Bandwidth Estimation in Broadband Access Networks

Karthik Lakshminarayanan

University of California at Berkeley

Venkata N. Padmanabhan
Jitendra Padhye

Microsoft Research

May 2004

Technical Report
MSR-TR-2004-44

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Bandwidth Estimation in Broadband Access Networks

Karthik Lakshminarayanan — University of California at Berkeley
Venkata N. Padmanabhan — Microsoft Research
Jitendra Padhye — Microsoft Research

***Abstract*— There has been much work on developing techniques for estimating the capacity and the available bandwidth of network paths based on end-point measurements. The focus has primarily been on settings where the constrained link can be modeled as a point-to-point link with a well-defined bandwidth, serving packets in FIFO order. In this paper, we point out that broadband access networks, such as cable modem and 802.11-based wireless networks, break this model in various ways. The constrained link could (a) employ mechanisms such as token bucket rate regulation, (b) schedule packets in a non-FIFO manner, and (c) support multiple distinct rates. We study how these characteristics impede the operation of the various existing methods and tools for bottleneck and available bandwidth estimation, and present a new available bandwidth estimation technique, *probegap*, that overcomes some of these difficulties. Our evaluation is based on experiments on actual 802.11a and cable modem links.**

I. Introduction

There has been much work on developing techniques for estimating the *capacity* and *available bandwidth*[1] of network paths based on end-point measurements. Capacity is defined as the bandwidth of the *narrow* link on a path while available bandwidth is defined as the headroom on the *tight* link. As noted in prior work [14], the motivation behind bandwidth estimation has been the potential for applications and end-host-based protocols to take advantage of bandwidth information in making intelligent choices on server selection, TCP ramp-up, streaming media adaptation, etc.

Previous work has assumed a simple model of network links. This model [23] assumes that the constrained link[2] along a path has a well-defined raw bandwidth that indicates the rate at which bits can be sent down the link. The link is assumed to be point-to-point with FIFO scheduling of all packets, including measurement probes and cross-traffic. Finally, the cross-traffic is assumed to be "fluid". We term this model the "traditional model".

In this paper, we argue that many of these assumptions made in the traditional model break down in the context of broadband access networks such as cable modem and 802.11 networks. Such networks are proliferating, and are likely to be the constrained link on paths to/from end hosts such as home computers (the anecdotal "last-mile" bottleneck), hence the deviation from the assumed link model becomes significant. There are a number of reasons why the traditional model breaks down:

- The link may not have a fixed or well-defined raw bandwidth, for instance, because of token-bucket rate regulation as in cable modems [3] or dynamic multi-rate schemes as in 802.11 [2]. A distinction needs to be made between the raw link bandwidth and the bandwidth seen by sustained streams.
- The scheduling of packets may not be FIFO, either because of a fully distributed contention-based MAC as in 802.11 or a centrally coordinated MAC as in the cable modem uplink.
- Multi-rate 802.11 links can interfere to create highly bursty cross-traffic patterns that result in a significant departure from the preferred fluid model for cross-traffic.

This paper makes three contributions. First, we identify the characteristics of broadband networks that present challenges to existing techniques for capacity and available bandwidth estimation. Second, we evaluate these problems through experiments in real broadband networks. We focus here on the broadband links in isolation, rather than as part of wide-area Internet paths, to be able to specifically evaluate the broadband issues. Third, we present a new available bandwidth estimation technique called *probegap* that shows promise in addressing some of these difficulties. The main idea behind this technique is to probe for "gaps" (i.e., idle periods) in the link by gathering one-way delay (OWD) samples. Beyond these specific contributions, we hope that our work will put the spotlight on a ripe and important area for future bandwidth estimation research.

The rest of this paper is organized as follows. In Section II, we survey related work on capacity and available bandwidth estimation. In Section III, we discuss the various characteristics of broadband networks that deviate from the link model assumed in previous work. We then describe the Probegap tool that we developed to address some of these challenges in Section IV. After describing our experimental testbed and methodology in Section V, we present an evaluation of existing tools as well as Probegap in broadband networks Section VI. We conclude in Section VII.

II. Related work

We survey the many tools and techniques that have been proposed for estimating the capacity and the available bandwidth of network links and paths. The literature offers multiple definitions of available bandwidth. For our discussion in this paper, we define available bandwidth to be the maximum rate that a new flow can send at without impacting the rate received by the existing flows on the tight link.

Many of the proposed capacity estimation schemes are based on the packet-pair principle [13], [15]. To alleviate the

[1] In this paper, we use the "bandwidth" to mean the data rate of links or paths, expressed in bits per second. We are *not* referring to the spectrum bandwidth at the PHY layer.

[2] We use the term "constrained link" to refer to both the narrow and tight links.

problem of cross-traffic interference, various refinements have been proposed, including sending trains of packets of various sizes (e.g., bprobe [7]) and better filtering techniques to discard incorrect samples (e.g., nettimer [16]). The filtering problem is complicated by the multi-modality of the distribution of packet-pair spacing [20] and the observation that the dominant mode may not correspond to the capacity [9]. In our experiments, we use the Pathrate tool [9], which employs a sophisticated filtering procedure to identify the correct mode even if it is not dominant.

An alternative to the packet pair/train approach is to infer link capacity from the relationship between packet size and delay, as in tools such as pathchar [11], clink [10], and pchar [18] . However, delay measurement relies on ICMP time-exceeded messages from routers, which limits both the applicability and the accuracy of these tools. On the other hand, these tools do not rely on the FIFO assumption made in the traditional model discussed earlier.

