Model-Based Identification of Dominant Congested Links

Wei Wei, *Member, IEEE*, Bing Wang, *Member, IEEE*, Don Towsley, *Fellow, IEEE, ACM*, and Jim Kurose, *Fellow, IEEE*

***Abstract*—In this paper, we propose a model-based approach that uses periodic end–end probes to identify whether a "dominant congested link" exists along an end–end path. Informally, a dominant congested link refers to a link that incurs the most losses and significant queuing delays along the path. We begin by providing a formal yet intuitive definition of dominant congested link and present two simple hypothesis tests to identify whether such a link exists. We then present a novel model-based approach for dominant congested link identification that is based on interpreting probe loss as an *unobserved* (virtual) delay. We develop parameter inference algorithms for hidden Markov model (HMM) and Markov model with a hidden dimension (MMHD) to infer this virtual delay. Our validation using *ns* simulation and Internet experiments demonstrate that this approach can correctly identify a dominant congested link with only a small amount of probe data. We further provide an upper bound on the maximum queuing delay of the dominant congested link once we identify that such a link exists.**

***Index Terms*—Bottleneck link, dominant congested link, end–end inference, hidden Markov model (HMM), Markov model with a hidden dimension (MMHD), network inference, network management, path characteristics.**

I. Introduction

MEASUREMENT and inference of end–end path characteristics have attracted a tremendous amount of attention in recent years. Properties such as the delay and loss characteristics of an end–end path [29], the minimum capacity and available bandwidth of a path [20], [25], [7], [15], [13], and the stationarity of the network [41] have been investigated. These efforts have improved our understanding of the Internet. They have also proved valuable in helping to manage and diagnose heterogeneous and complex networks.

Manuscript received July 27, 2009; revised March 02, 2010 and July 27, 2010; accepted July 28, 2010; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Dovrolis. Date of publication September 07, 2010; date of current version April 15, 2011. This work was supported in part by the National Science Foundation under Grants ANI-0085848, ANI-9980552, ANI-9973092, ANI-9977635, EIA-0080119, and EIA-0087945; CAREER Award 0746841; and a by subcontract with the University of Florida under Grant UF-EIES-0205003-UMA. An earlier version of this paper appeared in the Proceedings of the 3rd ACM SIGCOMM Internet Measurement Conference (IMC), Miami Beach, FL, October 27–29, 2003.

W. Wei, D. Towsley, and J. Kurose are with the Computer Science Department, University of Massachusetts Amherst, Amherst, MA 01003 USA (e-mail: weiwei@cs.umass.edu; towsley@cs.umass.edu; kurose@cs.umass.edu).

B. Wang is with the Computer Science and Engineering Department, University of Connecticut, Storrs, CT 06269 USA (e-mail: bing@engr.uconn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2010.2068058

In this paper, we study a specific end–end path characteristic, namely whether a *dominant congested link* exists along an end–end path. Informally, a dominant congested link is one that produces most of the losses and significant queuing delays on an end–end path. A formal definition is deferred to a later section. We avoid using the term "bottleneck link" since it has been defined in many different ways in the literature and there is no consensus on its meaning. Later in the paper, we relate our definition of dominant congested link to the notion of bottleneck link.

Identifying the existence of a dominant congested link is useful for traffic engineering. For example, when there are multiple paths from one host to another and all are congested, improving the quality along a path with one dominant congested link may require fewer resources than those along a path with multiple congested links. Identifying whether a path has a dominant congested link also helps us understand and model the dynamics of the network since the behavior of a network with a dominant congested link differs dramatically from one with multiple congested links.

When a dominant congested link exists, identifying the existence of such a link requires distinguishing its delay and loss characteristics from those of the other links. Achieving this goal via direct measurements is only possible for the organization in charge of that network. However, commercial factors often prevent an organization from disclosing the performance of internal links. Furthermore, as the Internet grows in both size and diversity, one organization may only be responsible for a subset of links on an end–end path. Some measurement techniques obtain internal properties of a path by using ICMP messages to query internal routers. *Traceroute* and *ping* are two widely used tools in this category. Some more advanced techniques use ICMP messages to measure per-hop capacity or delay [14], [8], [4] and pinpoint faulty links [24]. These approaches, however, require cooperation of the routers (to respond to ICMP messages and treat them similarly as data packets). Contrary to direct measurements using responses from routers, a collection of network tomography techniques infers internal loss rate and delay characteristics using end-end measurements [2], [6], [9]. Most tomography techniques, however, require observations from multiple vintage points.

In this paper, we propose a novel *model-based* approach to identify whether a dominant congested link exists along an end–end path using end–end measurements. We periodically send probes from one host to another so as to obtain a sequence of delay and loss values. The key insight in our approach is to utilize the queuing delay properties of the *lost* probes. For

example, if one link along the path is solely responsible for all losses, then all lost probes have the property that they "see" a full queue at this link.[1] We interpret a loss as an *unobserved delay* and discretize the delay values. Afterwards, we model the discretized delay sequence of all probes including those with missing values to infer whether a dominant congested link exists.

Our model-based approach has the following advantages. First, it utilizes delay and loss observations *jointly* for inference instead of the common approach of treating them separately. Second, it utilizes the correlation in the *entire* observation sequence instead of the very limited temporal correlation present in back-to-back packets. As we will see, the identification procedure only requires a short probing duration (in minutes).

The following are the main contributions of this paper.

- We present a formal yet intuitive definition of dominant congested link and provide two simple hypothesis tests to identify whether a dominant congested link exists along a path.
- Our model-based approach fully utilizes the information from the probing packets and enables very fast identification. Validation using *ns* simulation and Internet experiments demonstrates that this approach can correctly identify the existence of a dominant congested link in minutes.
- We provide a statistical upper bound on the maximum queuing delay of a dominant congested link once we identify such a link exists.

The rest of the paper is organized as follows. We review related work in Section II. In Sections III and IV, we provide a formal definition of dominant congested link and describe a methodology to identify whether such a link exists along a path. Section V presents our model-based approach. Section VI validates our approach using *ns* simulation and Internet experiments. Finally, Section VII concludes the paper, describes future work, and discusses other related issues.

II. Related Work

A dominant congested link is a link that produces most losses and significant queuing delays on an end–end path. Since most applications (TCP-based or real-time applications) are adversely affected by losses and delays, a dominant congested link is a form of "bottleneck link." Our definition of dominant congested link, however, differs from the notion of "bottleneck link" in the literature. One notion of bottleneck link is *tight link*, i.e., the link with the minimum available bandwidth; another notion is *narrow link*, i.e., the link with the minimum capacity [16]. Several studies focus on locating tight or narrow links [12], [3], [11], [14], [18]. Hu *et al.* design *Pathneck*, which combines closely spaced measurements and load packets to locate a tight link [12]. Akella *et al.* propose *BFind*, which gradually increases the sending rate of a UDP flow to locate a tight link [3]. Harfoush *et al.* use a packet train that contains packets of different sizes to measure the bandwidth of any segment of a network path, which can then be used to locate narrow links [11]. *Pathchar* estimates the capacity of each link on a network path and can naturally locate the narrow link of the path [14]. *MultiQ* discovers multiple bottleneck capacities along a path based on passive measurements of TCP flows [18]. Since a tight or narrow link may not be the one that produces most losses and significant queuing delays, our work complements others in identifying another form of bottleneck link along a path. We precisely define dominant congested link and differentiate it from other notions of bottleneck link in Section III. After identifying a dominant congested link, we further derive an upper bound of the maximum queuing delay of that link, which is an important path characteristic and is complementary to other tools that estimate the available bandwidth or the minimum link capacity of a path [20], [25], [7], [15], [32], [36], [13], [35].

Network tomography infers internal link properties through end–end measurements. A rich collection of network tomography techniques have been developed in the past (see [2] and [6] for a review). Many techniques rely on correlated measurements (through multicast or striped unicast probes). More recently, several studies use uncorrelated measurements to detect lossy links [28], [9], [5], [26], estimate loss rates [42], [27], or locate congested segments that have transient high delays [37]. Most tomography techniques, however, require *many* vintage points, while we only need measurements between two end-hosts along a *single* path.

The work closest in spirit to ours is the loss pair approach that is used to discover network properties [21], [22]. A loss pair is formed when two packets are sent close in time and only one of the packets is lost. Assuming that the two packets experience similar behaviors along the path, the packet not lost in a loss pair is used to provide insights on network conditions close to the time when loss occurs. Although our work also uses properties of lost packets, our objectives differ tremendously from those in [21] and [22]. More specifically, the study of [21] starts by assuming that a bottleneck link exists along the path and uses loss pairs to determine the maximum queuing delay of the bottleneck link. The study of [22] uses hidden Markov models to classify whether a packet loss occurs at a wired or a wireless part of the network based on the measurements of loss pairs. Our work focuses on determining whether a dominant congested link exists along a path. Furthermore, our model-based approach differs significantly from the loss pair approach: Our approach *infers* the properties of the lost packets by utilizing delay and loss observations jointly and the correlation in the entire observation sequence, instead of using direct measurements from the loss pairs. As we shall see (Section VI), our approach provides much more accurate results than the loss pair approach.

Lastly, the studies of [17], [33], and [19] detect *shared* congested links over *multiple paths*, while our study identifies dominant congested link along a *single* path.

III. Definition of Dominant Congested Link

In this section, we formally define dominant congested link and relate it to the widely used term "bottleneck link." For ease of reference, the key notation is summarized in Table I.

Consider K links/routers along an end–end path, as shown in Fig. 1. Each link/router is modeled as a droptail queue with a processing rate equal to the link bandwidth, and the maximum queue size equal to the buffer size of the router. Let Q_k denote

[1]We assume droptail queues and losses are caused by buffer overflow. A discussion on other scenarios is in Section VII.

TABLE I
KEY NOTATION

Notation	Definition
K	Number of links/queues along the path
Q_k	The maximum queuing delay at queue k
D_t^k	Queuing delay for virtual probe t at link k
D_t	Aggregate queuing delay for virtual probe t over all the links along the path
L_k	Set of virtual probes marked as lost at link k
L	Set of virtual probes with loss marks
F_k	Set of virtual probes that experience the maximum queuing delay Q_k at link k
F	Set of virtual probes that experience the maximum queuing delay at some link along the path

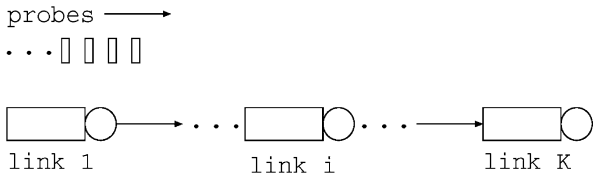


Fig. 1. Periodic probes are sent along a path with K links to identify the existence of dominant congested link.

the maximum queuing delay at queue k, i.e., the time required to drain a full queue. Then Q_k is determined by the buffer size and the link bandwidth of queue k. Probes are sent periodically from the source to the destination in a time interval $[t_1, t_2)$. We assume that the loss and delay characteristics experienced by the probes are stationary. Our goal is to determine whether a dominant congested link exists along the path based on measurements of the probes.

