Analysis of Discarding Policies in High-Speed Networks

Yael Lapid, Raphael Rom, and Moshe Sidi

***Abstract*—Networked applications generate messages that are segmented into smaller, fixed or variable size packets, before they are sent through the network. In high-speed networks, acknowledging individual packets is impractical; so when congestion builds up and packets have to be dropped, entire messages are lost. For a message to be useful, all packets comprising it must arrive successfully at the destination. The problem is therefore which packets to discard so that as many complete messages are delivered, and so that congestion is alleviated or avoided altogether.**

In this paper, selective discarding policies, as a means for congestion avoidance, are studied and compared to nondiscarding policies. The partial message discard policy discards packets of tails of corrupted messages. An improvement to this policy is the early message discard that drops entire messages and not just message tails.

A common performance measure of network elements is the effective throughput which measures the utilization of the network links but which ignores the application altogether. We adopt a new performance measure—goodput—which reflects the utilization of the network from the application's point of view and thus better describes network behavior.

We develop and analyze a model for systems which employ discarding policies. The analysis shows a remarkable performance improvement when any message-based discarding policy is applied, and that the early message discard policy performs better than the others, especially under high load. We compute the optimal parameter setting for maximum goodput at different input loads, and investigate the performance sensitivity to these parameters.

I. Introduction

MODERN networks differ substantially from the traditional networks in many aspects—the most important of which is flow control. Modern networks are typically high-speed networks, deploying high-capacity links, and integrating multiple services. Service integration means that the network has to support both services that need (and are willing to pay for) reserved resources, and those which cannot reserve resources and must rely on the available resources whenever the need arises. High transmission speed and high capacity mean that large amounts of data may be in transit through the network, which implies inability of the source to react in time to feedback coming from the network.

Indeed, much attention has been given recently to flow control in high-speed networks, in general, and ATM networks, in particular [1], [8]. As is evident from these, the most severe flow control problem arises from applications that generate data fairly irregularly, cannot reserve network resources, and are sensitive to data loss.

Many high-speed network applications generate messages which must be transported to a similar application at the other end of the network. These messages are segmented into smaller fixed or variable size packets which are then conveyed by the network. To be useful, all packets comprising a message must arrive successfully at the destination and be reconstructed into the original message. Hence, the network can only charge for the delivery of complete messages. Packet-by-packet acknowledgments and retransmissions are clearly impractical, and thus acknowledgments and retransmissions must be applied by the applications, at the message level.

The problem is therefore to deliver full messages by a network that handles and delivers packets. In other words, it seems beneficial to control the flow of packets with respect to messages boundaries, so as to accomplish the delivery of as many complete messages as possible.

Message misdelivery happens mainly due to congestion at network elements which causes buffers to overflow and packets to be dropped. Congestion may be built in network elements that are based on statistical multiplexing, especially when noncooperative clients who have not reserved resources introduce high loads to the network. In ATM networks, this service is known as *best-effort service*—a term related to traffic in noncontracted quality of service sessions. In this kind of session, the network does not guarantee any quality of service (e.g., percentage of lost packets) and the user does not have to comply with a certain data rate. The user introduces to the network as much data as it chooses, yet only part of it may be properly delivered. Because of the lack of coordination and commitment from both the user's and the network's side, high loads and congestion at various places in the network may evolve.

In case of congestion, and due to the lack of any commitment, the network can and will discard packets that belong to this type of service if congestion requires to discard packets. In spite of this lack of commitment on either side, the user cannot be charged for any message which is partially delivered, and thus it behooves the network to deliver complete messages.

Manuscript received December 1997; revised February 1998. This work was supported by the Consortium for Broadband Communication administered by the Chief Scientist of the Israeli Ministry of Commerce and Industry. Part of this work was presented in Infocom'97, Japan.

Y. Lapid is with the Department of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel and with the Department of Video Networking, ECI Telecom Ltd.

R. Rom is with the Department of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel, and with SUN Microsystems Labs, Mountain View, CA 94043 USA.

M. Sidi is with the Department of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel.

Publisher Item Identifier S 0733-8716(98)04105-5.

The user is therefore required to designate message boundaries, and the problem becomes that of which packets to discard so that as many complete messages are delivered and so that congestion is alleviated or avoided altogether.

A. Selective Discarding Policies

Selective discarding policies, as a means for congestion avoidance, were presented in [4] and [5]. These methods are implemented at network elements and do not depend on the cooperation of users, or the behavior of other network elements. By choosing which packets to deliver and which to drop, a network element tries to transmit as many complete messages as possible. In this paper, we study message-based selective discarding policies, which select packets to be discarded with respect to application message boundaries.

One discarding policy is partial message discard (PMD). According to this policy, the network element discards (drops) packets that belong to messages that were already damaged, that is, experienced a packet drop in the network element. In other words, if the buffer is full when a packet arrives to the network element, this packet, and all successive packets that belong to the same message, are discarded.

An improvement of this policy is the early message discard (EMD) in which, in addition to the forced discard executed when the buffer overflows (as in PMD), a threshold is defined at a certain buffer occupancy level. If a message *begins* to arrive when the buffer occupancy is above this threshold, the message is not accepted to the network element (i.e., all its packets are dropped). In this method, entire messages are discarded, while in PMD, only "tails" of messages are discarded where the beginnings are transmitted wastefully.

In [9], PMD and EMD are studied. Turner shows that the need for high queue capacity, in order to achieve high efficiency, grows with the number of virtual circuits. The buffer fill level over time is analyzed, and an EMD with Hysteresis algorithm is suggested, to achieve high efficiency with smaller queue capacities. The fair EMD with hysteresis algorithm is introduced in an attempt to achieve a level of fairness among the competing virtual circuits, when their rates differ significantly.

B. Performance Objectives

In [5] and in [9], the objective was to compare between discarding policies and noncontrolled systems based on the *effective throughput*. Effective throughput is the ratio of good packets on the outgoing link to the total outgoing flow. Good packets are those that belong to messages that were successfully transmitted in their entirety. However, the yardstick of effective throughput only shows how much of the transmitted traffic is not a waste, but disregards the quality of service the user gets, i.e., the percentage of its traffic that is transmitted successfully. Messages that were completely discarded by the network do not affect the throughput, yet make a big difference to the application. In other words, a high effective throughput (near 1) can be achieved even for situations where a very small percentage of the user's data is successfully transmitted by the network element (causing it to be retransmitted many times further increasing the load).

In this paper, we define another performance yardstick—*goodput*. Goodput is the ratio of good packets out of the total number of packets that arrive at the network element's input. This performance measure represents the percentage of user's data that is successfully transmitted by the network element, and that the network can charge for. This objective better represents both the quality of service the user gets and the utilization of the network element (i.e., *out-flow/in-flow*, where *out-flow* comprises only the useful part of the outgoing data flow).

In this paper we develop and analyze a model for systems using PMD and EMD policies. From an analytic point of view, the main contribution of this paper is the introduction of a novel recursion for the computation of the goodput. The analysis shows a remarkable goodput improvement when any message-based discarding policy is applied, and that the EMD policy performs better then PMD, especially under high loads. We also compute an optimal EMD threshold for maximum goodput at different input loads. We then extend the basic model to include fixed-size cells, multistage models, and on–off sources.

II. The Model

The model we use in this paper to study the behavior of various discarding policies is based on the dispersed message model introduced in [2]. According to this model, a message consists of a block of consecutive packets, which corresponds to a higher layer protocol data unit (application). The arrival epochs of the packets are dispersed over time, i.e., the packets that compose the message arrive to the system at different time instants. TCP/IP-based systems [3] are examples in which the application message is segmented into packets which are then transmitted through the network. At the receiving end, the transport protocol reassembles these packets back into a message before the delivery to higher layers takes place.

We consider systems with variable length *messages*, that is, the packets that arrive to the system belong to messages whose length is geometrically distributed with parameter q (independent from message to message). Thus, the mean length of a message is $1/q$ packets. Variable message size is typical in data applications where the message can be a document, an e-mail message, or an arbitrary file. This model also assumes a variable size packet which may correspond to some natural partition of the message (e.g., sections of a document, paragraphs of the e-mail message, etc.). In the application layer, a session can be defined in which one or more messages are transmitted through the network from a source to a destination. We assume that packets that belong to a specific session arrive according to a Poisson process with rate λ and the transmission time of a packet is exponentially distributed with rate μ. In Section V-A we will consider the case of fixed-length packets that is typical to ATM networks. In Section V-C we will consider on–off sources.

The network element in our model has a single finite input queue that can contain at most N packets (either buffered or being transmitted). When a packet arrives at the network element, it enters the input queue, if space is available. Otherwise, the packet is discarded (dropped). A packet leaves the queue when the server is available, i.e., after the service of the previous packet is completed. A packet is transmitted by the server of the network element through its service time. Hence, in terms of packets, the network element can be viewed as an $M/M/1/N$ model, with arrival rate λ and service rate μ. The *load* on the network element is defined as $\rho = \lambda/\mu$.

