, and the reserved traffic volume from the Estimator at each visited LSR.

Step 6. Upon receiving all cost estimates from all paths, LSR X selects the n LSPs with the least distribution costs that collectively can carry the volume of the original flow, and distributes the new or existing traffic along the new T_{ij} flows, $j=1,2,\ldots n$. (See the calculation below).

Step 7. After flow distribution, each Estimator compares the actual cost and the estimated cost and recalculates its own E-parameter for subsequent flow distribution [12]. It can be proven similarly to Ref. [12] that the newly selected LSPs are optimal in terms of the distribution cost and overall packet losses. When the traffic burstiness falls below a certain threshold value, the distributed traffic flows are collapsed into the original single flow. If the flow's resource requirement exceeds the LSR's resources before the selection of the ''least-cost'' LSPs is completed, then X makes the decision immediately based on data from previous flow distributions.

5. Simulation method

A network system is a set of network elements [18], such as routers, switches, links, users, and applications working together to achieve some tasks. The state of a network system is the set of relevant variables and parameters that describe the system at a certain time that comprise the scope of this study. Instead of building a physical model of a network, we build a mathematical model representing the behavior and the logical and quantitative relations between network elements. By changing the relations between network elements, we can analyze the model without constructing the network physically, assuming that the model behaves similarly to the real system, i.e., it is a valid model. For instance, we can calculate the utilization of a link analytically, using the formula $U=D/T$, where D is the amount of data sent at a certain time and T is the capacity of the link in bits per second. This is a very simple model that is very rare in real world problems. Unfortunately, the majority of real world problems are too complex to answer questions using simple mathematical equations. However, in highly complex cases, simulation technique is more appropriate. Simulation models can be classified in many ways. The most common classifications are as follows:

- □ Static and dynamic simulation models: A static model characterizes a system independently of time. A dynamic model represents a system that changes over time.
- □ Stochastic and deterministic models: If a model represents a system that includes random elements, it is called a stochastic model. Otherwise it is deterministic. Queuing systems, the underlying systems in network models, contain random components, such as arrival time of packets in a queue, service time of packet queues, output of a switch port, etc.
- □ Discrete and continuous models: A continuous model represents a system with state variables changing continuously over time. Examples are differential equations that define the relationships for the extent of change of some state variables according to the change of time. A discrete model characterizes a system where the state variables change instantaneously at discrete points in time. At these discrete points in time some event or events may occur, changing the state of the system. For instance, the arrival of a packet at a router at a certain time is an event that changes the state of the port buffer in the router.

In our study, we assume dynamic, stochastic, and discrete network models. We refer to these models as discrete-event simulation models.

Due to the complex nature of computer communications, network models tend to be complex as well. The development of special computer programs for a certain simulation problem is a possibility, but it may be very time consuming and inefficient. Recently, the application of simulation and modeling packages has become more customary, saving coding time and allowing the modeler to concentrate on the modeling problem in hand instead of the programming details.

In a world of more and more data, computers, storage systems, and networks, the design and management of systems are becoming an increasingly challenging task. As networks become faster, larger, and more complex, traditional static calculations are

