Probing Available Bandwidth in Radio Access Networks

Sven Krasser, Henry L. Owen
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250, USA
Email: {sven, owen}@ece.gatech.edu

Jochen Grimminger, Hans-Peter Huth, Joachim Sokol
Siemens AG, CT IC2 Corporate Technology
81730 Munich, Germany
Email: {jochen.grimminger, hans-peter.huth, joachim.sokol}@mchp.siemens.de

Abstract—**This paper presents a way of measuring whether a certain amount of bandwidth is available on a path in a radio access network or in other kinds of networks with DiffServ support. The basic concept is to fill up the bandwidth that is not used by normal data traffic with special probe packets, which are only forwarded if there are no data packets that could be forwarded. When probing multiple paths the interaction of different probe flows can be exploited to signal bandwidth requirements and to reserve bandwidth. In contrast to other approaches, the available bandwidth is not estimated by using statistical properties of the transmission of trains of packets but determined by measuring the actual throughput of a stream of low-priority probe packets.**

I. Introduction

Traffic engineering in radio access networks (RANs) is an important means to utilize unused resources, to improve service, or to use available resources more efficiently, especially to be able to cope with high point loads. In RANs, we can exploit the fact that the network is relatively small and maintained by a single carrier. However, this research also applies to networks for that similar assumptions hold. To do effective traffic engineering, knowledge about the network state is crucial. To gather this sort of information, network nodes can propagate state information to peer nodes or nodes can actively probe different alternative paths [1]. We will focus on the latter. The path properties are estimated by measuring the characteristics of probe packets (or more briefly referred to as *probes*) transmitted over the path. Several parameters can be obtained by such path probing. While packet loss rate, delay, or jitter probing can be implemented in a straightforward fashion, it is more complicated to get information about the *available* bandwidth on a path. Previous probing techniques based on packet dispersion like cprobe [2] use probes that strongly interfere with actual data traffic, which is undesired if probing has to be performed on a frequent basis since such probing consumes a significant amount of bandwidth by itself, increases queuing delay, and introduces additional jitter. Furthermore, an effective algorithm should give accurate results without probing the path for a long time. This research points out a new approach of probing available bandwidth with minimal intrusiveness to competing traffic, which also has properties useful for traffic engineering in radio access networks. The basic concept is to use dedicated queues on nodes for probe packets. Related approaches have been used in previous research on probe-based end-to-end admission control [3] or scalable resource reservation [4].

The rest of this paper is organized as follows: In Section II we describe the idea our proposed probing mechanism is based on. Section III shows an evaluation of the performance and intrusiveness in a simple network topology with a single probed path. Section IV addresses the issues that have to be taken into account if multiple paths that are sharing links are probed. Section V concludes the paper.

II. Unintrusive Bandwidth Probing

In contrast to the Internet, a RAN most likely will have some means to differentiate different traffic classes. We assume a similar RAN setup as in our previous research [5, 6], namely, DiffServ and MPLS support throughout the RAN. To minimize the impact of probe packets on normal data packets, we can set up a DiffServ codepoint with an associated per-hop behavior especially for probes. The basic concept is to flood the path of interest with probes containing that certain codepoint. Nodes on the path are configured to forward these probe packets only if no other packets are available. An ingress router on the rim of the traffic engineering domain sends out such probes to an egress router. The egress router measures the rate of the incoming probes and sends the result back in a notification packet to the ingress router. Thus, the suggested scheme needs only support at the rim of the network, if we assume that all routers in the inner network are DiffServ-enabled and have support for the special probe codepoint. We refer to those routers as core routers, while the routers on the rim are referred to as edge routers. Edge routers are ingress and egress points for traffic.

However, this scheme does not guarantee zero-intrusiveness. If there are only probes queued at a router's link, the router starts to forward a probe. A new data packet can arrive after the transmission of a probe has started. Nevertheless, the transmission of the probe packet is completed. This is apparently a delay problem that becomes more severe the larger the probe packets get. The minimum packet size is dictated by the underlying link layer technology.