In terms of messages, the behavior of the network element is more complicated. Naturally, all packets that belong to a specific message have to be transferred successfully in order for the message to be useful at the receiving end. Therefore, in most applications, even if a single packet of a message is discarded, the whole message has to be retransmitted. This implies that it is wasteful to forward packets that belong to a corrupted message (a message with at least one discarded packet). In order to reduce the waste of network resources, we consider two policies that discard packets even if the buffer of the network element is not full.

The first policy is the PMD. According to this policy, whenever a packet of a specific message is discarded since it arrived to a full buffer, all subsequent packets that belong to the same (corrupted) message are also discarded, irrespective of the state of the buffer upon their arrival. It is clear that this policy avoids sending packets that are clearly of no use. This also allows the network element some time to empty its input buffer, and increases the chances of the next message being successfully transmitted. Note, however, that the PMD policy is still wasteful since all packets that belong to the corrupted message, and have been accepted to the buffer before the first packet of the message that was discarded, will be transmitted (some of them may have been transmitted already upon the first discard), although they can be of no use at the receiving end.

The second policy, called the EMD, attempts to overcome the above drawback by rejecting whole messages that are unlikely to make it. To that end, the network element fixes a fill-level threshold K (K is an integer, $0 \leq K \leq N$). Instead of discarding packets only when the buffer is full, the network element discards all packets that belong to messages that started to arrive when the contents of its buffer had been above the threshold K. Note that while the network element discards entire messages that are in danger of becoming corrupted, it may discard messages that will not have been corrupted.

The performance measure used in this paper to compare the discarding policies is the *goodput* of the network element. Goodput is defined as the ratio between the amount of "good" packets on the outgoing link of the network element and the total amount of incoming packets. A "good" packet is a packet that belongs to a noncorrupted message. The goodput represents the percentage of user's traffic that is of value to the user, and that the network can charge for.

The setting of the parameter K that maximizes the goodput depends on the load at the network element's input. For a moderate load (i.e., $\rho < 1$), setting K too low prevents the usage of a significant part of the buffer and increases the chances of discarding messages that will not have been corrupted. On the other hand, for high-load situations, setting K too high (near N) may cause accepting messages that are highly probable to contain discarded packets. Setting K at different queue levels for different input loads allows us to maximize the chances of an entire message to be accepted to the buffer and be successfully transmitted by the network element (thus maximize the network element's goodput). Our analysis shows that there exists an optimal threshold K that can be found for any given load, and that for moderate loads, the PMD policy (i.e., $K = N$) is best.

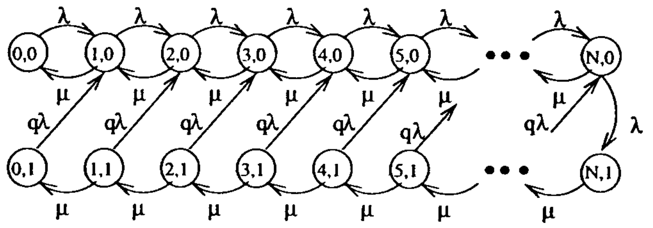

Fig. 1. A network element under the PMD policy.

III. Analysis

A. Discarding Policies Analysis

In this subsection we present queuing models with which we analyze the various discarding policies. The actual goodput derivation is deferred to the next subsection.

A network element that employs no discarding policy (other than discarding packets that arrive when the buffer is full) is modeled as an $M/M/1/N$ queue with arrival rate λ and service rate μ $(\rho = \lambda/\mu)$. A packet that arrives at an element that has N queued packets is discarded (not admitted to the queue). Let P_j $(0 \leq j \leq N)$ be the steady-state probability of having j packets in the system. Then it is well known [7] that $P_j = \rho^j / \Sigma_{i=0}^{N} \rho_i, 0 \leq j \leq N$. With these probabilities, the goodput of the network element can be derived, as is described in the next subsection.

For the PMD policy we recall that if a packet arrives when the queue is full, it is discarded and all subsequent packets that belong to the same message are also discarded until a head-of-message packet (a new message) arrives. To model this we must distinguish between two modes: the *normal mode* in which packets are admitted, and *discarding mode* in which arriving packets are discarded. The state transition diagram for this policy is given in Fig. 1. In the diagram, a state $(j, 0)$ describes the system having j packets operating in the normal mode, while a state $(j, 1)$ describes the system with j packets operating in the discarding mode. In particular, when the system is in state $(N, 0)$, the buffer is full; a packet that arrives at this state is discarded and the system enters state $(N, 1)$. Once a packet is discarded, the following packets belonging to the same message must be discarded. Since the length of the message is geometrically distributed, each of the subsequent packets belongs to the message with probability $p = 1 - q$ and hence is discarded with that probability. A head-of-message packet arrives with probability q. If the queue level upon that arrival is $j < N$, the packet is admitted to the queue and the system returns to normal mode, to state $(j + 1, 0)$. Let $P_{j,l}$ $(0 \leq j \leq N, l = 0, 1)$ be the steady-state probability of having j packets in the system and the system is in mode

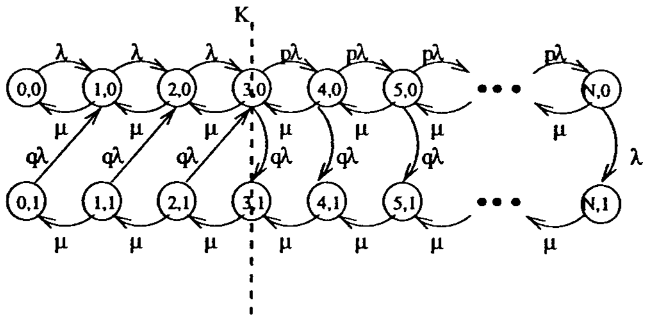

Fig. 2. A network element under the EMD policy.

l ($l = 0$ normal; $l = 1$ discarding). Then from Fig. 1 we have the following set of equations whose solution yields the steady state probabilities (with $\Sigma_{i=0}^{N}\ (P_{i,0} + P_{i,1}) = 1$):

$$\begin{aligned} \lambda P_{0,0} &= \mu P_{1,0}; \qquad q\lambda P_{0,1} = \mu P_{1,1} \\ (\lambda+\mu)P_{i,0} &= \lambda P_{i-1,0} + \mu P_{i+1,0} + q\lambda P_{i-1,1} \\ &\qquad 1 \le i \le N-1 \\ (q\lambda+\mu)P_{i,1} &= \mu P_{i+1,1} \qquad 0 \le i \le N-1 \\ (\lambda+\mu)P_{N,0} &= \lambda P_{N-1,0} + q\lambda P_{N-1,1} \\ \mu P_{N,1} &= \lambda P_{N,0}. \end{aligned} \tag{1}$$

The EMD policy, as mentioned before, is similar to the PMD, except that an additional threshold level is defined, say K. If a message starts to arrive when the system contains more than K packets, then all the packets of that message are discarded. State diagram for this policy is given in Fig. 2. As for the PMD, a state $(j, 0)$ describes the system when there are j packets in the buffer and arriving packets enter the buffer, while a state $(j, 1)$ describes the system when there are j packets in the buffer and each arriving packet is discarded. In particular, if a head-of-message arrives (with probability q) when the queue level is $j \ge K$, the packet is not admitted to the queue and the system enters state $(j, 1)$ in the discarding mode. The system remains in that mode, as described for the PMD model, until another head-of-message packet arrives (arrival with probability q). If that packet arrives when the queue level is $j < K$, then the packet is accepted and the system enters state $(j + 1, 0)$ in the normal mode. If the queue level at that arrival was $j \ge K$, the system stays in the discarding mode, that packet and all subsequent packets that belong to the new message (arrivals with probability $p = 1 - q$) are discarded. Then from Fig. 2 we have the following set of equations whose solution yields the steady-state probabilities (with $\Sigma_{i=0}^{N}\ (P_{i,0} + P_{i,1}) = 1$):

$$\begin{aligned} \lambda P_{0,0} &= \mu P_{1,0}; \qquad q\lambda P_{0,1} = \mu P_{1,1} \\ (\lambda+\mu)P_{i,0} &= \lambda P_{i-1,0} + \mu P_{i+1,0} + q\lambda P_{i-1,1} \\ &\qquad 1 \le i \le K \\ (\lambda+\mu)P_{i,0} &= p\lambda P_{i-1,0} + \mu P_{i+1,0} \\ &\qquad K+1 \le i \le N-1 \\ (\lambda+\mu)P_{N,0} &= p\lambda P_{N-1,0}; \qquad \mu P_{N,1} = \lambda P_{N,0} \\ (q\lambda+\mu)P_{i,1} &= \mu P_{i+1,1} \qquad 0 \le i \le K-1 \\ \mu P_{i,1} &= \mu P_{i+1,1} + q\lambda P_{i,0} \qquad K \le i \le N-1. \end{aligned} \tag{2}$$