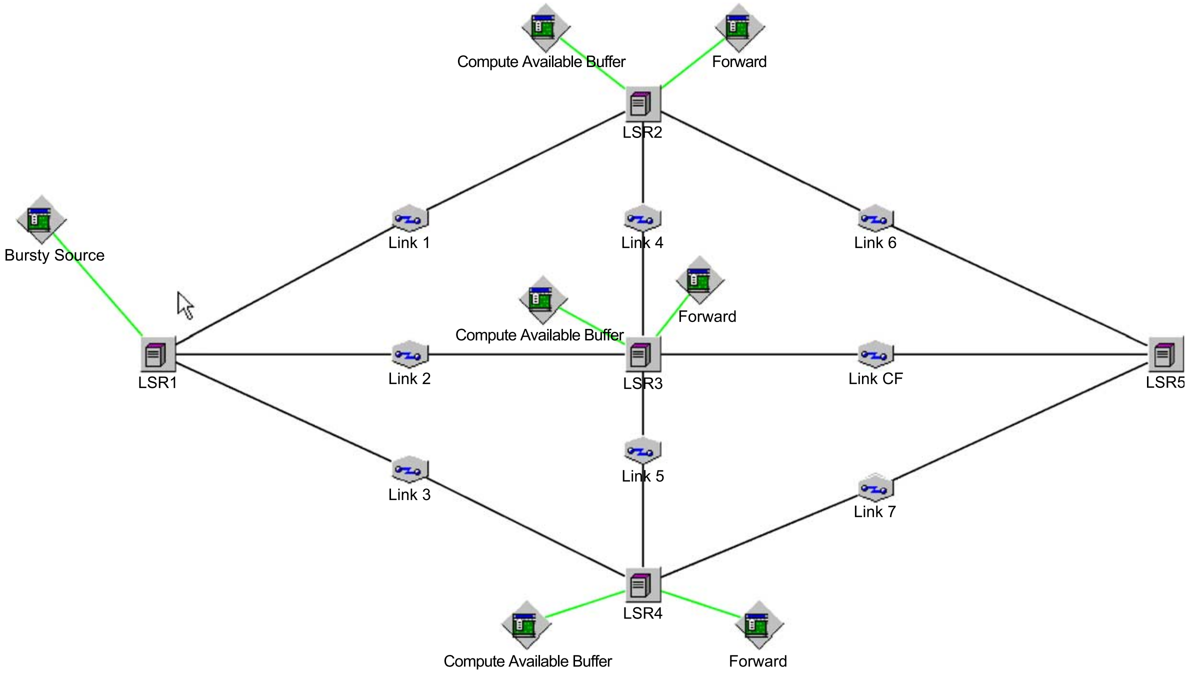
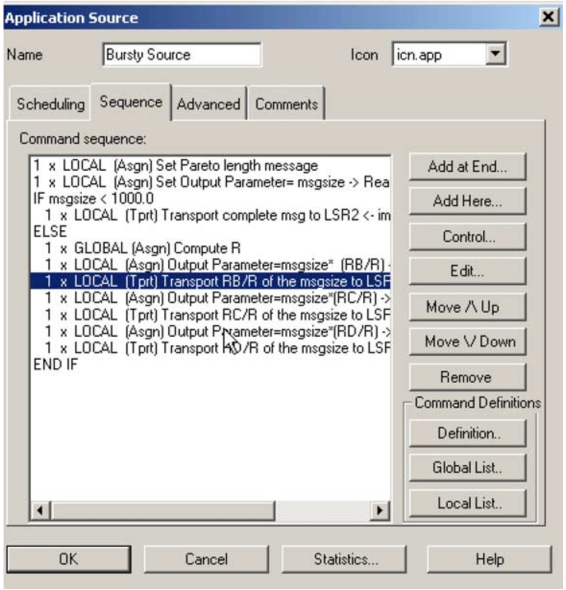


Fig. 3. The COMNET model of an IP network.

no longer reasonable approaches for validating the implementation of a new network design and multi-million dollar investments in new network technologies. Complex static calculations and spreadsheets, bottleneck analysis and/or queuing analysis are not appropriate tools any more due to the stochastic nature of network traffic and the complexity of the overall system.

Organizations depend more and more on new network technologies and network applications to support their critical business needs. As a result, poor network performance may have serious impacts on the successful operation of their businesses. In order to evaluate the various alternative solutions for a certain design goal, network designers increasingly rely on methods that help them evaluate several design proposals before the final decision is made and the actual systems is built. A widely accepted method is performance prediction through simulation. A simulation model can be used by a network designer to analyze design alternatives and study the behavior of a new system or the modifications to an existing system without physically building it. A simulation model can also represent the network topology and tasks performed in a network in order to obtain statistical results about the network's performance.

Simulation of large networks with many network elements can result in a large model that is difficult to analyze due to the large amount of statistics generated during simulation. Therefore, it is recommended to model only those parts of the network which are significant regarding the statistics we are going to obtain from the simulation. It is crucial to incorporate only those details that are significant for the objectives of the simulation.