If only one path is probed, the probing router can send out probe packets at a rate greater or equal to the bottleneck bandwidth of this path. The egress router then measures the complete available bandwidth on that path by computing the throughput of the probe flow. If multiple paths are probed and these paths share links, the probes will compete with each other and the available bandwidth measured is lower than the actual available bandwidth. This property may be problematic for the general case, but is rather useful for traffic engineering in RANs. The idea is that routers do not probe for the maximum available bandwidth, they probe for the amount of additional bandwidth they anticipate to use in the near future. In the simplest case, this anticipation can be based on the bandwidth the router is already using on this path. Using bandwidth in this sense means to forward traffic for which this router is the ingress point into the core network, not the traffic that the router forwards on behalf of other routers in the core network. The ingress router learns how much bandwidth of the bandwidth it is asking for is available in the network based on the needs of other routers by a notification packet sent to it by the endpoint of the path, the egress router. Such notification packets containing measurement results are sent by egress routers in fixed time intervals. This implies that every router gets a fair share of the resources of the network. A hotspot cannot completely cut off another part of the network. In this sense routers can reserve bandwidth by probing for it. Another implication is that race conditions are avoided. Consider a case in that two routers know about a certain amount of available bandwidth. Both routers admit new calls and start using a vast amount of this unused bandwidth. The network core, however, might not be able to support the traffic from both routers because of shared links on paths. In our scheme this problem is addressed. Routers just replace probes with data packets up to the rate the egress router receives the probes.

III. Impact of Probing on actual Data Traffic

To get an impression about the severity of the effects of the outlined bandwidth probing mechanism on actual data traffic, we present simulation results obtained using a modified version of ns-2 version 2.1b9a, which is augmented with additional means to support fast simulation of probes. This is crucial since the time a simulation takes is based on the number of events, which is in turn related to the number of packets simulated. Our probing scheme increases the latter enormously.

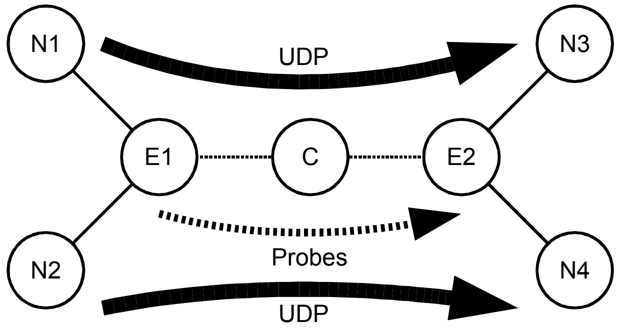

Fig. 1. Topology used to measure intrusiveness.

We are using ns-2's DiffServ module to set up an additional queue just for probes, as described in the previous section. With regard to further extensions we use weighted round-robin scheduling for all queues in the DiffServ domain rather than priority-based scheduling. The queue used for probes has a weight of zero. This implies that this queue is only considered if all the queues for the other traffic classes are empty and therefore yielding their time slot.

For this evaluation, we assume a simple network setup as shown in Fig. 1. All links have a propagation delay of 10 msec and a bandwidth of 100 KBits/sec. The latter has been chosen to limit the number of events. If the link bandwidth is made larger, more probes have to be generated to fill up the additional bandwidth. The simulation of a network with 1 GBit/sec links can therefore take up to about 10,000 times longer. As shown in Fig. 1, there are UDP senders sending from node N1 to node N3 and from node N2 to node N4 respectively. The senders are configured in a way that the link utilization of the DiffServ network core, the dashed links in Fig. 1, is around 60%. We choose a small packet size of 50 bytes for the UDP packets since the negative effect of probes becomes more apparent for smaller data packets. Probe packets are sent from edge router E1 on the rim of the core network to edge router E2. There is only one core router, labeled "C" in