B. Goodput Analysis

We recall that the goodput $\mathcal{G}$ is the ratio between total packets comprising good messages exiting the system and the total arriving packets at its input. Let $\mathcal{W}$ be the random variable that represents the length (number of packets) of an arriving message. Let $\mathcal{V}$ be the random variable that represents the success of a message, $\mathcal{V} = 1$ for a good message, and $\mathcal{V} = 0$ for a message which has one or more dropped packets. Then

$$\mathcal{G} = \frac{\sum_{n=1}^{\infty} n \cdot P(\mathcal{W} = n, \mathcal{V} = 1)}{\sum_{n=1}^{\infty} n \cdot P(\mathcal{W} = n)}. \tag{3}$$

Since the length of an (arbitrary) arriving message is geometrically distributed with parameter q, we have

$$\mathcal{G} = q \cdot \sum_{n=1}^{\infty} n \cdot P(\mathcal{W} = n, \mathcal{V} = 1). \tag{4}$$

The probability of an incoming messages of n packets to be transmitted successfully, can be expressed as follows:

$$\begin{aligned} &P(W = n, \mathcal{V} = 1) \\ &\quad = P(\mathcal{V} = 1 | \mathcal{W} = n) P(\mathcal{W} = n) \qquad n \ge 1. \end{aligned} \tag{5}$$

The second element in the product is the distribution of the length of arriving messages, i.e., $P(\mathcal{W} = n) = q(1-q)^{n-1}$. The first element is found from the conditional probability of the success of a message of length n given that its first packet arrived when there were i packets in the queue (or $(N - i)$ empty places). Let $\mathcal{Q}$ be the random variable representing the queue occupancy at the arrival of a head-of-message packet. Then

$$\begin{aligned} &P(\mathcal{V} = 1 | \mathcal{W} = n) \\ &\quad = \sum_{i=0}^{N} P(\mathcal{V} = 1 | \mathcal{W} = n, \mathcal{Q} = i) P(\mathcal{Q} = i) \end{aligned} \tag{6}$$

where $P(\mathcal{Q} = i) = P_{i,0} + P_{i,1}$ and $P_{i,j}$ are taken from the solution of (1) for PMD and the solution of (2) for EMD. This is true since the head-of-message packet is an arbitrary packet (that sees the stationary probabilities upon arrival), and since the length of an arriving message is independent of the queue state.

From the above we get

$$\begin{aligned} &\sum_{n=1}^{\infty} n \cdot P(\mathcal{W} = n, \mathcal{V} = 1) \\ &\quad = \sum_{n=1}^{\infty} n \cdot P(\mathcal{W} = n) \\ &\qquad \cdot \sum_{i=0}^{N} P(\mathcal{V} = 1 | \mathcal{W} = n, \mathcal{Q} = i) P(\mathcal{Q} = i) \end{aligned}$$

which yields the following as the expression of the goodput:

$$\begin{aligned} \mathcal{G} &= q \sum_{n=1}^{\infty} n \cdot q(1-q)^{n-1} \\ &\quad \cdot \sum_{i=0}^{N} P(\mathcal{V} = 1 | \mathcal{W} = n, \mathcal{Q} = i) P(\mathcal{Q} = i). \end{aligned} \tag{7}$$

To complete the calculation of the goodput, we therefore have to evaluate the conditional probabilities $S_{n,i} \triangleq P(\mathcal{V}=1|\mathcal{W}=n, \mathcal{Q}=i)$. These probabilities are computed recursively as follows.

Consider first a system that employs the PMD policy. Assume that the head-of-message packet arrives at a system at state $Q=i$, and the message is of length $n \leq N$. Then, if $i \leq N-n$, there is enough space in the buffer to accommodate the whole message and the message is guaranteed to be good. This is stated in the following equation:

$$S_{n,i}=1 \qquad 0 \leq i \leq N-n, \qquad 1 \leq n \leq N. \tag{8}$$

If $i=N$, i.e., the system is full, then the head-of-message packet is not admitted and the message is not a good one. Hence

$$S_{n,N}=0 \qquad 1 \leq n \leq N. \tag{9}$$

The above two equations establish the boundary (initial) condition for the recursion. Continuing with larger values, we have for $N-n+1 \leq i \leq N-1$ and $1 \leq n \leq N$

$$S_{n,i}=(1-r)S_{n-1,i+1}+rS_{n,i-1} \tag{10}$$

where $r \triangleq \mu/(\mu+\lambda)$ is the probability that a departure occurs before an arrival. The explanation of (10) is simple. If the next event following the arrival of the head-of-message packet is an arrival of a packet (which happens with probability $1-r$), the probability that the message is successful is $S_{n-1,i+1}$, since this packet can be viewed as the head-of-message packet of a message of length $n-1$ that arrives at a system with $Q=i+1$ packets. If the event following the arrival of the head-of-message packet is a departure of a packet (which happens with probability r), the probability that the message is successful is $S_{n,i-1}$, since the situation is as if the head-of-message packet had arrived at a system with $Q=i-1$ packets.

Combining (8), (9), and (10), we have that for $1 \leq n \leq N$

$$S_{n,i} = \begin{cases} 1 & 0 \leq i \leq N-n \\ (1-r)S_{n-1,i+1}+rS_{n,i-1} & N-n+1 \leq i \leq N-1 \\ 0 & i=N. \end{cases} \tag{11}$$

For messages of length $n > N$, there is no situation where success is guaranteed from the outset, and success depends more heavily on the evolution of the system after the arrival of the head-of-message. For the same reason as explained above, (10) holds for $1 \leq i \leq N-1$. A slightly different relation holds when the head-of-message packet arrives at an empty system $(i=0)$. If the head-of-message arrived at an empty system and the next event is a departure (which happens with probability r), the system is empty again and no further departures are possible; thus the probability that the message is successful is $S_{n-1,0}$ since the arrival of the next packet can be viewed as an arrival of the head-of-message packet of a message of length $n-1$ to an empty system. Thus, for $n > N$ we have

$$S_{n,i} = \begin{cases} (1-r)S_{n-1,i+1}+rS_{n-1,i} & i=0 \\ (1-r)S_{n-1,i+1}+rS_{n,i-1} & 1 \leq i \leq N-1 \\ 0 & i=N. \end{cases} \tag{12}$$

The recursions (11) and (12) are computed in ascending order of n $(n=1,2,\cdots)$ and ascending order of i $(i=0,1,2,\cdots,N)$.

In a system that employs the EMD policy, the above recursions remain correct only when the head-of-message packet arrives at the system when the number of packets is below the EMD threshold, i.e., $\mathcal{Q}=i < K$. If the head-of-message packet arrives when the system occupancy is above this threshold, the message as a whole is not admitted to the system, and hence is not a good one. We thus get for EMD policy the following probabilities:

$$\hat{S}_{n,i} = \begin{cases} S_{n,i} & i < K \\ 0 & K \leq i \leq N. \end{cases} \tag{13}$$

IV. Numerical Results

The parameters we use in our examples were set to correspond to realistic ratios between queue size and mean message length. Ratios of 1:20 (i.e., queue size of 20 times the mean message length) to 1:2, were checked. From the geometric message length distribution, it follows that the mean message length is $len = 1/q$ packets. In our examples, we set the queue size to $N=120$, and calculated the goodput for messages of mean length of 6, 15, 30, and 60. The traffic loads on the network element (ρ) are in the range of 0.8–2.2 where loads of $\rho < 1$ are referred to as moderate loads, while higher loads correspond to congestion buildups or noncooperative users.

Fig. 3 shows the goodput of the network element for mean message lengths of 6 and 30 packets, as a function of the offered load and for different policies: without any discarding policy, when PMD policy is introduced in the network element and when EMD policy is applied with a fixed threshold at half the queue size. Fig. 4 is a zoom of Fig. 3 for average message size of 30 packets, at moderate loads with PMD or EMD applied.

It is evident that when high loads are introduced, both discarding policies perform much better than a system with no discarding policy. At moderate loads, the PMD policy and the EMD policies perform similarly, with a slight advantage to the PMD policy. For heavy traffic loads, the EMD outperforms the PMD by up to 20% in terms of the goodput, and improves the network element's performance by a factor of up to 6. It also appears that the behavior of the system is not sensitive to changes in average message lengths. Furthermore, a controlled system (where some discard policy is implemented) is less sensitive to the message length, and the EMD is, again, better than the PMD in that perspective. In all cases, systems with shorter mean message length yield better goodput.

Given the superior performance of the EMD policy, it is natural to investigate the optimal threshold K_{opt} with respect to loads, message lengths, and buffer sizes. Fig. 5 depicts K_{opt} for a queue of size $N=120$, mean message lengths of 6, 15, 30, and 60, and loads in the range of 0.8 to 2.2. Fig. 5 shows that the optimal threshold is not very sensitive to the average messages length.

