Robust Identification of Shared Losses Using End-to-End Unicast Probes*

Khaled Harfoush harfoush@cs.bu.edu
Azer Bestavros best@cs.bu.edu
John Byers byers@cs.bu.edu

Computer Science Department
Boston University
Boston, MA 02215

Abstract

Current Internet transport protocols make end-to-end measurements and maintain per-connection state to regulate the use of shared network resources. When two or more such connections share a common endpoint, there is an opportunity to correlate the end-to-end measurements made by these protocols to better diagnose and control the use of shared resources. We develop packet probing techniques to determine whether a pair of connections experience shared congestion. Correct, efficient diagnoses could enable new techniques for aggregate congestion control, QoS admission control, connection scheduling and mirror site selection. Our extensive simulation results demonstrate that the conditional (Bayesian) probing approach we employ provides superior accuracy, converges faster, and tolerates a wider range of network conditions than recently proposed memoryless (Markovian) probing approaches.

1. Introduction

One of the defining principles of the network protocols used in the Internet lies in their ability to manage and share network resources fairly across competing connections. This is a notable engineering achievement, especially in light of the fact that individual connections exert distributed control over their transmission rates. But this fine-grained autonomy that connections exert coupled with our limited understanding of the interactions that multiple (TCP) connections impose limits the degree to which network resources can be tightly controlled. In our ongoing work as part of the Mass project [27], we investigate circumstances in which better diagnosis of network resources can be obtained, which we hope will lead to improved control mechanisms.

In this paper, we explore the effects of *concurrency* on diagnosing network conditions. As an example, a popular Internet server (e.g. Web server, proxy server, content distribution outlet, streaming media server, etc.) may potentially command a large number of concurrent connections. While most of these connections are likely to be to *different* clients, many may in fact be traversing the *same* set of congested resources. If connections sharing common congested resources can be identified, then improved network resource usage can be achieved through judicious allocation of bandwidth. In particular, rather than controlling connections traversing congested network resources independently, an Internet server could apply an aggregate control mechanism to such connections. Examples of such mechanisms include the aggregate congestion management technique proposed under the Congestion Manager framework [2] and the ATCP protocol [3]. Applications of this technique could extend well beyond the domain of congestion control to QoS admission control, selecting multiple mirror sites in parallel [5] and improved connection scheduling at webservers. But in order for any such control strategies to be practical, an endpoint must be able to quickly and accurately identify whether or not a set of its connections to remote locations traverse the same set of congested resources.

Helpfully, the end-to-end measurements made in the course of normal operations by most transport protocols provide a wealth of information about the end-to-end characteristics of a path in the network. For example, although the nodes comprising the path may not be known, end-to-end bottleneck bandwidth rates, round-trip times and packet loss statistics can all be inferred from the dynamics of a TCP connection [1]. In this paper, we show that in addition to the above connection-specific parameters, end-to-end measurements from different connections can be correlated in order to identify connections that share similar network conditions. What constitutes "similar conditions" depends on the purpose of the identification process. For the purpose of this paper, we specify two possible problem statements, defined below.

In each of the problem domains, we consider a scenario in which there is a single server, which has active connections (e.g. TCP flows) to two distinct clients,

*This work was partially supported by NSF research grants CCR-9706685 and ANI-9986397.

both experiencing steady-state packet loss rates of at least ϵ, for some constant $\epsilon > 0$. We assume that the paths from server to the clients form a tree, which from the server's perspective consists of a sequence of *shared* links followed by a sequence of *disjoint* links, in which the shared portion of the sequence may be empty.

Loss Sharing: For these two connections, determine if the incidence of packet loss on the shared portion of the tree is at least $\frac{\epsilon}{k}$, for a fixed constant $k \geq 1$.

Bottleneck Equivalence: For these two connections, determine if the incidence of shared loss is greater than the incidence of disjoint loss.

We have formulated these problem statements as yes-no questions, but note that the techniques we develop extend to the related question of estimating the incidence of shared loss. Also, it should be clear that while Bottleneck Equivalence implies Loss Equivalence, the converse does not hold.

Paper Contributions: This paper proposes an analytical technique for the robust determination of both loss and bottleneck equivalence for pairs of unicast connections emanating from the same server. Our technique relies solely on end-to-end loss information available at the server as a result of passive monitoring or of active probing. We present extensive simulation results that demonstrate the effectiveness of our approach as compared with the recently proposed approach of Rubenstein, Kurose, and Towsley [26] and the robustness of our technique to a wider variety of network and cross-traffic characteristics than previous work considered.

2. Related Work

Inference and prediction of network conditions is of fundamental importance to a range of network-aware applications, so it is no surprise that numerous research efforts are underway in this space. We classify and survey these research efforts in the context of our current work.

One widely adopted strategy is to mine the data collected by network-internal resources, such as BGP routing tables, to generate performance reports [12, 15, 18, 9, 16]. This approach is best applied over long-time scales to produce aggregated analyses such as Internet weather reports, but does not lend itself well to providing answers to the fine-grained questions we propose here.

Another approach is statistical inference of network internal characteristics based on end-to-end measurements of point-to-point traffic [4, 8, 28, 17, 24, 23, 20]. We adopt this general approach because information is gathered at the appropriate granularity (on a per-connection basis) and at the appropriate time scale to address the questions we study. These approaches can be further classified as *active* approaches, which introduce additional probe traffic into the network, and *passive* approaches, which make inferences only from existing network traffic. The benefit of the former approach is flexibility: one can make measurements at those locations and times which are most valuable; while the benefit of the latter approach is that no additional bandwidth and network resources are consumed solely for the purpose of data collection.

Cutting across other dimensions, one can also classify approaches as either receiver-oriented or sender-oriented, depending on where inferences are made; and multicast-driven or unicast-driven, depending on the model used to transmit probe traffic. Use of multicast traffic is appealing, as losses and delay within the multicast tree induces correlated behavior at receivers, which can streamline inference-making and produce results with higher confidence. Unfortunately, passive probing in an environment where multicast traffic is not present makes such a strategy infeasible.

Table 1 illustrates the above taxonomy with references to studies and projects that fall within each of its different categories. The work we present in this paper is identified as [X]; it is sender-based and is targeted for unicast environments. It works under both passive and active probing assumptions, albeit with different accuracy and convergence properties.

Packet-Pair Probing: One of the essential techniques in our constructions is the use of "packet-pair" techniques, originally used by Keshav [17], and subsequently refined by Carter and Crovella [9] and Paxson [21, 23, 22], to determine bottleneck bandwidth on a network path. In our work, we use a packet pair probe to a pair of *different* receivers to introduce loss and delay correlation, much the same way a multicast packet to these two receivers introduces correlation. A challenge associated with this approach, especially in passive probing, is inter-packet spacing and the time scales over which we can expect correlations to be present. The strategies we employ follow early work by Bolot [4] and recent work by Yajnik, Moon, Kurose and Towsley [28] which study the temporal dependence in unicast and multicast packet losses, respectively.