Name: Bursty Source

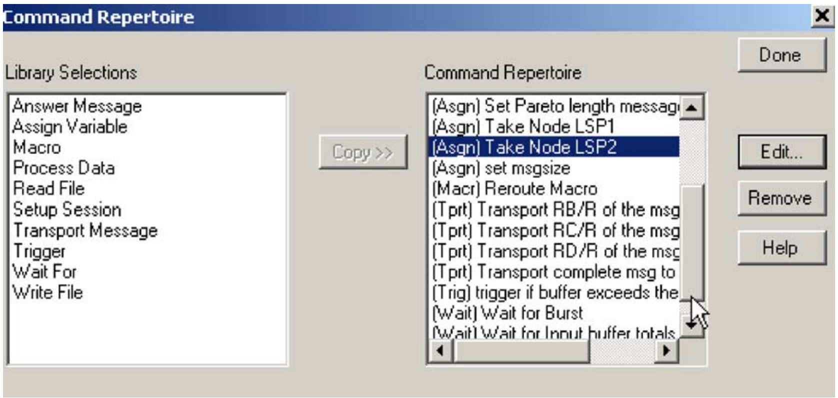
Icon: icn.app

Command sequence:

```
1  x  LOCAL  (Asgn) Set Pareto length message
1  x  LOCAL  (Asgn) Set Output Parameter= msgsize -> Rea
IF msgsize < 1000.0
  1  x  LOCAL  (Tprt) Transport complete msg to LSR2 <- im
ELSE
  1  x  GLOBAL (Asgn) Compute R
  1  x  LOCAL  (Asgn) Output Parameter=msgsize*  (RB/R) -
  1  x  LOCAL  (Tprt) Transport RB/R of the msgsize to LSF
  1  x  LOCAL  (Asgn) Output Parameter=msgsize*(RC/R) ->
  1  x  LOCAL  (Tprt) Transport RC/R of the msgsize to LSF
  1  x  LOCAL  (Asgn) Output Parameter=msgsize*(RD/R) ->
  1  x  LOCAL  (Tprt) Transport RD/R of the msgsize to LSF
END IF
```

Fig. 5. Application bursty source in the simulation.

Depending on the objectives, the same network might need different simulation models. For instance, if the modeler wants to determine the overhead of a new service of a protocol on the communication links, the model's links need to represent only the traffic generated by the new service. In another case, when the modeler wants to analyze the response time of an



Library Selections:

Answer Message
Assign Variable
Macro
Process Data
Read File
Setup Session
Transport Message
Trigger
Wait For
Write File

Command Repertoire:

(Asgn) Set Pareto length messag
(Asgn) Take Node LSP1
(Asgn) Take Node LSP2
(Asgn) set msgsize
(Macr) Reroute Macro
(Tprt) Transport RB/R of the msg
(Tprt) Transport RC/R of the msg
(Tprt) Transport RD/R of the msg
(Tprt) Transport complete msg to
(Trig) trigger if buffer exceeds the
(Wait) Wait for Burst
(Wait) Wait for Input buffer totals

Fig. 4. The command repertoire in the simulation.

Table 1
Number of packets dropped and input buffer usage

Node	# of packets blocked		Max. buffer use in bytes	
	Without traffic distribution	With traffic distribution	Without traffic distribution	With traffic distribution
LSP2	127	0	4,992,000	2,202,849
LSP3	0	0	0	2,202,849
LSP4	0	0	0	2,202,849
LSP5	0	0	0	4,405,698

application under maximum offered traffic load, the model can ignore the traffic corresponding to the new service of the protocol analyzed in the previous model.

Another important question is the granularity of the model, i.e., the level of details at which a network element is modeled. For instance, we need to decide whether we want to model the internal architecture of a router or we want to model an entire packet switched network. In the former case, we need to specify the internal components of a router, the number and speed of processors, types of buses, number of ports, amount of port buffers, and the interactions between the router's components. However, if the objective is to analyze the application level end-to-end response time in the entire packet switched network, we would specify the types of applications and protocols, the topology of the network and link capacities, rather then the internal details of the routers. Although the low level operations of the routers affect the overall end-to-end response time, modeling the detailed operations do not significantly contribute to the simulation results when looking at an entire network. Modeling the details of the routers' internal operations in the order of magnitude of nanoseconds does not contribute significantly to the end-to-end delay analysis in the higher order of magnitude of microseconds or seconds. The additional accuracy gained from higher model granularity is far outweighed by the model's complexity and the time and effort required by the inclusion of the routers' details.

Simplification can also be made by applying statistical functions. For instance, modeling cell errors in an ATM network does not have to be explicitly modeled by a communication link by changing a bit in the cell's header, generating a wrong CRC at the receiver. Rather, a statistical function can be used to decide when a cell has been damaged or lost. The details of a cell do not have to be specified in order to model cell errors. These discussions demonstrate that our goal of network simulation in this study is to reproduce the functionality of a network pertinent to a certain analysis, not to emulate it.