Fig. 6 depicts K_{opt}/N for queue sizes of $N=30,60,90$, mean message length of 6, and loads in the range of 0.8 to 2.2. We consider the ratio of the optimal threshold to the queue size

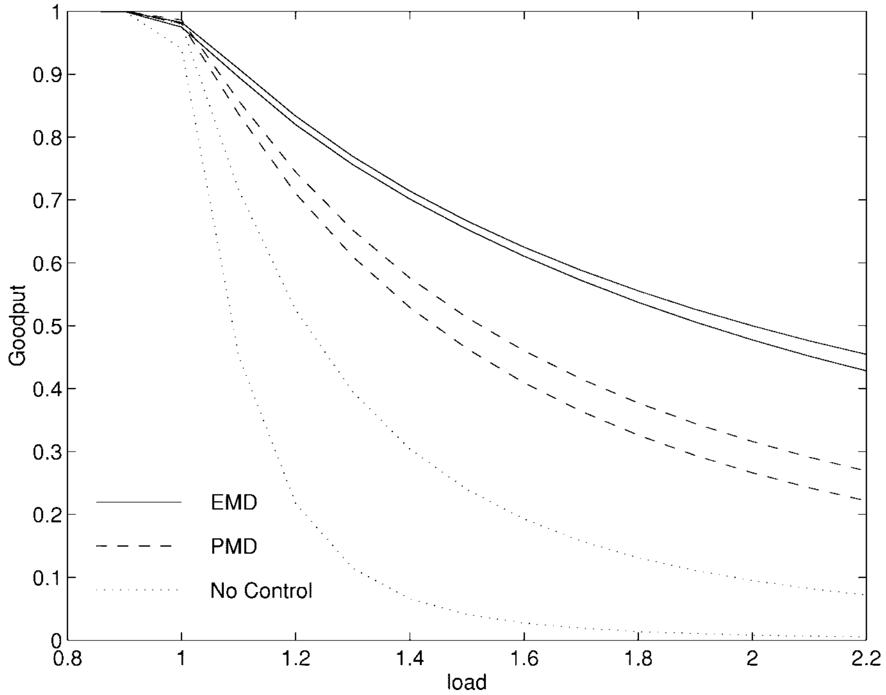

Fig. 3. Goodput versus load for EMD/PMD/no control [$N = 120; K = 60; 1/q = 6, 30$].

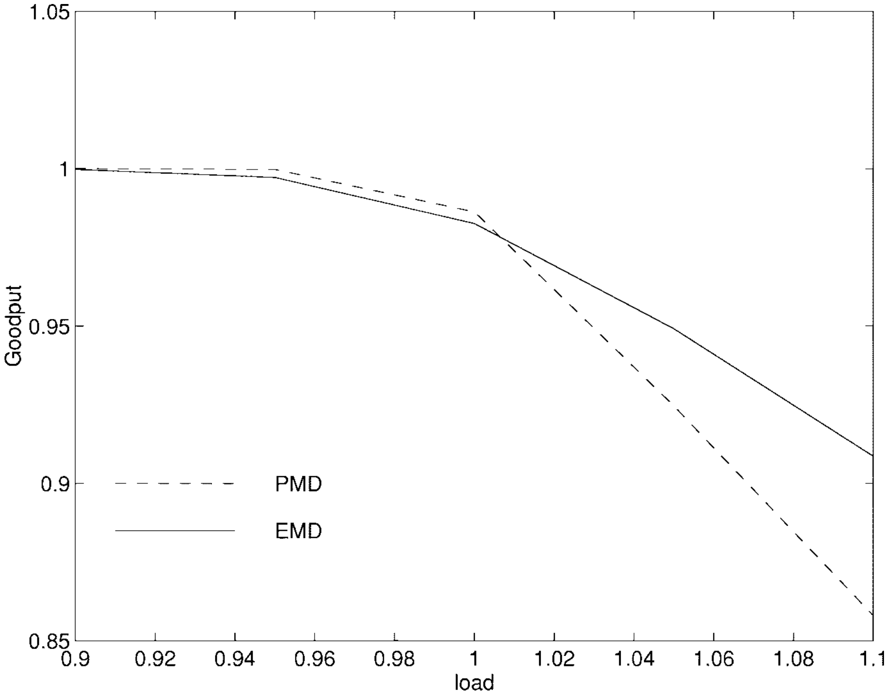

Fig. 4. Goodput versus load for EMD/PMD policies [$N = 120; K = 60; 1/q = 30$].

since we are interested in proportions rather than absolute values. The figure demonstrates that changes in the queue size for a fixed average message size hardly affect the optimal threshold. This is an encouraging result since it means that one can easily approximate the optimal threshold for a given system.

Fig. 7 shows the dependence of goodput on K for a system employing the EMD policy with several loads (the value of the goodput obviously decreases as the load increases). The figure shows that the goodput is rather insensitive to the EMD threshold, in a very wide range of thresholds (not too low and not too high). This is a remarkable result as it means that the EMD policy is rather robust and the choice of the threshold is not a crucial one. An optimum threshold, as expected, does exist but is not significant at all in terms of goodput (and is therefore invisible in these graphs). The explanation of this phenomenon is as follows. Clearly, when the value of the threshold is set high, the system behaves almost as with the PMD policy and loses its relative advantage. Similarly, when

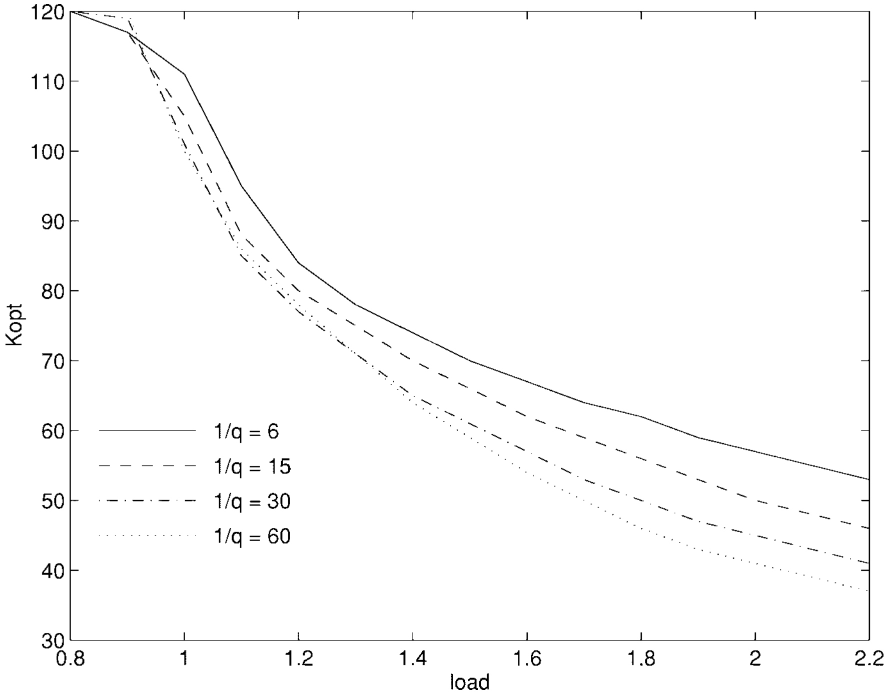

Fig. 5. K_{opt} versus load for the EMD policy $[N = 120; 1/q = 6, 15, 30, 60]$.

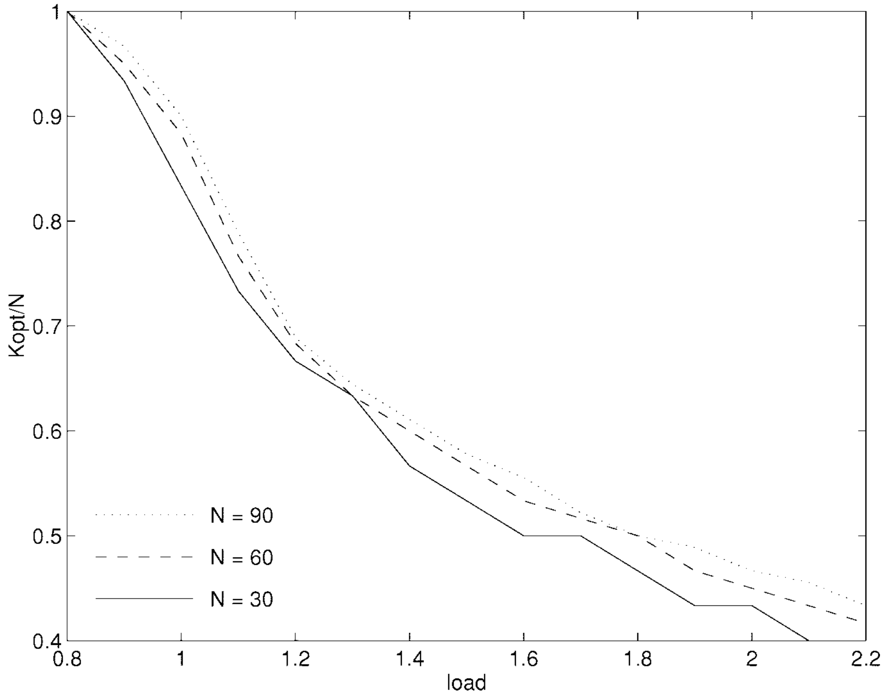

Fig. 6. K_{opt}/N versus load for the EMD policy $[N = 30, 60, 90; 1/q = 6]$.

the threshold is set too low, the buffer is not well utilized since many messages that could have been accepted are discarded. In a medium setting, at relatively high loads, the system will operate most of the time with a buffer occupancy around K. Lowering K means that longer messages are more likely to make it, but these messages are rather rare and therefore their effect on the goodput is negligible. Thus, although an optimal threshold does exist, its effect on the goodput is nonessential.

