

A Measurement-Friendly Network (MFN) Architecture

Sridhar Machiraju^{*}
Sprint Advanced Technology Labs (ATL)
Burlingame, California, U.S.A
Machiraju@sprint.com

Darryl Veitch
ARC Special Research Centre on Ultra-Broadband
Information Networks, CUBIN is an affiliated
program of National ICT Australia (NICTA)
Dept. of Electrical and Electronic Engineering
University of Melbourne, Australia
d.veitch@ee.unimelb.edu.au

ABSTRACT

Using active techniques to measure networks, that is by injecting probe packets, has proved to be quite challenging for properties beyond simple end-to-end delay and loss. Some of the greatest difficulties have resulted from our inability to design techniques robust to multi-hop queueing effects. This difficulty is only compounded by the need to keep measurements non-intrusive, that is to minimally affect ongoing data flows. In this paper, we show that novel network primitives based on hop-dependent priority queueing are very effective in addressing these challenges. By enabling these primitives, network operators can perform a variety of active measurements accurately. Such measurement-friendliness results from many factors including ease of applying fundamentally single-hop methods, better measurement capabilities, and easier clock synchronization. Other advantages of our architecture include ease of deployment, simplicity, low overhead and generality, i.e., no constraints on scheduling policies for data packets. We also discuss the challenges faced, for example, in coping with small but unavoidable inaccuracies and with exposing the primitives to end-users.

Categories and Subject Descriptors

C.2.3 [Computer-Communication Networks]: Network management; C.4 [Performance of Systems]: Measurement techniques

General Terms

Design, Management, Measurement, Performance.

Keywords

Active Measurement, Measurement-Friendly Network, MFN, Probing, Priority Queueing.

^{*}When this work was done, the author was also a student at UC, Berkeley.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCOMM'06 Workshops September 11-15, 2006, Pisa, Italy.
Copyright 2006 ACM 1-59593-417-0/06/0009 ...\$5.00.

1. INTRODUCTION

In recent years, the shortcomings of the current Internet architecture in allowing accurate measurements have become painfully obvious. Much work has been done on developing passive measurement infrastructures that use appropriate data collection mechanisms at some or all routers [22]. *Active measurements*, which use the delays and losses experienced by actively-injected probe packets, are an alternative way of measuring networks. However, they have traditionally [22] been viewed as inaccurate for all but the simplest tasks and mostly for end-users who cannot access measurement data collected at routers.

In this paper, we seek to understand if and how the accuracy of active measurements can be improved. Most prior works are focused on designing active techniques under the current network architecture. We investigate the *converse* problem - how can existing networks be converted into *Measurement Friendly Networks (MFNs)*, i.e., networks in which a variety of active measurements can be accurately performed. Unlike some previously-proposed network primitives [13, 16] that essentially allow probe packets to access clock information from individual routers, our goal is to use forwarding primitives based on the data plane. We focus on the estimation of end-to-end and per-hop performance metrics of unicast paths. These include delays [18], losses [20], queueing delays [5], busy periods [7], cross-traffic [15], capacities [5, 11, 17] and available bandwidth [4, 9, 21].

We start by using prior works to understand why it has been challenging to obtain accurate estimates via active measurements. For all but directly observable end-to-end metrics, the most difficult challenges arise from multi-hop queueing effects. For instance, IP-layer capacity estimation techniques such as Capprobe [10] need to send multiple pairs of back-to-back probes hoping that at least one of the pairs encounters no queueing from cross-traffic at *all* hops. Most available bandwidth techniques (for example, [9, 21]) and delay-based congestion control algorithms such as TCP Vegas [4] are also affected by multi-hop queueing effects [12]. Such effects also make it difficult for techniques which are fundamentally based on single-hop models or heuristics, to make the transition to multi-hop paths. This is fundamentally due to the fact that careful probe design is disturbed or even thwarted by queueing effects over the upstream nodes, and over the downstream passage to the receiver, the information encoded in probe delays is distorted in a highly non-linear fashion, or even entirely wiped out. Examples where this is important include techniques to estimate per-hop

round trip times (traceroute), capacities [5, 11, 17], queueing delays [5], cross-traffic [15], and bottleneck locations [8]. Robustness to multi-hop queueing effects is only complicated by the need to keep measurements as non-intrusive as possible. Recent work [12] showed such a trade-off for the case of available bandwidth estimation.