6. Simulation results

For the sake of simplicity and for the illustration of our model, we implemented a slightly modified version of the distribution cost calculation in the simulation. Let V denote the average traffic volume in bits per second that is going to be distributed. Let $C_1 \leq C_2 \leq ,\ldots \leq C_k$ denote the cost estimates along k

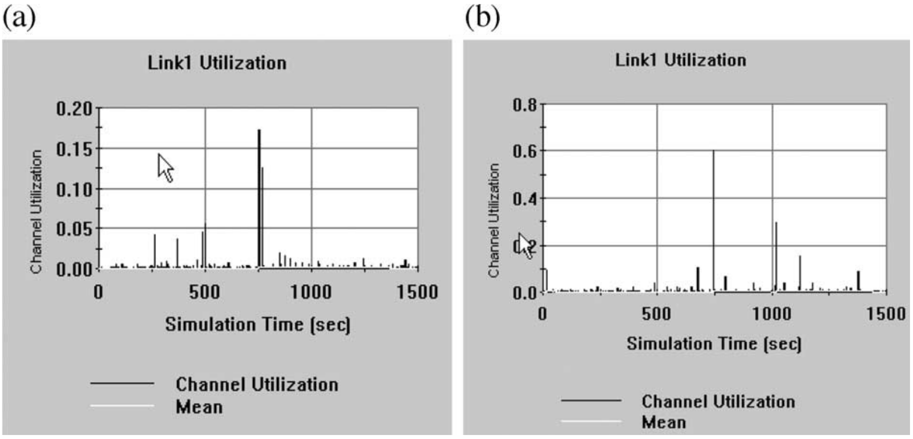


Fig. 6. Utilization of link AB (a) without and (b) with traffic distribution.

Table 2
Average and maximum utilization of Link1

Link1	Average utilization (%)	Maximum utilization (%)
Without traffic distribution	0.2600	60.0
With traffic distribution	0.0848	17.8

Table 3
Average and maximum message delay between LSR1 and LSR5

Message delay between LSR1 and LSR5	Average delay (msec)	Maximum delay (msec)
Without traffic distribution	46.83	1517
With traffic distribution	10.815	261.8

LSPs. LSR X will choose the first n LSPs, $n \leq k$, for which $V = V/C_1 + V/C_2 + \ldots + V/C_n$, and $1/C_1 + \ldots + 1/C_n = 1$.

In our simulation, we want to make sure that the traffic be distributed among the LSRs routers proportionally to the available buffer sizes. Let the cost estimate $C_i = 1/R_i$, where R_i is the available buffer size in bytes at LSR$_i$. Let us arrange the available buffer sizes $R_1 \geq R_2 \geq, \ldots \geq R_k$ and compute the sum $R_1 + R_2 + \ldots + R_k = R$. The simulation model distributes the flow proportionally to the available buffer size:

$$V = VR_1/R + VR_2/R + \ldots + VR_n/R, \text{ where}$$

$$R_1/R + \ldots + R_n/R = 1.$$

The motivation of LSR X's decision is that the higher an LSR's cost estimate the lesser amount of traffic will be distributed via that LSR.

We implemented the flow distribution protocol in the discrete event simulation system COMNET [7] based on real traffic traces captured in a large private network with bursty traffic. With slight modifications, similar models have been used in various papers and contexts to illustrate the affect of burstiness on different network protocols, such as DiffServ and IntServ, etc. [13–15,24,32]. In this paper, we assume that the selection algorithm above has already identified the paths with the least cost estimates. In order to demonstrate our distribution algorithm, we constructed the following simplified model in COMNET.

In Fig. 3, LSR1 through LSR5 denote the routers of the MPLS network connected by OC-3 links: Link1, Link2, etc. LSR1 is the origin; LSR5 is the destination of one or more Label Switched Path (LSP). There are three objects, so-called application sources, attached to the LSRs: "Bursty Source," "Compute Available Buffer," and "Forward". Application sources implement the behavior of an application by creating simple scripting language commands in the node where the application sources are attached to. For instance, the node LSR1 implements the commands in Fig. 4. Commands are taken from a Library of commands specific to a node and customized according to the application source using these commands.