Estimation of Network Parameters Using End-to-End Measurements: The specific problem of identifying and characterizing bottleneck equivalence classes is motivated in part by recent work on topological inference over *multicast* sessions [6, 7, 10, 24]. By making purely end-to-end observations of packet loss at endpoints of multicast sessions, Ratnasamy and McCanne [24] and Cáceres *et al.* [7] have demonstrated how to make unbiased, maximum likelihood estimation inferences of (a) the multicast tree topology and (b) the packet loss rates on the edges of the tree, respectively. They demonstrate that an observer with access to a complete record of arrivals and lost packets for each destina-

	Network Measurement		End-to-End Measurement			
	Active	**Passive**	**Active**		**Passive**	
Multicast	[18]	[12, 15]		[8, 7, 24, 28]		
Unicast	[16, 9]	[12, 15]	[17, 4, 26], [X]	[23, 28, 26], [X]	[26], [X]	[26]
			Sender	**Receiver**	**Sender**	**Receiver**

Table 1. A Taxonomy of Efforts to Characterize Network Conditions.

tion can make unbiased inferences about the underlying tree from that record. Their work is made possible by the fact that only one copy of a packet traverses any edge of the multicast tree. Thus, if two receivers share a common edge in the multicast tree, and the packet is dropped in the queue prior to traversing that shared edge, *both* downstream receivers will lose that particular packet. With sufficiently many measurements, this correlated behavior makes the inferences above possible.

The work most closely related to ours is that of Rubenstein, Kurose and Towsley[26]. Their work uses end-to-end probing to detect shared points of congestion (POCs). By their definition, a point of congestion is shared when a set of routers are dropping and/or delaying packets from both flows. Their technique for identifying POCs uses Poisson probe traffic to both remote endpoints and cross-correlation measures computed between pairs of packets from these flows. Our techniques differ from theirs by using packet pairs to exploit temporal dependence, our strategies for estimating parameters of the bottleneck queue and our ability to make accurate assessments when multiple congested gateways may exist along a path. In the experimental work section, we also demonstrate the improved accuracy and faster convergence of our approach.

3. Bayesian Probing

In this section, we describe the technique we propose for detecting shared losses. We start by describing the basic definitions, reviewing the overall objective and providing the motivation for the techniques that we propose. Then, we provide the algorithmic and analytical details of the underlying technique, which we illustrate on a one-server, two-clients scenario.

3.1. Basic Definitions and Notation

Consider the set of links used to route unicast traffic between a server and two different clients. Together these links form a tree T rooted at the server, with the clients at the leaves and routers at the internal nodes. The flows of packets sent from the server to each of the two clients share some of T's links and then continue on separate links en route to the different clients. A link L_i is the link whose downstream node is node i as illustrated in Figure 1. We refer to the set of links en route to client A as L_A, the set of links en route to client B as L_B and the set of links that they share as L_S.

Our objective is to define a binary diagnostic test that would identify whether or not significant packet loss is occurring on the set of links shared by client flows. To calibrate the level of loss which warrants a *shared losses* diagnosis, we define the following parameter of our BP approach.

Definition 1 *For a diagnostic procedure, the sensitivity constant c is the maximum loss probability allowed on the shared portion of the paths to multiple receivers while producing a "no shared loss" diagnosis.*

The value of the sensitivity constant c determines the tolerable level of shared losses that the BP technique will allow under a "no shared losses" diagnosis. Thus, in effect, the value of the sensitivity constant c can be used to tune the eagerness of our BP technique to reach a "shared losses" diagnosis. To achieve our objective we introduce two types of probe sequences:

Definition 2 *A* 1-packet probe sequence $S_i(\Delta)$ *is a sequence of packets destined to client i such that any two packets in $S_i(\Delta)$ are separated by at least Δ time units.*

Definition 3 *A* 2-packet probe sequence $S_{i,j}(\Delta, \epsilon)$ *is a sequence of packet-pairs where one packet in each packet-pair is destined to i and the other is destined to j, and where the intra-pair packet spacing is at most ϵ time units and the inter-pair spacing is at least Δ time units.*

The intuition behind the 1-packet probe sequence is to provide a baseline loss rate over each of the two end-to-end paths while the 2-packet probe sequence is used to provide a distinguishing mechanism to measure correlated loss over the shared links. The key insight is that because of their temporal proximity, we expect packets within a packet pair to have a high probability of experiencing a shared fate on the shared links. If the incidence of shared loss on the shared links is high, this leads to an increased probability of witnessing coupled losses within a packet pair. The values of Δ and ϵ in the above definitions of probe sequences are chosen empirically to make

it likely that the probes experience independent and dependent packet loss events, respectively. While we will describe appropriate settings of Δ and ϵ in our experimental section, we will generally require ϵ to be on the order of a millisecond and Δ to be on the order of a second, to achieve high dependence and ensure independence, respectively.

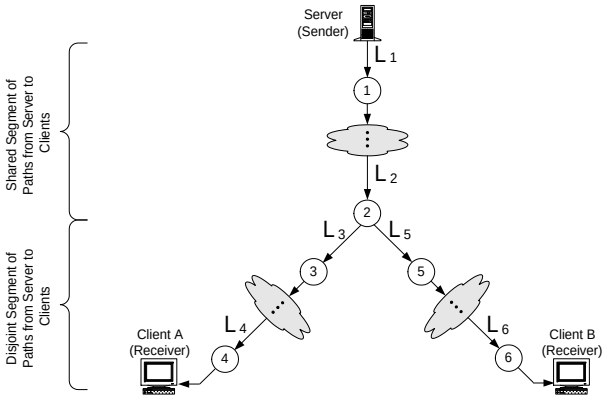

Figure 1. Notation used to describe the topology between a server and two clients

3.2. Diagnosis of Loss Sharing Using BP

We now return to the tree depicted in figure 1 to illustrate the basic premise of our proposed unicast probing technique and its associated analysis. With our packet probe sequences, there are four experimental outcomes which we use in our analysis: successful probes in the 1-probe sequences, successful packet-pair probes in the 2-probe sequence, and unsuccessful probes in the 2-probe sequence in which *both* packets in a pair are lost. The following notation will be useful throughout our analysis. Let g_A and g_B denote the fraction of the 1-packet probes in $S_A(\Delta)$ and $S_B(\Delta)$ respectively which were successfully received. Similarly, let $g_{A,B}$ denote the fraction of the 2-packet probes in $S_{A,B}(\Delta, \epsilon)$ that were successfully received by *both* clients A and B and let $b_{A,B}$ denote the fraction of the 2-packet probes that were lost en route to *both* clients A and B. Note that $g_{A,B} + b_{A,B}$ may be less than 1 due to pairs of probes in which one probe is lost en route to one client while the other probe arrives successfully at the other client.

To establish a relationship between outcomes of probes and network queues, we use the following terminology and notation. Any individual queue can accomodate zero, one, or more than one fixed-size probe packets at any time instant. In general, we define p_i^k be the steady-state probability that the queue at L_i can store exactly k probe packets, and p_i^{k+} be the probability that the queue at L_i can store k or more probe packets. From this definition, p_i^{1+} is the probability that a single probe packet sent over L_i at an instant chosen at random will successfully traverse L_i and p_i^0 is the probability that such a probe will be lost over L_i. With this notation, we can establish the following relationships between probe sequences and queue sizes.