Motivated by the above discussion, we investigate how robustness to multi-hop queueing and non-intrusiveness can be addressed. We show that this can be naturally achieved if measurement packets are endowed with hop-dependent priorities (data traffic has normal priority). Priorities higher than normal priority allow probes to bypass undesirable queueing effects while probes with lower priority (than normal packets) are not only non-intrusive, but also intrinsically measure the normal data queue because they are transmitted only when the latter is empty. We use hop-dependent priorities to design our MFN architecture and show that it enables a wide range of accurate per-hop and end-to-end active measurements.

Our MFN architecture simplifies network management by making it possible to exploit the unique advantages of active measurements, namely, their ability to directly measure what normal data packets experience and ease of deployment/use, without compromising measurement accuracy. Moreover, our architecture is quite general because it does not constrain the scheduling policies of normal data packets. One limitation is that non-preemption and cross-traffic persistence results in measurement inaccuracies and biases which remain inherently difficult to counter. However, we derive upper bounds on these biases and show that, often, they can be overcome or ignored. We also discuss additional questions related to exposing our proposed primitives to end-users and multiple simultaneous measurements.

This paper is organized as follows. In Section 2, we discuss our goals and assumptions, motivate hop-dependent priority queueing, and provide an overview of our proposed MFN architecture. In Section 3, we describe how various metrics can be measured in our proposed MFN architecture, and discuss how non-preemption and cross-traffic persistence generate measurement errors. Important issues related to practical deployment are discussed in Section 4. We summarize and present future directions in Section 5.

2. OVERVIEW

In this section, we state our goals and assumptions. Then, we show how priority queueing is a natural choice for addressing the main challenges of active measurements. We end by providing an overview of our MFN architecture.

2.1 Goals and Assumptions

In the previous section, we used prior work to motivate the need for mechanisms that address issues related to multi-hop queueing effects while controlling intrusiveness. Prior works (for example, [5, 8, 17]) have exploited TTL-expiry and ICMP generation to deal with the multi-hop nature of paths. Since the original rationale behind these primitives was network stability and error reporting, their utility is limited. Mahajan, et al. [16] suggested augmenting the timestamping mechanisms in ICMP for better user-level (fault diagnosis) measurements. Such mechanisms essentially provide access to data collected at routers and do not completely address the effect of multi-hop queueing on estimation bias. Luckie, et al. [13] proposed similar mechanisms

in a new measurement protocol, IPMP. Approaches such as these allow routers to provide more information to probe packets. In contrast, we investigate dataplane mechanisms that only forward measurement packets differently from normal data packets.

Network architecture design necessarily assumes a willingness for change. In recent years, security problems and competitive concerns have caused network operators to make their networks less transparent by, for instance, filtering ICMP messages. Yet, MFN architectures, which are more transparent, are of practical interest if they allow network operators to better measure their own networks. Moreover, the push for reduced transparency has not been universal especially in non-commercial networks. End-user access to better measurement primitives are also of interest to operators provided they can control access to them, and they are non-intrusive.

We assume that all the queues are visible at the IP layer. This is not true with networks that use MPLS. The impact of MPLS and similar technologies is out of the scope of this paper. However, given the use of non-FIFO schedulers today, we do not desire, and do not assume, any restriction on the scheduling of normal data traffic. The metrics that we consider are end-to-end and per-hop. They measure queueing delays, minimum delay, jitter, busy period durations, losses, capacities, cross-traffic and available bandwidth. Moments of these metrics, detailed statistics and even joint statistics are of interest. Later, we briefly comment on network-wide metrics such as traffic matrices and finer metrics such as flow size distributions.

2.2 Design Motivation

We consider multi-hop queueing effects and non-intrusiveness, and develop network primitives based on priority queueing that naturally address these two challenges.