Turning to available bandwidth estimation, early techniques such as cprobe [7] measured the asymptotic dispersion rate [9] rather than the available bandwidth. Many of the recently proposed techniques fall into two categories: packet rate method (PRM) and packet gap method (PGM). PRM-based tools, such as pathload [14], pathchirp [24], PTR [12], and TOPP [6], are based on the the observation that a train of probe packets sent at a rate lower than the available bandwidth would be received at the sending rate (on average). However, if the sending rate exceeds the available bandwidth, the received rate would be lower than the sending rate, and the probe packets would tend to queue up behind each other, resulting in an increasing OWD trend. Available bandwidth can be estimated by observing the sending rate at which a transition between the two modes occurs. We pick Pathload as the representative PRM-based tool for our experiments.

PGM-based tools, such as Spruce [23], Delphi [21] and IGI [12], send pairs of equal-sized probe packets, spaced apart according to the transmission time of the probes on the bottleneck link. If no cross-traffic gets inserted between the probes, then the inter-probe spacing is preserved at the receiver. Otherwise, the increase in the spacing is used to estimate the volume of cross-traffic, which is then subtracted from the capacity estimate to yield the available bandwidth. Unlike PRM, PGM assumes that the tight link is also the narrow link, and is susceptible to queuing delays at links other than the tight link. We pick Spruce as the representative PGM-based tool for our experiments.

Gunawardena *et al.* [8] have proposed an alternative approach to available bandwidth estimation based on measuring the RTT of probe packets, and the change thereof when a known amount of additional traffic is introduced. While this technique also uses delay information, it differs from Probegap (the tool that we introduce in Section IV) in how it operates, is much more heavy-weight (since it needs to introduce enough additional traffic to measurably affect the RTT of the probe packets), and is susceptible to asymmetry in link and cross-traffic characteristics given its dependence on RTT (rather than OWD as in Probegap).

While most of the above techniques assume the traditional link model, there has been some work recognizing and addressing issues that arise in settings where this model breaks down. Paxson [20] proposes techniques to mitigate the effect of multi-channel bottleneck links such as ISDN links. In a survey paper [22], Prasad *et al.* briefly discuss the impact traffic regulation and multi-rate on the definition of capacity. A recent macroscopic measurement study of broadband hosts [17] has also pointed issues, such as token bucket rate regulation, that might affect capacity estimation. However, we are not aware of a detailed measurement-based study that considers the bandwidth estimation issues that arise in the context of broadband links, which is the focus of this paper.

III. Broadband Network Issues

In this section, we discuss the characteristics of broadband networks that have an impact on the definition of and estimation techniques for bottleneck bandwidth and available bandwidth. We focus on two types of broadband access network technologies: cable modems and 802.11-based wireless; the latter is being used increasingly as the access technology in wireless hotspots, community wireless networks, etc. While DSL is also a major broadband access technology, our experiments show that DSL links conform to the traditional model, so we exclude such links from our discussion here.

A. Traffic Regulation

It is assumed that a link has a well-defined raw bandwidth that indicates the rate at which bits can be sent down the link. However, this assumption breaks down when a traffic regulation scheme is used. Typically, ISPs divide up a physical access link into smaller pieces that they then parcel out to customers. For example, the raw bandwidth of a typical DOCSIS-compliant cable modem network in North America is 27 Mbps downstream and 2.5 Mbps upstream (both per channel) [3]. However, the bandwidth that a customer is “promised” is typically an order of magnitude smaller.

To parcel out bandwidth in this way, a traffic regulation scheme is employed at the ISP end (e.g., the Cable Modem Termination System (CMTS), or cable “head-end”) and/or the customer end (e.g., the Customer Premises Equipment (CPE), or cable modem). The mechanism used usually in cable modem networks is a token bucket [1], which specifies the mean rate (in bits per second) as well as the maximum burst size (in bytes). Although the rate achievable by a sustained transfer is constrained by the mean rate, it is possible to send data corresponding to the token bucket depth at a rate equal to the raw link bandwidth.

Thus we need to make a distinction between the raw link bandwidth and the maximum achievable rate for sustained transfers. It is possible that packet pairs or even short packet trains will measure the former whereas applications may be more interested in the latter.

B. Non-FIFO Scheduling and Contention

The traditional model assumes that all packets arriving at a link are serviced in FIFO order. So a probe packet is assumed to experience a queuing delay commensurate with the *total* volume (in bytes) of the yet-to-be-serviced cross-traffic that preceded it in the queue. The size of the *individual* cross-traffic packets is therefore assumed to be not critical.

However, this FIFO model breaks down in broadband network settings. In an 802.11 wireless network, the stations contend for access to the channel in a distributed fashion. In a cable modem system, the CMTS periodically sends out a control message indicating the time slots assigned to the various stations and inviting the stations to contend for unused slots. So, in both settings, packets waiting at the different stations would not be transmitted in FIFO order.

One consequence of non-FIFO scheduling is that it may become harder in high-load situations to ensure that a packet pair goes through back-to-back (i.e., without any intervening cross-traffic). This is especially so when the MAC protocol tries to ensure fairness, either through explicit scheduling as in the cable modem uplink or through a distributed mechanism as in 802.11 (where a station that just finished transmitting a frame has a lower probability of winning the next round of contention compared to other stations that may have already partially counted down their backoff counters). The difficulty of sending packets back-to-back may impede the operation of capacity estimation techniques.

Another consequence of non-FIFO scheduling is that a new probe packet enqueued at one of the stations might in fact be transmitted sooner than the older cross-traffic packets waiting at other stations. So the probe packet may not experience a delay commensurate with the total volume of cross-traffic, leading to underestimation of the volume of cross-traffic and over-estimation of the available bandwidth.

The situation is complicated by the fact that contention/scheduling typically happens on a per-frame basis, regardless of frame size. Competing flows with different packet sizes would tend to get shares of bandwidth commensurate with their packet size. So the estimate produced by an available bandwidth estimation procedure might depend on the relative packet sizes of the probe traffic and the cross-traffic.

Finally, a contention-based MAC typically results in a loss of efficiency commensurate with the number of contending stations. So for a given aggregate volume of cross-traffic, the available bandwidth would tend to be lower when the number of stations is larger. However, this effect is significant only when the number of stations is large [5], and we do not evaluate this issue in our experiments.