We next define dominant congested link formally using the concept of *virtual probes*, which is introduced for ease of understanding (identifying whether a dominant congested link exists using *real probes* is deferred to Section V). A virtual probe is an imaginary probe that goes through all the links along the path and records the delay (both propagation and queuing delay) at each link. If it "sees" a full queue when reaching router k, it records the maximum queuing delay Q_k and marks itself as lost. Otherwise, it calculates the queuing delay from the current queue length and the link bandwidth. The end–end delay for a virtual probe is the sum of its delays over all the links along the path. A virtual probe differs from a real probe in that it traverses all the links even if it is "lost" at some link. Furthermore, it does not occupy any position in the queue and hence does not affect packets that arrive afterwards. We refer to a virtual probe that is marked as lost at some link as a probe with a *loss mark*. Note that to be consistent with the fact that a real probe can only be lost once, a virtual probe can be marked as lost at most once.

Consider an arbitrary virtual probe sent at time t from the source, $t \in [t_1, t_2)$. We use the sending time t to index the virtual probe. That is, virtual probe t refers to the virtual probe that is sent at time t from the source. Let D_t^k denote the queuing delay for virtual probe t at link k, $1 \le k \le K$. Let D_t denote the aggregate queuing delay for virtual probe t over all the links along the path. That is, $D_t = \sum_{k=1}^{K} D_t^k$. Let L_k denote the set of virtual probes marked as lost at link k. Define $L = \bigcup_{k=1}^{K} L_k$, the set of all virtual probes with loss marks. For virtual probe t, $t \in L_k$ indicates that this probe is marked as lost at link k; $t \in L$ indicates that this probe has a loss mark. We further define F_k to be the set of virtual probes experiencing the maximum queuing delay Q_k at link k. That is, $F_k = \{t \,|\, D_t^k = Q_k\}$. Since a probe is very small relative to the full queue size, we assume that the queuing delay for a probe taking the last available buffer position at router k equals Q_k. Therefore, F_k contains all the probes that are either marked as lost at link k or take the last free buffer position at link k. Since $t \in L_k \Rightarrow D_t^k = Q_k$, we have $L_k \subseteq F_k$. Define $F = \bigcup_{k=1}^{K} F_k$, the set of the virtual probes that experience the maximum queuing delay at some link along the path. We then have $L \subseteq F$.

Definition 1: Link k is a **strongly dominant congested link** in time interval $[t_1, t_2)$ if and only if for a virtual probe sent at any time $t \in [t_1, t_2)$, the following two conditions are satisfied:

$$P(t \in L_k \,|\, t \in L) = 1 \tag{1}$$

$$P\left(D_t^k \ge \sum_{i \ne k} D_t^i \,\middle|\, t \in F_k \right) = 1. \tag{2}$$

In other words, link k is a strongly dominant congested link if and only if it is responsible for all the losses, and if a virtual probe experiences the maximum queuing delay at link k, this delay is no less than the aggregate queuing delays over all the other links. It is easy to see from this definition that a strongly dominant congested link is unique.

The above definition considers both loss and delay, reflecting our sense that a dominant congested link is one that causes most losses and leads to significant queuing delays. Note that the condition on queuing delay is defined over the virtual probes that experience the maximum queuing delay at link k instead of over all virtual probes. This definition accounts for the dynamic nature of the network since even a congested link may sometimes have very low queue occupancy. We next relax the strict delay and loss requirements in Definition 3.1 and define a weaker notion of dominant congested link.

Definition 2: Link k is a **weakly dominant congested link** with parameters θ and ϕ, where $0 \le \theta < 0.5$ and $0 \le \phi < 1$, in time interval $[t_1, t_2)$ if and only if for a virtual probe sent at any time $t \in [t_1, t_2)$, the following two conditions are satisfied:

$$P(t \in L_k \,|\, t \in L) \ge 1 - \theta \tag{3}$$

$$P\left(D_t^k \ge \sum_{i \ne k} D_t^i \,\middle|\, t \in F_k \right) \ge 1 - \phi. \tag{4}$$

In other words, link k is a weakly dominant congested link if and only if a virtual probe is lost at link k with a probability no less than $1 - \theta$, and if a virtual probe experiences the maximum queuing delay at link k, this queuing delay is no less than the aggregate queuing delays over all the other links with a probability no less than $1 - \phi$. Since $0 \le \theta < 0.5$, that is, more than half of the losses occur at a weakly dominant congested link, a weakly dominant congested link is unique.

Note that the lower the values of θ and ϕ, the more stringent are the requirements on being a weakly dominant congested link. In particular, the definition of a weakly dominant congested link is the same as that of a strongly dominant congested link

when $\theta = \phi = 0$. A link identified as a weakly dominant congested link with θ and ϕ is also a weakly dominant congested link with θ' and ϕ', where $\theta' \geq \theta$ and $\phi' \geq \phi$. In particular, a strongly dominant congested link is a weakly dominant congested link with any $\theta \geq 0$ and $\phi \geq 0$.

Lastly, the definitions of strongly and weakly dominant congested link can be generalized by introducing a parameter ρ in the delay conditions [39]. In this paper, we focus on dominant congested link as defined in Definitions 3.1 and 3.2.

A. Dominant Congested Link Versus Bottleneck Link

A bottleneck link is typically defined to be a link with high loss rate, long queuing delay, high utilization, low available bandwidth, or low link capacity. It is not a dominant congested link if it does not satisfy the conditions in Definition 3.1 or 3.2. Several other differences between bottleneck link and dominant congested link are the following.

- Whether or not a link is a dominant congested link is relative. A link with a low loss rate is a dominant congested link as long as it satisfies the corresponding delay and loss requirements, despite the low loss rate.
- By definition, dominant congested link is unique if it exists, while there may exist multiple bottleneck links along a path.
- Neither strongly nor weakly dominant congested link can describe links that do not have losses. Therefore, a link with the lowest capacity, available bandwidth, or highest utilization is not a dominant congested link if no loss occurs at that link.

IV. Identification of Dominant Congested Link

In this section, we first describe two hypothesis tests to identify whether a dominant congested link exists along a path. We then describe how to obtain an upper bound on the maximum queuing delay of a dominant congested link after detecting its presence.

A. Hypothesis Tests

Our hypothesis tests utilize the queuing delays of the virtual probes with loss marks, i.e., virtual probes in L. We next use an example to illustrate why these queuing delays are helpful for dominant congested link identification. Suppose the null hypothesis is that there exists a strongly dominant congested link k. Then, if this hypothesis holds, the queuing delay of any virtual probe in L must satisfy the following two properties based on Definition 3.1. First, by Condition (1), it must be no less than Q_k, the maximum queuing delay at link k. Second, it must satisfy Condition (2) since all probes in F must satisfy this condition and L is a subset of F. If one of the two conditions does not hold, we can reject the null hypothesis.

We next describe the identification methodology in detail. Let W be a random variable representing the discretized end–end queuing delay of virtual probes in L. The discretization is as follows. Let D_0 denote the end-end propagation delay along the path. Let $D_{\max}$ denote the largest end-end delay of all virtual probes sent in the time interval $[t_1, t_2)$ (including those with and without loss marks). The maximum queuing delay is therefore $D_{\max} - D_0$. We divide the range of queuing delay, $[0, D_{\max} - D_0]$, into M equal length bins with bin width $b = (D_{\max} - D_0)/M$. Then, W takes value in $\{1, 2, \ldots, M\}$, where i corresponds to an actual delay value between $(i-1)b$ and ib. Let $F_W(w)$ represent the cumulative distribution function (CDF) of W. That is, $F_W(w) = P(D_t \leq w \,|\, t \in L)$ for any virtual probe sent at time $t \in [t_1, t_2)$. Then, $F_W(w)$ satisfies the following properties (their proofs are found in Appendix A).

Theorem 1: Let $D = \min\{w \,|\, F_W(w) > 0\}$, where $w = 1, 2, \ldots, M$. If link k is a strongly dominant congested link, then $D \geq Q_k$ and $F_W(2D) = 1$.[2]

Theorem 2: Let $D = \min\{w \,|\, F_W(w) > \theta\}$, where $w = 1, 2, \ldots, M$. If link k is a weakly dominant congested link with parameters θ and ϕ, then $D \geq Q_k$ and $F_W(2D) \geq (1-\theta)(1-\phi)$.

These two theorems form the basis of the hypothesis tests for identifying dominant congested link. In particular, the hypothesis test for strongly dominant congested link is based on Theorem 1. We refer to this test as *SDCL-Test* and summarize it in Fig. 2. In this test, the null hypothesis H_0 is that a strongly dominant congested link exists along a path. When the property in Theorem 1 is violated, we reject H_0. Otherwise, we accept it. Similarly, we have a hypothesis test for a weakly dominant congested link based on Theorem 2. This test is described in Fig. 3 and is referred to as *WDCL-Test*.

H_0: A strongly dominant congested link exists along the path.

Step 1: From $F_W(w)$, find $D = \min\{w \mid F_W(w) > 0\}$, where $w = 1, 2, \ldots, M$.
Step 2: If $F_W(2D) < 1$, we reject H_0. Otherwise, we accept H_0.

Fig. 2. SDCL-Test: Hypothesis test for a strongly dominant congested link.

H_0: A weakly dominant congested link with parameters θ and ϕ exists along the path.

Step 1: From $F_W(w)$, find $D = \min\{w \mid F_W(w) > \theta\}$, where $w = 1, 2, \ldots, M$.
Step 2: If $F_W(2D) < (1-\theta)(1-\phi)$, we reject H_0. Otherwise, we accept H_0.

Fig. 3. WDCL-Test: Hypothesis test for a weakly dominant congested link.