Fact 1 *The quantities g_A and g_B are unbiased estimators for $\prod_{i \in L_A} p_i^{1+}$ and $\prod_{i \in L_B} p_i^{1+}$, respectively.*

Fact 2 *The quantity $g_{A,B}$ is an unbiased estimator for $\prod_{i \in L_S} p_i^{2+} \prod_{i \in (L_A \bigcup L_B) \setminus L_S} p_i^{1+}$.*

Fact 1 holds because a single probe successfully arrives at the destination if and only if each queue en route has availability for at least one probe packet. Likewise, Fact 2 follows since a packet pair successfully arrives at the destination if and only if each shared queue has availability for both packets in the pair and disjoint queues have availability for at least one probe packet.

Establishing a similar relationship for $b_{A,B}$ is considerably more complex by virtue of the number of ways in which both packets in a packet pair may be lost. Either both packets are lost on the shared links; or exactly one packet is lost on the shared links, while the other is lost on the disjoint part of the tree; or both are lost independently on the disjoint links. Letting q_A be a shorthand for the probe loss probability over only the disjoint links to client A, i.e. defining $q_A = 1 - \prod_{i \in L_A \setminus L_S} p_i^{1+}$, and defining q_B similarly, we can enumerate these possibilities to establish the fact that:

Fact 3 *The quantity $b_{A,B}$ is an unbiased estimator for $\left(1 - \prod_{i \in L_S} p_i^{1+}\right) + \left(\prod_{i \in L_S} p_i^{1+} - \prod_{i \in L_S} p_i^{2+}\right)(q_A + q_B) + \prod_{i \in L_S} p_i^{2+} q_A q_B$.*

From these three facts, we can obtain an unbiased estimate for a quantity which occupies a central location in Fact 3 and which we define as follows:

$$X = \prod_{i \in L_S} p_i^{1+} - \prod_{i \in L_S} p_i^{2+}. \quad (1)$$

X can be interpreted as the probability of a packet pair encountering a situation on the shared links in which all shared queues have space for one probe packet, but not all queues have space for two packets. We next prove that we can obtain the following (surprisingly simple) estimate for X:

Lemma 1 *The quantity $g_A + g_B + b_{A,B} - g_{A,B} - 1$ is an unbiased estimator for X.*

Proof: Using Fact 2, we relate $g_{A,B}$ to X:

$E[g_{A,B}] = \prod_{i \in L_S} p_i^{2+} \prod_{i \in (L_A \bigcup L_B) \setminus L_S} p_i^{1+}$

$$= \left(\prod_{i\in L_S} p_i^{1+} - X\right) \prod_{i\in L_A\setminus L_S} p_i^{1+} \prod_{i\in L_B\setminus L_S} p_i^{1+}$$
$$= \prod_{i\in L_A\bigcup L_B} p_i^{1+} - \left(X \prod_{i\in L_A\setminus L_S} p_i^{1+} \prod_{i\in L_B\setminus L_S} p_i^{1+}\right)$$
$$= \prod_{i\in L_A\bigcup L_B} p_i^{1+} - X(1-q_A)(1-q_B)$$

Combining this equation with Facts 1 and 3 and by the linearity of expectation, we can write:

$$E[b_{A,B} + g_A + g_B - g_{A,B} - 1]$$
$$= \prod_{i\in L_A} p_i^{1+} + \prod_{i\in L_B} p_i^{1+} - \prod_{i\in L_S} p_i^{1+} + X(q_A+q_B) + \prod_{i\in L_S} p_i^{2+} q_A q_B - \prod_{i\in L_A\bigcup L_B} p_i^{1+} + X(1-q_A)(1-q_B)$$
$$= X + q_A q_B X + \prod_{i\in L_A} p_i^{1+} + \prod_{i\in L_B} p_i^{1+} - \prod_{i\in L_S} p_i^{1+} - \prod_{i\in L_A\bigcup L_B} p_i^{1+} + \prod_{i\in L_S} p_i^{2+} q_A q_B$$

It now suffices to demonstrate that the quantity $q_A q_B X$ cancels with the remaining terms. By applying the definitions we have:

$$q_A q_B X = q_A q_B \left(\prod_{i\in L_S} p_i^{1+} - \prod_{i\in L_S} p_i^{2+}\right)$$
$$= \left(1 - \prod_{i\in L_A\setminus L_S} p_i^{1+}\right)\left(1 - \prod_{i\in L_B\setminus L_S} p_i^{1+}\right)\left(\prod_{i\in L_S} p_i^{1+}\right) - \prod_{i\in L_S} p_i^{2+} q_A q_B$$
$$= (1 - \prod_{i\in L_A\setminus L_S} p_i^{1+} - \prod_{i\in L_B\setminus L_S} p_i^{1+} + \prod_{i\in (L_A\bigcup L_B)\setminus L_S} p_i^{1+}) \prod_{i\in L_S} p_i^{1+} - \prod_{i\in L_S} p_i^{2+} q_A q_B$$
$$= \prod_{i\in L_S} p_i^{1+} - \prod_{i\in L_A} p_i^{1+} - \prod_{i\in L_B} p_i^{1+} + \prod_{i\in L_A\bigcup L_B} p_i^{1+} - \prod_{i\in L_S} p_i^{2+} q_A q_B$$

Therefore the desired cancellation does take place, yielding the result. ■

3.3. The X Factor

We now motivate the reason for which obtaining an unbiased estimate of the value of X is valuable. As we mentioned in the preceding section, an estimate of X is an estimate of the probability that one of the queues on the shared links has room for a single probe packet. As the following example clearly demonstrates (and as one might imagine), the magnitude of this value, which is an analogue of p^1, tends to be highly correlated with the magnitude of packet loss on that link.

Figure 2 (left) shows how the values of p^0, p^1 and p^{2+} on a single link interact in an $M/M/1/K$ queueing system with queue size $K = 20$ as a function of the traffic load ρ. Under light load (ρ not much larger than 1), the values of p^0 and p^1 are almost identical. Under heavy load (ρ much larger than 1), the value of p^0 becomes larger than the value of p^1. The experiment depicted in Figure 2 (right) demonstrates similar phenomena in a bursty traffic model which we describe in detail in Section 4. The figure suggests that the value of p^1 increases in tandem with the value of p^0 as the background traffic rate increases. This trend is a key to our proposed technique and points to the value of an unbiased estimate for X.

To summarize, we can efficiently compute a running estimate of X using the 1-packet and 2-packet probe sequences sent from the server to the two clients. If $X > c$ (for some empirically-determined sensitivity constant c) we conclude that there are "significant" losses on the shared part of the path between the server and the clients. Otherwise we conclude that losses are primarily due to packet losses on the disjoint part of the path between the server and the clients.