The application source "Bursty Source" attached to LSP1 represents an MPLS path and implements the Path Distribution Protocol. Bursty Source sends mes-

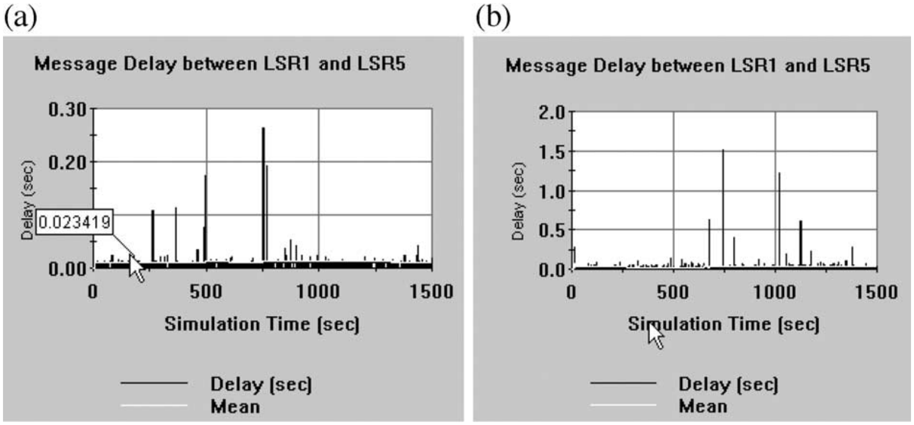


Fig. 7. Message delay (a) without and (b) with traffic distribution.

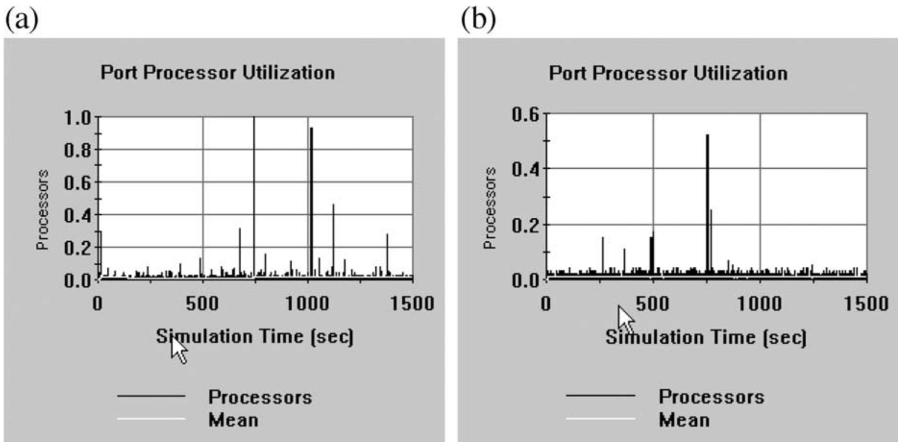


Fig. 8. Port processor utilization (a) without and (b) with traffic distribution.

sages to the destination with varying TFPs. If the traffic burstiness is less than a threshold, LSR1 will choose a path via LSR2. Otherwise, it will distribute the traffic among the adjacent routers identified by the selection algorithm shown in Fig. 5.

The "Compute Available Buffer" takes samples of the current size of buffer capacities of the LSPs in every second and transmits it to LSP1. The sizes of the available buffers are needed for the path distribution protocol. We can assume that routers can determine the buffers available for a certain traffic flow. The "Forward" application source implements the forwarding function of the LSPs.

We ran the simulation for 8000 s with and without traffic distribution and measured the number of packets dropped at the routers. The simulation statistics show (in Table 1) that 127 packets have been dropped without traffic distribution and no packets have been dropped with traffic distribution due to the lack of buffer space.

For Link1, as shown in Fig. 6 and Table 2, the average utilization was 0.26% and 0.09% without and with the traffic distribution respectively. The maximum utilization for Link1 could reach 60% when the traffic distribution method was not implemented.

Fig. 7 is the results for the corresponding message delay between LSP1 and LSP5. The message delay gets longer if no traffic distribution is implemented. Longer message delay (response time) can critically affect users' productivity. Table 3 demonstrates that

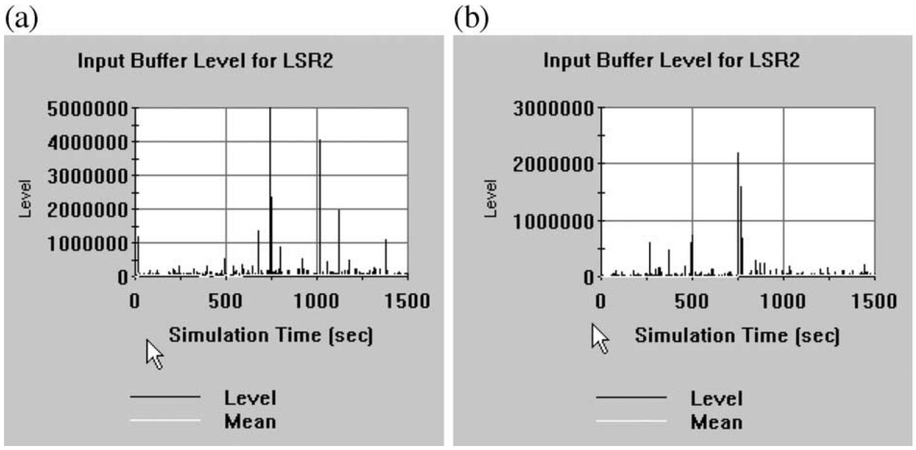


Fig. 9. Input buffer level for LSR2 (a) without and (b) with traffic distribution.