Consider the problem of estimating the queue sizes seen by normal data packets on the hops of a multi-hop path. For a single hop, the delay of a probe is proportional to the queue size of that hop. However, on a multi-hop path, the end-to-end delay includes the sum of the queue sizes over all hops. Hence, the single-hop technique based on end-to-end delay is not easily extended to multi-hop paths. Even if a multi-hop path contains a single predominant bottleneck (with much larger queues than the other hops), the noise from the other hops cannot be quantified. The single-hop technique is applicable, though, if probe packets experience queueing delay only at a single, specified hop h of the path. This is possible if probes are treated with higher priority than data traffic at all hops except h . In other words, such hop-dependent priority queueing endows measurement packets with a *hop isolation property*.

Next, consider the challenge of non-intrusive measurements. By definition, non-intrusiveness is achieved if probes do not cause the forwarding of normal data packets to be affected. This motivates a priority-queueing primitive opposite to the one above, namely, assigning lower priority to probes. Low priority packets possess what we refer to as the *inherent measurement property*. This is because a low priority packet is transmitted only when there is no normal priority packet and therefore, its transmission time correspond to the onset of what would have been idle periods of the normal data queue of the unperturbed system (without probes).

2.3 Measurement-Friendly Network Architecture

The above discussion motivates us to use hop-dependent low and high priority queueing for our MFN architecture. Data traffic is considered to have normal priority and hence, data packets are also called N-packets. Measurement packets may be treated with high, normal or low priority at *each* hop and the priority at different hops need not be the same. Thus, at each hop, a probe packet is a H-probe, N-probe or L-probe respectively. We use a string of letters from $\{L, H, N\}$ to denote the priorities of measurement packets along a path. The first letter indicates the default priority and uses the subscript d to denote this. The rest of the letters apply to individual hops, specified by a subscript, and represent exceptions to the default. For example, $H_d N_2 L_5$ denotes that a measurement packet has high priority at all hops except hop 2 and 5 where it has normal and low priority respectively. We use priority queueing in the *strict* sense, namely that packet transmission follows the order H, N, L, and packet dropping occurs in the order L, N, H.

3. MEASUREMENT TECHNIQUES

In this section, we investigate what can be measured using one of high, normal or low priority at a hop h and (applying the hop isolation property) using high priority at every other hop. These simple scenarios provide insights into how more complicated measurements can be performed. We assume a single measurement entity and the ability to signal hop-dependent priorities. For this section, we use *ns-2*[1] to simulate the topologies shown in Figure 1. Cross-traffic arrives at Poisson epochs. By default, we use a packet-size distribution that is trimodal with the three modes located at 40, 576 and 1500 bytes.

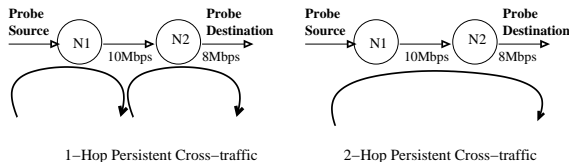


Figure 1: The two-hop systems we use - 1-hop (left) and 2-hop persistent (right) cross-traffic.

3.1 Minimum End-to-End Delay

Consider a H_d -probe in our MFN architecture. Such a probe is given high priority by *all* hops, and hence should not see any queueing from normal data traffic (a caveat to this is discussed shortly). Since we are assuming a single measurement entity, such probes would not see any queueing from other H_d -probes as long as they are widely spaced. Thus, H_d -probes of size p can be used to estimate the minimum end-to-end delay d along a path, which is the sum of the propagation time (D_i) and probe size dependent transmission time (p/C_i) over all hops:

$$d(p) = \sum_i D_i + p \sum_i 1/C_i \quad (1)$$

In the current network architecture, large numbers of probes may be needed to estimate $d(p)$ reliably, and it cannot be observed at all if one or more hops are persistently congested.

The ability of H_d -probes to access minimum delays has many uses (see Table 1), both in a per-hop sense as we describe later, and for paths. An important one is clock synchronization. Synchronization algorithms rely on successfully filtering out one-way delay and/or round-trip variability, which effectively distorts the timestamps received over the network from the reference clock. By eliminating this variability, H_d -probes will greatly enhance synchronization accuracy and robustness [23]. Given the pervasive need for clock synchronization, network operators could provide it as a service to customers in our MFN architecture.