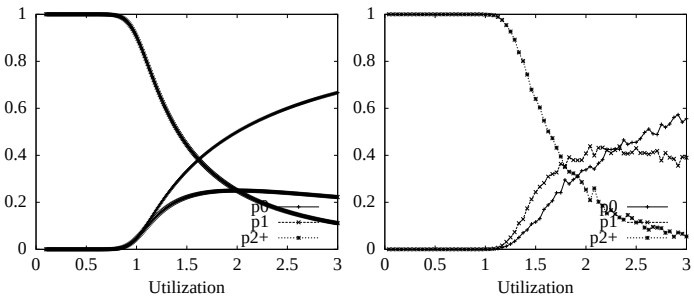

Figure 2. Values of P_0, P_1 and P_{2+} when $K = 20$ for different values of ρ: M/M/1/K Analysis (left) and ns simulation results with 64 Pareto ON/OFF UDP flows (right).

3.4. Basic Assumptions

A basic premise of our work is that while we assume the loss rate on all links in our topology may have substantial short-term variability (as is to be expected with self-similar background traffic), the mean packet loss rate on each link is stationary over longer time scales. This stationarity requirement is needed to allow a diagnostic procedure to converge. Thus, stationarity is required only over time scales that are comparable to the time it takes the diagnostic procedure to converge. In the next section we show that the BP technique possesses superior convergence, making it quite effective even when stationarity can only be assumed for short intervals (on the order of few seconds).

In the analysis presented above, we have made the following additional assumptions which we enumerate and discuss here:

1. Losses on the links occur only due to queue overflows.
2. $\forall i, j$: Losses on link L_i are independent from losses on link L_j.
3. A reliable feedback mechanism enables the sender to determine with certainty whether a given probe packet was lost.
4. The temporal constraints imposed on probe sequences (whether 1-packet or 2-packet probe sequences) are preserved throughout the journey of the probes from sender to receivers.

Assumption 1 reflects the current DropTail behavior present in most Internet routers today. We consider the negative consequences of RED gateways on our technique in the experimental section. Assumption 2 allows us to ignore any spatial correlation between link losses, and thus ignore any additional correlation terms. Assumption 3 enables us to assume that the server is able to accurately identify the outcome of the probing process, i.e. which packets of a 1-packet or 2-packet probe sequences were lost.

Assumption 4 is our most significant assumption, since it ensures that the individual packets within each packet-pair of a 2-packet probe sequence $S_{A,B}(\Delta, \epsilon)$ are separated by at most ϵ time units on *all* traversed links. Moreover, we must be assured that ϵ is sufficiently small that two packets of a packet-pair are close enough to each other on all traversed links to enable an accurate sampling of the state of a queue at the time the 2-packet probe reaches that queue. In particular, we need to use p_i^{2+} as the probability that the two packets of a packet-pair have traversed link i. Ideally, we would desire that the two packets reach the queue with an inter-arrival time $\epsilon = 0$. If the packets in a pair become substantially separated from one another in flight, our estimates $g_{A,B}$ and $b_{A,B}$ will be biased. We have studied the effects of $\epsilon > 0$ on the performance of our BP technique. Our findings (presented in the next section) confirm that the bias introduced by small amounts of separation and/or long paths is not excessive.[1]

4. Experimental Results

In this section we present results of extensive simulations that (1) compare our Bayesian Probing (BP) technique to the Markovian Probing (MP) technique proposed and evaluated in [26], and (2) establish the robustness of the BP technique to various parameters and conditions.

4.1. Techniques Evaluated

Bayesian Probing Technique: Recall from our presentation in section 3, the BP technique requires the specification of the Δ and ϵ parameters of the temporal constraints imposed on 1-packet and 2-packet probe sequences.

In the experiments we present in this section, probes were sent at a mean rate R. However, to alleviate synchronization effects, we imposed additive random noise on the interpacket spacing so that it was uniformly distributed over the range $[\frac{1}{R} - 5ms, \frac{1}{R} + 5ms]$, thus $\Delta = \frac{1}{R} - 5ms$. In our experiments, we set the value of ϵ to 0; that is packets within a packet-pair were sent back-to-back, with no time separation. Also, to normalize the losses on the shared links experienced by both receivers, the 2-packet probes in $S_{A,B}(\Delta, \epsilon)$ alternate between the two possible packet orderings.

Another parameter of our BP technique is the value of the sensitivity constant (c). Recall that the value of the sensitivity constant c determines the level of shared losses that the BP technique will tolerate while indicating a "no loss sharing" diagnosis. In our experiments, the value of c was fixed at 0.04. This value was chosen empirically based on experiments discussed later in this section.

Markovian Probing Technique: The MP technique described in [26] relies on the use of two Poisson processes for sending probe sequences f_1 and f_2 from the sender to the two receivers. To detect shared losses the MP technique depends on the calculation of the *Auto-Correlation* and the *Cross-Correlation* functions. The Auto-Correlation function (M_{a1}) is the conditional probability that a packet from f_1 is lost, given that the previous packet from f_1 is lost. The Cross-Correlation function (M_{x12}) is the conditional probability that a packet from f_1 is lost, given that the preceding packet from f_2 was lost. Given M_{a1} and M_{x12}, the MP technique described in [26] suggests the following test for identifying shared losses.

MP1 test: f_1 and f_2 are diagnosed to having shared losses if $M_{x12} > M_{a1}$; they are diagnosed as having no shared losses otherwise.

It is important to note that the MP test (above) could be reformulated by reversing the roles of the probes sent on the f_1 and f_2 paths.

MP2 test: f_1 and f_2 are diagnosed to having shared losses if $M_{x21} > M_{a2}$; they are diagnosed as having no shared losses otherwise.

In our experiments, we noted that the MP1 and MP2 tests yielded similar diagnosis when losses were symmetric along the non-shared links (or equivalently when losses are either all shared or all on the independent links—a central assumption of the MP technique described in [26]). However, the MP1 and MP2 tests yielded quite different diagnosis when this condition seized to hold true. Since we were interested in loosening this assumption (by allowing losses on both the shared and idependent portions of the paths), we combined the above tests into the following alternative test.

MP test:* f_1 and f_2 are diagnosed to having shared losses if $M_{x21} > M_{a2}$ **OR** $M_{x12} > M_{a1}$; they are diagnosed as having no shared losses otherwise.[2]

[1] We have measured the separation between probe packets for paths consisting of a large number of hops. The results consistently pointed to the validity of our temporal constraint assumption above and, consequently, the robustness of our BP technique.

[2] In private communications with the first author of [26], we also considered a conjunctive test for the identification of shared losses as opposed to the disjunctive test we propose here. Our

Our experimental results, which we present later in this section, show that the MP* test improved the accuracy of the MP technique significantly. Thus in the remainder of this paper, and unless otherwise noted, we will use the MP* test as the "default" test for the Markovian Probing (MP) technique.