We next give an example to illustrate SDCL-Test. Consider a path with K links and at least two of them are lossy. By Definition 3.1, there exists no strongly dominant congested link along this path. We next show that SDCL-Test indeed provides a correct result. Let I denote the set of lossy links. We assume all the links are independent and $P(D_t^k > 0) \in (0, 1), 1 \leq k \leq K$. Therefore, as time goes to infinity, for all the virtual probes with loss marks, the smallest queuing delay is $\min_{i \in I} Q_i$. That is, $D = \min_{i \in I} Q_i$ by the definition of D in Theorem 1. We will also observe a queuing delay of $\sum_{i \in I} Q_i + \epsilon$, where $\epsilon > 0$ is the sum of queuing delays from all the links not in I. It is clear that $\sum_{i \in I} Q_i + \epsilon > 2D$ since there are at least two lossy links and the links are independent. Therefore, $F_W(2D) < 1$, which indicates that there is no strongly dominant congested link. This

[2]We slightly abuse notation here and let Q_k also represent the discretized maximum queuing delay at link k. Whether Q_k represents the actual value or the discretized value should be clear from the context.

example also shows that SDCL-Test provides correct identification asymptotically. In practice, we do not require time to go to infinity, but require that the duration be "sufficiently long," an issue we will return to in Section VI.

B. Upper Bound of the Maximum Queuing Delay at a Dominant Congested Link

Suppose link k is a strongly dominant congested link. We estimate an upper bound of its maximum queuing delay Q_k as follows. From $F_W(w)$, we find the smallest value D such that $F_W(D) > 0$. Since all losses occur at link k, by the definition of $F_W(w)$, $D \geq Q_k$. Therefore, D is an upper bound of Q_k (note that D is a discretized delay value; the corresponding actual delay value is $(D-1)b$, where b is the bin width).

For a weakly dominant congested link k with parameters θ and ϕ, we can obtain an upper bound on its maximum queuing delay Q_k in a similar manner. More specifically, from $F_W(w)$, we find the smallest value D such that $F_W(D) > \theta$, then D can be used as an upper bound of Q_k since $D \geq Q_k$ by Theorem 2 (again the actual delay bound is $(D-1)b$, where b is the bin width). For link k with a very small value of θ, we can apply the following heuristic to obtain a tighter bound on Q_k. When plotting the probability mass function (PMF) of W, we choose the number of bins so that the resulting PMF has a connected component with most of the mass, and the rest of the components are as separated from it as possible. We can then use the smallest delay value that has probability significantly larger than 0 in this connected component as an upper bound of Q_k (this value can be easily located from the PMF; we do not define precisely a threshold for being significantly larger than 0). This method is illustrated using an example in Section VI-A2.

The rationale for the above heuristic is as follows. For ease of exposition, let us first consider the case where link k is a weakly dominant congested link with $\theta = 0$, that is, all losses occur at link k. Then, each instance of W is the sum of Q_k and the queuing delays over the rest of the links. Therefore, when choosing the number of bins properly, the PMF of W has a single connected component, where the lowest delay value is an upper bound of Q_k. Let us now consider the case where link k is a weakly dominant congested link with $\theta > 0$, that is, at least $(1-\theta)$ of the losses occur at link k. For very small θ, almost all losses occur at link k, and hence almost all instances of W are Q_k plus the queuing delays over the rest of the links, which form a connected component with most of the mass in the PMF of W (when choosing the number of bins properly). Therefore, we choose the lowest delay value with probability significantly larger than 0 in this connected component as an upper bound of Q_k.

V. Model-Based Identification of Dominant Congested Link

Our methodology for dominant congested link identification in Section IV relies on $F_W(w)$, queuing delay distribution of virtual probes with loss marks. In practice, virtual probes do not exist, and therefore we need to obtain $F_W(w)$ using *real* probes. In this section, we describe a novel model-based approach for this purpose. We first define *virtual queuing delay* for *lost* probes (it is equivalent to queuing delay of virtual probes with loss marks), and then describe how to obtain its distribution using a model-based approach.

A. Virtual Queuing Delay

A real probe differs from a virtual probe in that it does not have a delay if it is lost, while a virtual probe has an end–end delay even if it is "lost" in the middle. Analogous to the queuing delay of a virtual probe, we associate a *virtual queuing delay* to a lost real probe as follows. Suppose the probe is lost at link k. We imagine that the probe experiences the maximum queuing delay of this link, and then goes to the next link, where it experiences a queuing delay based on the queue occupancy at the arrival time, and then goes to the next link. This process repeats until it reaches the sink. The end–end virtual queuing delay for the probe is the (virtual) arrival time of the probe at the sink minus the sending time at the source.

By definition, virtual queuing delay of a lost probe is equivalent to queuing delay of a virtual probe with a loss mark. We therefore also use W to represent virtual queuing delay, and use $F_W(w)$ represent its CDF. As in Section IV, W is represented using discretized values. Denote the smallest and the largest end–end delays of all the probes that are not lost as $D_{\min}$ and $D_{\max}$, respectively. If the end–end propagation delay along the path, D_0, is known, we divide the range $[0, D_{\max} - D_0]$ into M equal length bins. Otherwise, we use $D_{\min}$ to approximate D_0 (our simulation and experiments in Section VI shows that the inaccuracy caused by this approximation is negligible when the probing duration is longer than several minutes).

Note that for an end–end path, its virtual queuing delay distribution may differ significantly from its observed queuing delay distribution because the former is from the *lost* probes while the latter is from the *observed* probes (i.e., probes that are not lost). This difference can be easily understood from the following example. Consider a path that contains a strongly dominant congested link k, while the rest of the links have no loss and negligible queuing delay. Then, the virtual queuing distribution concentrates on a single value, equal to the maximum queuing delay of link k, Q_k, while the observed queuing delay distribution is spread out between the value of 0 and Q_k.

B. Obtaining Virtual Queuing Delay Distribution

Broadly, we can use two types of approaches to obtain $F_W(w)$: empirical approach and model-based approach. One example of empirical approach is using loss pairs [21]. It assumes that two probes in a pair experience the same queuing delay, and uses the queuing delay of the probe that is not lost as the virtual queuing delay of the lost probe. This approach, however, is not always accurate. As shown in [21] and our experiments (Section VI), cross traffic can cause two probes in a pair to experience significantly different queuing delays.

In this paper, we focus on model-based approach, which infers virtual queuing delay distribution $F_W(w)$ using both measurements and a model. In particular, we investigate two models: hidden Markov model (HMM) [31] and Markov model with a hidden dimension (MMHD) [38]. Both models consider N hidden states and M observation symbols (corresponding to the M discretized queuing delay values). Using hidden states provides much more flexibility and can significantly reduce the

size of the state space compared to a Markov model [31], [38], [34].

The key insight of our model-based approach is that we interpret a loss as a delay with a *missing value* and develop expectation and maximization (EM) algorithms to obtain $F_W(w)$. We find that MMHD provides accurate results in all the cases we investigated (Section VI), while the results from HMM are not always accurate (see one example in Section VI-A3). This is because MMHD captures correlation between delay observations more accurately [38]. We next briefly describe MMHD, and then present an inference procedure to obtain $F_W(w)$ using this model (the EM algorithm for HMM is by extending that in [31] to deal with missing values and is omitted).

MMHD differs from the traditional HMM in that its state space contains both observations and hidden states (the state space of HMM only contains hidden states). Let Z_t denote the state of the model at time t. Then, Z_t contains two components, i.e., $Z_t = (X_t, Y_t)$, where $X_t \in \{1, 2, \ldots, N\}$ represents the hidden state, and $Y_t \in \{1, 2, \ldots, M\}$ represents the delay symbol at time t. Let π denote the initial distribution of the states. Let P denote the probability transition matrix. An element in the transition matrix P is denoted as $p_{(i,j)(k,l)}$, which represents the transition probability from state (i, j) to state (k, l). Note that the model degenerates to a Markov model when $N = 1$ since, in this case, every state in the model contains the same hidden state and only differs in the delay symbol. Let y_t be the observation value for Y_t. If the observation at time t is a loss, we regard it as a delay with a *missing value*, and use $y_t = *$ to denote it. Let $s(j)$ be the conditional probability that an observation is a loss given that its delay symbol is j. That is, $s(j) = P(y_t = * \mid Y_t = j)$.

Let $\lambda = (P, \pi, s)$ denote the complete parameter set of the model. We develop an EM algorithm to infer λ from a sequence of T observations. It is an iterative procedure and terminates when a certain convergence threshold is reached. A detailed description of the EM algorithm is in Appendix B. After obtaining the model parameters, we obtain the PMF of W, $f_W(w) = P(Y_t = w \,|\, y_t = *)$, as

$$f_W(w) = \frac{s(w) \sum_{t=1}^{T} \mathbf{1}(y_t = w)}{\sum_{t=1}^{T} \mathbf{1}(y_t = *)} \tag{5}$$

where $\mathbf{1}(\cdot)$ is the indicator function. This equation follows from Bayes formula: The numerator corresponds to the probability that a loss has delay symbol of w, and the denominator corresponds to the probability of loss in the sequence of T observations.

After obtaining $f_W(w)$, we obtain $F_W(w)$ directly from $f_W(w)$. Note that (5) relies on $s(w)$, which is obtained from the EM algorithm that uses the entire observation sequence (as shown in the derivation in Appendix B). Therefore, $F_W(w)$ is obtained using the information in the entire observation sequence, not only the loss observations.

VI. Validation

In this section, we validate the model-based identification method using both *ns* simulations and Internet measurements. We further explore the impact of various parameters (e.g., M and N in the models, the convergence threshold in the EM algorithms, and probing duration) on identification results.

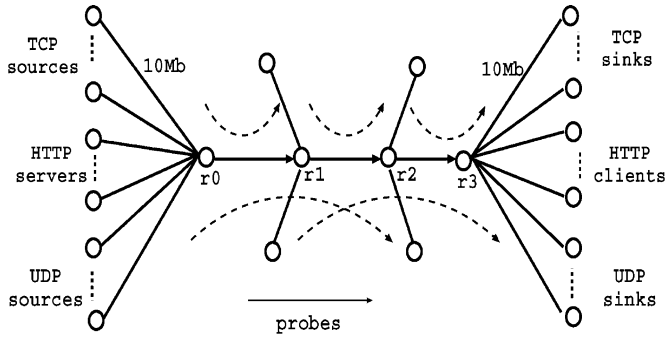


Fig. 4. Topology used in *ns*.