Another important application is in delay-based TCP, where the accuracy of estimates of minimum delays directly influences the stability and performance of the flow control. A “TCP friendly” network service could be provided whereby each flow would be allowed (from time to time only, to avoid abuse) to use a H_d probe to access minimum delays. Participation would be optional and the benefits independent of other users.

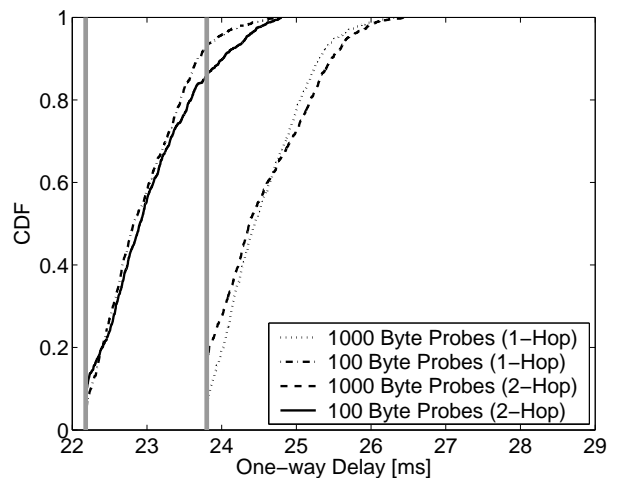


Figure 2: CDFs of end-to-end delay of H_d -probes of size 100 and 1000 bytes under 1-hop and 2-hop persistent cross-traffic scenarios. Gray vertical bars are the true minimum end-to-end delay.

In Figure 2, we plot the CDF of delays experienced by H_d -probes along the two-hop paths of Figure 1. The four distributions clearly separate into two groups according to the p dependence from Equation 1. However, there are two important additional observations which we now expand on: not all probes experience the minimum delay, and the CDF is influenced by the persistence properties of the cross-traffic.

3.1.1 Non-Preemption

Although the spread of values for each CDF in Figure 2 is narrow, clearly not all packets experience the minimum delay. The reason for this is *non-preemption*: at each hop, any packet already being serialized (if any) must complete its transmission before the H_d -probe can begin its own, causing the probe delays to suffer a non-negative noise component, ϵ . Non-preemption also causes in-transmission L-packets to be intrusive by delaying N-packets.

For a H_d -probe on a n -hop path, non-preemption noise will be 0 if the probe packet encounters no data packets on any hop. An upper bound corresponds to a probe arriv-

ing just after the largest packet (of size p_{max}) at each hop. Hence ϵ takes values in

$$\epsilon \in \left[0, \frac{p_{max}}{C_1} + \dots + \frac{p_{max}}{C_n} \right]. \quad (2)$$

Thus, noise is mostly due to the lowest-speed links on a path. For instance, assuming a maximum packet size of 1500 bytes, a 100Mbps link would contribute noise up to $120\mu s$ whereas a 2.4Gbps link would contribute less than $5\mu s$. Details of the probability distribution depends on various factors including cross-traffic packet sizes, arrival processes and persistence.

There are two important points here: First, the range of the noise is bounded, and the bound could be roughly estimated in practice reasonably easily. Second, the noise range in practice will in general be much lower than the bound, and will be far narrower than typical delays in the current architecture, where the full queue size is summed over each hop, rather than simply residuals of (at most) a single packet at each. Although the probability of actually achieving the minimum delay is in fact essentially the same in both architectures, the greatly reduced noise range when using H_d -probes should enable minimum delay estimates to be reliably obtained using very few probes.