C. Multi-rate Links

Links may operate at and switch between multiple rates. For instance, 802.11b supports dynamic rate adaptation that allows a radio link to switch between 1, 2, 5.5, and 11 Mbps rates by switching modulation schemes depending on channel quality, which can vary dynamically due to mobility or environmental changes. Likewise, 802.11a supports rates ranging from 6 Mbps to 54 Mbps. Thus the raw bandwidth of the link between say an access point (AP) and a wireless station could change abruptly.

Even if the link rate for each station does not change frequently, different stations in the same region could be operating at different rates, while still sharing the same wireless spectrum [19]. Thus the impact that a given volume of (cross-) traffic on the link to one station has on the available bandwidth on another link depends on the rate at which the former link is operating. For example, consider two clients associated with an AP. The client trying to estimate available bandwidth can communicate with the AP at 54 Mbps, while the other client that is generating cross-traffic can communicate only at 6 Mbps (say because it is further away). Since 802.11 contention happens on a per-frame basis, a single packet of cross-traffic sent on the 6 Mbps link would appear as a large burst of 9 back-to-back packets from the viewpoint of the 54 Mbps link.

This has the potential of impacting both the PRM- and PGM-based techniques for available bandwidth estimation. These techniques work best when the cross-traffic conforms to the fluid model (i.e., with infinitely small packet size) so that it gets interspersed uniformly with the probe packets. The highly bursty cross-traffic pattern might make it harder for a PRM-based technique such as Pathload to detect a clear increasing trend when the probing rate exceeds the available bandwidth. Likewise, the burstiness might make it harder for a PGM-based technique such as Spruce to obtain an accurate sample of the cross-traffic.

D. Is Available Bandwidth Still Interesting?

The discussion of non-FIFO scheduling in Section III-B raises an interesting question. If the (non-FIFO) MAC protocol were perfectly fair, then it may in fact be feasible to estimate the fair share of a new flow, yielding an approximation of the throughput that a new TCP connection would receive. Given that, is it still interesting to estimate the available bandwidth?

We believe that available bandwidth remains an interesting metric since it indicates the level to which a flow can quickly ramp up without negatively impacting existing traffic. For instance, if the fair share of a new TCP flow is 3 Mbps but the available bandwidth is only 1 Mbps, the appropriate behavior would be to quickly ramp up to 1 Mbps and then use the standard TCP congestion control algorithm to gradually attain the fair share of 3 Mbps. Ramping up to 3 Mbps right away would likely be disruptive to the existing flows.

As another example, in an 802.11-based in-home digital A/V network, a key question is that of admission control: can a new stream (say from the home media center to a TV) be admitted without impacting the existing streams? So the quantity of interest is the available bandwidth. Knowing the fair share is not as useful since it does not indicate whether the new stream would negatively impact existing streams in its attempt to attain its fair share.

IV. Probegap

We have discussed the problems that non-FIFO scheduling and frame-level contention present for existing techniques. We have developed the *probegap* tool for available bandwidth estimation to alleviate these problems. The idea is to estimate the fraction of time that a link is idle by probing for “gaps” in the busy periods, and then multiply by the capacity to obtain an estimate of the available bandwidth.

Probegap estimates the idle time fraction by gathering samples of one-way delay (OWD) over the link[3] The sender sends Poisson-spaced 20-byte probes containing a local timestamp and the receiver subtracts this from the received time to compute the OWD. The clocks need not be synchronized; they just need to maintain a constant offset. If a probe finds the link to be free, it would experience a low OWD. On the other hand, if it needs to wait for packets in transmission or ahead of it in the queue, it would experience a larger OWD. As illustrated in Figure 6, the distribution of OWD samples shows two distinct regions, the lower one corresponding to an idle channel and the higher one corresponding to a busy channel. Thus the knee in the CDF of OWD samples identifies the fraction of time that the channel is idle.

We identify the knee using the following heuristic. Each point on the CDF curve is considered as a candidate for identifying the knee. Each such point divides the CDF curve into two sections. We compute the "average" slope of each section as the slope of the line segment connecting the candidate point to the first and last points, respectively, on the CDF curve. A candidate point that locally maximizes the ratio of the slopes of the two segments identifies a knee. (We use a threshold of a 10% drop in the ratio on either side of the local maximum, to filter out spurious local maxima due to noisy data.) In general, the CDF curve may have multiple knees; we only consider the first one for our purposes here.

Probegap is lightweight, involving only about 200 20-byte probe packets sent over a 50-second interval in our experiments. It is more immune to the effects of non-FIFO scheduling since a cross-traffic packet in transmission is likely to be "noticed" (i.e., cause a measurable increase in OWD) regardless of which packet is scheduled for transmission next. However, it is not entirely immune to the non-FIFO effects since there is a small chance that a probe packet will arrive exactly during the idle period between successive cross-traffic packets and win the following round of contention, thereby experiencing a small OWD. This problem could be alleviated by having Probegap send probes in small bunches of say 2-3 back-to-back packets and pick the *maximum* OWD in each bunch as the correct sample. If the channel is in fact idle, we would still measure a small OWD, given the small size of the probe packets. But if the channel is busy, it is very unlikely that all of the probes in a bunch will slip through with a small OWD, so we will probably measure a large OWD. We defer the evaluation of this enhancement of Probegap to future work.

There are issues of concern with Probegap. First, relative drift between the sender and receiver clocks can mask any increase in OWD, especially if the cross-traffic packets are small and the link speed is high. However, in our experiments, the relative drift was no more than a few tens of microseconds over a 50-second measurement interval. In contrast, the transmission time of a 300-byte packet over a 6 Mbps channel is 400 microseconds. Second, like Spruce but unlike Pathload, Probegap is susceptible to delay variation due to links other than the tight link, which would be an issue, for instance, in wide-area paths with multiple congested links. However, in this paper we sidestep this issue by focusing on the broadband link in isolation. Furthermore, we believe that available bandwidth estimation even in such local-area settings with a dominant tight link is of interest. For instance, consider the home A/V admission control scenario discussed in Section III-D.