A. Validation Using ns Simulations

We use a topology containing four routers, r_0, r_1, r_2, and r_3, in *ns* simulation, as shown in Fig. 4. Link (r_i, r_{i+1}) denotes the link from router r_i to r_{i+1}, where $0 \le i \le 2$. The bandwidth and the buffer size of link (r_i, r_{i+1}) are varied to create different scenarios. All the other links (from a source or a sink to its corresponding router) have bandwidth of 10 Mb/s and buffer size sufficiently large so that no loss occurs. The propagation delay of link (r_i, r_{i+1}) is 5 ms. The propagation delay from a source or a sink to its corresponding router is uniformly distributed in $[10, 20]$ ms. We create three types of traffic conditions. The first type only has TCP-based traffic (in particular, FTP and HTTP traffic) from router r_i to r_j. The number of FTP flows ranges from 1 to 10, and the HTTP traffic is generated using the empirical data provided by *ns*. The second type only has UDP on-off traffic on link (r_i, r_{i+1}). The third type has both TCP-based and UDP ON–OFF traffic. The utilization of link (r_i, r_{i+1}) varies from 28% to 95% in different scenarios. We only present results under the third type of conditions; results under the other two types are similar (indeed, our scheme relies on virtual queuing distribution and is not sensitive to whether the congestion is caused by TCP or UDP traffic). In each experiment, we send UDP probes periodically along the path from r_0 to r_3 at an interval of 20 ms. Each probe is 10 bytes. Therefore, the traffic generated by the probing process is 4 kb/s, much smaller than the link bandwidths used in the simulation.

We use four methods to obtain virtual queuing delay distributions. The first method obtains the *actual* virtual queuing delay for each lost probe from the traces logged in *ns*. The second method uses loss pairs [21] (by sending two back-to-back probes from r_0 to r_3 at an interval of 40 ms; we use a 40-ms interval because it leads to the same number of probes as sending a single probe every 20 ms). The last two methods are model-based, using HMM and MMHD, respectively. For both models, unless otherwise specified, the number of delay symbols $M = 5$, the number of hidden states N is in the range of 1 to 4, and the convergence threshold in the EM algorithms is 10^{-4} or 10^{-5} (these two thresholds lead to similar results; we only present results using threshold 10^{-5}). For HMM, the initial values of the EM algorithm are chosen based on guidelines in [31]. For MMHD, the initial values in the transition matrix P

TABLE II
STRONGLY DOMINANT CONGESTED LINK: BANDWIDTHS AND CORRESPONDING LOSS RATES OF LINK (r_0, r_1)

Bw (Mbps)	Loss rate	Max. queuing delay (ms)		
		ns	model	loss pair
0.1	3.3%	200	200	205
0.2	2.5%	100	101	101
0.4	0.04%	50	51	53
1.0	0.02%	20	22	21

are chosen randomly; the initial distributions of π and s follow a uniform distribution. Unless otherwise stated, model-based approach in the rest of the paper refers to that using MMHD since it achieves accurate results in all the settings we investigate.

After determining that a dominant congested link exists, we further estimate an upper bound on the maximum queuing delay of that link. More specifically, we first use our model-based approach to obtain virtual queuing delay distribution, and then use the approach in Section IV-B to obtain an upper bound. For this purpose, our models use $M = 40$ (instead of $M = 5$) since a larger number of delay symbols provides finer granularity in the estimate. For comparison, we also obtain an upper bound of the maximum queuing delay using loss pairs [21].

We next report simulation results in three settings: when there exists a strongly or weakly dominant congested link, and when no dominant congested link exists. Each simulation runs for 2000 s. By default, we use the trace between 1000 and 2000 s to identify whether a dominant congested link exists. In Section VI-A4, we vary the duration of the probing process to investigate what duration is needed for accurate results. At the end, we report simulation results for routers using Active Queue Management (AQM) (all the other results are for drop-tail queues). For all the results presented below, we obtain the minimum and maximum end–end delay, $D_{\min}$ and $D_{\max}$, from the probing interval that is used to obtain the identification results. The propagation delay along the path, D_0, is unknown, and we use $D_{\min}$ to approximate it.

1) Strongly Dominant Congested Link: We first investigate settings in which a strongly dominant congested link exists. In particular, we set the various parameters so that losses only occur at link (r_0, r_1) (we also investigate several settings where losses only occur at link (r_2, r_3), and obtain accurate results in those settings). The buffer sizes at routers r_0, r_1, and r_2 are 20, 80, and 80 kb, respectively. The bandwidth of link (r_0, r_1) is varied from 0.1 to 1 Mb/s. The bandwidths of links (r_1, r_2) and (r_2, r_3) are both 10 Mb/s. Table II lists the bandwidths and corresponding loss rates of link (r_0, r_1) for four settings. In all the settings, our model-based approach (using MMHD) correctly accepts the null hypothesis that a strongly dominant congested link exists. We further estimate an upper bound on the maximum queuing delay of the strongly dominant congested link. Table II lists the actual maximum queuing delay (obtained directly from *ns*) and the estimates from MMHD and the loss pair approach. The estimates from both approaches are accurate: The maximum errors are 2 and 5 ms, respectively.

We next describe one setting in Table II in detail. In this setting, the bandwidth of link (r_0, r_1) is 1 Mb/s. Fig. 5 plots PMFs of the virtual queuing delays directly from *ns* (marked as "ns virtual") and from MMHD ($N = 1, 2$). The distributions from MMHD match the actual distribution, all concentrating on the discretized delay of 5. For illustration purpose, we also plot the observed queuing delay distribution (marked as "observed") in Fig. 5. As explained in Section V-A, the observed queuing delay distribution differs dramatically from virtual queuing delay distribution: The former has discretized delay values from 1 to 5, while the latter concentrates on the value of 5. Lastly, when using MMHD, we observe that $D = 5$ is the minimum delay such that $F_W(D) > 0$. Since $2D = 10 > M = 5$, we have $F_W(2D) = 1$. By SDCL-Test, we accept the null hypothesis that a strongly dominant congested link exists.

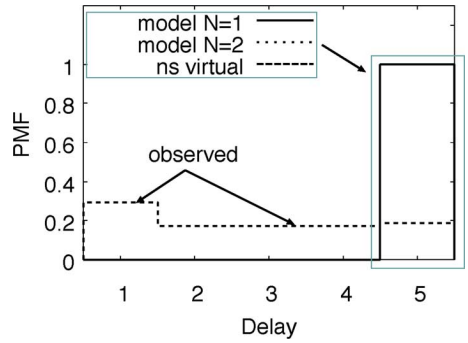


Fig. 5. Distributions of the observed and virtual queuing delays for a setting in which link (r_0, r_1) is a strongly dominant congested link.

TABLE III
WEAKLY DOMINANT CONGESTED LINK: BANDWIDTHS (IN Mb/s) AND CORRESPONDING LOSS RATES OF LINKS (r_0, r_1) AND (r_2, r_3)

(r_0, r_1)		(r_2, r_3)		Max. queuing delay (ms)		
Bw	Loss rate	Bw	Loss rate	*ns*	model	loss pair
0.7	0.2%	0.2	3.8%	128	128	164
1.0	0.1%	0.1	3.8%	256	258	281
0.2	0.04%	0.3	0.9%	85	90	90
0.5	0.05%	0.3	0.7%	85	86	136

2) Weakly Dominant Congested Link: We next investigate settings in which a weakly dominant congested link exists. In particular, we set the parameters such that losses occur at links (r_0, r_1) and (r_2, r_3), and the loss rate at (r_2, r_3) is significantly larger than that at (r_0, r_1). The buffer sizes at routers r_0, r_1, and r_2 are 25.6, 76.8, and 25.6 kb, respectively. The link bandwidth of (r_1, r_2) is 1 Mb/s. The link bandwidths of (r_0, r_1) and (r_2, r_3) with their corresponding loss rates are listed in Table III.

The null hypothesis is that there exists a weakly dominant congested link with $\theta = 0.06$ and $\phi = 0$. That is, the requirements on the weakly dominant congested link are the following: At least 94% of the losses occur at this link; furthermore, when a probe experiences the maximum queuing delay at this link, 100% of the time this queuing delay is no less than the aggregate queuing delay over other links. Our model-based approach (using MMHD) accepts the null hypothesis for all the settings. We further estimate an upper bound on the maximum queuing delay at the weakly dominant congested link. Table III lists the actual maximum queuing delay (directly from *ns*) and the estimates from our model-based approach and the loss pair approach. The estimates from our model-based approach are accurate (with a maximum error of 5 ms), while the loss pair ap-

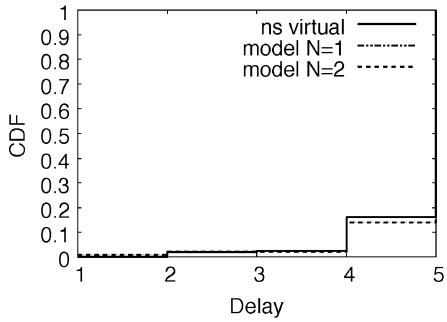


Fig. 6. Virtual queuing delay distribution for a setting in which link (r_2, r_3) is a weakly dominant congested link.

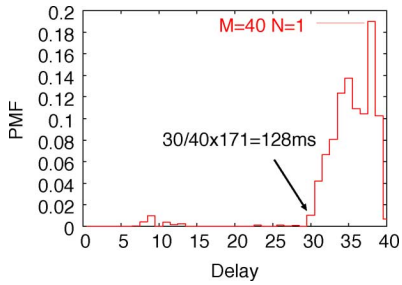


Fig. 7. Example to estimate an upper bound on the maximum queuing delay for the weakly dominant congested link (r_2, r_3).

proach can lead to large errors (the maximum error is 51 ms) since it is sensitive to queuing delays at links other than the weakly dominant congested link.

We next describe one setting in Table III in detail. In this setting, the bandwidths of (r_0, r_1) and (r_2, r_3) are 0.7 and 0.2 Mb/s, respectively. The average loss rate on link (r_2, r_3) is 3.8%, which accounts for 95% of the losses. Fig. 6 plots the virtual queuing delay distributions obtained directly from *ns* and from MMHD $(N = 1, 2)$. We observe that the distributions from our model are very similar to that directly from *ns*. From Fig. 6, $D = 1$ is the minimum delay such that $F_W(D) > 0$ ($F_W(1) = 0.01$, which is not quite observable from the figure; $F_W(5) = 1$). Since $F_W(2D) = F_W(2) = 0.02 < 1$, by SDCL-Test, no strongly dominant congested link exists along the path. For $\theta = 0.06$ and $\phi = 0$, $D = 4$ is the minimum delay such that $F_W(D) > \theta$. Since $2D = 8 > M = 5$, we have $F_W(2D) = 1$. By WDCL-Test, we accept the hypothesis that there exists a weakly dominant congested link with $\theta = 0.06$ and $\phi = 0$. When using $\theta = 0.02$ and $\phi = 0$ as the parameters, we reject the hypothesis, which is correct since no link in this setting is responsible for more than 98% of the loss.