the average message delay between the source and destination LSPs is significantly less with traffic distribution.

The traffic distribution method also results in significant reduction of the port processor's utilization as it is shown in Fig. 8. The port utilization is very close to 100% without flow distribution. Very high processor utilization may drastically decrease the performance of the LSRs.

As a consequence of high port processor utilization, the buffer usage levels of the LSPs are also higher without traffic distribution. The result of high buffer usage level is in longer waiting times in packet queues and increasing number of dropped packets. Fig. 9 shows the buffer level of LSR2 with and without flow distribution. Due to space limitations, we do not include all the details, such as the message interarrival time, traffic burstiness threshold, buffer sizes, the full command sequence, etc. The details are available from the authors upon request.

In summary, the simulation results demonstrate that the flow distribution improves the overall network performance in link utilization, port processor utilization, message delay, number of dropped packets, and buffer usage level.

7. Conclusion

The paper presented an extension of MPLS traffic engineering to reduce the number of discarded packets during congestions caused by bursty, long-range dependent traffic. When the traffic burstiness of a packet flow exceeds a certain threshold at an LSR, the router selects n new LSPs to distribute the original long-range dependent traffic over these LSPs. The selection is based on cost estimates of the traffic distribution and the effectiveness of the LSPs in reducing the number of discarded packets in previous traffic distributions. We implemented the flow distribution protocol in a discrete event simulation model and demonstrated that our proposed MPLS extension can minimize the packet loss caused by long-range dependent traffic. The current version of MPLS applies label distribution protocols, such as RSVP that reserves the network resources along an LSP in advance. An LSP may not be established in case of long-range dependent traffic that requires excessive network resources. In our proposed extension version, an LSP can still be established for long-range dependent traffic that would not be possible in the current version of MPLS. The main limitation of this study, however, is the inadequate capabilities of the current network routers. The routers have to implement the functions needed to make decisions on various network parameters, such as burstiness threshold, dividing the amount of traffic flow into sub flows, etc. The algorithm can be included in standards only if the router/switch vendors agree on a widely accepted standard and can provide enough processing power in the routers and switches.

Appendix A

Results in Refs. [1,24] have proven that the M/Pareto model is appropriate for modeling long-range dependent traffic flow characterized by long bursts. Originally, the model was introduced in Ref. [20] and applied in the analysis of ATM buffer levels. The M/Pareto model was also used to predict the queuing performance of Ethernet, VBR video, and IP packet streams in a single server queue [24,25]. We apply the M/Pareto model not just for a single queue, but also for predicting the performance of an interconnected system of links, switches and routers affecting the individual network elements' performance. We make use of some of the calculations presented in Refs. [24,25].

The M/Pareto model is a Poisson process of overlapping bursts with arrival rate λ. A burst generates packets with arrival rate r. Each burst, from the time of its interval, will continue for a Pareto-distributed time period. The use of Pareto distribution results in generating extremely long burts that characterize long-range dependent traffic. The probability that a Pareto-distributed random variable X exceeds threshold x is:

$$P(X > x) = \left\{ \begin{array}{ll} \left(\frac{x}{\delta}\right)^{-\gamma}, & x \geq \delta \\ 1, & \text{otherwise} \end{array} \right\} \qquad (1)$$

$1 < \gamma < 2, \delta > 0.$

The mean of X, the mean duration of a burst $\alpha = \delta\gamma/(\gamma - 1)$ and its variance is infinite [26]. Assuming a t

time interval, the mean number of packets M in the time interval t is:

$$M = \lambda tr\delta\gamma/(\gamma - 1), \text{ and} \tag{2}$$

$$\lambda = M(\gamma - 1)tr\delta\gamma \tag{3}$$

The M/Pareto model is described in Refs. [20,24] as asymptotically self-similar and it is shown that for the Hurst parameter the following equation holds:

$$H = (3 - \gamma)/2 \tag{4}$$

7.1. Implementation of the M/Pareto model in Comnet

Comnet models a transaction by a message source and a destination, the size of the message, and the communication devices, and links along the path. The rate at which messages are sent is specified by an interarrival time distribution. The M/Pareto model's Poisson distribution represents the number of message arrivals in a certain time interval. In Comnet, this information is expressed as the time interval between successive arrivals, hence, we use the Exponential distribution as the traditional way to specify interarrival time. Using the Exponential distribution will result in an arrival pattern characterized by the Poisson distribution. The interarrival time in the model is 1 s.