3.1.2 Cross-traffic Persistence

Our second observation is that the CDFs in Figure 2, generated using the same path, capacities, probe stream and cross traffic rate, depend on the cross-traffic topology, that is, on cross-traffic persistence. Consider the probability of observing zero non-preemption noise on a path. With 1-hop persistent cross-traffic, cross-traffic arrival processes at consecutive hops are independent. Hence, this probability is given by $\Pi_h(1 - \rho_h)$ where ρ_h is the utilization of that hop. With persistent cross-traffic, however, the queue occupancy at consecutive hops is dependent. For example, if a probe finds the first hop empty, it is more likely to find subsequent ones empty, increasing the probability of seeing the minimum delay, in agreement with the results of Figure 2 for both the $p = 100$ and $p = 1000$ groups. Moreover, this dependence causes ϵ to be dependent on probe size. To see why, consider our 2-hop persistent topology. Assume that the second hop has an empty queue, and the first hop is transmitting a 576-byte data packet which delays a H_d -probe just arriving. The latter will be delayed by the same data packet at the second hop, to an extent depending on its size (and the ratio of the capacities of the two hops). The resulting dependence is easily observed in Figure 2 where the CDFs corresponding to the two 1-hop persistent cases are identical up to a simple horizontal translation (corresponding to the difference in probe transmission time), whereas in the 2-hop persistent case the CDF shape actually changes with packet size.

3.2 Per-hop Queueing

Consider sending a H_dN_h -probe. This probe experiences normal queueing delay at hop h and experiences only non-preemption noise at other hops where it has high priority. Thus, these probes allow queueing delays at individual hops to be measured (see Table 1).

In Figure 3, we plot the true and measured CDFs of queue sizes at the second hop of the topologies shown in Figure 1. For reasons to be explained soon, we use 1500 byte cross-traffic packets only. To calculate the queue sizes encoun-

tered by probes at the second hop, we took a set of delay measurements using H_dN_2 -probes, and subtracted the minimum value found over the set from each member. We used the Ground Truth Calculator (GTC) [2, 14] to calculate the true CDFs of queue size. The true and estimated CDFs differ from each other. Suspecting non-preemption to be the reason, we plot the “noisy” versions of the true CDFs too. These noisy versions are obtained by convolving the true CDF with the non-preemption noise at the first hop given by a uniform distribution between 0 and the transmission time of a 1500-byte packet. We see that the noisy CDF coincides with the measured CDF in the 1-hop persistent case, confirming as we expect that, up to non-preemption noise, the queue size at the second hop can be extracted using such probes in this case.

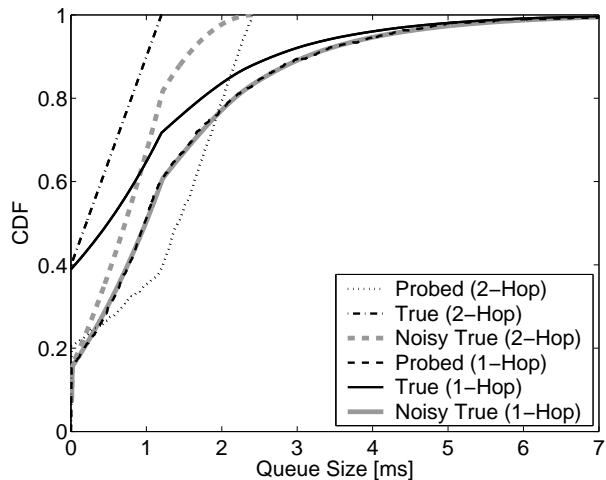


Figure 3: CDFs of estimated queue sizes at the second hop using H_dN_2 -probes under 1-hop and 2-hop persistent cross-traffic. The gray curves represent the true CDFs along with non-preemption noise.

With 2-hop persistent cross-traffic, the noisy version and true CDF do not coincide, because of the inter-hop dependence. Formally, if $Q_h(\cdot)$ represents the time-dependent queue size of hop h , then we would like our H_dN_h probes to reach hop h at times T_i . In practice, non-preemption causes the probes to reach hop h at different times and also, adds a non-negative error to the observed delay. Thus, we use the following approximation -

$$Q_h(T_i) \approx Q_h(T_i + \epsilon_{h,1}) + \epsilon_{h,1} + \epsilon_{h,2} \quad (3)$$

where $\epsilon_{h,1}$ is the non-preemption noise until and excluding hop h and $\epsilon_{h,2}$ is the noise from hop $h + 1$ onwards. Each of these have a range of the form of Equation 2. The $Q_h(\cdot)$ terms will have the same distribution if cross-traffic is 1-hop persistent in which case only non-preemption noise is an issue. With k -hop persistent cross-traffic however, an *unavoidable sampling bias* results. Persistence *and* non-preemption together cause this bias. Neither of them alone can cause it.