4.2. Experimental Setup

We used the Network Simulator (ns) [19] to simulate the topology illustrated in Figure 3. This topology represents a server and two clients. The shared portion of the paths between the server and the two clients is modeled by a single link (L1), whereas the disjoint portions of the paths between the server and the two clients are modeled by links (L2) and (L3), respectively. Both techniques were simulated from the server side by implementing a new *ns* "agent" that sends 200-byte probe packets to the receivers and waits for an acknowledgment for each probe sent. Probes are annotated with sequence numbers. The agent uses the absence of a probe acknowledgment as an indication of the probe's loss on the way to its destination. Also, the agent keeps some statistics about the probe losses and based on these statistics esimates whether there are shared losses or not by using either the BP, MP1 or the MP* techniques described above.

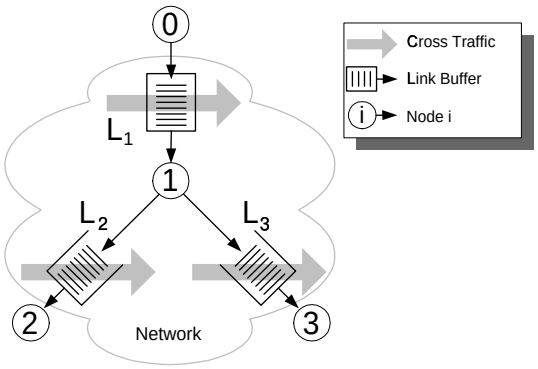

Figure 3. Topology used in our experiments

Baseline Model: Each one of the three links in Figure 3 is modeled by a single DropTail queue. The link delays were all set to 40ms and the link buffer sizes were all set to 20 packets. Each of these links was subjected to background traffic resulting from a set of Pareto ON/OFF UDP sources with a constant bit rate of 36Kbps during the ON times with a packet size of 200 bytes. The average ON and OFF times were set to 2 seconds and 1 second, respectively. The Pareto shape parameter (α) was set to 1.2. After a "warm-up" period of 10 seconds, the probing processes (and associated diagnostic processes) are started.

experimental evaluation indicated that a disjunctive test yielded better results, and is thus adopted in this paper.

To represent the various levels of congestion that any of these links may exhibit, we have chosen three sets of parameters that result in "High", "Mild", and "Low" levels of congestion. The baseline parameter settings for these congestion levels (and the resulting loss rates) are tabulated in Table 2.

Parameter Setting	Congestion Level		
	High	Mild	Low
Link Bandwidth	1Mb/sec	1Mb/sec	100Mb/sec
Background Flows	60	56	8
Observed Loss Rate	7-15%	$< 7\%$	$< 0.1\%$

Table 2. Settings used (and resulting loss rates) for the 3 congestion levels considered

Basic Test Cases: In order to evaluate the diagnostic abilities of the above techniques, we define four possible scenarios, featuring different levels of congestion along the shared and disjoint portions of the paths between the server and its clients. Table 3 enumerates these four scenarios.[3]

Scenario #	Congestion			Sharing Condition		Correct Diagnosis
	L1	L2	L3	Losses	B'neck	
(I)	H	L	L	Yes	Yes	Yes
(II)	H	M	L	Yes	Yes	Yes
(III)	M	H	L	Yes	No	Yes
(IV)	L	M	H	No	No	No

Table 3. Scenarios considered in this paper

Scenario (I) represents a situation in which a highly congested link exists on the shared portion of the path to the two clients, and no congestion exists on the disjoint portion of the paths. Scenario (IV) represents a situation in which losses are only possible on the disjoint portion of the path to the two clients. Scenarios (I) and (IV) represent the "litmus test" cases that must be diagnosed correctly by any technique that aims at identifying shared losses (or lack thereof).

Scenario (II) represents a situation in which a highly congested link exists on the shared portion of the path to the two clients, and a lesser congested link exists on one of the disjoint portion of the paths. Scenario (III) represents a situation in which a highly congested link exists on one of the disjoint portions of the paths to the clients, and a lesser congested link exists on the shared portion of the path to the two clients.

It is important to note that scenarios (II) and (III) violate one of the assumptions of the Markovian Probing technique of Rubenstein, Kurose, and Towsley—namely, that losses on a given path are the result of exactly one

[3] Results from additional scenarios we have tested were consistent with the results we present for the four scenarios in Table 3.

congested link on that path. We have included these scenarios to highlight the robustness of our Bayesian probing technique—in particular its ability to converge to a correct diagnosis when losses on a given path are the result of multiple congestions along that path. Notice that the existence of multiple congested gateways on a single path over an extended period of time is quite possible (due in part to the documented scaling phenomena of network traffic) [11, 25].

4.3. Performance Metrics

We consider three main metrics: (1) Accuracy, (2) Settling time, and (3) Convergence Ratio. We define each of these metrics next. In each of the definitions below, we assume that the diagnosis process starts at time $t = 0$ and that $1 \leq i \leq N$ refers to the diagnosis experiment under consideration.

Definition 4 *The accuracy of a diagnostic strategy at time t is defined as the probability that the diagnostic strategy will yield a correct diagnosis at time t.*

To measure the accuracy of a diagnostic strategy at time t, we measure the percentage of simulation experiments in which a correct diagnosis was reached at time t.

Definition 5 *For an experiment i, the settling time $S_i(t)$ of a diagnostic strategy is defined as the latest time $t' \leq t$ at which a wrong (or inconclusive) diagnosis was made for that experiment. The mean settling time $S(t)$ of a diagnostic strategy is defined as the expected value of the settling time at time t.*

The above definiton implies that the (mean) settling time is a monotonically non-decreasing function of t. To measure the mean settling time $S(t)$, we averaged the settling time for all simulation experiments at time t.

$$S(t) = \frac{\sum_{i=1}^{N} S_i(t)}{N}$$

In the remainder of this paper, we use settling time to imply mean settling time. This settling time as a function of t can be used to characterize the convergence of a diagnostic strategy (or lack thereof). We do so next.

Definition 6 *For an experiment i, the convergence ratio $C_i(t)$ of a diagnostic strategy is defined as the ratio between the time ellapsed since settling and t—namely*

$$C_i(t) = \frac{t - S_i(t)}{t} = 1 - \frac{S_i(t)}{t}$$

The mean convergence ratio of a diagnostic strategy $C(t)$ is defined as the expected value of the convergence ratio at time t.

One can easily show that a random diagnosis strategy yields a convergence ratio that approaches 0 as t increases. Thus, one can view the convergence ratio as a measure of "how much better" a diagnostic strategy is compared to a random diagnosis. The closer the convergence ratio is to zero, the slower the convergence; and, the closer the convergence ratio is to one, the faster the convergence.

The value of the convergence ratio for large enough values of t can be used to characterize the likelihood of convergence. In particular, if the convergence ratio approaches a constant r $(0 \leq r \leq 1)$ as t approaches infinity, then it follows that the probability that the diagnostic strategy will converge in an infinitely long experiment is r.

In our presentation below, and unless otherwise specified, we use the term "convergence ratio" to mean the convergence ratio at time $t = T_{max}$, where $T_{max} = 300$ seconds is the simulation time of our experiments.

4.4. Baseline Results for BP versus MP Techniques

Accuracy: Figure 4 (left) shows the accuracy achieved over time for the four basic scenarios we considered. Clearly, our Bayesian Probing (BP) approach yields a consistently higher accuracy than that achieved by the Markovian Probing (MP) approach.