We next describe how we obtain an upper bound of the maximum queuing delay at the weakly dominant congested link in this setting. To obtain an accurate estimate, we discretize delays more finely and use $M = 40$. Fig. 7 plots the PMF of the virtual queuing delays (using MMHD, $M = 40$, and $N = 1$). According to the heuristic described in Section IV-B, we first find the connected component with most of the mass, which is the rightmost component in Fig. 7. In this component, $D = 30$ is the minimum delay that has probability significantly larger than 0. The queuing delay range is $[0, 171]$ ms. Therefore, an upper bound on the maximum queuing delay at link (r_2, r_3) is

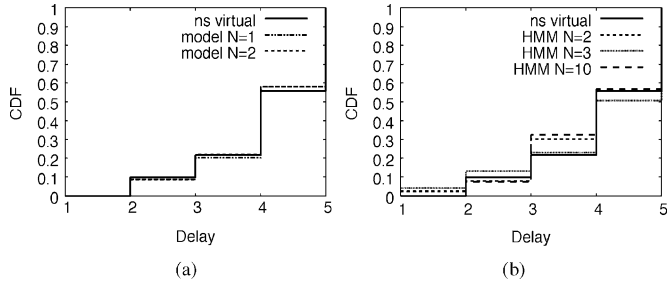


Fig. 8. Virtual queuing delay distribution for a setting with no dominant congested link. (a) MMHD. (b) HMM.

TABLE IV
NO DOMINANT CONGESTED LINK: BANDWIDTHS AND CORRESPONDING LOSS RATE OF LINKS (r_0, r_1) AND (r_2, r_3)

(r_0, r_1)		(r_2, r_3)	
Bw (Mbps)	Loss rate	Bw (Mbps)	Loss rate
0.4	0.26%	0.8	0.04%
0.2	0.3%	0.5	0.2%
0.1	2.2%	0.3	1.0%
0.1	2.3%	0.2	2.0%

$30/40 \times 171 = 128$ ms, which is very accurate (it equals to the actual maximum queuing delay).

3) No Dominant Congested Link: We next investigate settings in which no dominant congested link exists. In particular, we vary the parameters such that losses occur at links (r_0, r_1) and (r_2, r_3), and the loss rates at these two links are comparable. The buffer sizes at routers r_0, r_1, and r_2 are 25.6, 128, and 25.6 kb, respectively. The link bandwidth of (r_1, r_2) is 1 Mb/s. The link bandwidths and average loss rates of (r_0, r_1) and (r_2, r_3) are listed in Table IV. The null hypothesis is that there exists a weakly dominant congested link with $\theta = 0.03$ and $\phi = 0$. For all settings, our model-based approach (using MMHD) correctly rejects the hypothesis.

We describe the results from one setting in detail. In this setting, the bandwidths of links (r_0, r_1) and (r_2, r_3) are 0.1 and 0.2 Mb/s, respectively. Their loss rates are similar (2.3% and 2.0%, respectively). We therefore have two lossy links and no dominant congested link. Fig. 8(a) and (b) plots the virtual queuing delay distributions from MMHD $(N = 1, 2)$ and HMM $(N = 2, 3, 10)$, respectively. In both figures, we also plot the distribution obtained from *ns*. We observe that the distributions from MMHD match the *ns* result very well, while the distributions from HMM deviate from the *ns* result even for large N $(N = 10)$, indicating that MMHD is a more suitable model. We observe from Fig. 8(a) that for $\theta = 0.03$ and $\phi = 0$, $D = 2$ is the minimum delay such that $F_W(D) > \theta$. However, $F_W(2D) = F_W(4) = 0.58 < (1 - \theta)(1 - \phi) = 0.97$. We therefore conclude that there is no weakly dominant congested link with $\theta = 0.03$ and $\phi = 0$. Of course, there is no weakly dominant congested link with lower values of θ and ϕ either.

4) Required Probing Duration for Accurate Identification: So far, for each experiment, we have been using a trace of 1000 s (with 50 000 observations). We next investigate the impact of probing duration on the accuracy of identification results (we only consider the settings with average loss rate above 1%). For this purpose, we randomly choose a segment from the 1000-s

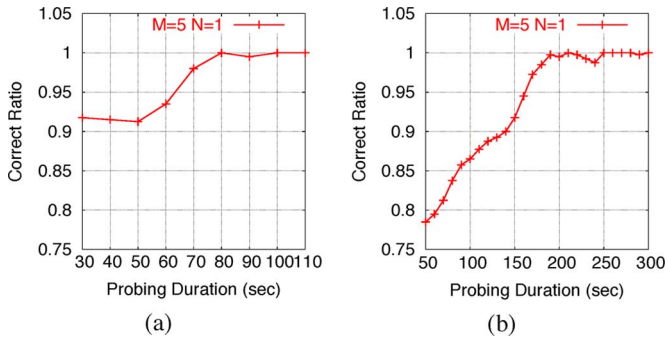


Fig. 9. Ratio of correct identification versus probing duration for two settings in *ns*. (a) Setting with a weakly dominant congested link. (b) Setting with no dominant congested link.

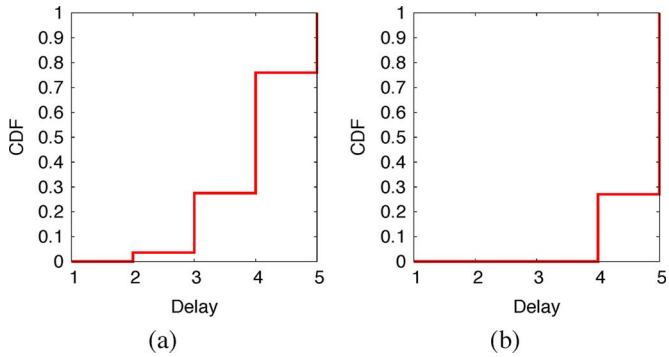


Fig. 10. Results under RED queues when there exists a strongly dominant congested link. The minimum threshold is (a) one-fifth and (b) half of the buffer size (i.e., 5 and 12 packets, respectively); $M = 5, N = 2$.

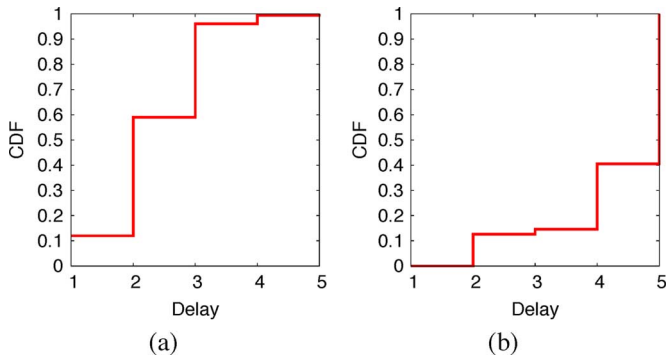


Fig. 11. Results under RED queues when there exists no dominant congested link. The minimum threshold is (a) 1/20 and (b) half of the buffer size (i.e., 5 and 50 packets, respectively); $M = 5, N = 2$.

trace as the probing sequence and identify whether a dominant congested link exists (using MMHD with $M = 5$ and $N = 1$). We repeat the process 400 times and obtain the fraction of correct identifications.

In the cases where a strongly dominant congested link exists, a probing duration of several tens of seconds suffices to achieve correct identification. In other cases (i.e., a weakly dominant congested link or no dominant congested link), the probing duration needs to be several minutes to achieve high accuracy. As an example, Fig. 9(a) and (b) plots the correct ratio versus probing duration for settings with and without a weakly dominant congested link (the settings described in detail in Sections VI-A2 and VI-A3), respectively. We observe that probing durations longer than 80 and 250 s are needed respectively for accurate results.

5) Routers Using Active Queue Management: A router that uses AQM can drop a packet even before the queue is full, which violates the droptail assumption in our scheme. We next investigate the performance of our scheme when routers use AQM, in particular when they use adaptive RED [10]. For such a router, the packet dropping rate increases linearly from 0 to $\max_p \in (0, 1]$ as the average queue size increases from a minimum threshold $\min_{\text{th}}$ to a maximum threshold $\max_{\text{th}}$. Furthermore, in the gentle mode (which we adopt), the dropping rate increases linearly from $\max_p$ to 1 as the average queue size increases from $\max_{\text{th}}$ to $2\max_{\text{th}}$.

We consider two examples, one with a strongly dominant congested link, and the other with no dominant congested link. More specifically, we consider the examples described in detail in Sections VI-A1 and VI-A3, respectively, and change the queuing discipline of all the links from droptail to adaptive RED. In both examples, all the links/queues have the same minimum threshold $\min_{\text{th}}$, the maximum threshold $\max_{\text{th}}$ is three times of $\min_{\text{th}}$, and $\max_p$ is chosen adaptively for all the links [10].

We first consider the example with a strongly dominant congested link. In this example, all losses happen at link (r_0, r_1). We consider two settings, where the minimum threshold of (r_0, r_1) is set to 5 and 12 packets, corresponding to $1/5$ and half of the buffer size, respectively. Fig. 10 plots the virtual queuing delay distribution for these two settings. Our identification is incorrect for small $\min_{\text{th}}$ ($F(2D) = F(4) < 1$ in Fig. 10(a), where $\min_{\text{th}}$ is five packets) and is correct for relatively large $\min_{\text{th}}$ ($F(2D) = F(8) = 1$ in Fig. 10(b), where $\min_{\text{th}}$ is 12 packets). The incorrect result under small $\min_{\text{th}}$ is expected since Theorem 1 does not hold for nondroptail queues. However, when $\min_{\text{th}}$ is large, a RED queue drops packets under large queue sizes and can hence behave similarly as a droptail queue. In that case, our scheme can make a correct identification (our approach uses discretized delay values and hence does not require that packet-drop only happens when the queue is full).

We next look at the example with no dominant congested link. In this example, links (r_0, r_1) and (r_2, r_3) have comparable loss rates, while the loss rate at link (r_1, r_2) is negligible. We again consider two settings, where the minimum threshold of both (r_0, r_1) and (r_2, r_3) is set to 5 and 50 packets corresponding to $1/20$ and half of their buffer size (they have the same buffer size), respectively. The resultant virtual queuing delays are plotted in Fig. 11. In both settings, our scheme correctly rejects the hypothesis that there exists a weakly dominant congested link with $\theta = 0.03$ and $\phi = 0$: $F(2D) = F(2) < 1$ in Fig. 11(a) and $F(2D) = F(4) < 1$ in Fig. 11(b). This is not surprising since the collective behavior of two congested RED queues differs from a (weakly or strongly) dominant congested queue.