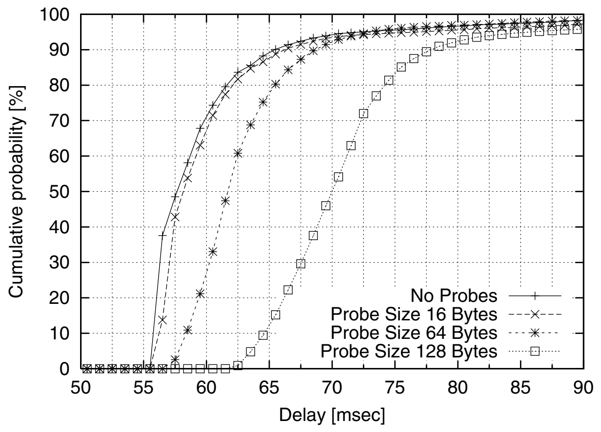

Fig. 2. Delay distribution.

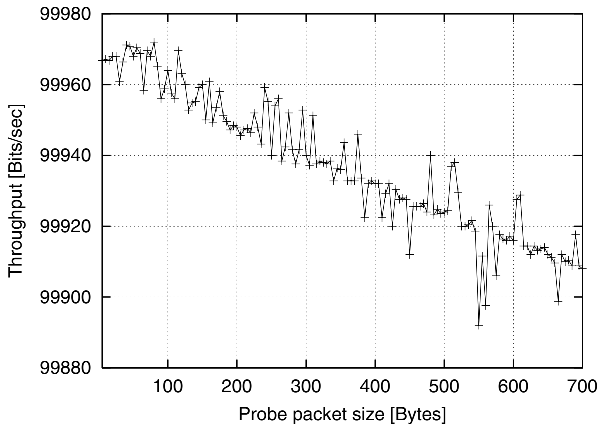

Fig. 3. Overall throughput vs. probe packet size.

Fig. 1. Probes are sent at the maximum rate, and therefore we see the worst-case effects of probes on data packets.

In Fig. 2, we present the resulting effect on the delay distribution. The plot shows the probability that the delay of a data packet is below a certain threshold. The 16-byte probes have nearly no effect on the delay characteristics, while the 128-byte probes introduce an additional delay of over 10 milliseconds. The 64-byte probes increase the delay by less than 5 milliseconds. Probes of about this size are reasonable to assume with regard to popular link-layer technologies.

Fig. 3 shows the overall throughput in the network. This is the aggregated throughput of all UDP senders plus the throughput of the probes. Furthermore, this throughput is closely related to the link utilization of the core links – the only difference is that the UDP throughput is measured on N3 and N4 rather than on E2 as it is done for the probes, which implies additional delay because of the extra link the packets have to traverse. Since the UDP throughput is constant and, in fact, shows only little dependence on the probe packet size, the difference between the 100,000 Bits/sec link capacity and the value of the graph in Fig. 3 is available bandwidth that has not been filled up with probes. Thus, this bandwidth is available but not measured. Fig. 3 shows that the measurement error gets worse as probes become larger. This occurs linearly. Nevertheless, even for probe packets sized 700 bytes, the error is still small, about 90 Bits/sec compared to the link capacity of 100,000 Bits/sec.

IV. Probing Multiple Paths with shared Links

When paths that share links are probed, probes start to interfere with probes on other paths. Probe traffic consists mainly of bursts of probes. Although probes are sent initially at a lower rate (as outlined in Section II) and hence with gaps between them, they are most likely compressed at the first hop and become a burst.

The probe queues should be big enough to be able to cope with multiple bursts. Nevertheless, if these queues are too big, probes might be queued for a long time until they are eventually transmitted to the next node. This measures the available bandwidth more exactly, but it has worse dynamic properties if the available bandwidth changes.

Due to these issues we first of all investigate the severity of such effects in an evenly loaded network. We use a slightly different topology for the next simulations, which is presented in Fig. 4. Initially, UDP senders are on the four leaf nodes, N1 to N4. To avoid flow synchronization packets are generated at random times at these senders. The average rate is set to 15 KBits/sec. Probing is done between the edge routers, E1 to E4. The probe rate is set to 25 KBits/sec. Every edge router sends probes to every peer edge router. The link capacity and the link delay are the same as in the previous section, 100 KBits/sec and 10 msec respectively.