By how much do $Q_h(T_i)$ and $Q_h(T_i + \epsilon_1)$ differ? Consider a two-hop path for which ϵ_1 is p_{max}/C_1 . The only 2-hop persistent cross-traffic that can arrive at the second hop in $(T, T + \epsilon_1)$ is the in-transmission packet at the first hop. Thus, assuming that all cross-traffic at the second

Metric	MFN Method	Brief Description
Minimum end-to-end delay	H_d -probes	Isolate queueing delays, e.g., TCP Vegas [4]. Eliminate path variability: simplifies clock synchronization.
Per-hop delay	H_dN_h -probes	Per-hop congestion information. Routing decisions by operator and end-users.
Per-hop “jitter”	H_dN_h -pair	Probes separated by time t . Determine if VoIP, etc., can be routed through hop.
Cross-traffic [15]	H_dN_h -pair	Ideal for methods not requiring fine timing, e.g., cross-traffic CDF [15]
Capacities [5, 10]	H_dN_h -probes	For narrow-link estimation, e.g., [10, 11, 17]
Per-hop (remaining) busy period [7]	H_dL_h -probes	Indicator of cross-traffic “burstiness”, i.e., how long congestion lasts. MFN method measures the <i>remaining</i> duration of busy period.
Per-hop busy period [7]	$2 H_dL_h$ -probes and a H_dN_h -probe	Inter-probe times t_1 and t_2 . If H_dN_h -probe is received before, between or after other two, busy period duration is in $(t_1 + t_2, \infty)$, $[t_1, t_1 + t_2)$ or $[0, t_1)$ respectively.
Losses	H_dN_h -probes H_dL_h -probes	Loss rate at hop h using, say, [20]. Receipt of H_dL_h -probes implies hop h not continuously congested. Useful with equation-based congestion control [6].
End-user perceived busy period	$2 N_dL_h$ -probes and a N_d -probe	Same as above except reordering tells us if packets sent t units apart would share busy period at hop h . Indicates if new packet can be sent without causing more congestion.
Worst-case congestion	L_d -probes	Completely non-intrusive. Delays are sum of remaining busy period duration at all hops. Receipt of probes indicates no continuously congested hop.
Available bandwidth [9]	L_d -probes	Use output rate for non-intrusive estimation of available bandwidth. Applicable for bottleneck detection [8]. Similar to [19].

Table 1: Some illustrative measurement methods in our MFN Architecture split into three classes. The first shows how single-hop methods can be used with multi-hop paths. The second show how low-priority queueing at a hop inherently measures it. The third shows what can be achieved *without* high-priority anywhere.

hop is 2-hop persistent, the difference between $Q_h(T_i)$ and $Q_h(T_i + \epsilon_1)$ is not more than the transmission time of a maximum-sized packet; This is confirmed in Figure 3 in which the true and probed CDFs are shifted only by 1.2ms, the transmission time of a 1500-byte packet. On a 3-hop path, a probe, which is delayed by an in-transmission 3-hop persistent cross-traffic packet, can get transmitted before that cross-traffic packet at the second hop especially if it is small. Thus, in-transmission packets at one hop can be effectively preempted at a later hop and the bias need not accumulate. We believe that the resulting inaccuracies due to persistence, for per-hop queueing and other per-hop metrics, are similar to Equation 2 and are currently working on showing if and when this is true.

3.3 Per-hop Busy Periods

Another indicator of congestion is busy periods at hops. These represent time intervals during which the queue is not empty [7]. In MFNs, busy period statistics can be estimated using (low priority) L-probes. In a single-hop case, an L-probe would be transmitted only when the current busy period ends. Thus, they measure the *remaining duration* of a busy period from an arbitrary point in time. This can be generalized to multi-hop paths by sending H_dL_h -probes. Due to lack of space, we do not show these results. They are present in [14]. As with per-hop queueing delays, non-preemption noise and bias due to persistence also affect these measurements.