For scenarios (I), (II), and (III) in which shared losses exist, our BP approach converges to 100% accuracy within a very short period of time. This is in sharp contrast to the MP approach, which oscillates considerably around the 75-90% accuracy range under scenario (I), around the 60-80% accuracy range under scenario (II), and around the 70-75% accuracy range under scenario (III).

For scenario (IV) in which there are no shared losses, our BP approach again converges rather quickly to 100% accuracy. Initially, the MP approach performs quite poorly (actually dropping to a 20-30% accuracy as late as 100 seconds into our experiments). However, over time, the MP approach does converge to almost 100% accuracy as well.

Convergence Characteristics: Figures 4 (center) and 4 (right) show the settling time and convergence ratio for the BP and MP approaches. Figure 4 (center) indicates that the settling time of our BP approach is decidedly lower than that of the MP approach under *all* test scenarios. Moreover, in three out of the four test scenarios—namely (I), (II), and (III)—the settling time function of the MP approach does not seem to level off, whereas the settling time function of the BP technique levels off in all four scenarios. The superior convergence properties of our BP approach are further confirmed in Figure 4 (right).

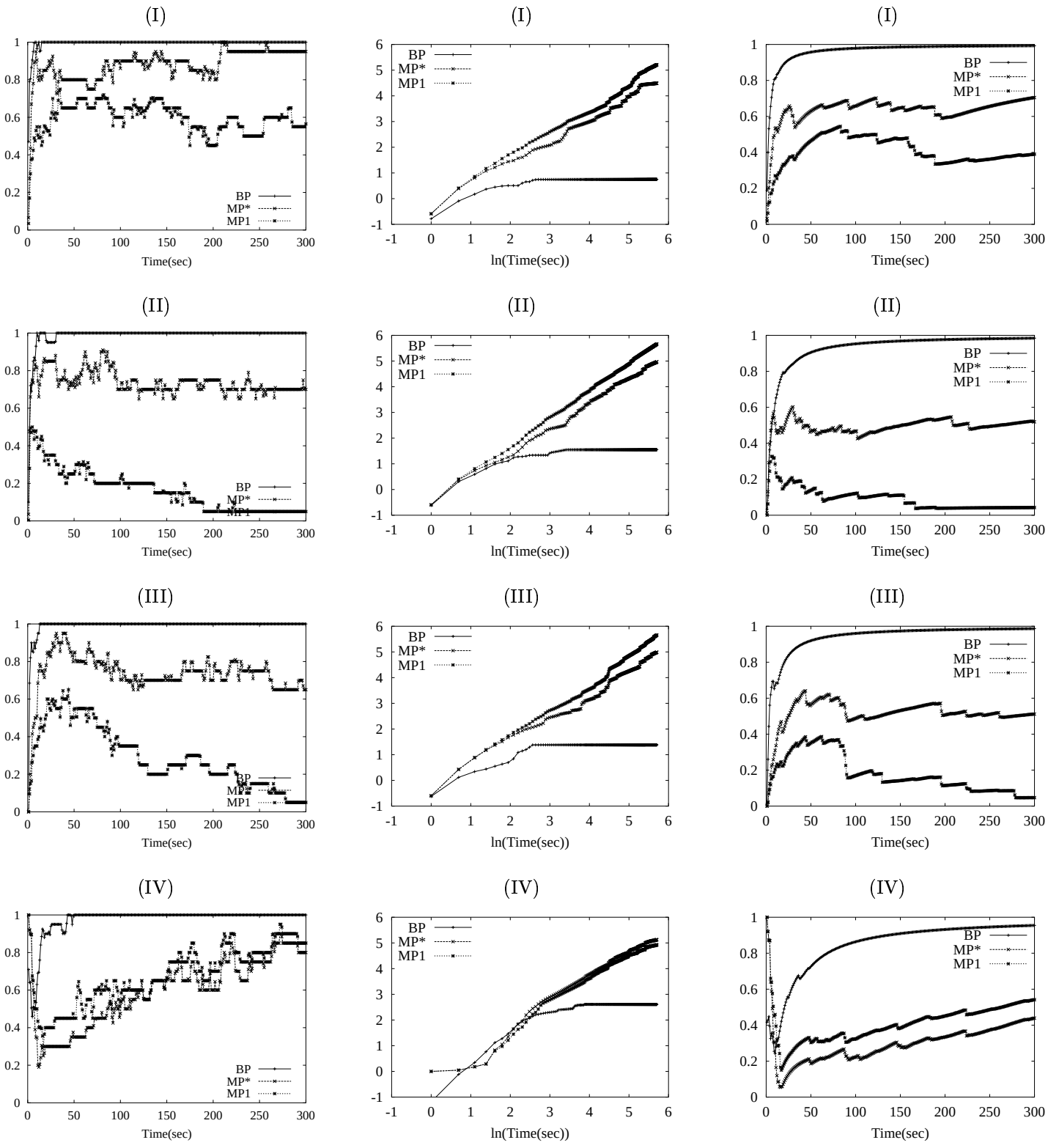

Figure 4. Accuracy (left), settling time on log-log plot (center), and convergence ratio (right) of BP vs MP for the basic test scenarios under the baseline model

4.5. Robustness of BP Technique

In the remainder of this section we summarize results of experiments we have conducted to evaluate the robustness of our BP technique to a host of parameters that may impact its performance characteristics. Readers interested in details of these experiments and our findings are refered to [14].

Effect of the BP Sensitivity Constant: As we noted earlier, the value of the sensitivity constant (c) used throughout our experiments was 0.04. We used this value after comparing the effect of c on the accuracy and the convergence ratio for the four baseline scenarios. This comparison is shown in figure 5.

As these figures indicate, setting c to 0.04 was a compromise between the accuracy and convergence ratios of scenarios (III) and (IV). Reducing the value of c leads to low performance for scenario (IV) (i.e. when losses are independent) since the BP approach tends to identify more "false positives". On the other hand, increasing the value of c leads to lower performance for scenario (III) (i.e. when shared losses exist but are not dominant for one of the receivers) since the BP approach's sensitivity to shared losses is reduced, resulting in a misdiagnosis.

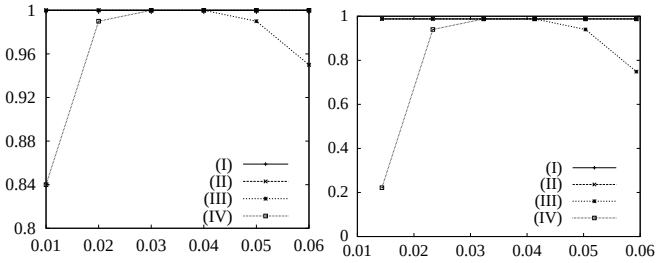

Figure 5. Effect of c on accuracy (left) and convergence ratio (right) for the four scenarios.