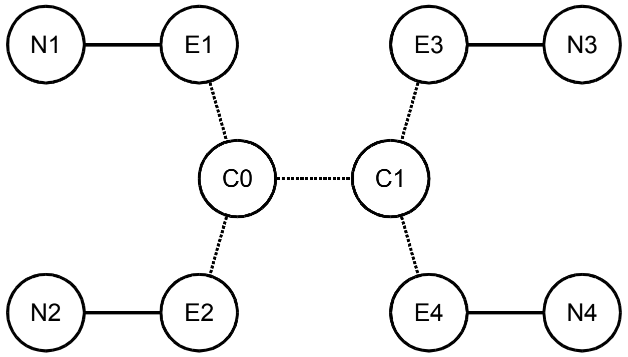

Fig. 4. Topology used for simulations of multiple-path effects.

TABLE I
Throughput measured

Src.	Dest.	Data rate	Probe rate
N1/E1	N2/E2	14895	19726
N2/E2	N1/E1	15189	19648
N3/E3	N4/E4	15003	19725
N4/E4	N3/E3	14846	19653
	anticipated	*15000*	*18333*
N1/E1	N3/E3	15056	10229
N3/E3	N1/E1	14941	9764
N1/E1	N4/E4	14927	9785
N4/E4	N1/E1	15111	10152
N2/E2	N3/E3	14859	9854
N3/E3	N2/E2	14980	10178
N2/E2	N4/E4	14966	10219
N4/E4	N2/E2	15088	9695
	anticipated	*15000*	*10000*

First of all, we show that the available bandwidth is divided up equally among the probe traffic flows. It turned out that the probe queue size only has little impact on the results. Table I shows the results gathered and the values anticipated if a fair share of bandwidth between flows is assumed. The probe queue is a droptail queue that can hold up to 20 probes. The values are averaged over 20 simulation runs to equal out the randomness introduced by the UDP senders.

In the next simulation setup N1 is sending data packets at various rates D_1 to N3. Likewise, E1 is sending probe packets at various rates R_1 to E3. Both rates are chosen such that they add up to 90 KBits/sec. This is to model the replacing of probes with data packets. Additionally, data packets are sent from N2 to N3 and to N4 at rates D_2 and D_3 of 10 KBits/sec each, and probes are sent from E2 to E3 and to E4 at rates R_2 and R_3 of 5 KBits/sec each.

Fig. 5 shows how the bottleneck link capacity (x-axis) is divided up ideally as more and more probe bandwidth is replaced with data packets (y-axis) by N1 and E1 (D_1 is increased at N1 and R_1 is decreased at E1). Note how the throughput of the probe flows originating at E2 decreases as N1 and E1 replace more probes with data. Fig. 6 shows this more precisely for the probe flow from E2 to E3. Ideally, the available bandwidth A is divided up among all probe flows sending at rate R_k according to

$$M_k = R_k \cdot \frac{A}{\sum R_i}, \tag{1}$$

where M_k is the measured probe throughput of probe flow k. If N1 and E1 replace an amount of Q of the original probe rate with data traffic such that the new data rate is $D_{1,new} = D_1 + Q$ and the new probe rate is $R_{1,new} = R_1 - Q$, the