V. Experimental Setting and Methodology

We describe the broadband network testbeds, the capacity and available bandwidth estimation tools, and validation methodology used in our experiments. As noted in Section I, we focus on the broadband links in isolation; we defer the evaluation of wide-area paths that include broadband links to future work.

A. Broadband Network Testbeds

The testbeds we considered include cable modem, wireless, and DSL links. However, since DSL conformed to the traditional model, we do not report those results here.

1) Cable modem testbed: Our cable modem testbed has two components, an on-campus experimental cable network and two commercial connections from Comcast in two different US cities. On the experimental cable network, the CMTS (cable "head-end") equipment was a Cisco uBR7246-VXR [4] and the CPE ("cable modem") was a Linksys BEFCMU10 EtherFast Cable Modem. Access to this experimental network offered us several advantages. We knew the actual rate control and token bucket settings used by the operator. We were able to place a measurement machine close to the CMTS (cable head-end), which allowed us to focus our measurements on just the cable link. We were also able to obtain two cable connections from the same CMTS that allowed us to confirm the non-FIFO scheduling and mutual interference of flows on the cable uplink through direct experiments.

2) Wireless testbed: The wireless testbed consists of 6 identical machines (2.4GHz Celeron, 256MB RAM), named M1 through M6, located within range of each other. Each machine is equipped with a Netgear WAG511 802.11a/b/g NIC and operating in ad hoc mode. We carry out all of our experiments in 802.11a mode to avoid interfering with the production 802.11b network. Unless otherwise specified, the link rate was set to 6 Mbps.

In all experiments, the bandwidth estimation tools are run between M1 and M2, while the other nodes are used to generate cross-traffic.

B. Tools

For capacity estimation, we use Pathrate [9], which subsumes much of the previous work on packet-pair- and packet-train-based capacity estimation. For available bandwidth estimation, we use Pathload [14], a PRM-based tool, and Spruce [23], a PGM-based tool. We also use our new Probegap tool for available bandwidth estimation. In our experiments,

[3] The reader may wonder why we cannot just sniff the channel and determine when it is busy and when it is not. This is difficult to do even in a local-area setting, because the end-host may not be attached directly to the wireless network. Even if it were directly attached, it may be difficult or impossible to assess whether the channel is busy near the intended peer node.

we ran Spruce over 1000 samples. We used a simple UDP traffic generator (udpload) to generate Poisson cross-traffic at various rates and packet sizes.

C. Validation Methodology

For validating the estimates yielded by the various tools, we needed to determine the true capacity and available bandwidth of the link in question.

In the case of the on-campus cable modem network, we knew the raw link speed and the token bucket rate, which we also validated by sending a stream of UDP packets back-to-back and observing the received rate. (The fact that this was an on-campus, experimental network afforded us the flexibility to run such intrusive experiments.) For wireless links, although we knew the nominal link speed, the IP-level capacity as estimated by the UDP stream was significantly lower. So we used the udpload numbers for various packet sizes as the true capacity. We discuss this issue in more detail in Section VI-B.

For validating available bandwidth estimates, we leverage the controlled nature of the wireless network and the lightly-loaded experimental cable network, which ensured minimal or no unwanted cross-traffic (we confirmed this for the cable network though usage statistics obtained from the operator). Thus we were able to control the level of cross-traffic for our experiments. Recall from Section II that available bandwidth is defined as the rate at at which a new flow can send traffic without affecting the existing flows (i.e., the cross-traffic). So we followed up each run of Pathload, Spruce, and Probegap with the following experiment with udpload, while preserving the same level of cross-traffic. We ramp up the rate of the udpload probe stream and determine the point at which the throughput of the cross-traffic begins to dip. The maximum throughput (i.e., receive rate) of the probe stream that leaves the cross-traffic unaffected yields our estimate of available bandwidth. We term this the “measured” available bandwidth.

VI. Experimental Results

In this section, we present experimental results to demonstrate and quantify the impact of the broadband issues described in Section III. The token bucket evaluation is done over the cable modem network whereas the remaining experiments focus on 802.11.

A. Impact of Rate Regulation in Cable Modem

We focus on the downlink of our experimental cable modem testbed. The raw channel rate is 27 Mbps, the token bucket rate is 6 Mbps, and the token bucket depth is 9600 bytes. The cross-traffic was directed at the same station (i.e., cable modem) as the probe traffic, so non-FIFO scheduling was not an issue in these experiments. Here is a summary of our findings:

1) Estimation of channel capacity: We ran Pathrate with various levels of cross-traffic ranging from 0 to 6 Mbps. In all cases (except when the cross-traffic rate was 6 Mbps, causing Pathrate to abort because of the loss of too many probes), Pathrate returned both low and high capacity estimates of 26 Mbps, which is a close match to the 27 Mbps raw channel speed. Basically, the token bucket is large enough (9600 bytes) to permit enough of the probes to go through back-to-back (despite the cross-traffic) that the capacity mode could be identified.

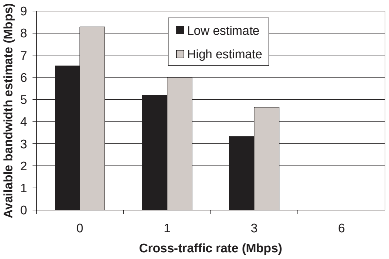

Fig. 1. The low and high available bandwidth estimates produced by Pathload for different levels of cross-traffic over the cable modem downlink. The bars corresponding to the 6 Mbps cross-traffic rate cannot be seen because Pathload’s low and high estimates were zero in this case.