Many other methods to measure delay, loss and bandwidth-related properties can be performed in MFNs. Due to lack of space, we briefly describe some of them in Table 1 (see [14] for a more detailed discussion). We divide them into three classes. The first shows how a variety of known and newer single-hop methods can be adapted to measure targeted hops. The second class illustrates the intrinsic measurement ability of low priority queueing. The third class shows what can be done without using high priority. Thus, these can be exposed to end-users without any additional rate limits. Notice that all methods are non-intrusive either because they use low priority everywhere or require infrequent probing only. The non-intrusiveness of low prior-

ity packets is especially useful in measuring loss properties of already-congested hops. All methods do suffer from arguably small inaccuracies due to non-preemption and persistence. For instance, per-hop jitter uses the following approximation similar to Equation 3.

$$(Q_h(T), Q_h(T + t)) \approx (Q_h(T + \epsilon_1) + \epsilon_1 + \epsilon_2, Q_h(T + t + \epsilon'_1) + \epsilon'_1 + \epsilon'_2).$$

Here, ϵ_1, ϵ'_1 and ϵ_2, ϵ'_2 represent the non-negative noise up to (and excluding) hop h and from hop $h + 1$ onwards respectively.

4. DISCUSSION

In this section, we discuss important issues that arise with our architecture in practice. Due to space constraints, we outline the important challenges and ways of addressing them.

4.1 Unavoidable Inaccuracies

Inaccuracies due to non-preemption and persistence are fundamental and cannot be completely circumvented. We believe that the simplicity of active measurements along with the following three reasons for ignoring/eliminating inaccuracies make our architecture appealing. First, non-preemption noise can be bounded and is mostly dependent (see Equation 2) on the transmission time of a maximum-sized packet on the narrow link of the path. We also provided some justification why the sampling bias due to non-preemption and persistent cross-traffic is likely to be small. We intend to quantify these errors in future. Second, the CDF of noise from the entire end-to-end path can also be estimated using H_d -probes (see left plot of Figure 2). Such estimates can in principle be used to “de-noise” measured CDFs. Third, the inaccuracies are not relevant in techniques that do not use high priority.

4.2 Deployment

There are two deployment issues with our architecture. The first issue, implementing priority queueing, requires only the necessary configuration options in current routers. This

is because most routers today have implementations of Differentiated Services [3] that allow at least three strict priority queues to be defined. Thus, a network operator only needs to change the configuration of a router to treat appropriately classified probes with the suitable priority. The second issue, that of probe packets indicating their priority, can be achieved in multiple ways. In the simplest case, network operators can just configure certain source/destination IP addresses and ports for each priority if measurements are only temporary. For a more permanent deployment, network operators can use any of the Type-of-Service (ToS) or even, address fields. If end-users need to signal arbitrary combinations of priorities, more bits needs to be used. However, given that measurement packets do not need arbitrary port numbers, we can leverage the 16-bit port numbers in addition to ToS bits.

4.3 Impact on Normal Data Traffic

One of our stated goals is to use non-intrusive measurements. Techniques that use high priority require very few probe packets and do not rely on observing queue buildups, such as in [9]. Techniques which use low priority packets are, of course, non-intrusive and do not affect existing data traffic. Therefore, our architecture is not intrusive if used by network operators internally. But, opening up MFN primitives to end-users, if the operator is motivated to make the network more transparent, requires a combination of policy and access control. Policy limiting the amount of probing traffic is essential if end-users are allowed to use high priority. Since none of our techniques rely on sending many high-priority packets back-to-back, using a small queue for the high-priority packets is acceptable. An additional issue with allowing end-users access to our primitives is that of multiple simultaneous measurements. For instance, high-priority probing packets from different measurement streams could cause undesirable (high-priority) queueing. There are two ways of solving this problem: (i) End-users infer the presence of multiple measurement entities by observing losses of H_d -probes (which are rate-limited and have small queues, as described above), (ii) Explicit access control mechanisms are used.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we showed that novel network primitives based on hop-dependent priority queueing can be used to design a Measurement-Friendly Network (MFN). We showed that a variety of performance metrics can be accurately estimated in our MFN architecture. Our architecture has many advantages including ease of deployment, simplicity and generality. By the latter, we mean that nothing in our architecture precludes the use of non-FIFO schedulers for normal data packets as long as high and low priority with respect to *all* data packets is possible. This is a very powerful advantage given that many access technologies use non-FIFO queueing disciplines. There are many avenues for future work. The more immediate ones are related to exploring, in detail, the various techniques sketched in this paper. In this context, we intend to quantify the estimation errors due to non-preemption and persistence. Another avenue for future work is that of estimating other metrics. Traffic matrix estimation is one such. Though it is a network-wide metric, it is inherently tied to cross-traffic persistence. We intend to explore if the dependence between two consecutive hops can be

used to estimate the amount of persistent traffic and therefore, also estimate the traffic matrix. It is unlikely, however, that fine-grained metrics such as flow sizes are going to be measurable using active measurements.