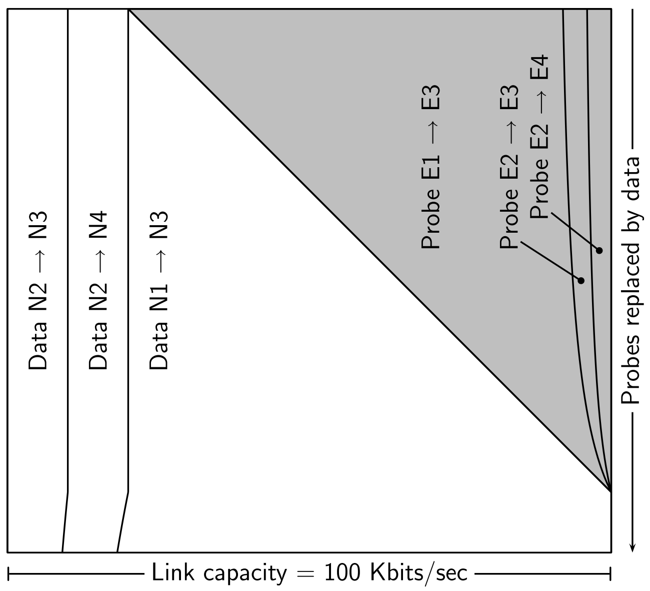

Fig. 5. Anticipated usage of the bottleneck link capacity.

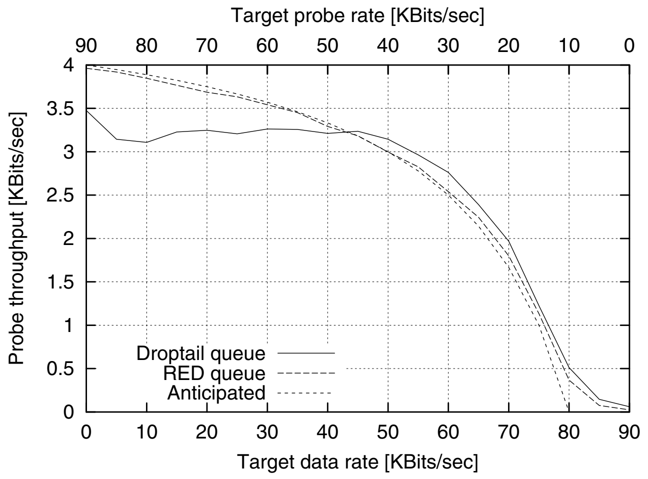

Fig. 6. Probe throughput of the E2 → E3 flow.

fraction in (1) changes and a different measured probe throughput is observed at all probe endpoints. At E3 the rate M_1 of incoming probes from E1 changes to a new value of

$$M_{1,new} = (R_1 - Q) \cdot \frac{A - Q}{\left(\sum R_i\right) - Q}. \tag{2}$$

In Fig. 6 we see that the droptail queue does not guarantee that probe flows share bandwidth fairly. Although RED queues have been developed with respect to congestion aware flows, they turn out to be useful to cope with the burstiness of probes. We use a RED queue with a maximum physical size of 40 packets and set both the minimum and maximum threshold to 20 packets. Note that this implies no early drops. For a description of RED parameters refer to [7].

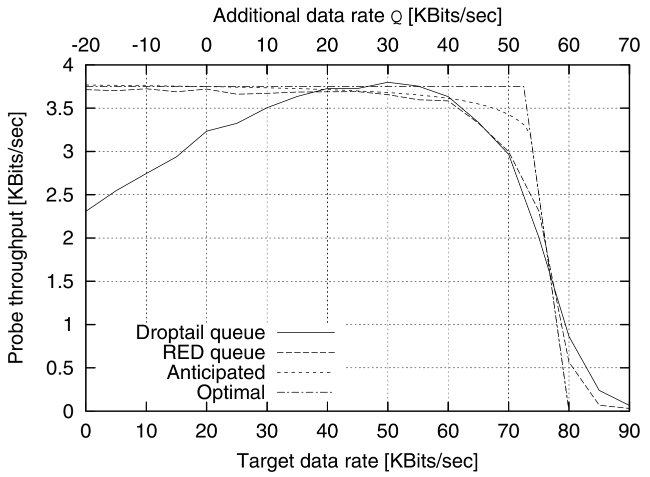

Fig. 7. Probe throughput of the E2 → E3 flow with probe flow E1 → E3 backing off.

For the purpose of bandwidth reservation we want the new measured value to equal the old value M_1 minus the probe bandwidth used for data, Q. Hence, the original probe rate R_1 has to be decreased by K, yielding

$$M_1 - Q = (R_1 - Q - K) \cdot \frac{A - Q}{\left(\sum R_i\right) - Q - K}. \tag{3}$$

By using (1) and (3), we can solve for K and obtain

$$K = Q \cdot \frac{R_1 - M_1}{M_1}. \tag{4}$$

In other words, if the data rate is increased by Q, the probe rate has to be decreased by $Q \cdot R_1/M_1$.