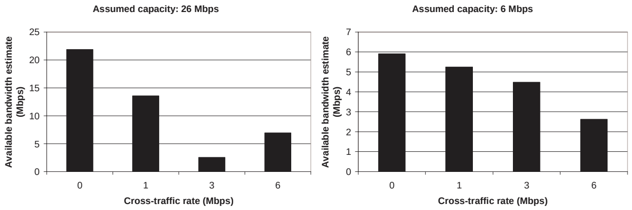

Fig. 2. The available bandwidth estimated by Spruce for different levels of cross-traffic over the cable modem downlink. The estimate differs depending on whether the capacity assumed by Spruce is the raw link bandwidth (26 Mbps) or the token bucket rate (6 Mbps).

2) Estimation of available bandwidth: We present the results for Pathload and Spruce.

Pathload: As shown in Figure 1, Pathload tends to overestimate the available bandwidth. With no cross-traffic, it reports available bandwidth to be in the range of 6.5-8.3 Mbps whereas we confirmed that the link could not sustain a rate higher than 6 Mbps. The reason for this overestimation is that Pathload uses 300-byte probe packets at these rates, and the 9600-byte token bucket size means that a large fraction of the probes could burst out at the raw channel rate, upsetting the OWD trend that Pathload looks for. As the cross-traffic rate is increased to 3 Mbps and 6 Mbps, its estimates drop down to 3.3-4.6 Mbps and 0 Mbps, respectively.

Spruce: Spruce needs an estimate of link capacity as input. It is unclear what link capacity Spruce should assume. We ran experiments assuming both the capacity returned by Pathrate (26 Mbps) and the token bucket rate (6 Mbps). Figure 2 shows the results. We see that in both cases, Spruce significantly overestimates the available bandwidth. The key problem is that depending on the state of the token bucket, the probe packets could go through at the assumed rate even in the presence of intervening cross-traffic, resulting in the overestimation.

We make a couple of interesting observations. First, when the capacity is assumed to be 26 Mbps, Spruce’s estimate corresponding to the 3 Mbps cross-traffic rate (2.5 Mbps) is consistently lower than that corresponding to the 6 Mbps cross-traffic rate (6.9 Mbps). The high level of congestion in the latter case causes 60-70% of Spruce’s probes to be lost, biasing the estimate towards probes that make it through when the cross-traffic is comparatively lighter. Second, when the

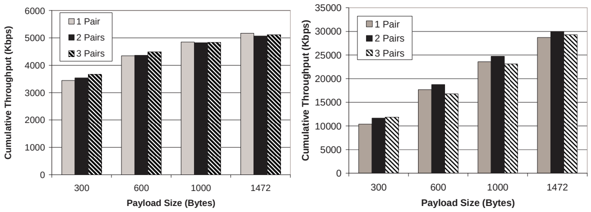

Fig. 3. Impact of packet size on maximum achievable throughput. The nominal channel capacity is 6Mbps for the left graph, and 54Mbps for the right graph. Each column represents average of 3 runs.

Total Cross traffic	0	3	4
Estimate	5.2 – 5.4	5.1 – 5.4	5.3 – 5.5

Fig. 4. Estimation of channel capacity using Pathrate under various loads. All numbers are in Mbps. Nominal channel capacity is 6Mbps.

cross-traffic rate is 3 Mbps, the estimate obtained assuming a 6 Mbps capacity (4.5 Mbps) is considerably larger than that obtained assuming a 26 Mbps capacity (2.5 Mbps). The reason is that in the latter case the increase in spacing between Spruce's probes due to cross-traffic is much higher *relative to* the input gap.

Summary: Our experiments indicate that both capacity and available bandwidth estimation are challenging because of the dichotomy between the raw link bandwidth and the token bucket rate. The problem is particularly acute in the case of a PGM-method like Spruce, since there is no "right" capacity estimate that can be assumed.

B. Impact of Packet Size in 802.11

Since packet transmission with a contention-based MAC such as 802.11 involves significant per-packet overhead (such as the preamble and the minimum spacing between successive packets), we quantify the impact of packet size on the maximum achievable throughput by using the udpload tool to blast a stream of back-to-back packets of various sizes. We also varied the number of simultaneously communicating node pairs from 1 to 3. Figure 3 shows the result when the wireless card rate was set to 6 Mbps and 54 Mbps. We see that in both cases, the cumulative throughput of the pairs increases significantly with the packet size, but does not depend strongly on the number of communicating pairs. Thus, we can conclude that the main source of throughput reduction is the MAC-layer overhead, and not OS overhead at the individual senders or receivers Otherwise the throughput should have increased with number of pairs.

C. Impact of Contention-based 802.11 MAC

For the experiments in this section, all wireless NICs operate at 6 Mbps (termed the "single-rate" case).

1) Estimation of channel capacity: We ran Pathrate between between M1 and M2, while machines M3-M6 were used to generate cross traffic at various rates and packet sizes. In all the runs, Pathrate produced a consistent estimate between 5.1 and 5.5 Mbps. This estimate is close to the maximum UDP traffic rate that the channel can support, as seen from Figure 3.

Cross Traffic		Estimate				Measured	
Rate	Payload (Bytes)	Pathload	Spruce	Probegap			
				300	1472	300	1472
1	300	2.9 – 2.9	3.7	2.4	3.5	2.5	3.7
	1472	3 – 3	4.2	2.7	3.9	2.7	4.2
2	300	2.2 – 2.3	3.2	1.6	2.3	1.7	2.3
	1472	2.2 – 2.3	3.5	2.1	3.0	2.0	3.3
3	300	2.3 – 2.3	3.8	0.8	1.1	0.4	0.6
	1472	1.6 – 1.6	1.5	1.4	2.1	1.4	2.4
4	300	2.3 – 2.3	3.7	0.4	0.6	0.1	0.1
	1472	0.9 – 0.9	1.2	0.9	1.2	0.7	1.1

Fig. 5. Single rate: Estimation of available bandwidth under various loads. All numbers are in Mbps. Nominal channel capacity is 6Mbps. The "Rate" column for cross traffic indicates offered rate, not achieved rate.