Fig. 8 depicts the probabilities of a message of length n packets to be discarded by the system. (Here the probability does not include the probability of such a message arriving at the queue.) In this example, the queue size is of length $N = 120$, and messages are an average length of 30 packets, the traffic load is $\rho = 1.2$. It is evident that the system with no control gives little chance for any message, especially the longer ones. The PMD control performs much better, and the loss

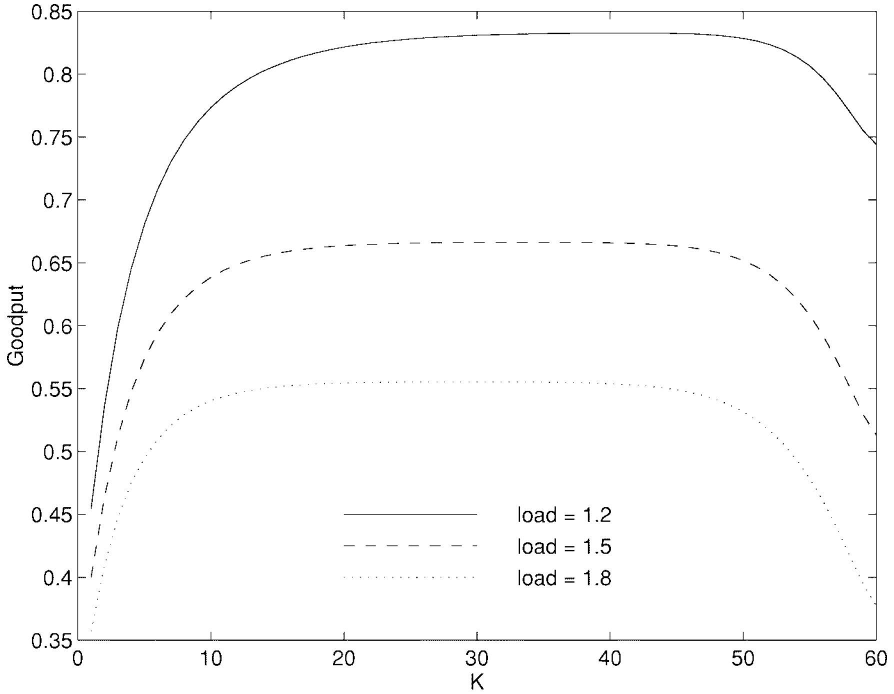

Fig. 7. Goodput versus K under EMD policy [$N = 90; 1/q = 6$; load $= 1.2, 1.5, 1.8$].

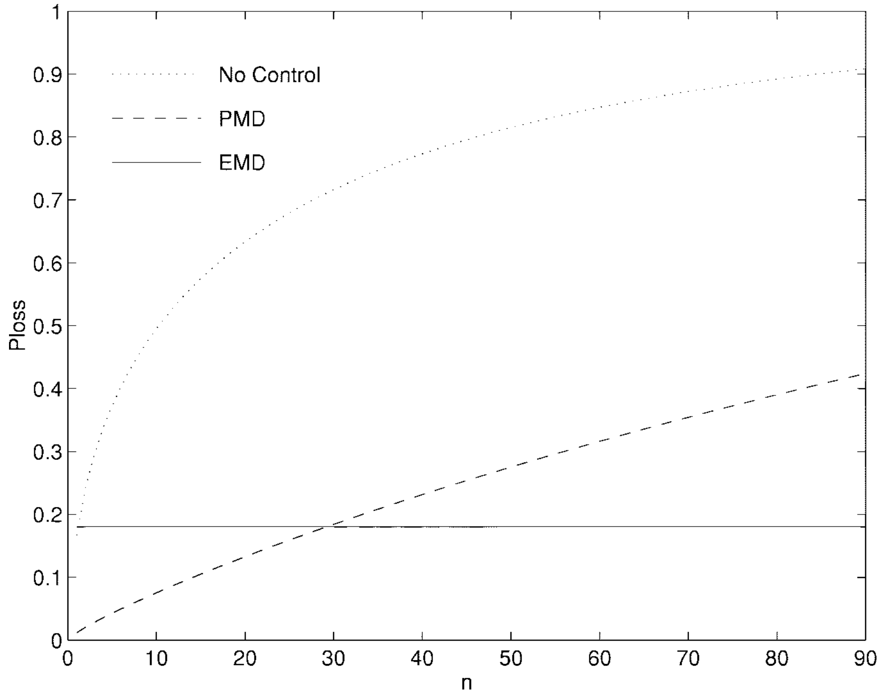

Fig. 8. Probability of message loss versus message length for the EMD/PMD/no control [$N = 120; K = 60$; load $= 1.2; 1/q = 30$].

probability curve rises much slower. But it is clear that EMD control is the most fair mechanism in terms of loss probability for a message of any length. In this figure we can see again that PMD, in comparison to EMD, performs better for short messages and worse for messages longer than the expected length.

Fig. 9 depicts the distribution of the length of a successful message. This example has the same parameters as the previous one. Again, a system with no control gives the poorest chances for any message to be transmitted successfully. The PMD mechanism gives better chances for shorter messages than the EMD mechanism. Larger messages (above the average length) will have better chances to be transmitted successfully under the EMD mechanism. As we can see, the differences in the success probabilities between PMD and EMD are not very significant. The improvement to the *goodput* with EMD is granted by the larger messages that, although

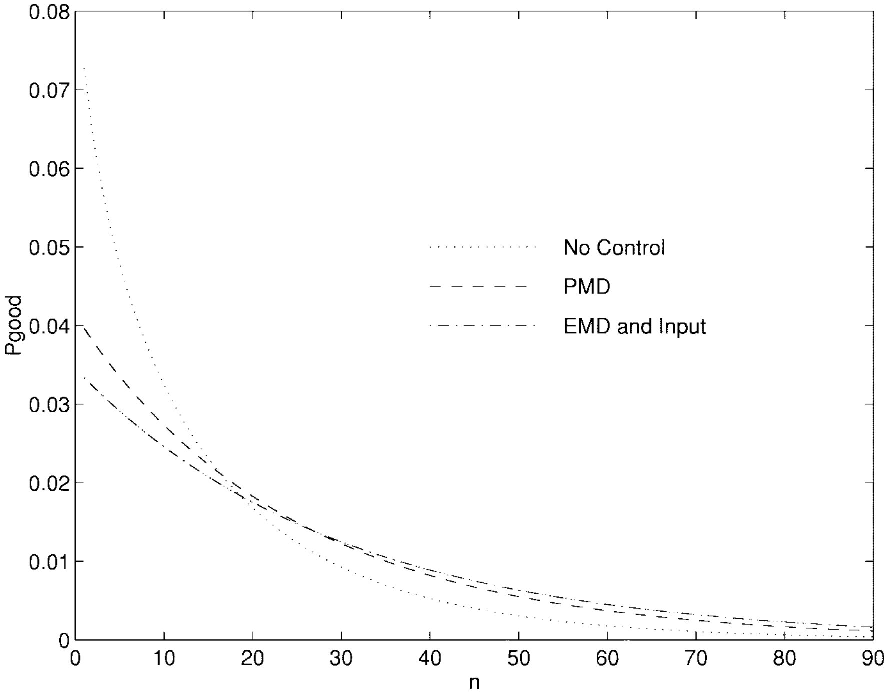

Fig. 9. Distribution of successful message length with EMD/PMD/No control $[N = 120; K = 60;$ load $= 1.2; 1/q = 30]$.

arriving more rarely, contribute many good packets to the output traffic. In the figure, the arriving message distribution is also given—the EMD graph unites with it, which means that the EMD mechanism is the most fair one for various messages' lengths, and that it preserves the message length distribution of the arriving traffic.

V. Extensions to the Basic Model

A. ATM Networks

In our model, we assumed that messages are segmented into exponentially sized packets, which translates into an exponential transmission (service) time. In ATM networks, however, messages are segmented into fixed size packets, called cells. To test the applicability of our model to ATM networks, we simulated an ATM network element, employing a discarding policy, and compared it to an equivalent exponential network element. The results exhibit an extremely tight match. Fig. 10 shows the goodput of an ATM network element employing EMD policy (the dashed curve) resulting from simulation, and the calculated goodput of the modeled system with the same parameters and an exponential packet size with mean identical to that of the ATM cell. It is clearly evident that as far as goodput (and throughput) is concerned, the difference between the exponential and deterministic packet sizes (with the same mean service time) is negligible. Therefore, the results of our model can be used to describe ATM network elements, as a specific case of message-based high-speed network elements.