The length of a message is specified by the Pareto distribution defined by two parameters in Comnet: the location and the shape. The location parameter corresponds to the packet arrival rate δ in a burst. The shape parameter corresponds to the γ parameter of the M/Pareto model calculated in the relation (1) as

$$\gamma = 3 - 2H$$

In the M/Pareto model, each burst will continue for a Pareto-distributed time period. In our test bed, we cannot measure the number of packets in a burst, only the number of packets in a message. Hence, we assume that a burst is equivalent to the number of bytes in a message sent or received in a second. Because the time period of each burst is proportional to the length of the message, we further assume that the length of the messages is also Pareto-distributed. So we derive the packet arrival rate δ in a burst not from the mean number of packets in a burst but from the mean length of messages denoted by M. The mean of the Pareto distribution is implemented in Comnet as:

$$M = \delta\gamma/(\gamma - 1)$$

and

$$\delta = M*(\gamma - 1)/\gamma$$

The relation (1) allows us to model bursty traffic based on real traffic traces by performing the following steps:

(a) Collect traffic traces using a network analyzer.
(b) Compute the mean arrival rate, the Hurst parameter H and the location parameter that is equivalent to the FOTS in the Fractal Point Process models in Ref. [28]. We used the Benoit package with the traffic trace as input to compute these parameters [3].
(c) Use the Exponential and Pareto distributions in the COMNET modeling tool with the parameters calculated above to specify the interarrival time and length of messages.
(d) Generate traffic according to the modified M/Pareto model and measure network performance parameters.

References

[1] R. Addie, M. Zukerman, T. Neame, Broadband traffic modeling: simple solutions to hard problems, IEEE Communications Magazine (1998) 88–95.

[2] M. Balakrishnan, R. Venkateswaran, QoS and differentiated services in a multiservice network environment, Bell Labs Technical Journal (1998 Oct–Dec) 222–239.

[3] Benoit 1.1, Trusoft Int'l, http://www.trusoft-international.com.

[4] Y. Bernet, The complementary roles of RSVP and differentiated services in the full-service QoS network, IEEE Communications Magazine (2000 February) 154–162.

[5] J. Bolot, A. Shankar, Dynamical behavior of rate based flow control systems, Computer Communication Review 20 (1990, Apr.) 35–49 (ACM SIGCOMM).

[6] P. Brittain, A. Farrel, MPLS Traffic Engineering: A Choice of Signaling Protocols, Analysis of the Similarities and differences between the two primary MPLS label distribution protocols: RSVP and CR-LDP, Data Connection Limited 2000.

[7] "COMNET III Application Notes: Modeling ATM Networks with COMNET III" CACI Products Company, 3333 North Torrey Pines Ct., La Jolla, California 92037, 1998.

[8] A. Durresi, S. Kota, M. Goyal, R. Jain, V. Bharani, Achieving

QoS for TCP traffic in satellite Networks with differentiated services, Space Communications 17 (2001) 125–136.

[9] G. Eichler, H. Hussmann, G. Mamais, I. Venieris, C. Salsano, S. Salsano, Implementing integrated and differentiated services for the internet with ATM networks: a practical approach, IEEE Communications Magazine (2002 January) 132–141.

[10] B.V. Ghita, S.M. Furnell, B.M. Lines, D. Le-Foll, E.C. Ifeachor, Network quality of service monitoring for IP telephony, Internet Research 11 (2001) 26–34.

[11] T. Gyires, Methodology for modeling the impact of traffic burstiness on high-speed networks, Proceedings of the 1999 IEEE Systems, Man, and Cybernetics Conference, October 12–15 Tokyo, Japan, 1999, pp. 980–985.

[12] T. Gyires, B. Muthuswamy, A bidirectional search approach for restoring circuits in communication networks, The Journal of Computer Information Systems (1997 Winter) 85–93.