To understand why Pathrate results are not affected by contention due to cross-traffic, we looked into the log files produced by Pathrate. We found that Pathrate was always able to find a mode between 5.1-5.3 Mbps, indicating that at least some probes go out back-to-back. This is because although 802.11's contention procedure has a bias against the node that just finished transmitting a packet, there is still a non-trivial probability that the same node will win the next round of contention, especially when the number of contending stations is small. The mode at 5.1-5.3 Mbps is not the dominant mode, especially under heavy cross-traffic. However, the asymptotic dispersion measurements usually generate a mode that includes at least part of the lower-rate mode(s), so these are de-emphasized. Thus, the tool always selects the higher mode, resulting in the correct capacity estimate.

2) Estimation of available bandwidth: We now perform experiments on each of the available bandwidth estimation tools, Pathload, Spruce, and Probegap. These tools are always run between M1 and M2, while cross traffic is generated between M3 and M4. We vary the rate of cross-traffic, from 1 to 4 Mbps in steps of 1 Mbps. We consider two packet sizes for the cross-traffic: 300 and 1472 bytes. Note from Figure 3 that the channel saturates at 3.5 Mbps with 300-byte packets and at 5.1 Mbps for 1472-byte packets. So, for instance, an offered cross-traffic load of 3 Mbps using 300-byte packets would constitute a heavy load whereas the same rate with 1472-byte packets would constitute only a moderate load.

For validation, we measure the available bandwidth using the validation technique described in Section V-C, using both 300 and 1472-byte packet sizes for the measurement stream. Note that for the rates we are considering, Pathload uses 300-byte probe packets. So we only compare Pathload's estimates with the validation numbers corresponding to 300-byte packets. Also, since Spruce uses 1472-byte probes, we specify the capacity as 5.1 Mbps and only compare Spruce's estimates with the validation numbers corresponding to 1472-byte packets. In contrast, we compare Probegap's estimates with the validation numbers corresponding to both packet sizes, for the reasons discussed later.

The results are summarized in Figure 5.

Pathload: Under low load conditions, Pathload's estimate agrees well with the available bandwidth measured with 300-byte packets (i.e., numbers in the penultimate column), irrespective of the cross-traffic packet size.

On the other hand, Pathload overestimates the available bandwidth when the cross traffic is high because it is a PRM-based tool. With a contention-based MAC, if a sender is sending at more than its fair share, and a second sender slowly starts ramping up its sending rate, then the first sender will eventually be "pushed back" by the MAC to its fair share, thereby giving the second flow its fair share as well. [4] While this happens, the output rate of the PRM probes matches their input rate, and there is no increasing trend in the OWDs of the probe packets. The net result is that the estimate tends to the fair share rather than the available bandwidth.

Spruce: Spruce's estimates are in good agreement with the measured available bandwidth when both the packet size used for cross-traffic and that used for validation are 1472 bytes (i.e., compare the Spruce estimates in the second row for each rate with corresponding numbers in the last column). This is because Spruce also uses 1472-byte packets to probe the channel. On the other hand, if the cross-traffic packet size is 300 bytes, Spruce tends to significantly overestimate available bandwidth. For example, with a cross-traffic of 4 Mbps comprising 300-byte packets, Spruce estimates the available bandwidth to be 3.7 Mbps whereas the available bandwidth measuring with 1472-byte packets ("1472" sub-column under the "Measured" column) was only 0.1 Mbps. This overestimation is due to contention happening on a per-packet basis, which results in only a small number (typically just one, due to MAC fairness) of the 300-byte cross-traffic packets being inserted between Spruce's pair of much larger probe packets.

Probegap: As noted in Section IV, Probegap estimates the fraction of time the channel is free and multiplies it by the capacity to estimate available bandwidth. However, since capacity depends on packet size, we pick the capacity value corresponding to the packet size that we wish to estimate available bandwidth for (3.5 Mbps for 300-byte packets and 5.1 Mbps for 1472-byte packets (Figure 3)). This mimics what an application interested in available bandwidth for *its* packet size might do.

From the results, we see that Probegap's estimates for a given packet size show a good match to the available bandwidth measured by our validation procedure for the corresponding packet size (i.e., compare the "300" column for Probegap with the penultimate column, and compare the "1472" column with the last column). However, Probegap overestimates the available bandwidth when the cross traffic is high. The reason is that even when the channel is saturated with cross-traffic, there is a small chance that the probe packet will arrive exactly during the idle period between successive cross-traffic packets and then win the contention round. This would result in a small OWD, which Probegap would mistake as indicating an idle channel. This effect can be seen in Figure 6, where even at 4 Mbps cross-traffic, about 10-20% of the probe packets go through with little increase in OWD.

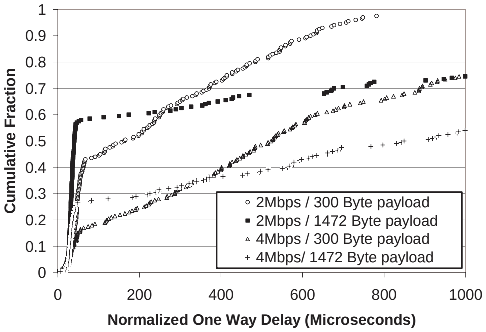

Fig. 6. CDF of one-way delay under various cross traffic conditions. The nominal channel capacity is 6Mbps. Probegap uses 20-byte probe packets. We "normalize" the OWDs to make the minimum OWD in each case equal to zero.

Cross Traffic		Estimate
Rate	Payload	
0	-	29 – 30
2	300	23 – 24
	1472	25 – 27
4	300	10 – 11.4
	1472	32 – 33

Fig. 7. Estimation of channel capacity using Pathrate under various loads. All numbers are in Mbps.