Effect of Temporal Separation ϵ: As we discussed in Section 3, an important assumption of our BP technique is that the separation (in time) between packet pairs in a 2-packet probe sequence (i.e., ϵ) is sufficiently small so as to keep the two packets of a packet-pair close enough to each other on all traversed links. This enables an accurate sampling of the state of a queue at the time the 2-packet probe reaches that queue.

Figure 6 shows the accuracy of our BP technique under the four baseline models and for various values of the BP sensitivity constant c. In general, our experiments show that the BP technique's accuracy and convergence are quite robust for values of ϵ less than 1 msec. Notice that a separation larger than 1 msec is unlikely[4] even when packet-pairs traverse long paths.

[4]If the two packets in a packet-pair are sent back-to-back, then it would be necessary for 12.5MB of cross traffic to interveen between these two packets on a 100Mbps link to achieve a separation of 1 msec.

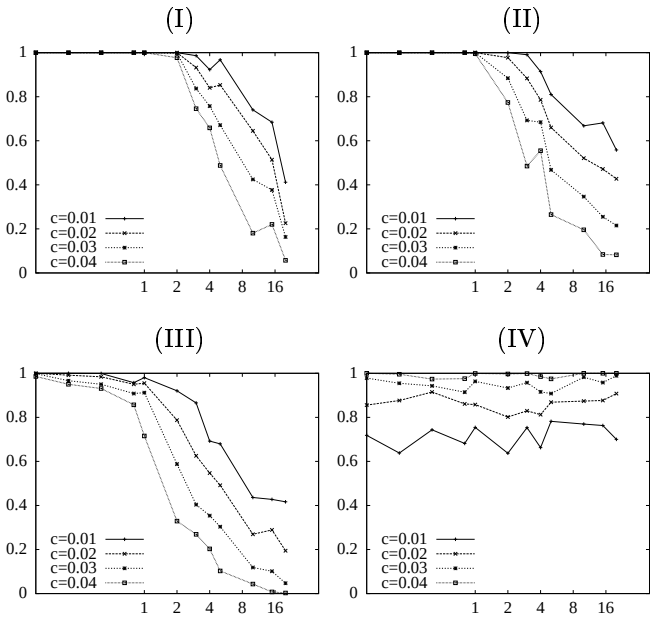

Figure 6. Effect of ϵ (in msec on x-axis) on accuracy of the BP technique.

The results in Figure 6 indicate that as the separation between packet-pairs in 2-packet probe sequences increases (i.e., as ϵ grows larger), the BP technique's ability to diagnose shared losses, i.e. under scenarios (I)-(III), decreases. Under scenario (IV) The accuracy and convergence of the BP technique are unaffected by ϵ. This is expected since 2-packet probe sequences are instrumental only for the detection of shared losses (which are not present under scenario (IV).

An interesting observation from the results shown in Figure 6 is the trade-off between the sensitivity constant (c) and the temporal separation between packet-pairs (ϵ). When ϵ is small (e.g. $\epsilon < 1$ msec), a larger value of c yields better accuracy and convergence for all scenarios. However, as ϵ grows larger (i.e. as the effectiveness of 2-packet probe sequences decreases), a decrease in c lead to better accuracy and convergence for scenarios (I), (II), and (III)—i.e. when a diagnosis of "shared losses" is warranted—but lead to a deterioration in both accuracy and convergence for scenario (IV)—i.e. when a diagnosis of "independent losses" is warranted.

Thus, if the value of ϵ cannot be guaranteed to remain within the $[0, 1msec]$ range, then the value of c should be chosen based on which misdiagnosis is safer—namely, misdiagnosing shared losses as independent, or misdiagnosing independent losses as shared.

Effect of Traffic Burstiness: In our baseline experiments, traffic burstiness was moderate with the ON/OFF times of the constant-bit-rate UDP background flows set to a Pareto distribution with $\alpha = 1.2$. Traffic burstiness may negatively impact the accuracy and convergence of a diagnostic strategy, since it may reduce (or increase) loss correlations.

Table 4 (left) shows the accuracy and the convergence ratio of the BP technique for the four baseline scenarios under various values of α (i.e. under different levels of background traffic burstiness). These figures show that the BP technique's accuracy and convergence are unaffected by traffic burstiness, except for very small values of α (namely $\alpha = 1.001$) under scenario (IV).[5]

Effect of Probing Rate: Another important parameter of the BP technique is the probing rate R. A higher probing rate is desirable because it implies a "faster" diagnosis (i.e. a shorter settling time). However, a higher probing rate results in smaller time separation between probes, and thus threatens to violate the assumption of probe independence, which is central in our derivation of the BP diagnostic test. Finally, in the context of active probes, a higher probing rate implies more probe traffic, which is not desirable.

Table 4 (right) shows the accuracy and the mean settling time of the BP technique for the four baseline scenarios under various probing rates (recall that the probing rate used in our baseline experiments was 15 probes per second). These figures show that BP's accuracy is quite robust (even for the highest rates we attempted). The advantage of higher probing is evident in the overall trend of lower settling times when probing rates are increased, especially for positive loss sharing diagnoses. For example, by quintupling the probing rate from 5 to 25, the mean settling time under scenario (I) is reduced by a factor of 5 from 12.34 to 2.29 seconds.

Effect of Queuing Discipline: The BP technique relies on an important property of the queuing discipline used on link buffers. Namely, it relies on the high probability of back-to-back losses of packet-pairs in a 2-packet probe when the link buffer is full (i.e. congested). This property is likely to hold for a DropTail queueing discipline, which is the discipline we have assumed for link buffer management in our experiments so far.

Figure 7 shows the accuracy and convergence of our BP technique when a Random Early Detection (RED) [13] queuing discipline is used. In these experiments, the parameters of RED that we used were: `minthresh`=5, `maxthresh`=15, and `maxp`=0.1.

The figure shows a definite deterioriation in performance under loss sharing scenarios, i.e. scenarios (I), (II), and (III). This is expected since RED tends to reduce loss correlation and thus is likely to adversely affect the effectiveness of 2-packet probes (since losses of the two packets in a packet pair will tend to be less well correlated). This results in a tendency of the BP technique to be biased towards making a "no loss sharing" diagnosis. Figure 7 shows that despite RED's negative impact, the BP technique was still robust enough to yield acceptable accuracy and convergence for all scenarios, except scenario (III), for which BP's performance was almost undinstinguishable from a random diagnosis.

It is important to note that the MP technique we evaluated earlier in this section suffers from the same disadvantage when a RED queuing discipline is deployed [26], which leaves the problem of robust shared loss identification in the presence of RED gateways an important open problem to the best of our knowledge.

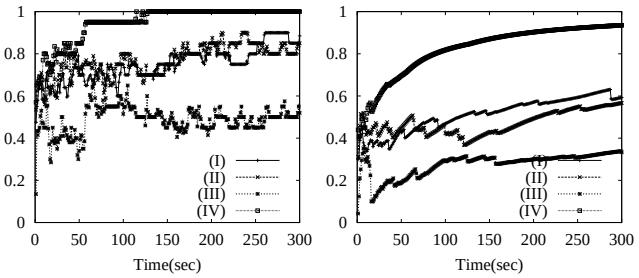

Figure 7. Effect of RED Queueing on the accuracy (left) and convergence ratio (right).