Implementation of a selective discard policy in ATM networks is suggested in [6] for applications that use AAL5 as an adaptation layer to the ATM layer. AAL5 uses a single bit in the ATM cell header to indicate the end-of-message cell. It is proposed that this bit be used to implement the discarding mechanism. When a cell is dropped at an intermediate network element, the virtual channel connection to which this cell belongs is kept in memory and all subsequent cells belonging to this connection are dropped, until (including) the end-of-message cell is encountered.

PMD and EMD implementations are quite similar. They differ only in the decision when to (start to) drop a message. In the PMD, it depends on the buffer size. In the EMD, it also depends, in a trivial manner, on the threshold. In both cases, the mechanism need not look at the cells' payload or the AAL header, but only at the specific bit in every *cell's* header, and it can therefore be easily implemented in hardware.

B. Multistage Model

In previous sections we studied the goodput improvements of an isolated network element deploying the EMD mechanism. We now turn to investigate the performance of a multistage subnetwork whose elements apply the EMD mechanism to the aggregated traffic at their inputs. We assume that each stage in the subnetwork is that presented and analyzed in the previous section. The output of each stage leaves the system with probability Γ and continues to the following stage with probability $1 - \Gamma$. Thus the input into a stage is comprised of a local source and a portion of the output of the previous stage. The local source generates messages comprised of packets as described in Section II. At every stage, the EMD mechanism is applied to every arriving message (both messages from previous stage and from the local source).

The *goodput* measure we use here is for a single stage and per data source, namely, the ratio between transmitted "good" packets and incoming packets, of a specific stream (this is an "application oriented" objective).

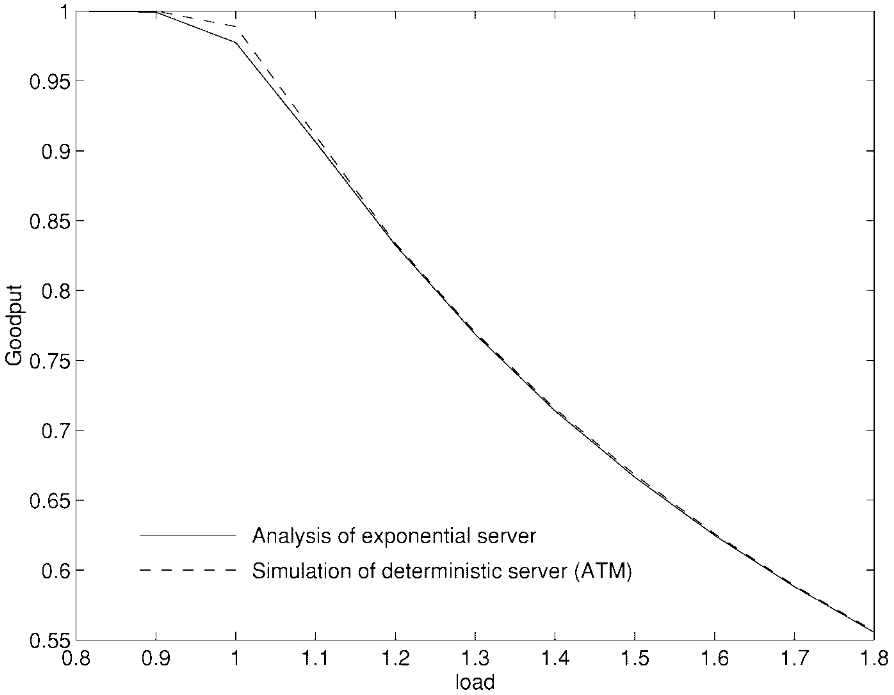

Fig. 10. Model comparison fixed versus varying packet size $[N = 100; K = 50; 1/q = 10]$.

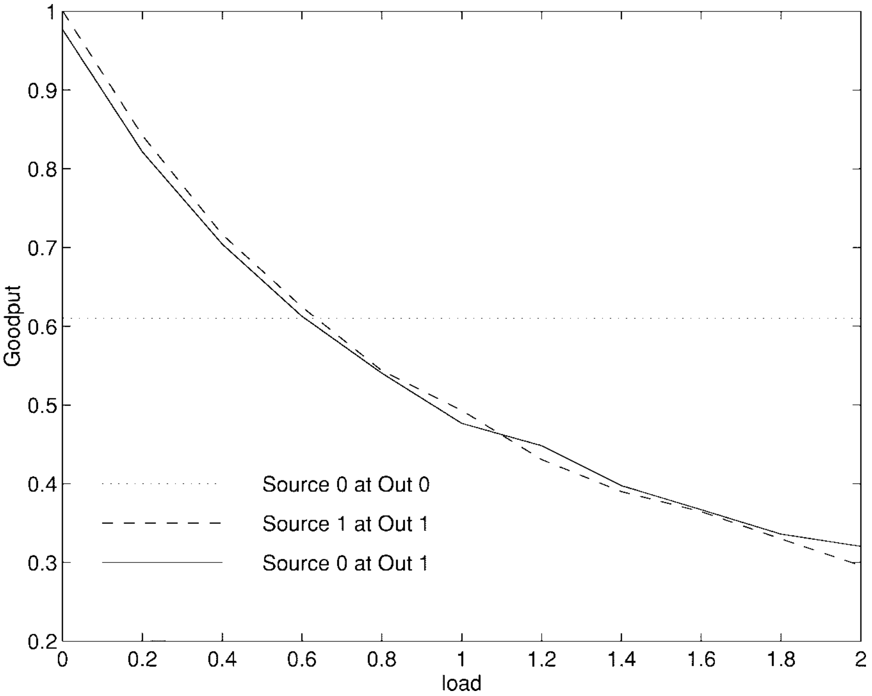

Fig. 11. Two EMD stages: goodput versus load $[N1 = N2 = 120; K1 = K2 = 60$; mean length $1/q1 = 1/q2 = 30; \rho 0 = 1.6]$.

Simulation results for two and for three stages in tandem are presented next. In Fig. 11 a two stage system is considered. The first stage serves traffic only from its local source (as in the basic model). This source generates messages of mean length 30 and its load is 1.6. The dotted line describes the goodput of the first stage. As the service rate is $\mu = 1$, it is clear that the goodput is around 0.6. The second stage receives all packets that leave the first stage $(\Gamma = 0)$, as well as those generated by the local traffic source (identical to the source at the first stage). It is seen that for a threshold at $K = 60$, the second stage gives the same goodput for both traffic streams (dashed and solid lines), varying with the load of its local source.

Fig. 12 describes the behavior of three stages in tandem implementing the EMD. The three stages have identical traffic sources with mean message length of 30 packets. Each of the three stages has queue of length 120 and the EMD threshold is set to 90. The loads of the first and second traffic sources are 0.8 (each) and the load of the third varies from almost 0 to 2.0. The dotted and dashed lines show the goodput of the

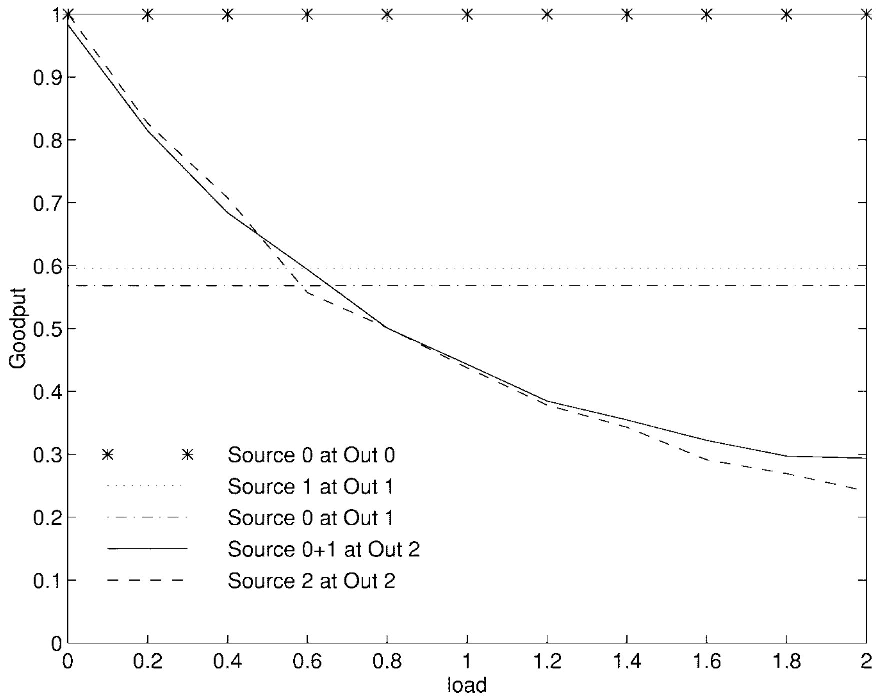

Fig. 12. Three EMD stages: goodput versus load [$N1 = N2 = N3 = 120; K1 = K2 = K3 = 90; 1/q1 = 1/q2 = 1/q3 = 30; \rho 0 = \rho 1 = 0.8$].

local sources at stage 1 and 2, respectively, the dash-dot and solid lines show the goodput of the traffic coming from the previous stage for the above. Again, it is seen that the third stage yields almost the same goodput for external and internal traffic.