D. Impact of Multirate Environment in 802.11

In this section, we present quantitative results showing the impact of the multirate environment (discussed in Section III-C) on estimates provided by all the tools. The setup for all tests is as follows. The NICs on machines M1 and M2 are set at 54 Mbps. All estimation tools run between these two machines. The NICs on machines M3 and M4 are set at 6 Mbps, and all cross traffic is generated between these two machines.

1) Estimation of channel capacity: The capacity estimates produced by Pathrate are shown in Figure 7. We see that the Pathrate estimate is consistent with the channel capacity in most cases. The one aberration is when the cross-traffic rate is 4 Mbps and its packet size is 300 bytes (representing a fully saturated channel). In this case, Pathrate consistently reported a significantly lower estimate of 11.4 Mbps. We currently do not have a satisfactory explanation for why Pathrate exhibits this behavior in the multi-rate setting but not in the single-rate setting (Figure 4).

2) Estimation of available bandwidth: We conducted experiments for the same set of parameters as in the single-rate case, but only report a subset of them in Figure 8.

Pathload: Consider the cross-traffic rate of 2 Mbps, generated using 300-byte packets. The estimate provided by Pathload is comparable to the measured available bandwidth when the validation test uses 300-byte packets (i.e., same as the probe packet size used by Pathload). However, at a cross-traffic rate of 4 Mbps generated again using 300-byte packets, Pathload significantly overestimates the available bandwidth. This is because of the tendency towards the fair share, as noted in Section VI-C.2.

Cross Traffic		Estimate				Measured	
Rate	Payload (Bytes)	Pathload	Spruce	Probegap			
				300	1472	300	1472
2	300	5.7 – 5.7	12	5	14.2	5.1	13.9
	1472	8.6 – 10.1	25.7	6.4	17.7	6.5	18
4	300	2.6 – 2.9	0	1.2	3.4	0.3	0.3
	1472	2.6 – 2.7	20.9	2.6	7.2	2.7	7.5

Fig. 8. Multi-rate: Estimation of available bandwidth under various loads. All numbers are in Mbps. The estimation ran between M1-M2 (54Mbps), while traffic was generated by M3-M4 (6Mbps).

[4] It is important to remember that the contention occurs on a per-frame basis, so fair share implies equal number of frames.

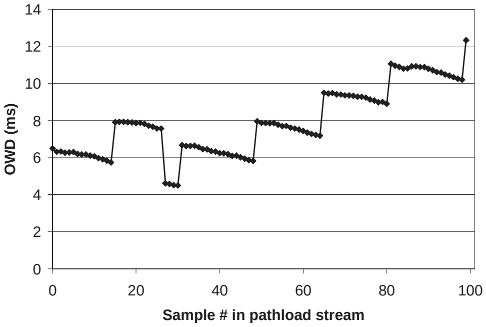

Fig. 9. OWD sequence for a Pathload stream of rate 9.79 Mbps sent on the 54 Mbps channel, with a cross-traffic of 2 Mbps comprising 1472-byte packets sent on the 6 Mbps channel.

When the cross traffic is generated using 1472-byte packets, Pathload overestimates the available bandwidth. For instance, when a cross-traffic of 2 Mbps is generated with 1472-byte packets, Pathload estimates the available bandwidth to be between 8.6-10.1 Mbps whereas the measured available bandwidth (using 300-byte packets) is only 6.5 Mbps. The reason for the overestimation is that the cross-traffic on the 6 Mbps channel appears as large bursts to the Pathload probes sent on the 54 Mbps channel. So the Pathload probes tend to queue up behind the large cross-traffic bursts, and when the channel becomes free, the probes go out back-to-back. Even when Pathload's probing rate exceeds the available bandwidth, typically there are only a few large steps in the OWD sequence, not the steady increasing trend that Pathload expects.

Figure 9 shows one such OWD sequence. Although Pathload's stream rate (9.79 Mbps) exceeds the available bandwidth, there is in fact a *decreasing* trend in OWD between successive steps. This happens because several of Pathload's probes queue up behind large bursts of cross-traffic, before being transmitted back-to-back. The end result is that Pathload is often unable to detect an increasing trend in OWD.

Spruce: Spruce tends to slightly overestimate available bandwidth at low cross-traffic rates. But it reports zero available bandwidth when 4 Mbps cross traffic is generated by 300-byte packets, which closely matches the measured available bandwidth of 0.3 Mbps. The reason Spruce'e estimate is zero in this case, whereas it was 3.7 Mbps in the single-rate case (Figure 5), is that each 300-byte cross-traffic packet appears as a large burst of cross-traffic on the 54 Mbps channel. Given the relative speeds of the two channels, a single such burst is comparable to or larger than the 1472-byte size of the probe packets sent on the 54 Mbps channel.[5] Since the cross-traffic saturates the 6 Mbps channel, there is always a cross-traffic packet waiting to be transmitted. Due to 802.11's attempt at MAC fairness, one cross-traffic packet (equivalent to a burst at least as large as a single 1472-byte probe packet) tends to get inserted on average between Spruce's pair of 1472-byte probes, resulting in the zero estimate. In contrast, in the single-rate case the amount of cross-traffic that gets inserted on average between the probes is only 300 bytes, so Spruce's available bandwidth estimate is higher (3.7 Mbps), although the channel is fully saturated in this case as well.