6. REFERENCES

- [1] NS-2(Network Simulator). <http://www.isi.edu/nsnam/ns/>.
- [2] F. Baccelli, S. Machiraju, D. Veitch, and J. Bolot. The Role of PASTA in Network Measurements. In *Proc. of ACM SIGCOMM*, 2006.
- [3] S. Blake, D. L. Black, M. A. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC 2475 - An Architecture for Differentiated Services, December 1998.
- [4] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance. In *Proc. of ACM SIGCOMM*, 1994.
- [5] A. B. Downey. Using Pathchar to Estimate Internet Link Characteristics. In *Proc. of ACM SIGCOMM*, 1999.
- [6] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-Based Congestion Control for Unicast Applications. In *Proc. of ACM SIGCOMM*, 2000.
- [7] N. Hohn, D. Veitch, K. Papagiannaki, and C. Diot. Bridging Router Performance and Queuing Theory. In *Proc. of ACM SIGMETRICS*, 2004.
- [8] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang. Locating Internet Bottlenecks: Algorithms Measurements and Implications. In *Proc. of ACM SIGCOMM*, 2004.
- [9] M. Jain and C. Dovrolis. End-to-end Available Bandwidth: Measurement Methodology Dynamics and Relation with TCP Throughput. *IEEE/ACM TON*, 11(4):537-549, 2003.
- [10] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi. CapProbe: A Simple and Accurate Capacity Estimation Technique. In *Proc. of ACM SIGCOMM*, 2004.
- [11] K. Lai and M. Baker. Measuring Link Bandwidths Using a Deterministic Model of Packet Delay. In *Proc. of ACM SIGCOMM*, 2000.
- [12] X. Liu, K. Ravindran, and D. Loguinov. Multi-Hop Probing Asymptotics in Available Bandwidth Estimation: Stochastic Analysis. In *Proc. of IMC*, October 2005.
- [13] M. J. Luckie, A. J. McGregor, and H.-W. Braun. Towards Improving Packet Probing Techniques. In *Proc. of IMW*, November 2001.
- [14] S. Machiraju. *Theory and Practice of Non-intrusive Active Network Measurements*. PhD thesis, University of California at Berkeley, 2006.
- [15] S. Machiraju, D. Veitch, F. Baccelli, A. Nucci, and J. Bolot. Theory and Practice of Cross-Traffic Estimation Poster. In *Proc. of ACM SIGMETRICS*, June 2005.
- [16] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level Internet Path Diagnosis. In *Proc. of ACM SOSP*, 2003.
- [17] A. Pásztor and D. Veitch. Active Probing using Packet Quartets. In *Proc. of IMW*, 2002.
- [18] V. Paxson. End-to-End Internet Packet Dynamics. In *Proc. of ACM SIGCOMM*, 1997.
- [19] M. Singh, S. Guha, and P. Francis. Utilizing Spare Network Bandwidth to Improve TCP Performance. In *Poster in Proc. of ACM SIGCOMM*, 2005.
- [20] J. Sommers, P. Barford, N. Duffield, and A. Ron. Improving Accuracy in End-to-End Loss Measurement. In *Proc. of ACM SIGCOMM*, 2005.
- [21] J. Strauss, D. Katabi, and F. Kaashoek. A Measurement Study of Available Bandwidth Estimation Tools. In *Proc. of IMC*, 2003.
- [22] G. Varghese and C. Estan. The Measurement Manifesto. In *Proc. of HotNets*, 2003.
- [23] D. Veitch, S. Babu, and A. Pásztor. Robust Synchronization of Software Clocks Across the Internet. In *Proc. of IMC*, 2004.