Furthermore, if we assume that no other nodes than N1 and E1 change their rates, we can use $M_{1,new}$ to obtain the available bandwidth and the total aggregated probe rate for the time before the rate changed:

$$A = \frac{M_1 Q \cdot (M_{1,new} - R_1 + Q)}{R_1 M_{1,new} - R_1 M_1 + M_1 Q}, \tag{5}$$

$$\sum R_i = \frac{R_1 Q \cdot (M_{1,new} - R_1 + Q)}{R_1 M_{1,new} - R_1 M_1 + M_1 Q}. \tag{6}$$

To obtain the current values, Q has to be subtracted from (5) and (6). However, to gather these values the router has to probe at an increased rate (by neglecting K) to measure $M_{1,new}$. Hence, other routers will measure a decreased probe rate, which is not desired in our scheme.

In Fig. 7 we show the results. The initial rate for the probe flow from E1 to E3 is 70 KBits/sec. For the data flow from N1 to N3 we choose an initial rate of 20 KBits/sec. M_1 is taken from the previous simulation setup where we measured a rate of 53512.448 KBits/sec. Based on this value we show an anticipated curve based on (3). The optimal curve in Fig. 7 can only be achieved if the measured value M_1 equals the fair-share value of 52500 KBits/sec. Like in Fig. 6, we can see in Fig. 7 that the RED queue performs better.

V. Conclusion

We have presented a new methodology to determine the availability of bandwidth in radio access networks or networks with a similar setup. Furthermore, this new approach enables routers to reserve parts of this bandwidth. Since our approach is based on low-priority probe packets, we achieve minimal intrusiveness with other packets.

The gathered information can be used for the purpose of admission control for bulk data transfers. However, a general admission control scheme also needs information about path delay, e.g. for voice connections. Admission control for such expedited forwarding traffic must therefore be focused on delay measurements as well.

The results gathered so far show the effectiveness in a simplistic network setup with a single bottleneck. The impact of multiple bottlenecks is not addressed. Multiple bottlenecks imply multiple congestable links. Therefore, some of the assumptions made in Section IV do not hold anymore. Additional research results regarding this can be found in [8].

References

[1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of Internet traffic engineering," *IETF RFC 3272*, May 2002.

[2] R. Carter, and M. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Proc. Performance '96*, Lausanne, Switzerland, vol. 27-28, pp. 297-318, 1996.

[3] V. Elek, G. Karlsson, and R. Rönngren, "Admission control based on end-to-end measurements," *Proc. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 623-630, 2000.

[4] W. Almesberger, T. Ferrari, and J.-Y. Le Boudec, "SRP: a scalable resource reservation protocol for the Internet," *Proc. Sixth International Workshop on Quality of Service*, pp. 107-116, 1998.

[5] D. Barlow, H. Owen, V. Vassiliou, J. Grimminger, H.-P. Huth, and J. Sokol, "Router-based traffic engineering in MPLS/DiffServ/HMIP radio access networks," *Proc. IASTED International Conference on Wireless and Optical Communications*, Banff, Canada, pp. 360-365, 2002.

[6] S. Krasser, H. Owen, D. Barlow, J. Grimminger, H.-P. Huth, and J. Sokol, "Evaluation of the local state fair share bandwidth algorithm," *Proc. International Conference on Telecommunications*, Papeete, French Polynesia, pp. 911-916, 2003.

[7] S. Floyd, and V. Jacobson, "Random early detection gateways for congestion avoidance," in *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397-413, 1993.

[8] S. Krasser, H. Owen, J. Grimminger, H.-P. Huth, and J. Sokol, "Distributed bandwidth reservation by probing for available bandwidth," accepted *International Conference on Networks*, Sydney, Australia, Sept. 2003.