However, if 1472-byte packets are used to generate the cross-traffic while holding the cross-traffic rate the same (4 Mbps), we find that Spruce significantly overestimates available bandwidth (it estimates available bandwidth to be 20.9 Mbps whereas the value measured with 1472-byte packets is 7.5 Mbps). There are two issues here. First, cross-traffic appears as very large bursts. Each 1472-byte cross-traffic packet appears as a burst roughly 5 times as large as that due to a 300-byte cross-traffic packet, and there could multiple such packets in a single burst. This burstiness makes it difficult for Spruce's sampling process to obtain an accurate estimate of the volume of cross-traffic. Second, the transmission time for the bursts of cross-traffic — often several milliseconds — runs foul of the threshold used in Spruce to disambiguate between genuine cross-traffic-induced packet gaps and gaps due to OS context switches. So the samples corresponding to the large bursts are ignored, resulting in the overestimate. We tried to rectify this problem by including all samples regardless of the OS context switch threshold, but that resulted in an estimate of cross-traffic exceeding the link capacity, implying that the available bandwidth was zero. This goes back to the basic problem that the burstiness in the cross-traffic caused by the interference between multirate links makes accurate sampling difficult.

Probegap: Probegap produces good estimates at low cross-traffic rates (viz., 2 Mbps cross-traffic regardless of the cross-traffic packet size and 4 Mbps cross-traffic generated with 1472-byte packets). However, it significantly overestimates available bandwidth when cross traffic rate is high (viz., 4 Mbps cross-traffic generated with 300-byte packets) for the reasons mentioned in the single-rate case (Section VI-C.2).

VII. Conclusion

In this paper, we have considered the challenges posed by broadband networks to existing techniques for capacity and available bandwidth estimation. We have specifically focused on rate regulation using token bucket in cable modem networks, and non-FIFO scheduling and burstiness caused by multi-rate links in 802.11 networks. Our key findings are that (a) the Pathrate capacity estimation tool estimates the raw link speed but not the token bucket rate, (b) the PGM method breaks down because of the dichotomy between the raw link speed and the token bucket rate, (c) non-FIFO scheduling and frame-level contention in 802.11 causes problems for both the PGM (Spruce) and PRM (Pathload) methods, and (d) interference between links operating at different rates in 802.11 can make cross-traffic appear bursty for the faster link, exacerbating the problems.

We have also introduced Probegap, a new one-way-delay based technique for estimating available bandwidth that alleviates the problems caused by non-FIFO scheduling. While this technique shows promise, evaluating it in wider settings remains a goal for future work.

Acknowledgements

Jeff Chirico, Joseph Heinz, Geoffry Nordlund, Ananth Rao, and Brian Zill helped us with various aspects of our cable

[5] The precise burst size would depend on the actual rather than nominal capacities of the two channels (Figure 3).

modem and wireless testbeds. Constantinos Dovrolis and Manish Jain provided clarifications regarding the Pathrate and Pathload tools while Jacob Strauss answered our questions regarding Spruce. Tom Blank helped with the clock drift issue. We would like to thank them all.

References

[1] http://qbone.internet2.edu/bb/Traffic.pdf.

[2] http://standards.ieee.org/getieee802/802.11.html.

[3] http://www.cablemodem.com/.

[4] http://www.cisco.com/warp/public/cc/pd/rt/ub7200/index.shtml.

[5] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordinated Function. *IEEE JSAC*, 18(3):535–547, March 2000.

[6] B. M. M. Bjorkman and P. Gunningberg. A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks. In *Proc. of Global Internet Symposium*, 2000.

[7] R. L. Carter and M. E. Crovella. Measuring Bottleneck Link Speed in Packet Swicthed Networks. *Performance Evaluation*, 27-28, 1996.

[8] P. K. Dinan Gunarwardena and L. Massoulie. Network Characteristics: Modelling, Measurements and Admission Control. In *IWQoS*, 2003.

[9] C. Dovrolis, D.Moore, and P.Ramanathan. What Do Packet Dispersion Techniques Measure? In *Proc. of IEEE INFOCOM*, 2001.

[10] A. Downey. clink: a Tool for Estimating Internet Link Characteristics, 1999. http://rocky.wellesley.edu/downey/clink/.

[11] A. B. Downey. Using pathchar to Estimate Link Characteristics. In *Proc. of ACM SIGCOMM*, 1999.

[12] N. Hu and P. Steenkiste. Evaluation and Characterization of Available Bandwidth Techniques. *IEEE JSAC Special Issue in Interet and WWW Measurement, Mapping and Modeling*, 2003.

[13] V. Jacobson and M. J. Karels. Congestion Avoidance and Control. In *Proc. of ACM SIGCOMM*, 1988.

[14] M. Jain and C. Dovrolis. End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. In *Proc. of ACM SIGCOMM*, 2002.

[15] S. Keshav. A Control-Theoretic Approach to Flow Control. In *Proc. of ACM SIGCOMM*, 1991.

[16] K. Lai and M. Baker. Measuring Link Bandwidths Using a Deterministic Model of Packet Delay. In *Proc. of ACM SIGCOMM*, 2000.

[17] K. Lakshminarayanan and V. N. Padmanabhan. Some Findings on the Network Performance of Broadband Hosts. In *Proc. of IMC*, 2003.

[18] B. Mah. pchar: A Tool for Measuring Internet Path Charateristics, 1999. http://www.employees.org/~bmah/software/pchar.

[19] Martin Heusse and Franck Rousseau and Gilles Berger-Sabbatel and Andrzej Duda. Performance Anomaly of 802.11b. In *Proc. of IEEE INFOCOM*, 2003.

[20] V. Paxson. *Measurement and Analysis of End-to-End Internet Dynamics*. PhD thesis, University of California at Berkeley, 1997.

[21] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk. Multifractal Cross-traffic Estimation. In *ITC*, 2000.

[22] R.S.Prasad, M.Murray, C.Dovrolis, and K.C.Claffy. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network*, 2003.

[23] J. Strauss, D. Katabi, and F. Kaashoek. A Measurement Study of Available Bandwidth Estimation Tools. In *Proc. of IMC*, 2003.

[24] Vinay J. Ribeiro and Rudolf H. Riedi and Richard G. Baraniuk and Jiri Navratil and Les Cottrell. pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In *Proc. of PAM Workshop*, 2003.