C. On–Off Source Model

1) Noncontrolled On–Off Model: Consider an on–off source, where during "on" periods it generates messages segmented into packets as described in the basic model (Section II). During "off" periods, the source does not generate any traffic. The number of messages generated during the "on" period is geometrically distributed with parameter α, i.e., with every head-of-message packet, the source remains in the "on" state with probability $(1-\alpha)$ and another message is generated, or the source switches to the "off" state with probability α. The "off" period is exponentially distributed with parameter β. The state transition diagram describing the noncontrolled system is given in Fig. 13. In the diagram, a state $(j, 0)$ describes the system having j packets operating in the "on" state, while a state $(j, 1)$ describes the system with j packets operating in the "off" state.

Let $\mathcal{P} = [P_{0,0}, P_{0,1}, P_{1,0}, P_{1,1}, \cdots, P_{N,0}, P_{N,1}]$ be the vector of the above state probabilities (at steady state) and let $\mathcal{R}$ denote the transition rate matrix. Then

$$\mathcal{P} \cdot \mathcal{R} = 0; \qquad \sum_{i=0}^{N} (P_{i,0} + P_{i,1}) = 1. \tag{14}$$

The transition rate matrix can be easily derived from the state diagram in Fig. 13 and the steady-state probabilities can be calculated.

The goodput analysis for the on–off model is similar to that of the basic model, with only slight changes. Here, a distinction should be made between messages that arrive during an "on" period and messages that arrive as the first message of an "on" period, and find the system in the "off" state, i.e., one of $(j, 1)$ states. In calculating the success probability of a message, given it has arrived when the buffer was at some state, we should consider the probability of this arrival (either within an "on" period, or as the ending of an "off" period). Let $\mathcal{M}$ be the random variable representing the state of the system, $\mathcal{M} = 0$ for the "on" state and $\mathcal{M} = 1$ for the "off" state. Let $\mathcal{N}$ be the random variable representing the number of messages arriving during an "on" period, and $\overline{\mathcal{N}}$ its average. The probability of a message finding the system in an "on" state with i packets in the queue is

$$\hat{P}_{i,0} = (\overline{\mathcal{N}} - 1)/\overline{\mathcal{N}} P(\mathcal{Q} = i, \mathcal{M} = 0) \tag{15}$$

where $P(\mathcal{Q} = i, \mathcal{M} = 0)$ are the steady-state probabilities for $(i, 0)$ from above. However, the probability of a message finding the system in an "off" period, i.e., as the first message of an "on" period with i packets in the queue, is

$$\hat{P}_{i,1} = 1/\overline{\mathcal{N}} \sum_{l=i}^{N} P(\mathcal{Q} = l, \mathcal{M} = 0) \alpha q (\mu/(\mu + \beta))^{(l-i)} \cdot (\beta/(\mu + \beta)). \tag{16}$$

Then (6) is replaced by

$$P(\mathcal{V} = 1 | \mathcal{W} = n) = \sum_{i=0}^{N} P(\mathcal{V} = 1 | \mathcal{W} = n, \mathcal{Q} = i) \Big(\hat{P}_{i,0} + \hat{P}_{i,1}\Big). \tag{17}$$

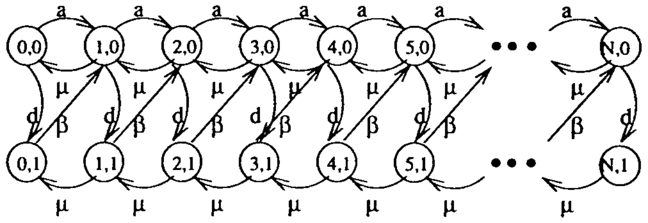

Fig. 13. The model of a network element with on–off source $[a = ((1-\alpha)q+p)\lambda; d = \alpha q\lambda]$.

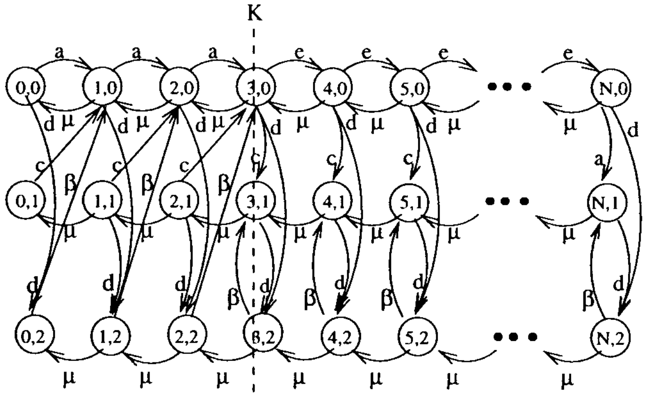

Fig. 14. On–off source—a network element under EMD policy $[a = ((1-\alpha)q+p)\lambda; c = (1-\alpha)q\lambda; d = \alpha q\lambda; e = p\lambda]$.

From the distribution of the number of messages in an "on" period, we get that $\overline{N} = 1/\alpha$. Substituting (15) and (16) into (17), and that into (5) and (4), gives the expression for thegoodput of this system:

$$\mathcal{G} = \frac{q}{\hat{P}_{i,0}+\hat{P}_{i,1}} \sum_{n=1}^{\infty} n \cdot q(1-q)^{n-1} \cdot \sum_{i=0}^{N} P(\mathcal{V}=1|\mathcal{W}=n, \mathcal{Q}=i)\Big(\hat{P}_{i,0}+\hat{P}_{i,1}\Big). \quad (18)$$

2) On–Off Model with EMD: When the EMD mechanism is deployed in the above described system, it has two modes of operation in the "on" state: normal mode and discarding mode. In the "off" state, there is only one mode of operation since no packets arrive at this state. Here, a state $(j,0)$ describes the system when there are j packets in the buffer, the source is in the "on" state, and arriving packets enter the buffer. A state $(j,1)$ describes the system when there are j packets in the buffer, the source is in the "on" state, but each arriving packet is discarded. In particular, if a head-of-message arrives (with probability q) when the queue level is $j \geq K$, the packet is not admitted to the queue and the system enters state $(j,1)$ in the discarding mode. As a head-of-message packet arrives during an "on" state, the system switches to the "off" state with probability α, no packet enters the queue, and the system enters state $(j,2)$. The "off" state ends with rate β and the system enters state $(j,0)$ or $(j,1)$ according to the queue level j. The state transition diagram of this system is given in Fig. 14.

Let $\mathcal{P} = [P_{0,0}, P_{0,1}, P_{0,2}, P_{1,0}, P_{1,1}, P_{1,2}, \cdots, P_{N,0}, P_{N,1}, P_{N,2}]$ be the vector of the above state probabilities, and let $\mathcal{R}$ denote the transition rate matrix. Then

$$\mathcal{P}\cdot\mathcal{R} = 0; \qquad \sum_{i=0}^{N}(P_{i,0}+P_{i,1}+P_{i,2}) = 1. \quad (19)$$

The transition rate matrix can be easily derived from the state diagram in Fig. 14, and the steady-state probabilities can be calculated.

With the state probabilities, the goodput of this system is calculated, as described for the noncontrolled bursty model, where in (15) $P(\mathcal{Q} = i, \mathcal{M} = 0)$ are the steady-state probabilities for the "on" state, i.e., $P_{i,0}+P_{i,1}$.

3) On–Off Models Results: Solving the equations for the state probabilities of the EMD and the noncontrolled systems with on–off sources gives goodput values for different system parameters. In Figs. 15 and 16, the improvements in goodput for systems with EMD versus the noncontrolled system are presented. Various parameters for the "on" period or "off" period length are given. In Fig. 15, α ranges from 0.01 to 0.99, and since α is the probability of ending an "on" period as α is increased, the goodput increases (and the positive effect of the EMD mechanism decreases). The explanation for this is that since α is the probability to end an "on" period, as it increases the system enters more often into an "off" state and the queue "has more time" to free space, and thus the probability of messages to be successful increases (hence goodput increases). EMD improves the goodput of the described system for all cases, but by a larger scale for small α, where the goodput of the noncontrolled system becomes very low for high loads (as in the basic "all-on" model). In Fig. 16, β ranges from 1 to 0.001. As β is the rate at which "off" periods end, the smaller it is, off periods are longer and the goodput is higher, and, again the significance of the EMD mechanism is diminished. This is expected since for long off periods, little or no congestion is developed and the need for a selective discarding policy is alleviated. Both figures show that EMD improves goodput significantly, especially under high loads (i.e., $\rho > 1$).