5. Conclusion

In this paper, we have presented a robust technique for determining whether a pair of connections emanating from the same node experience shared losses. We presented results of extensive simulations that confirm the robustness of our methodology and its effectiveness as compared with the recently proposed memoryless probing technique of Rubenstein, Kurose, and Towsley [26] which we termed "Markovian Probing". Specifically, our technique converges very quickly to a correct diagnosis under a wide variety of network conditions.

The work presented in this paper is part of a larger effort by the Mass Group at Boston University [27], which aims to harness the interplay between on-line network diagnosis and control for massively accessed Internet servers. For such servers—in which thousands of connections (or flows) may be managed concurrently—it is desirable to diagnose network conditions at a wider variety of resolutions than were considered in this paper. We are looking into a number of possibilities, including the identification of long-term "shared bottlenecks" and "loss topology" between a sender (mass server) and a possibly large number of receivers (clients).

[5] Note that in this experiment, our packet-pair probes were sent "back-to-back" (i.e. $\epsilon = 0$). The impact of traffic burstiness is likely to be more pronounced when $\epsilon > 0$.

	Accuracy for $150 \leq t \leq 300$				Convergence @ $t = 300$ sec			
	$\rightarrow 1$	1.10	1.20	1.80	$\rightarrow 1$	1.10	1.20	1.80
I	1.00	1.00	1.00	1.00	0.99	0.99	0.98	0.98
II	1.00	1.00	0.99	1.00	0.99	0.98	0.94	0.98
III	1.00	0.99	1.00	1.00	0.61	0.99	0.98	0.94
IV	0.71	0.98	1.00	0.97	0.50	0.91	0.95	0.85

BP performance for various values of α

Accuracy for $150 \leq t \leq 300$				Settling Time @ $t = 300$ sec			
5	10	20	25	5	10	20	25
1.00	1.00	1.00	1.00	12.35	8.45	2.63	2.29
1.00	1.00	1.00	1.00	6.75	8.02	4.71	2.89
0.98	0.99	1.00	1.00	25.46	57.54	8.44	4.52
0.99	1.00	1.00	1.00	28.99	37.17	13.59	14.85

BP performance for various values of R

Table 4. Effect of traffic burstiness (left) and probing rate (right) on BP performance

References

[1] M. Allman and V. Paxson. On Estimating End-to-End Network Path Properties. In *Proceedings of ACM SIGCOMM '99*, 1999.

[2] H. Balakrishnan, H. Rahul, and S. Seshan. An Integrated Congestion Management Architecture For Internet Hosts. In *SIGCOMM '99*, Cambridge, MA, September 1999.

[3] A. Bestavros and O. Hartmann. Aggregating Congestion Information Over Sequences of TCP Connections. Technical Report BUCS-TR-98-001, Boston Univ, CS Dept, January 1998.

[4] J. C. Bolot. End-to-end Packet Delay and Loss Behavior in the Internet. In *SIGCOMM '93*, pages 289–298, September 1993.

[5] J. W. Byers, M. Luby, and M. Mitzenmacher. Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads. In *Proceedings of IEEE INFOCOM '99*, pages 275–83, March 1999.

[6] R. Cáceres, N. Duffield, S. Moon, and D. Towsley. Inferring Link-level Performance from End-to-End Multicast Measurements. Internal report. Available at ftp://gaia.cs.umass.edu/pub/Caceres99gi99.ps.Z.

[7] R. Cáceres, N. G. Duffield, J. Horowitz, D. Towsley, and T. Bu. Multicast Based Inference of Network-Internal Characteristics: Accuracy of Packet-Loss Estimation. In *INFOCOM '99*.

[8] R. Cáceres, N. G. Duffield, S. B. Moon, and D. Towsley. Inference of Internal Loss Rates in the MBone. In *IEEE Global Internet (Globecom)*, Rio de Janeiro, Brazil, 1999.

[9] R. L. Carter and M. E. Crovella. Measuring bottleneck link speed in packet switched networks. In *PERFORMANCE '96, the International Conference on Performance Theory, Measurement and Evaluation of Computer and Communication Systems*, October 1996.

[10] N. Duffield, V. Paxson, and D. Towsley. MINC: Multicast-based Inference of Network-Internal Characteristics. http://www-net.cs.umass.edu/minc/.

[11] A. Feldmann, A. C. Gilbert, and W. Willinger. Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic. In *Proceedings of SIGCOMM '98*, pages 42–55, October 1998.

[12] Felix: Independent Monitoring for Network Survivability. http://ftp.bellcore.com/pub/mwg/felix/index.html.

[13] S. Floyd and V. Jacobson. Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4), August 1993.

[14] K. Harfoush, A. Bestavros, and J. Byers. Robust Identification of Shared Losses Using End-to-End Unicast Probes. Technical Report BUCS-TR-2000-013, Boston University, Computer Science Department, May 2000.

[15] IPMA : Internet Performance Measurement and Analysis. http://www.merit.edu/ipma.

[16] V. Jacobson. `Pathchar`: A Tool to Infer Characteristics of Internet Paths. ftp://ftp.ee.lbl.gov/pathchar.

[17] S. Keshav. *Congestion Control in Computer Networks.* PhD thesis, University of California at Berkeley, September 1991.

[18] Mtrace: Tracing multicast path between a source and a receiver. ftp://ftp.parc.xerox.com/pub/netsearch/ipmulti.

[19] `ns`: Network Simulator. http://www-mash.cs.berkeley.edu/ns/ns.html.

[20] V. Padmanabhan. Optimizing Data Dissemination and Transport in the Internet. Presented at the BU/NSF Workshop on Internet Measurement, Instrumentation and Characterization, September 1999.

[21] V. Paxson. End-to-end Routing Behavior in the Internet. In *SIGCOMM '96*, Stanford, CA, Aug 1996.

[22] V. Paxson. End-to-end Internet Packet Dynamics. In *SIGCOMM*, 1997.

[23] V. Paxson. *Measurements and Analysis of End-to-end Internet Dynamics.* PhD thesis, U.C. Berkeley and Lawrence Berkeley Laboratory, 1997.

[24] S. Ratnasamy and S. McCanne. Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements. In *Proceedings of IEEE INFOCOM '99*, pages 353–60, March 1999.

[25] R. H. Riedi and J. L. Vehel. Multifractal properties of TCP traffic: a numerical study. Technical Report 3129, INRIA, March 1997.

[26] D. Rubenstein, J. Kurose, and D. Towsley. Detecting Shared Congestion of Flows Via End-to-end Measurement. In *ACM SIGMETRICS '00*, Santa Clara, CA, June 2000.

[27] The Boston University MASS Project. Diagnosis and Control of Network Variability by MASS Servers. http://www.cs.bu.edu/groups/mass, 2000.

[28] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modelling of the temporal dependence in packet loss. In *Proceedings of IEEE INFOCOM '99*, pages 345–52, March 1999.