[13] T. Gyires, Active agents algorithm for differentiated services in networks with bursty traffic, Proceedings of the Communication Systems, Networks and Digital Signal Processing (CSNDSP' 2002), Third International Symposium, Stafford, UK, July 15–17, Co-Sponsored by: IEEE, IEE, EURASIP, BCS, 2002, pp. 74–77.

[14] T. Gyires, Simulation of the harmful consequences of self-similar network traffic, The Journal of Computer Information Systems (2002 Summer) 94–111.

[15] T. Gyires, Software Agents Architecture for Controlling Long-range Dependent Network Traffic, accepted for publication in the Special Issue of Mathematical & Computer Modelling, Elsevier Science.

[16] P. Herrmann, H. Krumm, O. Drogehorn, W. Geisselhardt, Framework and tool support for formal verification of high-speed transfer protocol designs, Telecommunications Systems 20 (2002) 291–310.

[17] M. Khan, Network management: don't let unruly internet apps bring your network to its knees, Communications News 37 (2000) 30–32.

[18] M. Law, W.D. Kelton, Simulation Modeling and Analysis, third edition, McGraw-Hill Higher Education, 2000.

[19] W. Leland, M. Taqqu, W. Willinger, D. Wilson, On the self-similar nature of Ethernet traffic, computer communications review 23, Proc. of the ACM/SIGCOMM'93, San Francisco, 1993 (September), pp. 183–193.

[20] N. Likhanov, B. Tsybakov, N. Georganas, Analysis of an ATM Buffer with Self-Similar ("Fractal") Input Traffic, Proceedings of the IEEE INFOCOM Conference, 1995, pp. 985–992.

[21] S. Mishima, L. Moy-Yee, G. Yee-Madera, E. Yousefi, Broadband packet switch processor, Space Communications 18 (2002) 91–95.

[22] M. Nadeau, S. Chalifour, QoS in the Relentless Pursuit of Customers, Telephony 5 (2000 June) 234–239.

[23] E. Neame, Investigation of traffic models for high speed data networks, Proceedings of ATNAC '95, 1995, pp. 145–151.

[24] T. Neame, M. Zukerman, Application of the M/Pareto process to modeling broadband traffic streams, Proceedings of ICON '99, Brisbane, Queensland, Australia, 28 September–1 October, 1999, pp. 53–58.

[25] T. Neame, M. Zukerman, R. Addie, A practical approach for multi-media traffic modeling, Proceedings of Broadband Communications '99, Hong Kong, 10–12, November, 1999, pp. 73–82.

[26] B.D. Reimers, In depth quality of service: Carriers expand QoS options, InternetWeek (2001 June 25) 37–38.

[27] Request for Comments: 2205, L. Zhang, S. Berson, S. Herzog, S. Jamin, Resource ReSerVation Protocol (RSVP), September 1997.

[28] B. Ryu, S. Lowen, Fractal traffic models for internet simulation, HRL laboratories and mclean hospital, Proceedings of IEEE Int'l Symposium on Computer Communications, 2000, pp. 287–291.

[29] C. Semeria, RSVP Signaling Extensions for MPLS Traffic Engineering, White Paper, Juniper Networks, 2000.

[30] C. Semeria, RSVP Signaling Extensions for MPLS Traffic Engineering, White Paper, Juniper Networks, 2000.

[31] M. Taqqu, V. Teverovsky, W. Willinger, Estimators for long-range dependence: an empirical study, Fractals 3 (1995) 785–788.

[32] W. Weiss, QoS with differentiated services, Bell Labs Technical Journal (1998 Oct–Dec) 48–62.

Tibor Gyires is a Professor in the School of Information Technology at Illinois State University, Normal, Illinois. He received his PhD degree in Mathematics at the Kossuth Lajos University, Hungary. Dr. Gyires' research areas include distributed artificial intelligence, search algorithms, high-speed networks, simulation modeling of networks, and network capacity planning and performance prediction. He has published several papers in referred journals, proceedings, and books. His hobbies are skiing, sailing, and playing basketball.

H. Joseph Wen is an Associate Professor of MIS and chairperson of the Department of Accounting and Management Information Systems at Donald L. Harrison College of Business at Southeast Missouri State University. He holds a PhD from Virginia Commonwealth University. He has published over 100 papers in academic refereed journals, book chapters, encyclopedias and national conference proceedings. Dr. Wen has received over 6 million dollars research grants from various State and Federal funding sources. His areas of expertise are Internet research, electronic commerce (EC), transportation information systems, and software development.