VI. Conclusions

This paper addresses selective discarding policies as the means to control congestion in high-speed networks. Selective discarding increases the percentage of user's data that are successfully transmitted by a network element, saving retransmissions and waste of bandwidth, and improving the quality of service even for best-effort traffic where no quality guarantees are given. Selective discarding policies require neither the cooperation of the users nor coordination with other network elements. Their introduction to the network is therefore simple and allows us to easily obtain substantial performance improvements.

In the paper, we developed an analytical model to examine the performance of systems with no discarding policies in place as well as systems that deploy the PMD and EMD policies. The results show that any message-based discarding policy mechanism provides a *remarkable* improvement in network performance compared to systems without any policy in place.

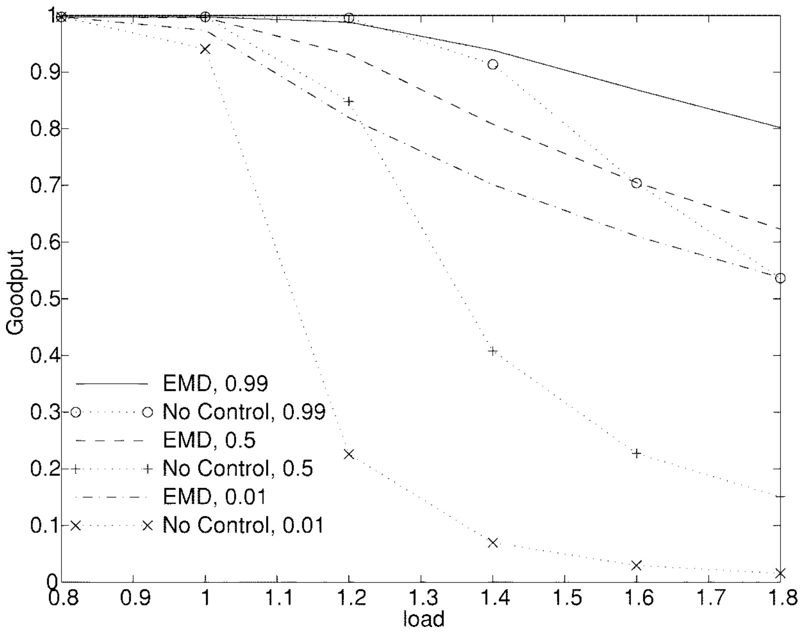

Fig. 15. On–off source: goodput versus load EMD/No control different On periods $[N = 120; K = 60; 1/q = 30; \alpha = 0.01, 0.5, 0.99; \beta = 0.1]$.

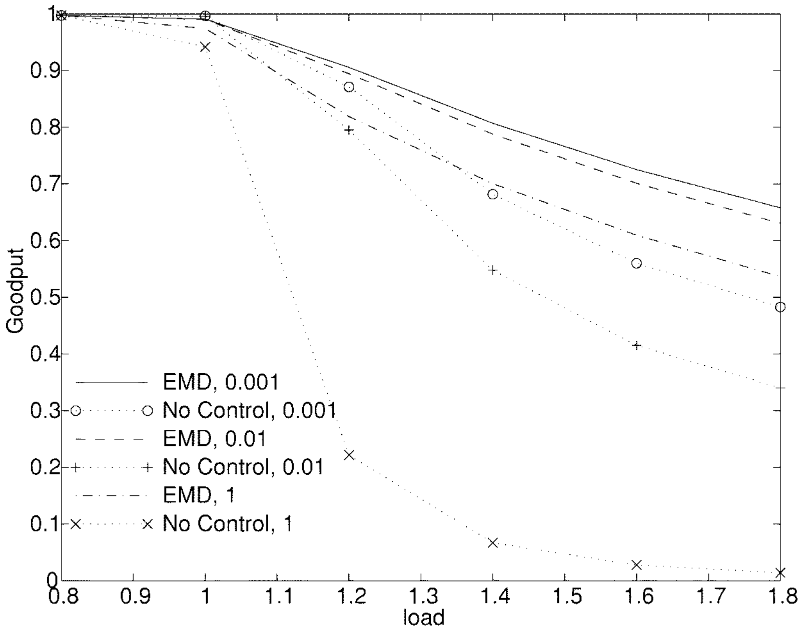

Fig. 16. On–off source: Goodput versus load EMD/No control different Off periods $[N = 120; K = 60; 1/q = 30; \alpha = 0.1; \beta = 1, 0.01, 0.001]$.

The two policies examined perform differently depending on network load. For moderate traffic loads, PMD policy gives the best performance of the network element's goodput. When the load increases and congestion is more likely to develop, the EMD policy performs better than the PMD with a significant improvement in goodput performance of the system, compared to the noncontrolled case. An optimum threshold for the EMD mechanism can be determined off-line, and set with respect to the introduced load at the network element's input queue. Analysis shows that maximal goodput is not sensitive to the setting of the optimal threshold. Furthermore, the optimum threshold is hardly sensitive to the typical size of the transmitted messages, and hence can serve various best-effort traffic applications, with no special adjustments. The adjustment of the threshold should only change with the applied load.

These results can also be applied to the specific case of ATM, where selective discarding can improve the quality of service of best-effort (or UBR) services. Our analysis shows that improvement in goodput with the EMD mechanism is also achieved for the case of an on–off source, in particular, when the basic system performs poorly. Finally, it may be interesting to further investigate a case of several sources subject to selective discarding, for their individual performance improvements and their mutual effects.

References

[1] F. Bonomi and K. W. Fendick, “Credit-based flow control for ATM networks,” *IEEE Network*, vol. 9, pp. 25–39, Mar./Apr. 1995.

[2] I. Cidon, A. Khamisy, and M. Sidi, “Analysis of packet loss processes in high speed networks,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 98–108, Jan. 1993.

[3] D. E. Comer, *Internetworking with TCP/IP*, vol. 1. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[4] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 25–39, Aug. 1993.

[5] S. Floyd and A. Romanow, “Dynamics of TCP traffic over ATM networks,” in *Proc. ACM/SIGCOMM’94*, Sept. 1994, pp. 79–88.

[6] A. E. Kamal, “A performance study of selective cell discarding using the end-of-packet indicator in AAL type 5,” in *Proc. INFOCOM’95*, Boston, Apr. 1995, pp. 1264–1272.

[7] L. Kleinrock, *Queueing Systems—Theory*, vol. 1. New York: Wiley-Interscience, 1975.

[8] H. T. Kung and R. Morris, “Credit-based flow control for ATM networks,” *IEEE Network*, vol. 9, pp. 40–48, Mar./Apr. 1995.

[9] J. S. Turner, “Maintaining high throughput during overload in ATM switches,” in *Proc. INFOCOM’96*, San-Francisco, Apr. 1996, pp. 287–295.

Yael Lapid was born in Haifa, Israel on October 11, 1967. She received the B.Sc. (cum laude) and the M.Sc. degrees from the Technion-Israel Institute of Technology, Haifa, Israel, in 1989 and 1996, respectively, all in electrical engineering.

During 1989–1994 she served in the Israeli Navy and was involved in R&D in electronic systems at Raphael Research Institute in Israel. Currently she is with ECI Telecom Ltd., Israel in the R&D department for high-speed network products. Her current interests are in system design and software design for ATM systems and networks.

Raphael Rom received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, and the Ph.D. degree in computer science from the University of Utah, Salt Lake City, UT.

He was a Senior Researcher on the research staff of SRI International in California, and subsequently joined the Faculty of Electrical Engineering in the Technion, Haifa, Israel. Since 1989 he has also been with Sun Microsystems where he recently lead and managed the high speed networking group of SunLabs. In addition he held visiting positions in IBM T. J. Watson Research Center and Stanford University. He is the coauthor of the book *Multiple Access Protocols: Performance and Analysis.* His areas of interest are algorithms for and performance analysis of data communication and wireless networks and the design of general data communication systems.

Moshe Sidi received the B.Sc., M.Sc., and the D.Sc. degrees from the Technion-Israel Institute of Technology, Haifa, Israel, in 1975, 1979, and 1982, respectively, all in electrical engineering.

In 1982 he joined the faculty of Electrical Engineering Department at the Technion. During the academic year 1983–1984 he was a Post-Doctoral Associate at the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology, Cambridge, MA. During 1986–1987 he was a visiting scientist at IBM T. J. Watson Research Center, Yorktown Heights, NY. He coauthored the book *Multiple Access Protocols: Performance and Analysis* (Springer Verlag, 1990). His research interests are in wireless networks and multiple access protocols, traffic characterization and guaranteed grade of service in high-speed networks, queueing modeling and performance evaluation of computer communication networks.

Dr. Sidi served as the Editor for Communication Networks of the IEEE TRANSACTIONS ON COMMUNICATIONS from 1989 until 1993, as the Associate Editor for Communication Networks and Computer Networks of the IEEE TRANSACTIONS ON INFORMATION THEORY from 1991 until 1994, and as an Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING from 1993 until 1997. Currently he serves as an Editor of the *Wireless Journal*.