B. Internet Experiments

We evaluate the performance of our scheme using Internet experiments conducted in February 2003 and June 2010. Results

for the former are described in [39]. We next only present results for the latter.

From June 21–25, 2010, we conduct two sets of experiments using hosts in PlanetLab [30]. The first set of experiments uses the same sender, located at Cornell University, Ithaca, NY, and different receivers located at Universidade Federal do Paraná (UFPR), Curitiba, Brazil; Seoul National University (SNU), Seoul, Korea; and Universidad de Sevilla (USevilla), Seville, Spain, respectively, all using Ethernet to access the Internet. The second set of experiments uses the same receiver, an ADSL host in the east coast of the United States, and different senders located at Cornell University, UFPR, SNU, and USevilla, respectively. In each experiment, we send periodic UDP probes from a sender to a receiver at the interval of 20 ms and run *tcpdump* [1] to capture the timestamps of the probes at these two hosts to obtain one-way delays (since the clocks of these two hosts are not synchronized, we use the method proposed in [40] to remove clock offset and skew in one-way delays). Each experiment lasts for 1 h.

For each experiment, we select a stationary probing sequence of 20 min for model-based identification (using MMHD). The queuing delays are discretized into five delay symbols, that is, $M = 5$. We vary the number of hidden states N from 1 to 4. For all the experiments, the null hypothesis is that there exists a weakly dominant congested link with $\theta = 0.05$ and $\phi = 0.05$. Note that it is very difficult to validate the results from the Internet experiments since we do not have access to the internal routers to measure per-hop delay and loss for each probe. We therefore use some existing measurement tools and verify whether our identification results are consistent with results from these tools. In particular, we use *pchar* [23] (a tool based on pathchar [14]) to estimate link bandwidth along a path, and use traceroute to obtain a crude estimate of the delays to all the routers along the path. We next describe the results for the two sets of experiments. The starting time of all the experiments described below are in Coordinated Universal Time (UTC).

1) Ethernet Receivers: In this set of experiments, the sender is at Cornell University, and the receivers are at UFPR, SNU, and USevilla, all using Ethernet to access the Internet. Only the path with the receiver at UFPR has losses. We therefore only report results of an experiment on that path. The experiment started at 12 p.m. on June 25, 2010. There are 11 hops along this path. The average loss rate of this probing sequence is 0.1%. Fig. 12 shows the inferred virtual queuing delay distributions using $N = 1, 2, 3$, and 4. The distributions under different values of N are very similar, all concentrating on discretized delay of 1. For $\theta = 0.05$ and $\phi = 0.05$, $D = 1$ is the minimum delay such that $F_W(D) > \theta$. Since $F_W(2D) = F_W(2) = 1 > (1-\theta)(1-\phi) \approx 0.90$, by WDCL-Test, we accept the hypothesis that there exists a weakly dominant congested link with $\theta = 0.05$ and $\phi = 0.05$. Results from *pchar* indicate that one link inside Brazil has much lower bandwidth than others, which is consistent with our identification.[3]

2) ADSL Receiver: In this set of experiments, the senders are at Cornell University, UFPR, USevilla, and SNU, and the receiver is an ADSL host in the east coast of the United States. The paths with the sender at UFPR, USevilla, and SNU have nonnegligible losses. We therefore only report the results of these three paths, which contain 15, 11, and 20 hops, respectively. Fig. 13(a)–(c) plots the inferred virtual queuing distributions of an experiment from UFPR, USevilla, and SNU, respectively, using $M = 5$, $N = 1, 2, 3$, and 4 (again, the inference results under different values of N are very similar). These three experiments started at 18:36 on June 21, 20:17 on June 21, and 13:46 on June 22, 2010, with the average loss rates of 0.1%, 0.7%, and 0.07%, respectively. In all the experiments, for $\theta = \phi = 0.05$, $D = 1$ is the minimum delay such that $F_W(D) > \theta$. For the experiments from UFPR and USevilla [Fig. 13(a) and (b)], $F_W(2D) = F_W(2) > (1-\theta)(1-\phi) \approx 0.90$, while for the experiment from SNU [Fig. 13(c)], $F_W(2D) = F_W(2) = 0.76 < 0.90$. Therefore, by WDCL-Test, we accept the null hypothesis that there exists a weakly dominant congested link with $\theta = 0.05$ and $\phi = 0.05$ along the former two paths, and we reject the null hypothesis for the last path. Results from *pchar* show a low bandwidth link close to the ADSL receiver. For the path from SNU, *pchar* also reveals a low bandwidth link in the middle (13th hop) of the path. These results are consistent with our identifications.

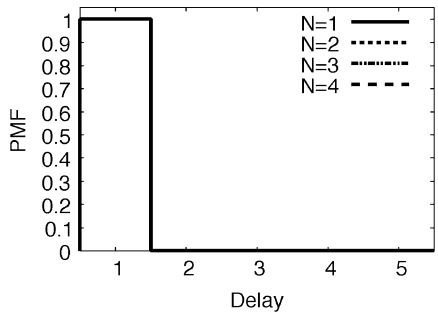


Fig. 12. Virtual queuing delay distribution of an experiment from Cornell University to UFPR (Brazil).

3) Effect of Probing Duration: The results above all use 20-min traces. We now vary the probing duration and investigate its impact on identification accuracy for the experiment from USevilla to the ADSL receiver (its average loss rate is 0.7%; the loss rates of the other experiments are too low). More specifically, we randomly choose a segment from the 20-min trace as a probing sequence to identify whether there exists a weakly dominant congested link with $\theta = \phi = 0.05$ using our model-based approach. When discretizing virtual queuing delays, we explore two cases: 1) using the minimum end–end delay in the probing sequence as the propagation delay D_0; and 2) using the minimum end–end delay in the entire trace (of 1 h, with over 10^5 probes) as D_0. We believe the latter case provides an estimate very close to the real D_0. Therefore, we refer to the former case as D_0 *unknown* and the latter as D_0 *known*. For each setting (choice of probing sequence, knowing or not knowing D_0), we check whether the identification result is consistent with that from the 20-min trace. We repeat the process 100 times and obtain the fraction of consistent identifications.

Fig. 14 plots the consistency ratio versus the probing duration for $M = 5$ and $N = 1$. We observe identical results when

[3]It is worth mentioning that the link bandwidth estimates from *pchar* (and other bandwidth estimation tools) alone cannot be used as a reliable basis for dominant congested link identification as explained in Section III-A.

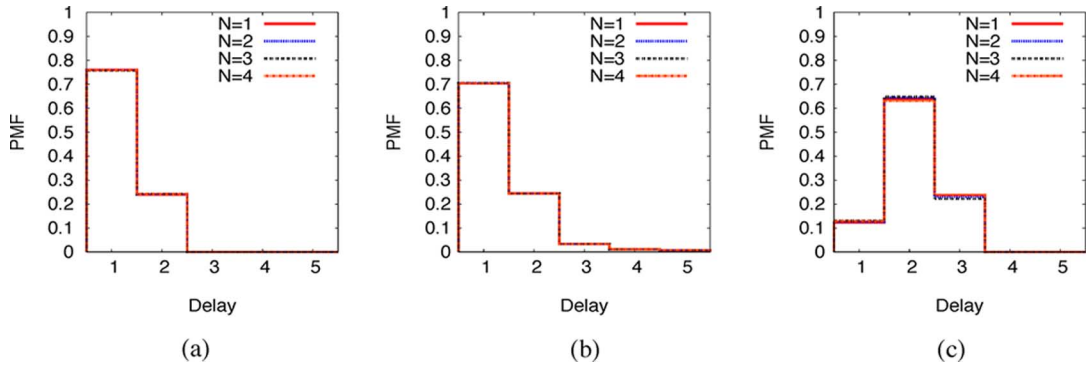


Fig. 13. Virtual queuing delay distributions of experiments with an ADSL host as the receiver. (a) Sender: UFPR. (b) Sender: USevilla. (c) Sender: SNU.

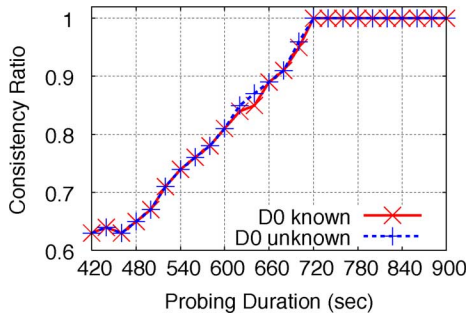


Fig. 14. Ratio of consistent identification of an experiment from USevilla, Spain, to a ADSL receiver in the U.S.; $M = 5, N = 1$.

knowing and not knowing D_0, indicating that using the minimum end–end delay in a probing sequence as D_0 provides good approximation. We also observe consistency ratio of 1 when the probing duration is above 12 min. This is longer than what we reported in [39] (where a probing duration of a few minutes suffices) due to the much lower average loss rate (the loss rate here is 0.7%, while the loss rate is above 4% in the experiments in [39]).

VII. Conclusion and Discussion

In this paper, we provided a formal yet intuitive definition of dominant congested link and proposed two simple hypothesis tests for identifying whether a dominant congested link exists along a path. We then developed a novel model-based approach for dominant congested link identification from one-way end–end measurements. Our validation in *ns* simulation and Internet experiments shows that the model-based approach requires only minutes of probing for accurate identification. As future work, we will investigate how to pinpoint a dominant congested link after identifying such a link exists.

Discussion: Our scheme assumes that the routers on an end–end path use droptail queues, and therefore a packet lost at a router "sees" a full queue at that router. When a router uses AQM, this assumption does not hold. As shown in Section VI-A5, our scheme may not provide correct identification in this case. For a path with a wireless link (e.g., a last-mile and/or first-mile IEEE 802.11 link), losses at this link can be due to interference and fading, which is not correlated with long queuing delays, and hence our approach does not apply.

Appendix A
Proof of Theorems 1 and 2

Before proving Theorem 1, we first prove the following lemma.

Lemma 1: If link k is a strongly dominant congested link, then $F_W(2Q_k) = 1$.

Proof: If k is a strongly dominant congested link, then virtual probe t satisfying $t \in L$ is lost at link k. This implies that it experiences the maximum queuing delay at link k. That is, $D_t^k = Q_k$. Therefore, the end–end queuing delay of this probe $D_t = Q_k + \sum_{i \neq k} D_t^i \in [Q_k, 2Q_k]$. Since the probe is arbitrary, we have $W \in [Q_k, 2Q_k]$. Hence, $F_W(2Q_k) = P(W \leq 2Q_k) = 1$.

We now prove Theorem 1. ■

Proof: If k is a strongly dominant congested link, then virtual probe t satisfying $t \in L$ experiences a queuing delay of Q_k at router k. Therefore, we have $W \geq Q_k$. Since D is the minimum delay value such that $F_W(D) > 0$, we have $D \geq Q_k$. Lemma 1 indicates that $F_W(2Q_k) = 1$. Since CDF $F_W(w)$ is a nondecreasing function, we have $F_W(2D) = 1$. ■

Before proving Theorem 2, we first prove the following lemma.

Lemma 2: If link k is a weakly dominant congested link with parameter θ and ϕ, then $F_W(2Q_k) \geq (1-\theta)(1-\phi)$.

Proof: For arbitrary virtual probe packet t, we have

$$
\begin{aligned}
&P(D_t \leq 2Q_k \,|\, t \in L) \\
&\quad = P(D_t \leq 2Q_k, D_t^k = Q_k \,|\, t \in L) \\
&\qquad + P(D_t \leq 2Q_k, D_t^k \neq Q_k \,|\, t \in L) \\
&\quad \geq P(D_t \leq 2Q_k, D_t^k = Q_k \,|\, t \in L) \\
&\quad = P(D_t \leq 2Q_k, t \in F_k \,|\, t \in L) \\
&\quad = P(t \in F_k \,|\, t \in L) P(D_t \leq 2Q_k \,|\, t \in F_k, t \in L) \\
&\quad \geq P(t \in L_k \,|\, t \in L) P(D_t \leq 2Q_k \,|\, t \in F_k, t \in L) \\
&\quad \geq (1-\theta) P(D_t \leq 2Q_k \,|\, t \in F_k, t \in L) \\
&\quad = (1-\theta) P\left(Q_k + \sum_{i \neq k} D_t^i \leq 2Q_k \,|\, t \in F_k, t \in L \right) \\
&\quad = (1-\theta) P\left(\sum_{i \neq k} D_t^i \leq Q_k \,|\, t \in F_k, t \in L \right)
\end{aligned}
$$

$$= (1-\theta)P\left(Q_k \geq \sum_{i\neq k} D_t^i \,|\, t \in F_k, t \in L\right)$$
$$= (1-\theta)P\left(D_t^k \geq \sum_{i\neq k} D_t^i \,|\, t \in F_k, t \in L\right)$$
$$\geq (1-\theta)(1-\phi).$$

The second inequality is because $L_k \subseteq F_k$; the third and the last inequality are due to the conditions on losses and delays for weakly dominant congested link, respectively. This implies $F_W(2Q_k) \geq (1-\theta)(1-\phi)$. ∎

We now prove Theorem 2.

Proof: We first prove $D \geq Q_k$ by contradiction. Suppose $D < Q_k$. Then, for an arbitrary virtual probe with loss mark that is sent at time t, i.e., $t \in L$

$$\begin{aligned}
&P(D_t \leq D \,|\, t \in L)\\
&\quad = P(D_t \leq D, t \in L_k \,|\, t \in L)\\
&\qquad + P(D_t \leq D, t \notin L_k \,|\, t \in L)\\
&\quad \leq P(D_t < Q_k, t \in L_k \,|\, t \in L)\\
&\qquad + P(D_t \leq D, t \notin L_k \,|\, t \in L)\\
&\quad = P(D_t \leq D, t \notin L_k \,|\, t \in L)\\
&\quad = P(D_t \leq D \,|\, t \notin L_k, t \in L) P(t \notin L_k \,|\, t \in L)\\
&\quad \leq P(t \notin L_k \,|\, t \in L)\\
&\quad = 1 - P(t \in L_k \,|\, t \in L)\\
&\quad \leq 1 - (1-\theta)\\
&\quad = \theta.
\end{aligned}$$

In the above, the first inequality is due to the assumption that $D < Q_k$, the second inequality is because the first probability is no more than 1, and the last inequality is due to the condition on losses for weakly dominant congested link. Since the probe is chosen arbitrarily from set L, we have $F_W(D) = P(W \leq D) \leq \theta$. However, by the definition of D, we have $F_W(D) > \theta$, a contradiction. Therefore, $D \geq Q_k$.

By Lemma 2, $F_W(2Q_k) \geq (1-\theta)(1-\phi)$. Since $F_W(w)$ is a nondecreasing function, we have $F_W(2D) \geq F_W(2Q_k) \geq (1-\theta)(1-\phi)$. ∎

APPENDIX B
EM ALGORITHM TO INFER λ OF MMHD

We next describe an EM algorithm to infer the parameter set, $\lambda = (P, \pi, s)$, from a sequence of T observations for MMHD. We first define several notations conforming to those used in [31]. Define $\alpha_t(i,j)$ to be the probability of the observation sequence up to time t and the state being in (i,j) at time t, given λ. That is

$$\alpha_t(i,j) = P(Y_1 = y_1, \ldots Y_t = y_t, Z_t = (i,j) \,|\, \lambda).$$

Define $\beta_t(i,j)$ to be the probability of the observation sequence from time $t+1$ to T, given state being in (i,j) at time t, given λ. That is

$$\beta_t(i,j) = P(Y_{t+1} = y_{t+1}, \ldots, Y_T = y_T \,|\, Z_t = (i,j), \lambda).$$

Define $\xi_t(i,j,k,l)$ to be the probability of state being in (i,j) at time t and in (k,l) at time $t+1$, given the observation sequence and λ. That is

$$\xi_t(i,j,k,l) = P(Z_t = (i,j), Z_{t+1} = (k,l) \mid Y_1 = y_1, \ldots, Y_t = y_t, \lambda).$$

Define $\gamma_t(i,j)$ to be the probability of being in state (i,j) at time t, given the observation sequence and λ. That is

$$\gamma_t(i,j) = P(Z_t = (i,j) | Y_1 = y_1, \ldots, Y_T = y_T, \lambda).$$

We derive $\xi_t(i,j,k,l)$ from $\alpha_t(i,j)$ and $\beta_{t+1}(k,l)$ as

$$\xi_t(i,j,k,l) = \frac{\alpha_t(i,j) p_{(i,j)(k,l)} \beta_{t+1}(k,l)}{\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^N \sum_{l=1}^M \alpha_t(i,j) p_{(i,j)(k,l)} \beta_{t+1}(k,l)}.$$

Observe that γ_t can be calculated from ξ_t as

$$\gamma_t(i,j) = \sum_{k=1}^N \sum_{l=1}^M \xi_t(i,j,k,l).$$

The EM algorithm is an iterative algorithm. Each iteration consists of two steps: the expectation step and the maximization step. During the expectation step, we compute the expected number of transitions from state (i,j), and the expected number of transitions from state (i,j) to state (k,l) using the model parameters obtained during the previous iteration. We also compute the expected number of times that a loss observation has delay symbol of j, and the expected number of symbol j. During the maximization step, we calculate a set of new model parameters from the expected values obtained from the expectation step. The iteration terminates when the difference between the parameters of the new model and the previous model lies below a certain convergence threshold.

A. Expectation Step

Without loss of generality, we assume y_1 and y_T are not losses. In the expectation step, we first calculate α and β using the procedures referred to as forward and backward steps, respectively [31]. The procedure to calculate $\alpha_t(i,j)$, where $t = 1, \ldots, T, i = 1, \ldots, N, j = 1, \ldots, M$, consists of the following steps.

1) Initialization

$$\alpha_1(i,j) = \begin{cases} \pi(i, y_1), & j = y_1 \\ 0, & j \neq y_1 \end{cases}.$$

2) Induction

$$\alpha_{t+1}(i,j) = \begin{cases} \sum_{k=1}^N \sum_{l=1}^M \alpha_t(k,l) p_{(k,l)(i,j)} s(j), & y_{t+1} = * \\ \sum_{k=1}^N \sum_{l=1}^M \alpha_t(k,l) p_{(k,l)(i,j)}, & y_{t+1} = j \\ 0, & \text{o.w.} \end{cases}$$

where $t = 1, 2, \ldots, T-1$.

The procedure to calculate $\beta_t(i,j)$, where $t = 1, \ldots, T, i = 1, \ldots, N, j = 1, \ldots, M$, contains the following steps.

1) Initialization

$$\beta_T(i,j) = \begin{cases} 1, & j = y_T \\ 0, & j \neq y_T \end{cases}.$$

2) Induction

$$\beta_t(i,j) = \begin{cases} 0, & y_t \neq *, j \neq y_t \\ \sum_{k=1}^{N} \sum_{l=1}^{M} p_{(i,j)(k,l)} \beta_{t+1}(k,l), & \text{o.w.} \end{cases}$$

where $t = T-1, T-2, \ldots, 1$.

After obtaining α and β, we calculate ξ and γ as shown before. Afterwards, we calculate the various expectations using ξ and γ, which is omitted here and can be found in the computation in the maximization step.

B. Maximization Step

The new model parameters are obtained in the maximization step as

$$\hat{\pi}_{(i,j)} = \gamma_1(i,j) \tag{6}$$

$$\hat{p}_{(i,j)(k,l)} = \frac{\text{expected number of transitions from } (i,j) \text{ to } (k,l)}{\text{expected number of transitions from } (i,j)} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j,k,l)}{\sum_{t=1}^{T-1} \gamma_t(i,j)} \tag{7}$$

$$\hat{s}(j) = \frac{\text{expected number of times that a loss has delay of } j}{\text{expected number of delay} j} = \frac{\sum_{t=1}^{T} \mathbf{1}(y_t = *) \sum_{i=1}^{N} \gamma_t(i,j)}{\sum_{t=1}^{T} \sum_{i=1}^{N} \gamma_t(i,j)}. \tag{8}$$

ACKNOWLEDGMENT

The authors thank S. Sen for helpful discussions; L. Golubchik, C. Papadopoulos, and E. A. de Souza e Silva for providing accounts for the Internet experiments; and X. Chen for his help with experiments in PlanetLab. They also thank the anonymous reviewers for their insightful comments, and C. Dovrolis for handling the paper. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

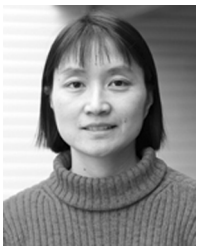
REFERENCES

[1] "tcpdump," [Online]. Available: http://www.tcpdump.org/
[2] A. Adams, T. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. L. Presti, S. Moon, V. Paxson, and D. Towsley, "The use of end-to-end multicast measurements for characterizing internal network behavior," *IEEE Commun. Mag.*, vol. 38, no. 5, pp. 152–159, May 2000.
[3] A. Akella, S. Seshan, and A. Shaikh, "An empirical evaluation of wide-area Internet bottlenecks," in *Proc. ACM SIGCOMM IMC*, Oct. 2003, pp. 101–114.
[4] K. G. Anagnostakis, M. B. Greenwald, and R. S. Ryger, "Cing: Measuring network-internal delays using only existing infrastructure," in *Proc. IEEE INFOCOM*, Apr. 2003, vol. 3, pp. 2112–2121.
[5] A. Batsakis, T. Malik, and A. Terzis, "Practical passive lossy link inference," in *Proc. PAM*, Mar.–Apr. 2005, pp. 362–367.
[6] M. Coates, A. O. Hero III, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Process. Mag.*, vol. 19, no. 3, pp. 47–65, May 2002.
[7] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?," in *Proc. IEEE INFOCOM*, Apr. 2001, vol. 2, pp. 905–914.
[8] A. B. Downey, "Using pathchar to estimate internet link characteristics," in *Proc. ACM SIGCOMM*, Aug. 1999, pp. 241–250.
[9] N. Duffield, "Network tomography of binary network performance characteristics," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5373–5388, Dec. 2006.
[10] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of REDs active queue management," Tech. rep., Aug. 2001. [Online]. Available: http://www.icir.org/floyd/papers/adaptiveRed.pdf
[11] K. Harfoush, A. Bestavros, and J. Byers, "Measuring bottleneck bandwidth of targeted path segments," in *Proc. IEEE INFOCOM*, Apr. 2001, vol. 3, pp. 2079–2089.
[12] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating Internet bottlenecks: Algorithms, measurements, and implications," in *Proc. ACM SIGCOMM*, Aug. 2004, pp. 41–54.
[13] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 6, pp. 879–894, Aug. 2003.
[14] V. Jacobson, "Pathchar—A tool to infer characteristics of Internet paths," Apr. 1997 [Online]. Available: ftp://ftp.ee.lbl.gov/pathchar
[15] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," in *Proc. ACM SIGCOMM*, Aug. 2002, pp. 295–308.
[16] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," in *Proc. PAM*, Mar. 2002, pp. 14–25.
[17] D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlenecks," in *Proc. ICCCN*, Oct. 2001, pp. 174–181.
[18] S. Katti, D. Katabi, C. Blake, E. Kohler, and J. Strauss, "MultiQ: Automated detection of multiple bottleneck capacities along a path," in *Proc. ACM SIGCOMM IMC*, Oct. 2004, pp. 245–250.
[19] M. S. Kim, T. Kim, Y. Shin, S. S. Lam, and E. J. Powers, "A wavelet-based approach to detect shared congestion," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 763–776, Aug. 2008.
[20] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," in *Proc. ACM SIGCOMM*, Aug. 2000, pp. 283–294.
[21] J. Liu and M. Crovella, "Using loss pairs to discover network properties," in *Proc. ACM SIGCOMM Internet Meas. Workshop*, Nov. 2001, pp. 127–138.
[22] J. Liu, I. Matta, and M. Crovella, "End-to-end inference of loss nature in a hybrid wired/wireless environment," in *Proc. WiOpt*, Mar. 2003.
[23] B. A. Mah, "pchar: A tool for measuring Internet path characteristics," 2005. [Online]. Available: http://www.kitchenlab.org/www/bmah/Software/pchar/
[24] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "User-level Internet path diagnosis," in *Proc. ACM SOSP*, Oct. 2003, pp. 106–119.
[25] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," in *Proc. IEEE GLOBECOM*, Nov. 2000, vol. 1, pp. 415–420.
[26] H. X. Nguyen and P. Thiran, "The boolean solution to the congested IP link location problem: Theory and practice," in *Proc. IEEE INFOCOM*, May 2007, pp. 2117–2125.
[27] H. X. Nguyen and P. Thiran, "Network loss inference with second order statistics of end-to-end flows," in *Proc. ACM SIGCOMM IMC*, Oct. 2007, pp. 227–240.
[28] V. N. Padmanabhan, L. Qiu, and H. J. Wang, "Server-based inference of Internet link lossiness," in *Proc. IEEE INFOCOM*, Mar.–Apr. 2003, vol. 1, pp. 145–155.
[29] V. Paxson, "End-to-end internet packet dynamics," *IEEE/ACM Trans. Netw.*, vol. 7, no. 3, pp. 277–292, Jun. 1999.
[30] "PlanetLab," [Online]. Available: http://www.planet-lab.org
[31] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.
[32] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "Pathchirp: Efficient available bandwidth estimation for network paths," in *Proc. PAM*, Apr. 2003.
[33] D. Rubenstein, J. Kurose, and D. Towsley, "Detecting shared congestion of flows via end-to-end measurement," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 381–395, Jun. 2002.
[34] K. Salamatian and S. Vaton, "Hidden Markov modelling for network communication channels," in *Proc. ACM SIGMETRICS*, Jun. 2001, pp. 92–101.
[35] A. Shriram and J. Kaur, "Empirical evaluation of techniques for measuring available bandwidth," in *Proc. IEEE INFOCOM*, May 2007, pp. 2162–2170.
[36] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proc. ACM SIGCOMM IMC*, Oct. 2003, pp. 39–44.

[37] A. Tachibana, S. Anoa, T. Hasegawa, M. Tsurub, and Y. Oie, "Locating congested segments over the internet by clustering the delay performance of multiple paths," *Comput. Commun.*, vol. 32, no. 15, pp. 1642–1654, Sep. 2009.
[38] W. Wei, B. Wang, and D. Towsley, "Continuous-time hidden Markov models for network performance evaluation," *Perform. Eval.*, vol. 49, no. 1–4, pp. 129–146, Sep. 2002.
[39] W. Wei, B. Wang, D. Towsley, and J. Kurose, "Model-based identification of dominant congested links," in *Proc. ACM SIGCOMM IMC*, Oct. 2003, pp. 115–128.
[40] L. Zhang, Z. Liu, and C. Xia, "Clock synchronization algorithms for network measurements," in *Proc. IEEE INFOCOM*, Jun. 2002, vol. 1, pp. 160–169.
[41] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proc. ACM SIGCOMM Internet Meas. Workshop*, Nov. 2001, pp. 197–211.
[42] Y. Zhao, Y. Chen, and D. Bindel, "Towards unbiased end-to-end network diagnosis," in *Proc. ACM SIGCOMM*, Sep. 2006, pp. 219–230.

Wei Wei (S'02–M'06) received the B.S. degree in applied mathematics from Beijing University, Beijing, China, in 1992; the M.S. degree in statistics from Texas A&M University, College Station, in 2000; and the M.S. degrees in computer science and applied mathematics and Ph.D. degree in computer science from the University of Massachusetts Amherst, in 2004 and 2006, respectively.

From November 2006 to August 2008, he was a Senior Research Engineer with the United Technologies Research Center, East Hartford, CT. From August 2008 to June 2009, he was a Visiting Assistant Professor with the Computer Science and Engineering Department, University of Connecticut, Storrs. He is currently a Senior Post-Doctoral Associate with the Department of Computer Science, University of Massachusetts Amherst. His research interests are in the areas of computer networks, distributed embedded systems, and performance modeling.



Bing Wang (M'02) received the B.S. degree in computer science from Nanjing University of Science and Technology, Nanjing, China, in 1994; the M.S. degree in computer engineering from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1997; and the M.S. degrees in computer science and applied mathematics and Ph.D. degree in computer science from the University of Massachusetts Amherst, in 2000, 2004, and 2005, respectively.

Afterwards, she joined the Computer Science and Engineering Department, University of Connecticut, Storrs, as an Assistant Professor. Her research interests are in computer networks, multimedia, and distributed systems.

Dr. Wang received the National Science Foundation (NSF) CAREER Award in 2008.

Don Towsley (M'78–SM'93–F'95) received the B.A. degree in physics and the Ph.D. in computer science from the University of Texas at Austin in 1971 and 1975, respectively.

From 1976 to 1985, he was a member of the faculty of the Department of Electrical and Computer Engineering, University of Massachusetts Amherst. He is currently a Distinguished Professor with the Department of Computer Science, University of Massachusetts Amherst. He has held visiting positions at the IBM T. J. Watson Research Center, Yorktown Heights, NY; Laboratoire MASI, Paris, France; INRIA, Sophia-Antipolis, France; AT&T Labs—Research, Florham Park, NJ; and Microsoft Research Laboratories, Cambridge, U.K. His research interests include networks and performance evaluation.

Prof. Towsley is a Fellow of the Association of Computing Machinery (ACM) and a Member of ORSA. He served as Editor-in-Chief of the IEEE/ACM TRANSACTIONS ON NETWORKING, serves on the Editorial Boards of the *Journal of the ACM* and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and has previously served on numerous other Editorial Boards. He was Program Co-Chair of the joint ACM SIGMETRICS and PERFORMANCE 1992 conference and the Performance 2002 conference. He is Chair of IFIP Working Group 7.3. He has received the 2007 IEEE Koji Kobayashi Award, the 2007 ACM SIGMETRICS Achievement Award, the 2008 ACM SIGCOMM Achievement Award, the 1998 IEEE Communications Society William Bennett Best Paper Award, and numerous best conference/workshop paper awards.

Jim Kurose (S'81–M'84–SM'91–F'97) received the Ph.D. degree in computer science from Columbia University, New York, NY, in 1984.

He is currently a Distinguished University Professor with the Department of Computer Science, University of Massachusetts Amherst. He has been a Visiting Scientist with IBM Research, INRIA, Institut EURECOM, the University of Paris, LIP6, and Thomson Research Labs. With Keith Ross, he is the coauthor of the textbook *Computer Networking: A Top–Down Approach*, 4th ed. (Addison-Wesley, 2007). His research interests include network protocols and architecture, network measurement, sensor networks, multimedia communication, and modeling and performance evaluation.

Prof. Kurose has served as Editor-in-Chief of the IEEE TRANSACTIONS ON COMMUNICATIONS and was the founding Editor-in-Chief of the IEEE/ACM TRANSACTIONS ON NETWORKING. He has been active in the program committees for IEEE INFOCOM, ACM SIGCOMM, and ACM SIGMETRICS for a number of years, and has served as Technical Program Co-Chair for these conferences. He has received a number of awards for his educational activities, including the IEEE Taylor Booth Education Medal.